

Тест: Дистрибутивы-малоежки

LINUX FORMAT

Главное в мире Linux

Август 2014 № 8 (186)

КАК UBUNTU

Перенастройте популярнейший дистрибутив

- » Пусть Unity приладится к вам
- » Выкиньте хлам и линзы Amazon
- » Играйте на повышенной частоте



ПЛЮС!
Одедем Linux как «Денди»
с. 56



Парень из LPI

« LPI всегда держал планку, но 2013 год остается рекордным даже сейчас »

Билл Куинн Обучим весь мир Linux-до с. 42

Управление пакетами

Apt-get ero!

» Таинственный мир репозитория Linux

Сети

Танцуем с Samba

» Windows и Linux в одной упряжке

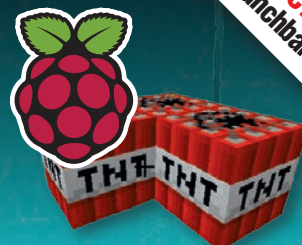
Облачные вычисления

OpenStack

» Ваш личный гипервизор повелевает облаками

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

Linux center
www.linuxcenter.ru



Строим пушку Pi в Minecraft

» Закодируем пушку на Python и пальцем на весь мир! с. 84

БЕСПЛАТНЫЙ DVD
4 ДИСТРИБУТИВА: Puppy, Crunchbang, Manjaro, Elive



Также в номере...

Мониторинг системы

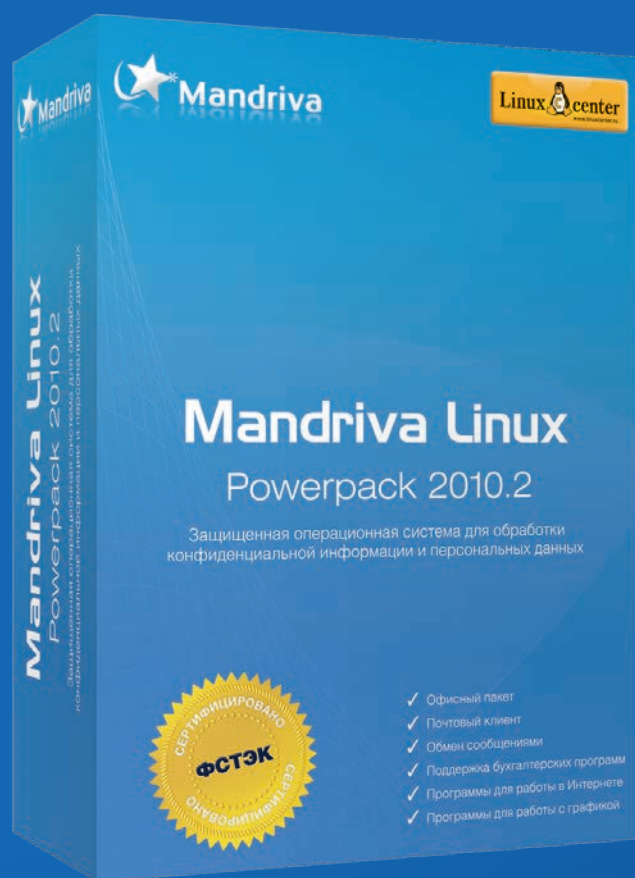
» Проследите за всем на своем ПК, благодаря нашему руководству с. 70



Mandriva Linux

Powerpack 2010.2

Дистрибутив может использоваться для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К1 включительно.



Простота
установки, работы
и администрирования

Легкость
освоения новичками
в GNU/Linux

Широчайший спектр
поддерживаемого
оборудования

Цена 4990 руб.

Дистрибутив Mandriva Linux Powerpack 2010.2
прошел сертификацию ФСТЭК и предназначен для создания
защищенной офисной рабочей станции или небольшого сервера.



www.linuxcenter.ru/shop/certified_fstek/mandriva_fstek/mandriva_2010-2_pp_certific_fstek/

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Что мы делаем

- » Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.
- » Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.
- » Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.
- » Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.



Кто мы

В этом номере мы показываем, как переделать Ubuntu. И мы спросили наших авторов: а что бы они изменили в Linux (если вообще меняли бы)?



Джонни Бидвелл

В некотором смысле Linux сам себя меняет. Я помню, как душно все было 10 лет назад. А сейчас все происходит автоматически. Но я ворчу на *Grub 2* за неумение распознать мои разделы GPT (в которых я сам не разбираюсь), и на *fontconfig* — за то, что не воспроизводит шрифты с точностью до пикселя.



Нейл Ботвик

Я могу удумать массу изменений, но ничего менять не стал бы. Лучше я пожалуюсь на форумах или в Твиттере... может, pošлю сообщение об ошибке... и подожду, пока кто-нибудь это не исправит. Ведь именно так и работает Open Source, разве нет?



Дэвид Хейвард

Во-первых, я бы уменьшил количество рабочих столов: выбор — дело хорошее, но в меру. Во-вторых, вколотил бы в головы изготовителей оборудования, что надо поддерживать драйверы. И, наконец, избавился бы от троллей, которые портят жизнь в Linux своими «ну, я-то знаю побольше вашего».



Лес Паундер

А надо ли менять Linux? Linux — это свобода, свобода выбирать себе дистрибутив и приспособлять его под себя. Linux вечно развивается, и наши любимые дистрибутивы впитывают новые идеи. Потому-то Linux и идеален как передний край технологий.



Маянк Шарма

Два слова: стандарты и стабильность. Разумеется, выбор — это фирменная марка экосистемы Open Source, но где бы мы оказались, если бы кто-то в 1992-м разветвил ядро? Я считаю, что сообществу разработчиков Linux следует выделить ключевые приложения и библиотеки и сосредоточиться на их улучшении.



Валентин Синицын

Я бы пересобрал SLS, добавив к нему кнопку «Пуск», и сделалась настольной ОС №1 еще до выхода Windows 95.



Эргономический тупик

» В начале была последовательность символов, вводимая с перфокарты, перфоленты, а потом — клавиатуры, и выводимая на бумагу или экран... Потом появился графический интерфейс — GUI, позволяющий указать на нечто на экране и нажать кнопку на манипуляторе. Дальнейшее зависело от того, какую кнопку и сколько раз...

В современных системах прекрасно уживаются оба подхода. Первый естественен для ввода текстов и команд, а второй более приближен к интерфейсам «реальной машинерии» — кнопкам, тумблерам и тому подобному.

Попытки создать единый GUI, одинаково удобный для работы с мышью и сенсорным экраном, пока не увенчались успехом, а темой этого номера стали очередные рекомендации по «допиливанию» Unity. Осмелюсь предположить, что задача не решается. По крайней мере, традиционными средствами.

«Нетрадиционные» интерфейсы активно развиваются, но пока не позволяют выйти из тупика. Управление голосом не годится для управления курсором и плохо справляется со вводом связанного текста, а сегодняшние интерфейсы «мозг-компьютер», мягко говоря, несколько неудобны.

Налицо очередной «кризис интерфейса», и следующий прорыв в ИТ, судя по всему, будет связан именно с его преодолением. Посмотрим...

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

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

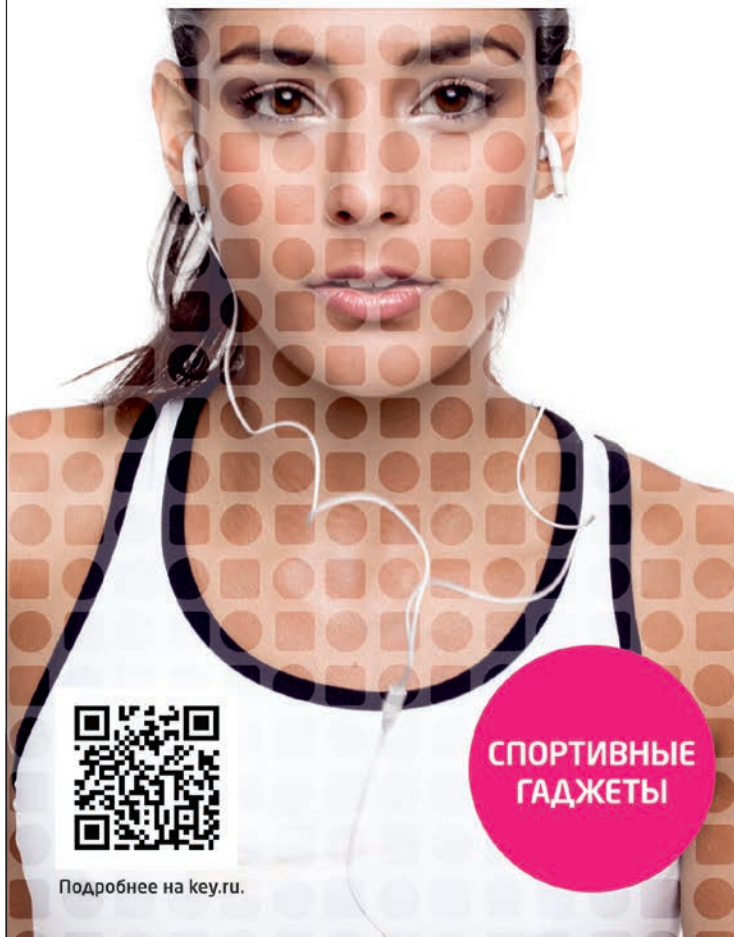
Вопросы распространения: sales@linuxformat.ru

Сайт: www.linuxformat.ru, группа «ВКонтакте»: vk.com/linuxform

» Адрес редакции: Россия, Санкт-Петербург, пр. Медиков, 5, корп. 7

» Телефон редакции: (812) 309-0686. Дополнительная информация на с. 112

ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы



Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



Шаромер Misfit Shine

СПОРТИВНЫЕ
ГАДЖЕТЫ



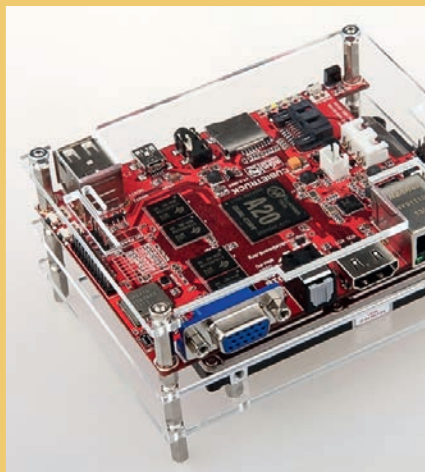
Подробнее на key.ru.

Товар сертифицирован. Количество товара ограничено.



Одноплатный компьютер Cubieboard3

Cubietruck



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Процессор: AllWinnerTech SOC A20, ARM Cortex-A7 Dual-Core
- Видео: ARM® Mali 400 MP2
- Выходы: HDMI, VGA 1080p
- Оперативная память: 2 Гб, DDR3, 480 МГц
- Встроенная память: 8 Гб NAND
- Внешние накопители: интерфейс SATA 2.0 для 2,5" HDD; возможно подключение 3,5" HDD с использованием внешнего питания 12 В
- Сеть: 10M/100M/1G Ethernet, Wi-Fi и Bluetooth
- Питание: DC 5 В, 2,5 А с подключенным HDD; возможно подключение

литиевого элемента питания

- Периферия: 2 × USB host, 1 × USB OTG, 1 × Toslink (SPDIF), 1 × IR, 4 × LEDs, 3,5" аудиовыход, 3 кнопки
- Поддерживаемые ОС: Android, Linux-based, BSD
- 54 контакта расширения I2S, I2C, SPI, CVBS, LRADC × 2, UART, PS2, PWM × 2, TS/CSI, IRDA, LINEIN & FMIN & MICIN, TVIN × 4
- Размер платы: 11 × 8 см

КОМПЛЕКТАЦИЯ

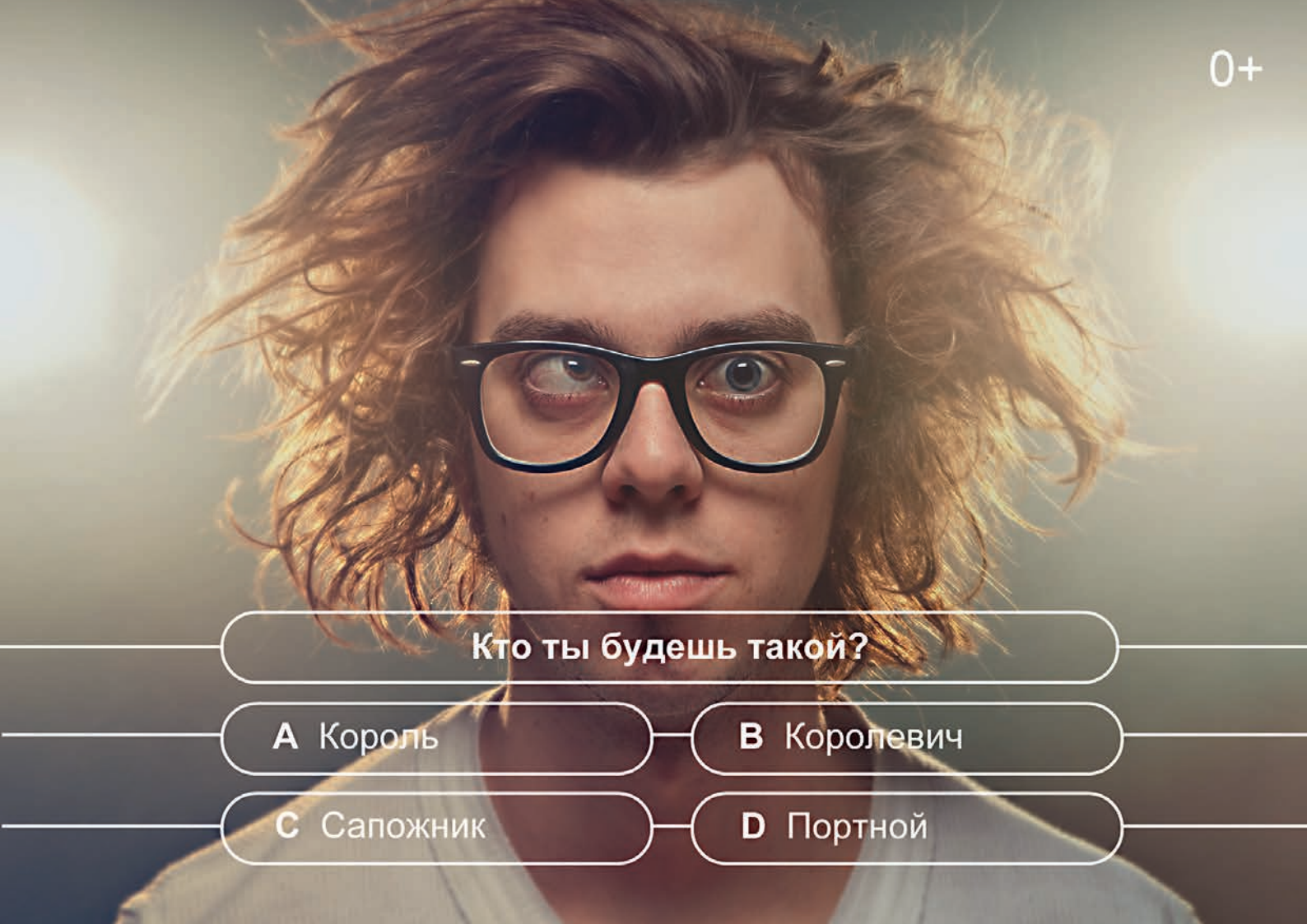
- Плата Cubietruck
- Прозрачный акриловый корпус
- Кабель питания USB
- Кабель MiniUSB
- Кабель SATA с питанием
- Радиатор для процессора

4395
руб.

www.linuxcenter.ru/shop/ops_hard/cubieboard/cubietruck/

«ГНУ/Линуксцентр»: +7 812 309-06-86 (Санкт-Петербург), +7 499 271-49-54 (Москва)





Кто ты будешь такой?

А Король

В Королевич

С Сапожник

Д Портной

5 причин пройти тест «Профориентация» от HeadHunter:

1. Вы узнаете о своих сильных и слабых сторонах, особенностях интеллекта.
2. Вы получите рекомендации к дальнейшему развитию.
3. Для вас будет сформирован список из 15 наиболее подходящих профессий со ссылкой на актуальные вакансии.
4. Вы можете пройти тест на сайте, не выходя из дома.
5. Вы можете прикрепить тест к резюме, чтобы работодатель видел сам, на что вы способны!

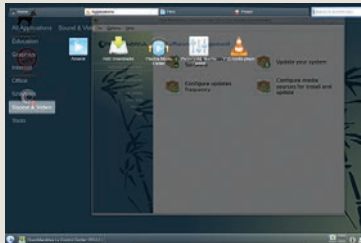
Содержание

И не только это не мешало бы хакнуть...

Обзоры

OpenMandriva 2014 14

Ответвление, восставшее из пепла Mandrake Linux, похоже, всерьез намеревается всех победить.



» OpenMandriva 2014: так что у нас со свежей версией?

OpenBSD 5.5 15

Вряд ли заманит новичков в Linux, но для продвинутых пользователей это мощная ОС, на которую можно положиться как на каменную стену.

Капо OS 16

Вы приобрели Raspberry Pi, чтобы выучиться программированию? Скорее хватитесь за эту альтернативу Raspbian!

PIR Alarm 17

Кто усторожит сторожа? Естественно, это практичное дополнение к вашему Raspberry Pi.

Intel NUC D54250 18

Этот бравый маленький компьютер с малопривлекательным названием произведен компанией Intel.



» Intel NUC D54250 невелик, но спроектирован идеально.

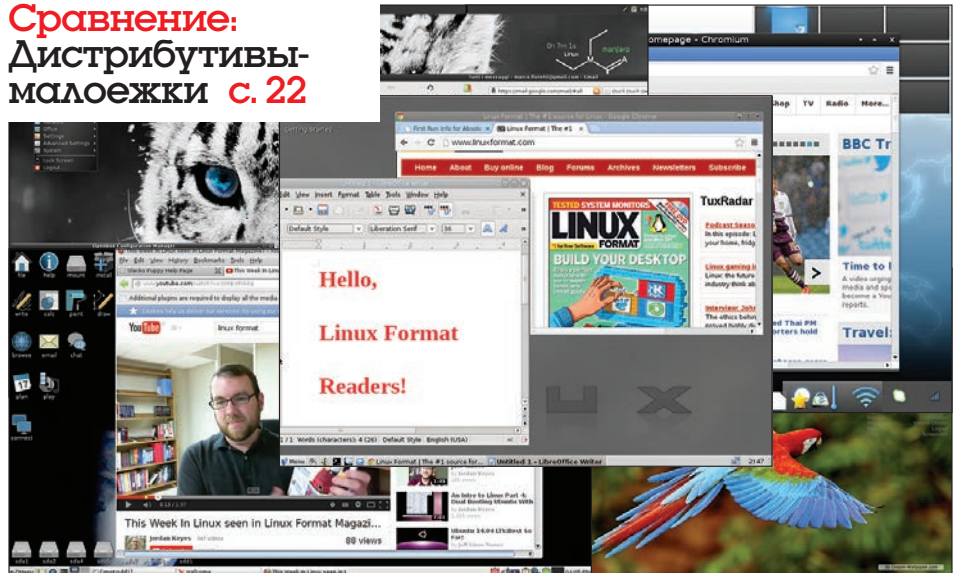
Dell Chromebook 19

Компания Dell взяла да и тоже выпустила ноутбук на ChromeOS: получился надежный и с симпатичной внешностью.

Недовольны последним релизом от Canonical? Переделайте его! с. 34

Хак Ubuntu

Сравнение: Дистрибутивы-малоежки с. 22



Люди говорят



« Следует изменить ИТ-стереотипы и сделать учебный план интереснее »

Билл Кинн об обучении и LPI с. 42

На вашем бесплатном DVD



Дистрибутивы-малоежки

» 5 лучших легковесов Linux

40 эффективных инструментов

» Практически все, что вам требуется

ПЛЮС: HotPicks и коды учебников **с. 106**

Ищите в этом номере

Экономичные программы 46

Извлеките все из своего 486-го процессора!

Управление пакетами 50

Мы приветствуем подход, принятый в Linux.

Спутник Интернета 54

Он висит над Европой, обеспечивая связь.

Эмулируем игры 56

Новая виртуальная жизнь игровых приставок.



Академия кодига

Python 84

Наш автор не в силах оторваться от *Minecraft*, и мы применили его энергию в мирных целях: помочь вам с OO-программированием.

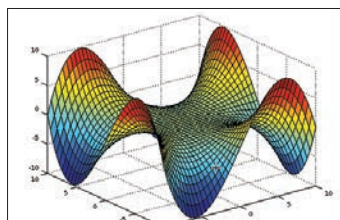
большой, чем аудитория «знатоков». Изучим рычаги управления вычислительной математикой и анализом.

Octave 88

Поместив всего один учебник по *Octave*, мы охватили диапазон

Erlang 92

Тестирование программ должно проводиться еще до их выпуска, в процессе разработки.



Учебники

Терминал

Основные программы 68

Руководство для начинающих по работе в терминале: какие команды надо знать в первую очередь.

Мониторинг системы

GKrellM 70

Этот полезный инструмент проследит за порядком в вашей системе. Причем его вид настраивается чуть ли не до каждого пикселя.



» **GKrellM:** на страже системы.

Шифрование жесткого диска

Dm-crypt 74

Truecrypt не лишен проблем — так что вот вам альтернативный вариант. *Dm-crypt* способен шифровать на уровне блоков!

Собираем облака

OpenStack 76

Автоматизируем сборку *OpenStack*, клонируем множество виртуальных машин из базовых образов.

Потоковое вещание

Mopidy 80

Превратим старый компьютер — а то и Raspberry Pi — в музыкальный сервер и заполним звуками весь дом.

Постоянные рубрики

Новости 6

Отечественную платформу — на общественных началах, борьба с анонимщиками и за импортозамещение, вышел новый CentOS, LinkedIn тайком пользуется Linux, профессор Таненбаум собрался на покой, Qualcomm ответила за базар, а Linux водит автомобили.

Новости Android 20

Sony озботилась физической сохранностью, мобильники ищут лекарства, Samsung открывает магазин, а очки Google норовят читать мысли.

Сравнение — и не одно 22, 28

Пять лучших дистрибутивов с малым потреблением ресурсов: Absolute Linux, Crunchbang, Elive, Manjaro, Pupy. ПЛЮС! На свете есть не только *Firefox* и *Chrome*: вот вам еще шесть web-браузеров, чтобы бороздить

просторы Интернета сообразно вашей матчасти. *Konqueror*, *Midori*, *Opera*, *Qupzilla*, *Rekonq*, *Seamonkey*.

Интервью LXF 42

Билл Куинн свидетельствует о бурном росте интереса к Linux.

Что за штука... GoboLinux 60

Дистрибутив с уникальной файловой системой, скрытой за привычным интерфейсом.

Рубрика сисадмина 62

Доктор исследует отходняк от печально прославившейся уязвимости Heartbleed и учит, как подружить Windows и Linux.

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик — про *I-Nex*, загрузку дистрибутивов с наших DVD,

параметры *Grub*, запуск программ MS-DOS, резервное копирование автоматом, *Wicd* и *Mint*.

HotPicks 100

Отвечайте горяченького! Лучшие в мире новинки свободного ПО: *Crossfire*, *GCC*, *http*, *Infolist*, *Micro Tetris*, *MongoDB*, *Nixos*, *QEMU*, *view3dscene*, *WordPress*, *Youtube-dl*.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Отличный номер был... Ну так закажите его, пока не поздно! Это же просто.

Через месяц 112

Покинем Windows, пригубим Wine, поищем лучший медиа-плеер... и многое другое.



ГЛАВНОЕ Отечественное ПО » Дела судебные » Конференции » Дистрибутивы
» Open Source Inside » Личности » Копирайт » Автомобили

ОТЕЧЕСТВЕННОЕ ПО

Стартует проект «Бета»

Компания «Диасофт» объявила о создании новой отечественной платформы.

Компания «Диасофт» — один из крупнейших российских разработчиков ПО — начинает новый проект «Бета», по созданию платформы для разработчиков на базе исключительно отечественных составляющих. Реализовать его предполагается вне рамок каких-либо государственных программ и не на бюджетные средства. По замыслу организаторов, работа может быть разбита на следующие локальные задачи:

- » создание ОС, работающей на серверах и рабочих станциях производства России;
- » выпуск средств виртуализации для нее;
- » создание СУБД с широкими возможностями горизонтального масштабирования;
- » создание отечественной среды разработки приложений и средств их отладки;
- » выпуск средств, автоматизирующих тестирование созданных приложений;
- » создание отечественного сервера приложений;
- » разработка средств балансировки нагрузки и других компонентов для создания кластеров серверов приложений;



» Проект по созданию отечественной платформы инициирован серьезным участником рынка.

- » создание средств проектирования, облегчающих разработку прикладных сервисов и программ;
- » разработка отечественных средств бизнес-аналитики и универсального генератора отчетов;
- » разработка средств моделирования и исполнения бизнес-процессов;
- » создание средств проектирования и создания пользовательских интерфейсов для работы на различных устройствах;
- » выпуск мобильной операционной системы, работающей на смартфонах и планшетах российских производителей;

- » разработка технологии установки/обновления создаваемых приложений;
- » разработка средств миграции для различных языков и технологий разработки, позволяющих перенести созданные с их использованием прикладные программные продукты на данную платформу, обеспечив тем самым наличие всех нужных заказчикам бизнес-приложений;
- » формирование методологического обеспечения процесса разработки, создание учебных курсов.

Спектр решаемых задач охватывает практически все предприятия ИТ-отрасли: проект «Бета» будет реализован методом массовой коллективной разработки в интересах всех отечественных компаний, ради создания реальной альтернативы применяемым в настоящее время импортным приложениям. Инициаторы проекта убеждены: в России есть всё необходимое для решения этой глобальной задачи — и квалифицированные разработчики, и наличие устойчивого спроса, и потребности государства. Они уверены, что объединив усилия, можно в разумные сроки создать первую версию отечественной платформы.

Сама компания «Диасофт» выделяет своё подразделение, занимающееся разработкой платформы Diasoft Framework, в отдельную независимую компанию — «Диасофт Платформа». Число присоединившихся к проекту пока неизвестно, но для пессимизма особых оснований нет.

ДИСТРИБУТИВЫ

Новый релиз CentOS

Опубликована новая версия клона Red Hat.

Вышла очередная версия дистрибутива CentOS 7.0, основанного на пакетной базе Red Hat Enterprise Linux 7.0. С официального сайта можно загрузить следующие сборки: DVD на 4 ГБ, NetInstall на 370 МБ, образ для USB Flash на 7 ГБ и Live-сборки с Gnome и KDE. Дистрибутив CentOS 7 доступен только для архитектуры x86-64.

Этот выпуск — первый, который подготовлен уже после перехода проекта под опеку компании Red Hat. То есть теперь его с полным основанием можно считать официальной бесплатной версией корпоративного дистрибутива.

Из Red Hat 7 унаследованы и основные изменения: полный переход на 64-разрядную архитектуру, использование XFS



» Live-сборки системы CentOS с рабочим столом Gnome.

как файловой системы по умолчанию, задействование системного менеджера *systemd* и применение Gnome 3 в качестве основного рабочего стола.

КОНФЕРЕНЦИИ

Переходим от слов к делу?

В Москве прошла конференция, посвящённая вопросам импортозамещения.

В Москве 25 июня состоялась организованная изданием PC Week/RE конференция «Стратегическая независимость государственных и корпоративных информационных систем. Импортозамещение в сфере ИКТ». В конференции приняли участие представители государства, ИКТ-компаний и отраслевых ассоциаций.

Главной идеей мероприятия стала необходимость перейти от слов к реальным делам. Поэтому участники не столько рассказывали о своих достижениях (хотя какая конференция обходится без доли саморекламы), сколько предлагали конкретные мероприятия и шаги, направленные на достижение информационной независимости нашей страны.

Большинство участников конференции сошлись во мнении, что конкурентоспособный продукт может быть создан только в рыночных условиях. Однако с учётом заметного отставания России от стран Запада разработка ПО с нуля представляется нереальной.

Таким образом, наиболее перспективный и прагматичный путь — это использование модели свободного ПО, когда продукт в основном разрабатывается сообществом, а российские компании только поддерживают его и доводят до требуемой конкретным заказчиком функциональности.

Что делать? Варианты.

1. **Использовать готовое зарубежное ПО.** 
2. **Создавать отечественное ПО «с нуля» полностью силами российских специалистов.** 
3. **Создавать отечественное ПО на основе открытых и свободных международных компонентов (open-source).** 

Президент отраслевой ассоциации РУССОФТ рассказал, что базовая система понятий уже создана усилиями трёх организаций — РУССОФТ, АРПП «Отечественный софт» и РАСПО. В согласованном документе даются точные и недвусмысленные определения российского производителя ПО и отечественной программной платформы, и предложены методы реализации эффективного импортозамещения. Однако внедрением никто не занимался, и страна уже отстала на четыре года.

Говоря о свободном ПО в контексте импортозамещения, президент и генеральный конструктор компании ROSA Владимир Рубанов подчеркнул, что технологическая независимость отнюдь

» С докладом на конференции выступил представитель РАСПО Владимир Рубанов.

не означает изоляции отрасли от мирового сообщества. Он считает, что следует обязательно создавать международную кооперацию посредством участия отечественных разработчиков в крупных международных проектах, но при этом постоянно увеличивать долю присутствия в них.

По его мнению, процесс импортозамещения надо проводить «без резких движений», сосредоточившись на критически важных областях, а в остальных использовать то, что есть, постепенно заменяя импортные приложения на отечественные. Такой подход позволит сделать переход максимально безболезненным для пользователей.

Разумеется, не все участники конференции согласились с тем, что надо обязательно придерживаться рыночного подхода. Так, председатель правления АНО «Центр компетенции по электронному правительству» Владимир Дрожжинов не видит ничего страшного в государственном протекционизме. Он считает, что для ускорения перехода целесообразно отказаться от рыночных условий, а поставить перед отраслью чёткие цели и создать условия для привлечения к работе команды с должной компетентностью. Когда цель будет достигнута, тогда можно постепенно переводить компании на рыночные рельсы.

ДЕЛА СУДЕБНЫЕ

Угроза анонимным сетям

Вчинен иск против разработчиков сети Tor.

Окружной суд Техаса будет рассматривать иск частного лица к проекту Tor. Суть дела такова: на сайте Pinkmeth, доступ к которому предоставляется через анонимную сеть, публикуются фотографии девушек в стиле «ню», причём без их ведома. Снимки могут быть получены самыми различными способами, в том числе посредством взлома частных компьютеров, на которых хранятся слишком откровенные фото хозяйки.

Пострадавшая считает, что должна получить в качестве компенсации один миллион долларов. Причём часть этих средств

должны выплатить разработчики сети Tor, поскольку именно они виновны в создании инструмента, позволяющего недобросовестным владельцам сайтам нарушать закон.

Впрочем, эксперты из правозащитной организации Electronic Frontier Foundation утверждают, что иск не имеет перспективы быть удовлетворённым. В соответствии с законодательством, никакие провайдеры не могут рассматриваться в качестве непосредственных распространителей информации, предоставляемой другими провайдерами.



» Анонимной сети Tor угрожает крупный штраф.

OPEN SOURCE INSIDE

Популярность СПО растет

Социальная сеть LinkedIn — разработчик свободного ПО.

Социальная сеть LinkedIn больше известна как место для поиска и установления деловых контактов. Или, если менее официально, как площадка для направленного на потенциального работодателя саморекламы. Однако у этой профессиональной сети есть и другие достижения, не столь известные широкой публике.

«Компания LinkedIn широко использует свободные решения.»

В своём недавнем интервью сайту [TheNextWeb.Com](#) вице-президент LinkedIn по технологиям Алекс Воути [Alex Vauthey] рассказал о том, что компания не только широко использует свободные решения, но и в меру своих сил способствует их развитию. На сегодняшний день фирма представила сообществу более двадцати собственных проектов.

В среде ИТ-специалистов получила известность *Voldemort*, распределённая

система хранения. Она работает по принципу «ключ–значение», поддерживает линейное масштабирование и хорошо приспособлена к большим нагрузкам.

Еще одно достаточно популярное решение — фреймворк кластеризации *Helix*. Его применяют при развёртывании больших распределённых систем при необходимости сохранения баланса нагрузки на значительное число узлов кластера при их линейном добавлении. Особенность решения в том, что оно полностью автома-

тизировано и может самостоятельно регулировать нагрузку при аварийном или плановом отключении какого-либо узла.

Наконец, наиболее известное свободное решение, развиваемое командой LinkedIn — механизм публикации и подписки *Apache Kafka*. В настоящее время этот проект имеет наивысший приоритет в Apache Software Foundation и используется в Twitter, Netflix, Spotify, Pinterest, Tumblr и Airbnb.

Apache Kafka
A high-throughput distributed messaging system.

download introduction uses documentation performance clients ecosystem faq project
 • wiki
 • bugs
 • mailing lists
 • committers
 • powered by
 • papers & talks
 developers
 • code
 • projects
 • contributing
 • coding guide
 • unit tests

Apache Kafka is publish-subscribe messaging rethought as a distributed commit log.

Fast
A single Kafka broker can handle hundreds of megabytes of reads and writes per second from thousands of clients.

Scalable
Kafka is designed to allow a single cluster to serve as the central data backbone for a large organization. It can be elastically and transparently expanded without downtime. Data streams are partitioned and spread over multiple machines to allow data streams larger than the capability of any single machine and to allow clusters of co-ordinated consumers.

Durable
Messages are persisted on disk and replicated within the cluster to prevent data loss. Each broker can handle terabytes of messages without performance impact.

Distributed by Design
Kafka has a modern cluster-centric design that offers strong durability and fault-tolerance guarantees.

► *Apache Kafka* — один из свободных проектов, развиваемых LinkedIn.

Разумеется, оно применяется и в самом LinkedIn. На его основе реализована система, используемая для анализа данных из хранилища записей. Та самая, при помощи которой пользователи сети получают всевозможные рекомендации типа «Люди, которых Вы можете знать».

ЛИЧНОСТИ

Создатель MINIX идет на пенсию

Но Эндрю Таненбаум продолжит чтение лекций.

Роль профессора Амстердамского университета Эндрю Таненбаума [Andrew Tanenbaum] в появлении Linux сложно переоценить. Именно он разработал ОС MINIX, которая вдохновила Линуса Торвальдса на создание свободной ОС, сделавшей революцию в мировом ИТ.

Разумеется, у профессора Таненбаума есть множество других заслуг, о которых сейчас самое время вспомнить:

- » его труды переведены на 21 язык и заслуженно считаются базовыми учебниками по операционным системам и сетям;
- » он имеет звание доктора философии;
- » он является членом Нидерландской королевской академии наук;
- » на протяжении 12 лет он был деканом исследовательской школы ASCI;
- » более 20 его учеников получили докторскую степень.



► Создатель MINIX Эндрю Таненбаум решил уйти на отдых.

И вот спустя 43 года работы в университете он заявил об окончании активной преподавательской деятельности — 23 октября 2014 года состоится его последняя лекция. Она будет открытой для всех желающих. На сайте университета открыта специальная страничка, где все могут почитать о достижениях этого человека и зарегистрироваться на его последнюю лекцию.

Впрочем, слово «последняя» в данном случае применяется очень условно. Заслуженный пенсионер вовсе не собирается прекращать активную деятельность: в его планы входит написание книг и статей, а также выступление с лекциями в разных странах мира. Точного списка этих стран пока нет, но хочется надеяться, что профессор Таненбаум найдёт время посетить и Россию.

Добро пожаловать в робототехнику!

ScratchDuino

Электронный комплекс на основе свободного аппаратного обеспечения для школ и вузов

ScratchDuino.Лаборатория

Плата расширения с датчиками ввода-вывода — как установленными на плате, так и подключаемыми. Предназначена для программирования взаимодействия компьютера с внешними устройствами.

Полная интеграция («из коробки») со средой программирования Scratch, предназначенной для детей.

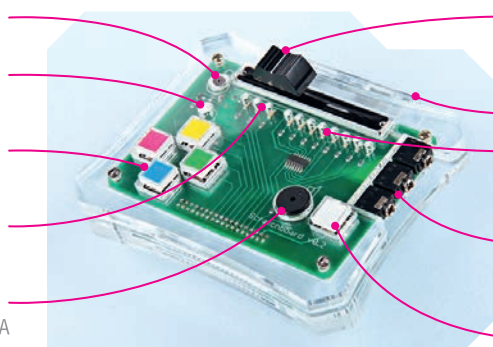
ДАТЧИК ЗВУКА*

ДАТЧИК ОСВЕЩЕННОСТИ*

КРЕСТОВИНА КНОПОК

ЦВЕТНЫЕ СВЕТОДИОДЫ

ДИНАМИК ДЛЯ ВЫВОДА ЗВУКА



ПЕРЕМЕННЫЙ РЕЗИСТОР (ПОЛЗУНОК)*

КАРТРИДЖ ARDUINO

СВЕТОДИОДЫ, УПРАВЛЯЕМЫЕ ЧЕРЕЗ РАЗРЯДНУЮ МАТРИЦУ

РАЗЪЕМЫ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ

КНОПКА*

* РАБОТАЮТ В РЕЖИМЕ ЭМУЛЯЦИИ RISCBOARD, НЕ ТРЕБУЯ ДОПОЛНИТЕЛЬНОГО ПРОГРАММИРОВАНИЯ.

ЦЕНА **5000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

ScratchDuino.Робоплатформа

Внешний робот-исполнитель, управляемый из среды программирования Scratch. Не требует навыков программирования на языках высокого уровня и может применяться для обучения, начиная с младших классов.

ScratchDuino.Робоплатформа

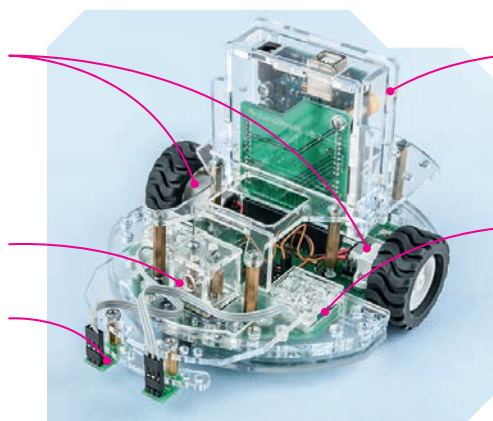
обеспечивает возможность:

- подключения внешних датчиков без механических соединений и пайки;
- расширения через последовательную шину RoboBus;
- установки деталей Lego Technics.

МОТОР-РЕДУКТОРЫ

ФОТОДАТЧИК

ДАТЧИК ЛИНИИ



КАРТРИДЖ ARDUINO

МАГНИТНЫЙ ЦИФРО-АНАЛОГОВЫЙ РАЗЪЕМ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ БЕЗ МЕХАНИЧЕСКИХ СОЕДИНЕНИЙ И ПАЙКИ*

* ВСЕГО НА РОБОПЛАТФОРМЕ 5 РАЗЪЕМОВ. ВОЗМОЖНО ПОДКЛЮЧЕНИЕ ДАТЧИКОВ КАСАНИЙ, ИК-ДАТЧИКОВ РАССТОЯНИЙ/ПРЕПЯТСТВИЙ И ДР.

ЦЕНА **10 000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

Назначение ScratchDuino

- Изучение взаимодействия компьютера с внешней средой
- Изучение изменения параметров внешней среды
- Изучение процессов передачи информации и принципов ее построения
- Изучение внешних устройств управления
- Моделирование устройств

Среда разработки

Lazarus (язык Pascal) ■ Scratch ■ Arduino IDE

ScratchDuino адаптирован для учебных заведений. Поставляется с комплектами учебно-методических материалов.

Продукт разработан при финансовой поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере

WWW.SCRATCHDUINO.RU

(812) 309-0686 (Санкт-Петербург) ■ (499) 271-4954 (Москва)

НА КОПИРАЙТНОМ ФРОНТЕ

Серьезная победа

Компания Qualcomm погорячилась с иском.

Копирайтные войны не думают затихать. В гуманитарных областях (кино, книги, музыка) пока побеждают сторонники жесткого контроля за соблюдением прав (их уже успели окрестить правоторговцами), но в сфере ИТ ситуация иная.

Qualcomm потребовала закрыть на GitHub 116 репозиторий, где якобы были объекты, нарушающие закон США об авторском праве на интеллектуальную собственность в цифровую эпоху (DMCA) — прежде всего репозиторий CyanogenMod и официальный репозиторий компании Sony Mobile для разработчиков альтернативных прошивок, но «под раздачу» попала и масса мелких проектов по Android. Казалось, репозиториям не жить. Но GitHub повёл себя грамотно: требования формально выполнил и нанес ответный удар.

У разместивших спорные файлы пользователей запросили комментарии, гарантировав снятие блокировки при удалении этих файлов; им также предложили самим решать вопрос с правообладателем, если

его доводы неубедительны. Параллельно GitHub провёл своё расследование и выяснил странное: в чёрный список попал даже репозиторий самой Qualcomm, не говоря уж о файлах, принадлежащих Linux Foundation. Т.е. проверка правообладателем была поверхностной, и выполнение его требования нанесло урон добросовестным пользователям. На что, разумеется, GitHub пойти не может.

В итоге Qualcomm, вынужденная признать, что часть файлов из блок-списка ей не принадлежит, отозвала запрос и извинилась за поспешность. По прочим файлам будет новое расследование, и по его итогам начнут работу с отдельными проектами, без блокировки репозиторий.

Мораль ясна. Тщательная проверка крупных систем на нарушения — вещь дорогая, и шансов на быстрый успех нет. Если атакуемый не испугается, а будет действовать по закону, скорее всего придётся извиняться. Такие процессы на руку лишь патентным троллям, а для серьёзных компаний велик риск замарать репутацию.

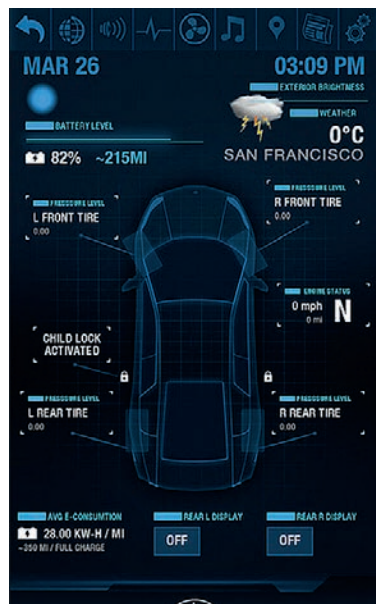
АВТОМОБИЛИ

Платформа для авто

Linux Foundation продемонстрировал систему Automotive Grade Linux.

Вышла первая версия платформы Automotive Grade Linux, которая включает всевозможные открытые компоненты, пригодные для применения в автомобильных подсистемах. Авторы подчёркивают, что AGL не является законченным решением, готовым для использования — это, скорее, некая эталонная система, требующая адаптации под конкретное оборудование.

AGL представляет собой дистрибутив Linux, использующий для взаимодействия с пользователем решение Tizen и содержащий многочисленные программы, созданные для автомобиля. Разработка платформ — прекрасный пример кооперации компаний из различных отраслей. Со стороны автопрома в проекте участвуют Nissan, Toyota, Hyundai и др., а ИТ-сегмент представляют Intel, Samsung, Fujitsu, NEC, NVIDIA и TI. **LXF**



Automotive Grade Linux — платформа для автомобильных подсистем.

Новости короткой строкой

» Вышла первая альфа-версия рабочего стола Moonlight, разрабатываемого на Qt5, для установки на устаревшие и слабые компьютеры. Источник: plus.google.com/u/0/

» Компания Oculus VR принялась за кроссплатформенный сетевой движок для игр RakNet под лицензией BSD, с публикацией кода на GitHub. Источник: www.oculusvr.com

» Организация Software Freedom Conservancy объявила о выходе системы управления репозиториями Kallithea, поддерживающей решения Git и Mercurial. Источник: sfconservancy.org/news/

» Ядру Linux 3.1 придан статус ветки с длительным сроком поддержки, обновления для которой будут выпускаться до августа 2016 года. Источник: lkml.org/lkml/2014/7/3/692

» Сообщество xda-developers представило проект по разработке СПО, способного полностью заменить проприетарные компоненты Google в системе Android. Источник: forum.xda-developers.com

» Разработчики системы Fedora сообщили, что переход дистрибутива на использование по умолчанию окружения Wayland произойдёт в лучшем случае не раньше выпуска Fedora 23. Источник: blogs.gnome.org/uraxus

» Разработчики Chrome OS сообщили о планах развития этой системы, которые предусматривают возможность запуска в ней приложений для Android. Источник: www.omgchrome.com

» Верховный суд США признал юридическую ничтожность патентования абстрактных идей, оформленных в виде инструкции для компьютера. Источник: www.fsf.org/news/

» Началось публичное тестирование кандидата в релизы очередной версии LibreOffice 4.3. Источник: blog.documentfoundation.org/

» Вместо удобной, но некрасивой темы Raleigh в GTK+ будет использоваться Adwaita. Источник: blogs.gnome.org-for-gtk

» Microsoft выпустила очередную версию Skype для Linux — результат годовой работы программистов. Источник: blogs.skype.com/2014

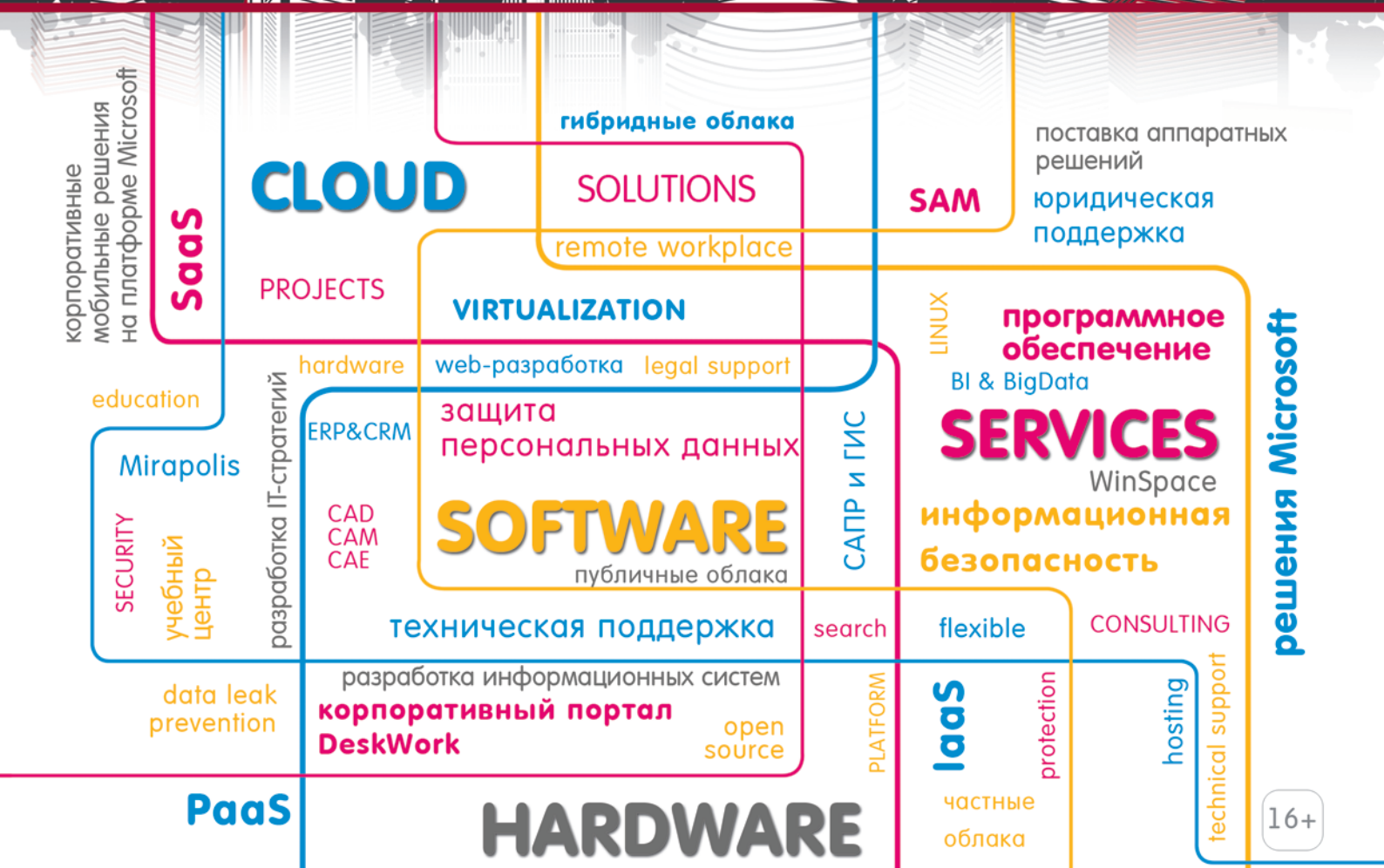
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2014

LINUX
FORMAT
Главное в мире Linux



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



Стоимость подписки Годовая — 2280 руб., полугодовая — 1230 руб. без учета стоимости доставки.
Адреса и телефоны редакции Санкт-Петербург, пр. Медиков, 5, корп. 7, тел. (812) 309-06-86.
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

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

shop.linuxformat.ru

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Systemd OS?

Как и следовало ожидать, *systemd*, возникнув просто как схема инициализации, нацеленная на распараллеливание запуска стартовых сервисов, разрастается, вбирая всё новые службы, вплоть до DHCP-сервера, а вскоре и DNS. По словам автора, она превращается в набор блоков для построения операционной системы. Похоже, что недалёк день, когда будет объявлено о создании Systemd OS, которая интегрирует в себя и графическую систему (давно обещанный *Wayland*), и рабочие среды типа Gnome и KDE. А заодно поглотит все основные дистрибутивы Linux'a, вытеснив оставшиеся, такие, как Slackware, вместе с BSD-системами, на обочину. И наступит эпоха всеобщей унификации и благоденствия — думают, что это повысит конкурентоспособность Linux как ОС общего назначения.

Однако — конкурентоспособность по отношению к кому? Ведь унифицированная система, обещающая благоденствие, существует давно, и называется она Windows. Имеется, в лице Mac OS, и система, интегрирующая в себе всё на свете (по крайней мере, такой она видится пользователю). А конкурентоспособность Linux'a на этом фоне как раз и была до сих пор обусловлена разнообразием его дистрибутивов и их «конструкторским» характером, позволяющим индивидуализировать систему практически безгранично. Утратив разнообразие и «конструктивизм», Linux рискует потерять существенную часть своих нынешних пользователей. А вот приобретёт ли новых, в лице любителей унификации и интеграции?
alv@posix.ru

Сегодня мы рассматриваем:

OpenMandriva 2014 14
Mandriva в настоящее время — это дистрибутив, упорно карабкающийся вверх. Ну и как себя чувствует свежая версия?

OpenBSD 5.5 15
Мы протестировали дистрибутив, предназначенный отнюдь не для новичков, но немного времени и терпения превратят его в превосходную — и супер-надёжную — ОС.

Kano OS 16
Забудьте Raspbian: для Raspberry Pi завелся новый дистрибутив, сосредоточенный на образовании.

PIR Alarm 17
Комплект для охраны вашего дома, а заодно и проект для вашего Raspberry Pi.

Intel NUC D54250 18
Приводит ли в восхищение последний из миниатюрных ПК от Intel? Запускаем и проверяем.

Dell Chromebook 19
Dell вступает на арену Chromebook'ов — вот и разберемся, стоила ли игра свеч.



› Intel NUC D54250: Ящик, наполненный лучшим от Intel. Нам понравился.



› Mandrake Linux — далекое прошлое; OpenMandriva — сила, с которой следует считаться.



› Рассматриваем операционную систему для Raspberry Pi, предназначенную деткам.

Сравнение: Дистрибутивы-малоежки

с. 22

Absolute Linux



Crunchbang



Elive



Manjaro



Puppy



Даже если ваше оборудование подвешало, вам незачем страдать по новейшим дистрибутивам с передовыми функциями. Мы протестировали 5 лучших дистрибутивов, нетребовательных к ресурсам.

OpenMandriva 2014

В очередной раз наткнувшись на еще один дистрибутив от Mandriva, Маянк Шарма мучается вопросом: а ну как Mandriva — это новый Ubuntu?

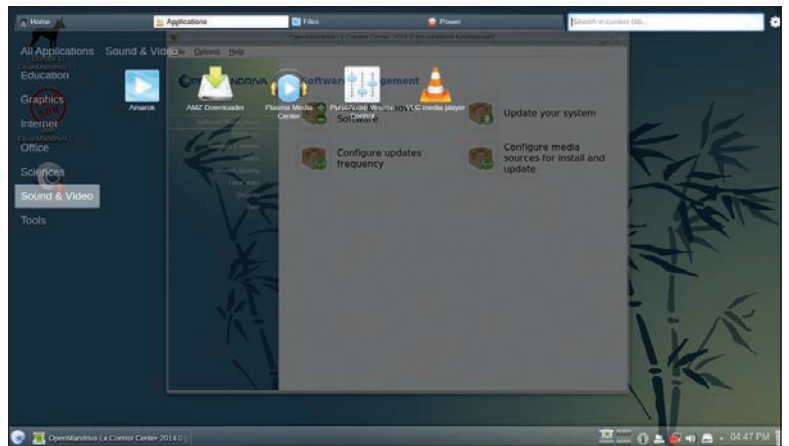
Вкратце

» Ответвление Mandriva, ориентированное только на рабочий стол KDE. См. также Mageia, PCLinuxOS.

С тоит вам подумать, что Mandriva пришел конец, как он всплывает снова. Урожденный Mandrake Linux разветвлялся так часто, что не упоминать Mageia и его некоммерческая организация казались логическим следствием для проекта, монетизация которого не удалась ни одной корпорации. Но сейчас есть дистрибутив OpenMandriva от разработчиков, трудившихся над ROSA — российским ответвлением проекта Mandriva под згидой некоммерческой организации OpenMandriva Association (Франция).

Как дистрибутив, Mageia — трудный пример для подражания, а у OpenMandriva 2014.0 лишь второй релиз. Но, несмотря на их общее происхождение, разница уже видна. Во-первых, OpenMandriva имеет рабочий стол только KDE, а всеохватывающая Mageia поддерживает все популярные (и не очень) рабочие среды. Добавим, что OpenMandriva основан на рабочем столе ROSA и поэтому унаследовал инструменты юного российского дистрибутива.

В версии использован свежий релиз KDE 4.12.4. Рабочий стол глубоко настраивается и по умолчанию использует программу для запуска полноэкранных приложений *Homerun*. Другое добавление KDE — *Plasma Media Center*, способный проигрывать локальные медиа-файлы, а также выборочный контент с онлайн-источников вроде YouTube и Flickr. Медиаплеер по умолчанию — *VLC*. Вы также найдете здесь популярные приложения KDE — *Krita*, *Kamoso*, *Kopete*, *Okular*, *Amarok*



» OpenMandriva применяет переделанный рабочий стол KDE с загрузчиком *Homerun*.

и *Kmail*, почтовый клиент по умолчанию, наряду с *Firefox 29* и *LibreOffice 4.2.3.3*.

За кулисами, это первый релиз OpenMandriva Lx с поддержкой загрузки EFI — правда, не на всех оснащенных UEFI машинах. Другая важная скрытая функция дистрибутива — специализированное ядро с улучшенной производительностью, включающее заплатки от Кона Коливаса [Con Kolivas], *UKSM* для лучшего управления памятью и *TuxOnIce* для лучшего сервиса приостановки работы компьютера и перехода в спящий режим. Даже live-образы загружаются быстрее, благодаря функции ядра *squashfs*, позволяющей полностью задействовать CPU.

Шаги прогресса

Инсталляция — в стандартном режиме Mandriva, хотя по окончании можно удалить неиспользуемую поддержку «железа» и локализации пакетов. Данная опция также доступна из свежевыпущенного приложения загрузки *Welcome* — это одна из лучших новых функций дистрибутива, и кроме ознакомления пользователей с дистрибутивом, *Welcome* предлагает ссылки на общие задачи администратора, такие как обновление, установка нового ПО и конфигурация рабочего стола.

В дистрибутиве изначально нет проприетарных кодеков, но их можно установить из приложения *Welcome* одним щелчком. В приложении также есть разделенные на категории ссылки для установки популярных приложений, таких как *Skype*, *Dropbox*, игровой клиент *Steam*,

Calligra Office Suite, пакет читалки *Calibre*, *Eclipse*, *GIMP*, *VirtualBox* и многое другое.

При общем лоске, кое-где нужна доработка. Скажем, устаревший брандмауэр *Drakfirewall* от Mandriva заменили новой утилитой *firewall-config*, на базе *firewall* (займ у проекта Fedora) — но это может отпугнуть от работы в группе: по умолчанию заложен максимум безопасности, и блокируется совместное использование *Samba* и *NES*. Пока не работает знаменитый родительский контроль Mandriva.

Сейчас дистрибутив способен удовлетворить поклонников дистрибутива ROSA, но не привлечет народ с других популярных веток Mandriva. Техническая основа здравая, но проекту нужна некая уникальная рыночная фишка для среднего пользователя настольной ОС. **LXF**

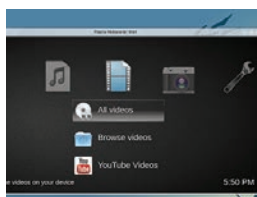


Свойства новскидку



Мастер Welcome

Знакомит с дистрибутивом, предоставляя ссылки на такие задачи, как настройка системы и установка приложений.



Plasma Media Centre

Это новое приложение умеет проигрывать локально хранящиеся медиа-файлы, а также видео и фото с популярных онлайн-сервисов.

LINUX FORMAT Вердикт

OpenMandriva Lx 2014.0

Разработчик: OpenMandriva Association
Сайт: www.openmandriva.org
Лицензия: Различные

Функциональность	8/10
Производительность	9/10
Удобство использования	8/10
Документация	5/10

» Быстрый, улучшенный KDE-дистрибутив, не лишенный достижений, но и не без шероховатостей.

Рейтинг **7/10**

OpenBSD 5.5

Маянк Шарма сходил налево ради несокрушимо безопасного дистрибутива BSD.

Вкратце

» Операционная система BSD с упором на безопасность, массово используемая как одиночный сервер, а для любителей — и как настольная. См. также: FreeBSD, NetBSD.

OpenBSD сделала себе имя одной из самых надежных ОС. Во всем мире для брендмауэров, систем обнаружения вторжений и даже серверов общего назначения многие обеспечивающие безопасность установки полагаются на строгий аудит кода OpenBSD и модель разработки, где защита прежде всего. OpenBSD — не ОС общего назначения; ее нельзя оценивать по тем же параметрам, что и среднестатистический настольный дистрибутив. OpenBSD — для опытных пользователей, знающих, с чем они имеют дело, и готовых справиться с ее причудами ради достойной работы.

Одна из таких причуд OpenBSD — текстовый инсталлятор. Но он не должен доставить особо много неприятностей опытным пользователям и по умолчанию предлагает разумные настройки. Чтобы справиться с ним, за плечами нужно иметь несколько установок Linux/BSD, но подсказки достаточно подробны. Новинка данного релиза OpenBSD 5.5 — добавление метода автоустановки с поддержкой сценариев для автоматической установки и обновлений. Это особенно полезно, если вы хотите развернуть или обновить несколько одинаковых машин. Другое улучшение инсталляции — теперь ISO-образы для i386 и x86-64 платформ можно писать на USB-диски.

По части платформ OpenBSD ввела поддержку с дюжины архитектур, в т.ч. Alpha от HP, системы Aviiion от Data General и ряда систем на ARMv7. В «железе» тоже уйма усовершенствований: новые драйверы, улучшающие виртуальную поддержку оборудования, новый беспроводной

драйвер для различных чипов Wireless-N для Intel Centrino, новый драйвер сенсорной площадки, поддерживающий мультитачтрекпады Broadcom, появившиеся на новых MacBook, MacBook Pro и ноутбуках MacBook Air от Apple. Также это первый релиз OpenBSD с поддержкой графических карт AMD Radeon, благодаря подключению кода установки режима ядра (KMS) из ядра FreeBSD.

После инсталляции вы получите «скелет» системы с некоторым количеством служб в вашем распоряжении и минимальный рабочий стол на базе *FVWM*, если вы выбрали его в процессе установки. Эта основа предоставляет все инструменты, необходимые, чтобы превратить ее в спецсервер или полноценную настольную ОС.

Защита нового уровня

Вдобавок ОС расширяет свои действия по безопасности криптографической подписью своих пакетов, позволяя верифицировать их до установки. Для этого в OpenBSD 5.5 дебютировала утилита *Signify*. ОС можно установить и без верификации пакетов, но, как легко представить, разработчики не одобряют такую практику.

Кроме того, модифицирована утилита *pkg_add* для инсталляции и обновления прекомпилированных пакетов с ПО: теперь она доверяет только подписанным пакетам. Подписание более чем 8000 пакетов для поддерживаемых архитектур было геркулесовой задачей! Точно так же разработчики проверили все дерево исходного кода, включая ядро, библиотеки и пользовательские приложения, чтобы обеспечить поддержку новой 64-битной переменной *time_t*. Использование 64-битной *time_t* на всех платформах сняло для OpenBSD проблему 2038 года [«сброс в 1901 г.» UNIX из-за переполнения разрядности системной переменной отсчета времени, — прим. ред.], введя новый диапазон дат, занимающий примерно 292 миллиардов лет, начиная с сегодняшнего дня. За эти новые возможности релиз OpenBSD 5.5 получил гимн «Свернуть время [Wrap in Time]».

Ошибка Heartbleed послужила разработчикам OpenBSD поводом ответить с от популярной библиотеки *OpenSSL*. Главная цель нового проекта *LibreSSL* —



» Репозиторий OpenBSD полон популярных рабочих столов, таких как *Gnome*, *KDE* и *Xfce*, и приложений типа *LibreOffice*, *Firefox* и *Thunderbird*.

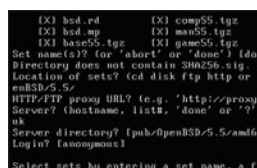
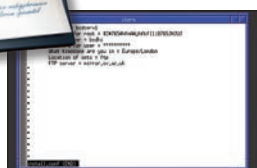
создать версию библиотеки, заслуживающую большего доверия, чем оригинал, коль скоро разработчик, неосторожно внесший ошибку Heartbleed, заявил, что у библиотеки *OpenSSL* не хватает глаз для отслеживания трудных скрытых ошибок.

Цель проекта *LibreSSL* — обеспечить полную замену *OpenSSL* «на ходу»: большую ее часть планируют переписать и проверить код *OpenSSL* на потенциальные уязвимости. Первоначально *LibreSSL* будет доступен только для OpenBSD. В настоящий момент планируется включение первой версии в релиз OpenBSD 5.6, который должен выйти где-то в ноябре.

Итак, OpenBSD 5.5 продолжает укреплять защиту. Но мы не рекомендуем ее настольным пользователям, даже тем, кто сумеет использовать ее таким образом. Но если кому нужна надежная серверная ОС, то OpenBSD — весьма заманчивый вариант. **LXF**



Свойства навскидку



Автоустановка

Скриптовый метод автоустановки допускает установку и обновления без вмешательства человека, с помощью файла ответов.

Безопасная установка

Релизы и пакеты имеют криптографические подписи, и установщик может верифицировать их перед инсталляцией.

LINUX FORMAT Вердикт

OpenBSD 5.5

Разработчик: The OpenBSD Foundation
Сайт: www.openbsd.org
Лицензия: BSD

Функциональность	8/10
Производительность	8/10
Простота использования	5/10
Документация	7/10

» Доработанная версия с новыми функциями для дальнейшего улучшения безопасной ОС.

Рейтинг 7/10

Kano OS

Лес Паундер испытывает стильную ОС для Raspberry Pi, разработанную для облегчения новичкам — особенно детям — знакомства с устройством.

Вкратце

» Новая ОС для Raspberry Pi, стремящаяся облегчить обучение детей эмпирическим путем. Основана на Raspbian, самом популярном дистрибутиве для Raspberry Pi.

С момента выхода Pi в 2012 году, Raspbian де-факто является стандартом дистрибутива Raspberry Pi. Но, как и у многих дистрибутивов Linux, его кривая обучения крутовата для новых пользователей, особенно для тех, кто, в общем, мало знаком с компьютерами. Здесь-то и выходит на сцену ОС Kano. Построенная на прочной базе Raspbian, Kano OS является частью набора, который был популярным проектом на Kickstarter в конце 2013 г. В проекте есть все необходимое для построения компьютерной лаборатории в классе, включая комплект документов и более модную клавиатуру.

В ОС Windows или Mac, Kano OS можно установить на 4-ГБ SD-карту, используя удобное приложение *Burner*; для Linux придется пока придерживаться нашего старого друга *dd*. При тестировании мы выяснили, что карта Class 10 показывает лучшую скорость.

При первой загрузке ОС представляет экран, похожий на терминал. Сначала для создания учетной записи пользователя назовите Kano свое имя. Затем вы переходите к простой игре-угадашке, по типу «Матрицы». После некоторой дальнейшей настройки и изучения, как обезвредить бомбу, вы попадаете на рабочий стол.

Первое впечатление, что рабочий стол очень приятный и отзывчивый — это большое достижение для тех, кто полагает, что Raspbian в своем нынешнем виде все еще малость тормозит. (Команда Kano



» Рабочий стол Kano по умолчанию яркий, красочный и простой в использовании, и позволяет детям экспериментировать и сосредоточиться на обучении.

обещает, что улучшения в установленном порядке будут направляться в Raspbian). Среди прочих улучшений — перекомпилированное ядро и урезанная система, представляющая только необходимое. ПО устанавливается обычным методом *apt-get*, но Kano идет с собственным менеджером обновлений, который вовремя освежает все установленные приложения.

Начинка из приложений

Kano поставляется с тем же пакетом приложений, что и Raspbian, но они тонко настроены на лучшую работу с Kano. Например, Sonic Pi (см. LXF185, стр. 17) адаптирован под дизайн ОС. Примечательное дополнение к предустановленному ПО — *Minecraft*, наряду со Scratch-подобным визуальным языком программирования, где каждый может создать новое содержимое мира *Minecraft*. В новое ПО, предоставленное командой, входит *Kano-Settings* — инструмент управления конфигурацией вашего Raspberry Pi, включающий разгон процессора.

Kano OS делает Raspberry Pi более дружелюбным в использовании, добавляя гладкий пользовательский интерфейс, с которым справятся даже дети — вот его основное преимущество. Он удобен для любого новичка, но в основном Kano разработан для того, чтобы дети экспериментировали и обучались на опыте.

Сама по себе Kano OS хороша, но для использования ее по максимуму не обойтись без вспомогательных материалов. На момент написания они ограничены

двумя книгами и проектами, входящими в ОС — к счастью, превосходными. Книжки, подобие инструкции к набору Lego, объясняют, как проще и методичнее использовать Kano и ПО. Со слов одного из основателей проекта, Алекса Кляйна [Alex Klein], будет и другой контент для пользователей, доступный через пакеты расширения.

В целом, Kano — отличный проект, хорошо отвечающий спросу и делающий обучение приятным опытом; включенные в его идеи делают его идеальной средой для учеников. Технически, очень впечатляет ускорение: снижена задержка, которой страдают другие дистрибутивы для Raspberry Pi. Однако продвинутым пользователям и серьезным разработчикам лучше придерживаться Raspbian, до тех пор, пока Kano не возмужает. LXF

Свойства новскидку

Minecraft для всех
Предустановлен *Minecraft*, плюс предлагаются проекты для программирования ваших собственных приключений во вселенной *Minecraft*.

Добавочное оборудование
Под гладким интерфейсом в Kano включено большинство приложений, по умолчанию входящих в ОС Raspbian.

LINUX FORMAT Вердикт

Kano OS
 Разработчик: Команда Kano
 Сайт: <http://kano.me/>
 Лицензия: Свободное ПО

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

» Идеальный дистрибутив для детей, желающих обучиться работе на компьютере, но пока еще не дистрибутив для серьезного хакинга.

Рейтинг 9/10

Набор PIR Alarm GPIO

Лес Паундер делает первые шаги к созданию собственной охранной системы с помощью хитроумного проекта для Raspberry Pi. Шаг 2: игрушечные пушки.

Вкратце

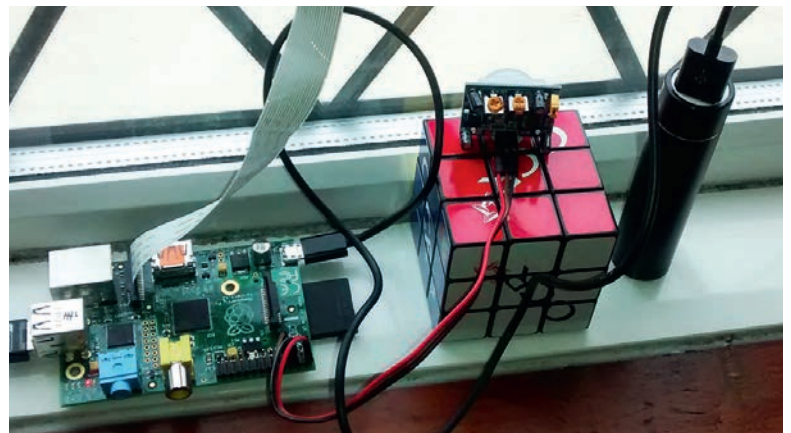
» Интересный проект с использованием инфракрасных сенсоров, Bluetooth и камеры Pi защитит ваш дом и сообщит вам по электронной почте о попытках проникновения.

Набор PIR (Passive Infrared sensor) Alarm от TR Computers позволяет создать собственную домашнюю охранную систему. В нём нет ни камеры, ни Raspberry Pi, зато есть инфракрасный датчик движения (PIR) на базе популярного чипа BISS0001, кабели мама-мама, пластиковые винты для крепления камеры и датчика к корпусу Raspberry Pi и схема сверления отверстий в корпусе Raspberry Pi, чтобы прикрепить готовый комплект к дверному косяку или ещё куда-нибудь.

Комплект собирается очень просто, согласно четким инструкциям на приложенном DVD. Труднее всего подключить датчик к GPIO (General Purpose Input Output) Raspberry Pi, но при тщательном следовании инструкциям особых проблем не будет. Для удобства, на датчике есть схема расположения контактов — под пластиковым куполом, который легко снимается; но будьте осторожны с конденсаторами по периметру платы, они очень хрупкие.

Собрав комплект, мы быстро протестировали его, следуя руководству, начав с отправки электронной почты с приложенным фото в случае сигнала тревоги. Скрипт также записывает 10-секундное видео, сохраняемое на Raspberry Pi для последующего извлечения.

Проектные файлы для этого набора написаны не на Python, характерном для Raspberry Pi, а в виде сценариев оболочки *Bash*. Чтобы активировать все функции отправки сообщений с вложениями, надо установить некоторые добавочные пакеты.



» Настройка прибора для слежения за всеми из окна. Сенсор придётся разместить снаружи, поскольку стекло не очень хорошо пропускает ИК-лучи.

Сделав это и добавив детали нашей учетной записи электронной почты во всплывающие файлы настройки, мы проверили правильность работы, отправив тестовое письмо из терминала.

Окрылённые успехом, мы пристроили устройство, чтобы оно контролировало входную дверь на предмет проникновения злоумышленников, и подключили портативный USB-аккумулятор. Через SSH мы дистанционно вошли в систему в сеанс *byobu* на Pi и запустили скрипт оболочки; затем на несколько часов разошлись по делам, оставив устройство работать.

Время испытаний

Мы весь день инициировали сигнал тревоги, выходя из дома и запихивая всякие предметы в почтовый ящик. Датчик реагировал на наши действия, заставляя устройство делать снимки, которые приходили на наш адрес, и фиксировать каждое действие на видео, которое сохранялось на самом Raspberry Pi.

На прилагаемом DVD есть масса других проектов: как, используя *ImageMagick*, добавить отметку времени на ваши медиа-файлы; как наладить работу с web-камерами Playstation 2 и 3 или USB-устройствами захвата видео; как, в случае тревоги, идентифицировать нарушителя с помощью сканирования средствами Bluetooth и по сигналу проверять, не пытается ли кто-то взломать вашу сеть. У последних двух проектов нет аналогов на рынке.

С комплектом легко работать, документация очень понятна и хорошо написана.

Других комплектов PIR-сигнализации в настоящее время нет, но в Интернете много самопальных проектов и инструкций; так стоит ли платить £11,99 за этот набор, особенно если учесть, что на eBay можно найти датчик BISS0001 в районе £4?

Стоит. Хотя бы ради входящих в него ресурсов и отличной документации, шаг за шагом проводящей вас по процессу. Бесспорно, подобный датчик можно приобрести в разы дешевле, но тогда вам либо понадобятся исходники, либо придётся самим создавать собственные скрипты для работы с ним.

Если вы не прочь повозиться с каким-нибудь простым аппаратным проектом Raspberry Pi, но не знаете, с чего начать или где взять необходимые компоненты, то это идеальный старт. **LXF**



Свойства навскидку



Сенсор PIR

Модуль BISS0001 PIR широко используется в разных проектах; пригодился также и в этом.



Что в коробке?

Хорошо, что есть материалы и по фото, и по видео. И по другим любопытным проектам!

LINUX FORMAT Вердикт

PIR Alarm GPIO Project

Разработчик: TR Computers
Сайт: <http://tri.co.uk/>
Цена: £11,99

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

» Простой набор для начинающих, желающих попробовать аппаратные проекты. Впереди — море открытий.

Рейтинг **9/10**

Intel NUC D54250

Этот маленький силач от Intel выводит HDTV Нейла Мора на новый уровень.

Вкратце

» Миниатюрный облегченный ПК с громким именем от Intel и внушительной для таких размеров производительностью и поддержкой 3D.

Спецификация

- » Intel Core i5 4250U, 1,3 ГГц
- » Intel HD Graphics 5000
- » 2 слота DDR3 SODIMM, 1,35 В
- » 2 мини-слота PCIe
- » v1.2 DisplayPort
- » v1.4a micro-HDMI
- » 4×USB 3.0, 2×USB 2.0
- » Gigabit LAN
- » Габариты, мм: 116×112×34,5

Хоть и с опозданием более чем на год, *LinuxFormat* должен приглядеться к Intel NUC. Две оригинальные модели NUC — невразумительно именуемые DC32171YE и DC3217BY — появились в конце 2012-го и, на шумев в отрасли, привлекли к себе достаточно внимания. Через год-полтора Intel обновила ассортимент за счёт архитектуры Haswell и переделанного корпуса, который стал еще меньше. Так в чём же секрет успеха?

Intel называли новую линейку 'NUC', что означает Next Unit of Computing [Новое слово в компьютерах]. Весьма претенциозное название, если учесть, что это всего лишь малый форм-фактор, проприетарный дизайн от Intel. Мы использовали ненавистное слово на букву П, хотя сейчас в мире техники стремятся уходить от индивидуальных схем. Но дизайн очень аккуратный и опрятный: размер материнской платы в NUC — всего



» После внутренней переработки на материнскую плату NUC накрутили ещё больше.



» Ещё меньше, ещё быстрее, ещё лучше укомплектованный, и ещё, и ещё...

10,16×10,16 см, а изящный корпус (116,6×112×34,5 мм) умещается в одной руке.

Это удалось Intel за счёт того, что материнская плата задумана как автономная; никакого расширения, разве что за счёт базовых слотов. Процессор припаян непосредственно к плате, как и уникальная система охлаждения, которая работает почти бесшумно.

Главные ценности

Процессор данной модели — Intel Core i5 4250U. 'U' указывает на низкое энергопотребление — максимальная мощность всего 15 Вт. Это двухядерный процессор 1,3 ГГц с Hyper Threading, а также поддержкой режима Intel Turbo до 2,6 ГГц, то есть официального автоматического разгона. Важно отметить, что эта модель Haswell также использует новую Intel HD Graphics 5000, и на ней пойдут современные игры с разрешением 720p с приемлемой (хотя и лишь чуть побольше 20 в секунду) частотой кадров. В дополнение к столь современному процессору — технологии Intel Quick Sync и Wireless Display, поддерживающие выход на новые 4K-дисплеи через micro-HDMI и Display Port.

Как раз прогрессивное оснащение может создавать проблемы менее прогрессивным пользователям: выходит, вам надо покупать micro-HDMI или Display Port-совместимый монитор. Объясняется это тем, что в NUC используется более новый тип памяти DDR3 SODIMM на 1,35 В, в отличие от более старого и широко поддерживаемого 1,5-вольтового. Два одинаковых слота SODIMM поддерживают до 32 ГБ памяти; также внутри имеются два слота PCIe для диска SSD типа mSATA и платы беспроводной связи, для которой в NUC

предустановлена спаренная беспроводная антенна. Также в новой версии есть незначительные, но полезные обновления более старых комплектующих. Имеется четыре порта USB 3.0 и два — 2.0, а также проводной порт Ethernet. Кроме того, в Intel сочли нужным добавить аналоговый выход для наушников и инфракрасный датчик на передней панели; и то, и другое не лишнее для домашних медиа-систем. Кроме того, есть внутренний коннектор USB 2.0 и порт STAT.

Цена Intel NUC слегка обманчива, ибо из коробки обычно продают облегченный вариант устройства. Так что от £279 цена быстро дорастает до менее сладких £390, за счёт памяти, накопителей и беспроводной карты. Плохо ли это? Для системы с такими функциями — нет. Но если вам нужен только медиа-центр, то это перебор: Pí за £30 делает то же самое. **LXF**

Сравнение навскидку

Тест	Zbox Nano	NUC DC33217	NUC D54250
Blowfish*	4,726	5,846	4,725
Cyptohash**	327,085	246,524	326,134
Fibonacci*	1,669	2,552	1,587
N-Queens*	7,346	6,438	5,082
FFT*	1,011	1,581	0,992
Raytracing*	6,202	7,826	6,198

* меньше — лучше, **больше — лучше

LINUX FORMAT Вердикт

Intel NUC D54250

Разработчик: Intel
Сайт: www.intel.com/nuc
Цена: от £279

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

» Хорошо спроектированная система, хоть и не расширяемая; мощная, но за кругленькую сумму.

Рейтинг **9/10**

Dell Chromebook 11

Допрыгнул ли Chromebook от Dell до планки? Выясняет **Кевин Ли**.

Вкратце

» Малобюджетный ноутбук с Chrome OS.

Спецификация

- » CPU: Intel Celeron 2955U, 1,4 ГГц (2 ядра, кэш 2 МБ)
- » Графика: Intel HD Graphics
- » ОЗУ: 2 ГБ DDR3L
- » Экран: ЖК, 11,6 дюйма, разрешение 1366×768
- » Хранилище: 16 ГБ SSD, плюс 100 ГБ от Google Drive на 2 года
- » Порты: 2×USB 3.0, HDMI, читалка SD-карт, гнездо для наушников/микрофона, гнездо для Kensington замка
- » Wi-Fi 802.11 a/b/g/n, Bluetooth 4.0
- » Камера: 720p

Официальный факт: производители ринулись на Chromebook'и. Не успели в Toshiba отпраздновать выход их первой машины на основе браузера, как и Dell туда же. Рост выбора ноутбуков всегда радует, да только Chromebook'и растут в одну породу. Chrome OS, операционная система на базе браузера и облака от Google, почти не изменилась со времени своего дебюта в мае 2011 г., за исключением новых реализаций Google Now и расширения магазина приложений. Как следствие, аппаратная часть также не изменилась. Помимо Chromebook Pixel, функциональность прежняя, с экранами, худо-бедно HD, и с упором на энергосбережение — в конце концов, это малобюджетные машины. Удастся ли Dell прервать монотонность серии и явить миру нечто новое и яркое?

Chromebook 11 от Dell сразу очаровывает простым, но элегантным дизайном. Покрытие 11,6-дюймового ноутбука в основном выполнено из матового пластика угольного цвета, с черным резиновым покрытием по краю лицевой панели экрана и клавиатуры, а также частично по бокам. Ничто в этом Chromebook не кричит о его бюджетности — он не особо отличается от других устройств того же производителя. Сборка достаточно добротная, с минимальным прогибом даже в середине клавиатуры.

По своим габаритам Dell Chromebook 11 находится посередине класса 11,6-дюймовых ноутбуков Chrome, с размерами 29,5 × 20 × 2,5 см (Ш × Д × В) и весом 1,3 кг. Это чуть легче, чем Acer C720P Chromebook, который на 0,5 см тоньше. И лишь чуть-чуть толще, но более чем на 200 г тяжелее, чем Chromebook 11 от HP.



» Первая вылазка Dell на территорию Chromebook весьма импозантна.

Отходя от внешних параметров, аппаратные спецификации — практически общие для всех Chromebook'ов, и устройство Dell — не исключение. Мы протестировали версию Dell Chromebook 11 на 2 ГБ, стоимостью £179. В нем 16 ГБ SSD — бережливость, присущая всё большему количеству машин на Chrome OS: ведь Chromebook'и живут на облачных хранилищах, со 100 ГБ бесплатного пространства на Google Drive в течение двух лет, а значит, в тяжелом и медленном жестком диске нет никакой необходимости.

Сам по себе Dell больше всего напоминает Acer C720P, с тем же разрешением экрана, 1366×768, и даже тем же процессором. Единственная придриска — экран имеет синюшный оттенок, и контрастность могла бы быть получше.

Батарея-долгожитель

Несмотря на все наши старания как можно быстрее истребить заряд батареи, Dell Chromebook 11 сопротивлялся паразитично долго: 8 часов 22 минуты. Даже при самых интенсивных нагрузках индикатор батареи никогда не опускался ниже прогнозируемых пяти часов. При обычном использовании — с яркостью экрана 65% работая на Google Drive под звуки Spotify — время автономной работы достигало 10 часов, или даже 12, за счет снижения яркости экрана и воздержания от YouTube. С таким ноутбуком можно выйти из дома, забыв зарядное устройство.

Dell Chromebook 11 — не прорыв в будущее для Chrome OS и не революция

в аппаратуре Chromebook. Помимо недочётов с экраном, это просто надёжный ноутбук, хорошо укомплектованный внутри и привлекательный снаружи. Дизайн Chromebook, наконец, устоялся, что позволяет производителям вроде Samsung, а теперь и Dell работать над эстетикой.

Тем не менее, ясно, что Chrome OS еще незрелая. Ей почти три года, а Google до сих пор не взялся почистить интернет-магазин Chrome — не говоря уже о том, чтобы привлечь больше сторонних приложений, если они питают хоть какую-то надежду вывести свою ОС на рынок пошире, чем рынок студентов и профессионалов, которым нужен недорогой ноутбук для мелкой офисной работы. Впрочем, всегда можно установить старый добрый Linux, ради большей функциональности. **LXF**

LINUX FORMAT Вердикт

Dell Chromebook 11

Разработчик: Dell
Сайт: www.dell.co.uk
Цена: £179

Функциональность	5/10
Производительность	9/10
Удобство использования	9/10
Оправданность цены	8/10

» Емкость аккумулятора, дизайн и добротная сборка отнюдь не как у малобюджетного ноутбука.

Рейтинг 9/10





БЕЗОПАСНОСТЬ

Особое внимание — физической защите

Компания Sony готовит защищённый смартфон.

Производители смартфонов среднего класса, как правило, мало задумываются о такой характеристике устройства, как возможность его полноценной физической защиты. Юзабилити — да, внешний вид — да, защита информации — тоже да, а вот физическая безопасность не считается сколько-нибудь значимым конкурентным фактором.

Безусловно, такая точка зрения ошибочна. Всё больше пользователей работают по модели BYOD — на их смартфонах обрабатывается не только личная,

но и служебная информация. Поэтому требования к безопасности устройства в целом постоянно повышаются.

Поскольку смартфон с точки зрения BYOD — вспомогательное устройство, приобретаемое только для оперативной связи, то его не оставишь в гостиничном сейфе, отправляясь на пляж или прогулку по лесу. Так что на практике есть только один метод защиты смартфона: никогда не выпускать его из рук (или, как минимум, не вынимать из кармана).

Не исключено, что менеджеры Sony это поняли и заранее объявили о том, что

на сентябрьской выставке IFA 2014 наряду с другими моделями будет показан влагозащищённый LTE-смартфон среднего ценового диапазона. То есть аппарат, явно ориентированный на применение в рамках концепции BYOD.

О технических параметрах смартфона ничего не говорится, что вполне объяснимо: главное в нём — не начинка, а корпус и цена. Не стоит лишний раз засорять мозги пользователя гигагерцами и гигабайтами, поскольку он заранее знает, что они будут вполне достаточными для решения его задач.



» Sony готовит непромокаемый смартфон.

ОТЕЧЕСТВЕННОЕ ПО

Подарок мобильнику от «Спутника»

Вышли Android-версии приложений «Мой дом» и «Лекарства».

Отечественная поисковая система «Спутник» выпустила для мобильных устройств два сервиса, имеющих важное социальное значение.

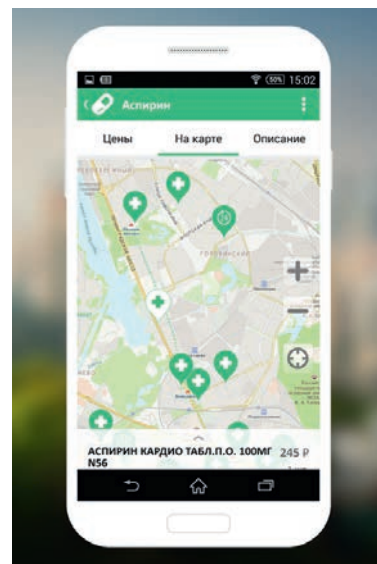
Первый — «Мой дом» — предоставит людям информацию обо всех обслуживающих население организациях, расположенных около заданного адреса: управляющих компаниях, почтовых отделениях, поликлиниках и т. д., и поможет проложить к ним оптимальный маршрут. По стандарту «Спутника», информация должна включать адрес, официальное название, часы работы и контакты. Любопытные могут уточнить год постройки здания, этажность и т. п. В базе данных программы хранится около 20 млн российских домов. Разумеется, большая их часть — в Москве и других мегаполисах. Но перспективы

у решения есть, и всё будет зависеть от активности региональных властей. Полезная функция приложения — сохранение истории поиска. Доступ к единожды востребованному можно получить без подклю-

«Всё будет зависеть от решения региональных властей.»

чения к Интернету. В наших условиях это бывает чрезвычайно важным.

Второй сервис — «Лекарства». Он радикально упрощает поиск ближайшей аптеки, а также получение информации о медицинских препаратах. Поиск возможен и по названию лекарства, и по входящему



» Мобильное приложение «Лекарства» поможет найти нужные медикаменты.

в него основному активному веществу. Теоретически сервис работает по всей РФ. На практике он полезен жителям крупных городов (хотя в небольших населённых пунктах легко обойтись без него). Его база данных включает более 15 тыс. аптек и постоянно пополняется. Работает и навигация — приложение построит оптимальный маршрут от дома до нужной аптеки.

GALAXY

От Samsung Apps к Galaxy Apps

Компания Samsung обновляет магазин приложений.

Компания Samsung открыла магазин приложений Galaxy Apps, который теперь заменит привычный Samsung Apps. Особых изменений в содержании не заметно — новыми стали только дизайн и структура.

Вероятнее всего, один из основных производителей смартфонов и планшетов на основе системы Android начинает потихоньку создавать собственную экосистему, формально обособленную от того, что делает Google. Хотя, разумеется, на совместимость приложений это никак не влияет — приложения можно устанавливать из любого источника.

Также не беспочвенно предположение, что в Samsung намерены как-то разграничить Android-устройства Galaxy и устройства, работающие на их собственной платформе Tizen. Сделать это лучше до того, как Tizen обретет популярность, иначе наступит путаница.

С прагматичной точки зрения магазин Galaxy App предпочтителен для владельцев соответствующих устройств — в нём

есть программы только для аппаратов Samsung, а на некоторые приложения предлагаются заметные скидки.

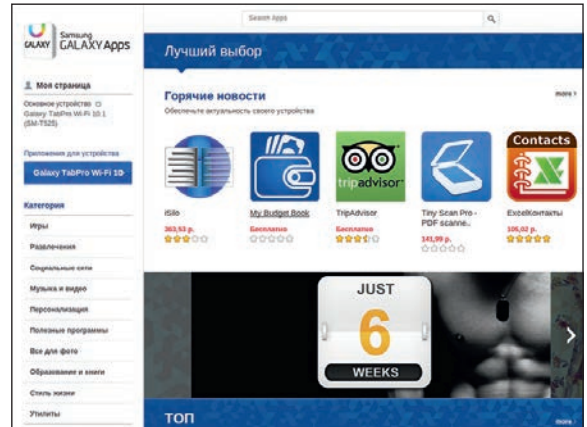
Компания Samsung вообще активно использует ПО для продвижения своих устройств. Например, использующие Galaxy игроки в знаменитую *FIFA 14* получают многочисленные бонусы, в том числе специальных «прокачанных» футболистов, характеристики которых несколько выше обычного. Шансы на победу, таким образом, возрастают.

Новый магазин состоит из четырёх секций. Они частично копируют те, что имеются в Google Play.

В «Горячих новостях» пользователи смогут установить рекомендованные производителем приложения. По понятным причинам он располагается в самом верху страницы.

В «ТОПЕ» собраны самые популярные программы. Причём все они уже представлены в Google Play.

А вот раздел «Для Galaxy» — это уже эксклюзивные предложения специально



► Компания Samsung выпустила новую версию магазина приложений.

для владельцев аппаратов производства Samsung. Он состоит из нескольких секций — Essentials (наиболее важная), Gifts (лучшие бесплатные программы, For Professionals (программы для корпоративного сектора), Specials (программы, написанные с применением SDK Samsung).

И, наконец, «Специальные предложения на сегодня» — раздел для ежедневного посещения, поскольку именно через него продаются программы со скидками. За этим разделом стоит последить: иногда скидка составляет 100%!

ТЕЛЕПАТИЯ

Google Glass становится телепатом

Очки дополненной реальности будут улавливать сигналы мозга.

Окружающая нас действительность становится всё больше похожей на фантастику. Пользователи ещё не успели привыкнуть к возможности управления устройствами при помощи голосовых команд, а лондонская фирма This Place уже сообщает о подготовке платформы MindRDR, которая сделает гаджеты самыми настоящими телепатами.

Правда, пока речь идёт только об одном устройстве — очках дополненной реальности Google Glass. Да и мысли читаются весьма условно и крайне ограниченно. Однако всё принципиально новое началось с маленького шага вперёд.

Чтобы управлять Google Glass силой мысли, потребуется дополнительное устройство — датчик мозговой активности Neurosky EEG. Оно закрепляется на голове пользователя и способно реагировать на повышение концентрации мозговой деятельности.

С практической точки зрения это пока ещё игрушка, поскольку устройство понимает только один мысленный приказ: сделать фото и разместить его в социальной сети. Логика разработчика в данном случае понятна. Наверняка они хотят набрать максимальное число пользователей (и тестеров) при помощи известного маркетинговым «вау-эффекта».

Обещано, что в ближайшее время доступ к проекту получат инженеры из других компаний; и, естественно, при подключении свежих сил дело должно двинуться повеселее. **LXF**



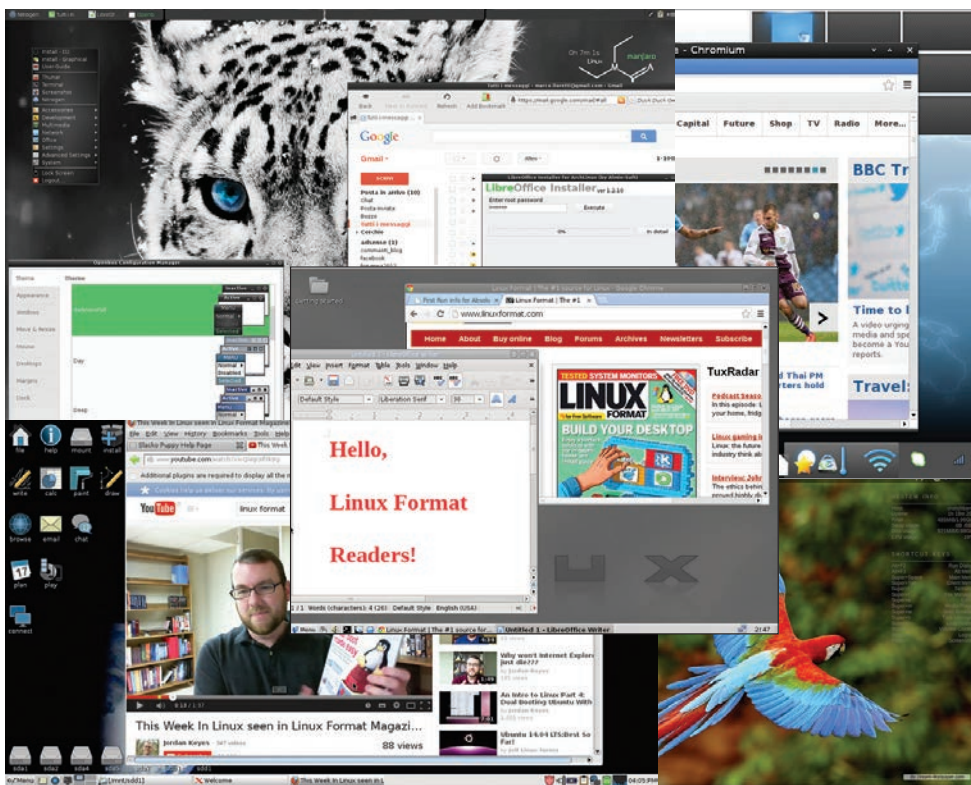
► Очками Google Glass можно будет управлять усилием мысли.

Сравнение

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

Дистрибутивы-малоежки

Живы ли не жадные к ресурсам настольные Linux? Марко Фиоретти тестирует пять малоизвестных дистрибутивов — шустрых, но и удобных тоже.



Про наш тест...

Мы не рассматривали Lubuntu и прочие «легкие» клоны Ubuntu по той простой причине, что скорее всего их и так хорошо знают новички в Linux, которым важен нетребовательный к ресурсам дистрибутив. После долгих исследований мы выбрали один дистрибутив с подлинно уникальной архитектурой (Puppy) и четыре легковесных производных более популярных систем, а именно: Slackware, Arch Linux и Debian. Затем мы запускали их в режиме live на маломощном ноутбуке и/или устанавливали на виртуальные машины с разными объемами ОЗУ, но не больше 2 Гб. Во всех случаях системы тестировались при обычной, повседневной настольной деятельности, от просмотра сайтов и работы в офисных пакетах до вывода фотографий. Мы также опробовали несколько рядовых задач системного администратора, от установки программ до добавления пользователей и смены общей настройки системы.

Наша подборка

- » Absolute Linux
- » Crunchbang
- » Elive
- » Manjaro
- » Puppy

За последние 20 лет компьютеры ушли далеко. Как и настольный Linux; но с ним несколько иначе. Современные компьютеры — в любом формате, от смартфонов до ноутбуков — на порядки мощнее своих предков из 1990-х, а настольный Linux вдесятеро быстрее не стал. Какие же у нас есть варианты ускорить свой Linux, особенно на «возрастном» оборудовании?

Многие разработчики создают т.н. «нетребовательные к ресурсам» дистрибутивы, съедающие намного меньше памяти и циклов процессора, чем Ubuntu и ему подобные «модерновые» продукты. Увы, это

«Нетребовательные к ресурсам дистрибутивы слишком часто бесполезны для большинства.»

слишком часто порождает настольные системы, по сути бесполезные для большинства. Так, Damn Small Linux (DSL) прекрасен для программера, но для многих ли «обычных» вещей он годится? Возьмете ли вы DSL, чтобы редактировать таблицы Excel, смотреть клипы на YouTube, работать с облачными сервисами на Ajax типа Gmail или без особого труда обновлять ПО?

Что и подвигло нас на это Сравнение. Мы не искали самонаименший на свете настольный Linux, или требующий меньше всех ОЗУ: нас интересовали дистрибутивы, в которых обычным, «непродвинутым» пользователям будет проще всего работать с офисными документами, проверять свою электронную почту и обращаться к популярным web-сервисам даже на старых ПК.

Настольная полноценность

Что, собственно, я могу сделать в этом дистрибутиве?

Основные причины, почему многие люди, заинтересованные в Linux, никогда не берутся за дистрибутивы «с малым потреблением ресурсов», вероятно, заключаются в отсутствии желания, поддержки и приложений. Тем, у кого мощный компьютер, проще не вникать. Новички в Linux тоже часто держатся подальше от «нишевых» дистрибутивов, поскольку, что вполне объяснимо, хотя и оставляется там, где есть справочники и другие пользователи. Мы обсудим эту проблему через страницу. А пока займемся жалобой на «недостаток приложений».

Итогом рассмотрения будет, что для наших дистрибутивов «недостатка приложений» нет, и вашим истинным ограничителем будет оборудование. Никакая программа не позволит редактировать видео с высоким разрешением или обрабатывать таблицы на тысячи строк при (скажем) 128 МБ ОЗУ.

Для традиционных офисных задач [SOHO] вы найдете приложения, которые в любой из наших систем выполняют практически все необходимое; иногда это те же инструменты, что и в более известных настольных Linux. Если приложения нет —

не проблема: вы все равно сможете установить его из предустановленных архивов всего парой щелчков или нажатий клавиш.

Док-панель по умолчанию в Elive все-ма впечатляет: тестируемая версия по-прежнему придерживается *LibreOffice 3*, но также предоставляются *Clementine*, *Skype*, *Scribus*, *Chromium*, *Blender*, *Inkscape*, *RawTherapee*, *GIMP* и *VLC*. И все это вы найдете отображенным на док-панели.

Puppy тоже впечатляет, но иным способом. Некоторые версии идут с *SeaMonkey* в качестве браузера, клиента электронной почты и программы для чтения RSS. В Slacko Puppy мы обнаружили *Firefox* и клиент электронной почты *Sylpheed*.

Кроме того, похоже, что меню Slacko создано в старой доброй традиции Unix — с таким изобилием мелких инструментов, включая малоизвестные, что мы просто сбились со счета. Часто у них винтажный вид и интерфейс и специализация в одной области деятельности.

Комбинируя их, вы сможете делать практически все, что хотите, если уделите время на обучение. Текстовый редактор по умолчанию — *AbiWord*, но легко добавить *OpenOffice 3.1*.



» Системное меню у Elive простое, но между ним и док-панелью есть все необходимое...

Absolute, Crunchbang и Manjaro находятся где-то между этими двумя крайностями, каждый по-своему. В Absolute имеется ряд спецприложений разработки (*HtmlPage*, *Qt Designer* и шестнадцатеричный редактор), и можно получить *LibreOffice 4.1.4*. Меню интернет-приложений включает *Filezilla* и просмотрщик YouTube.

Crunchbang идет с браузером *IceWeasel*, который работает прекрасно, хоть иногда и подтормаживает. Выбор Мультимедиа ограничен регулировкой громкости, *VLC* и *Xfburn* для мастеринга DVD.

В Manjaro, содержащем *LibreOffice 4.1.5*, сходная ситуация: микшер *Alsa* и мало что кроме него. Вдобавок это единственный здесь дистрибутив, неспособный с ходу воспроизвести видео с YouTube.

Вердикт

- Elive ★★★★★
- Crunchbang ★★★★★
- Manjaro ★★★★★
- Puppy ★★★★★
- Absolute ★★★★★

» У Elive наиболее полный набор функций, но остальные дистрибутивы умеют делать практически то же самое...

Установка

Легко ли начать работать с этими системами?

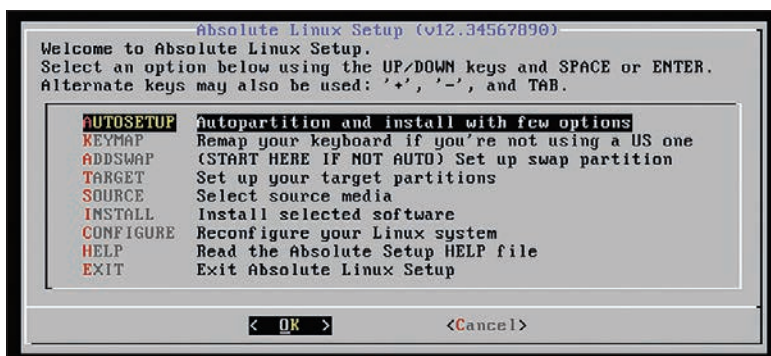
В общем и целом, установка этих пяти дистрибутивов или использование их как live-систем не должны представлять сложностей, если только вы не установите их на совершенно новое, более или менее закрытое оборудование, или при наличии сложной комбинации жестких дисков.

Например, программа установки Absolute Linux чисто текстовая, с загрузчиком *LILO*. Если вы не знали, что Linux возник в 1994 году, попробуйте Absolute, чтобы увидеть, каково было его устанавливая в те времена. Но не переживайте — пускай все это выглядит не гламурно, использует только клавиатура, а мышь — никогда,

и чтение надписей на некоторых панелях отнимает больше времени, чем хотелось бы... реальных сложностей здесь по сути нет. Оборудование прекрасно определяется, и можно спокойно принять по умолчанию все настройки и ответы на вопросы, которые вы не поняли сразу.

Самые дружелюбные установщики (или загрузчики, для версий live) — графическая программа у Manjaro, и того лучше, у Elive. Первая, именуемая *Thus*, наверное, больше всего похожа на Ubuntu, и умеет также шифровать разделы. У второй отлично отлажен выбор языка и клавиатуры и есть интересные опции в окне Choices.

Загрузчик Slacko Puppy мало озабочен своей милотностью, зато работает гладко. А основная настройка системы после входа в нее передается дружелюбному мастеру. Здесь и выбираются код страны, время, раскладка клавиатуры, сетевые настройки, графика и пр. Программа даже тявкает, одобряя ваши действия.



» Установщик Absolute Linux вызывает вопрос, какой нынче год, но работать с ним легко.

Вердикт

- Elive ★★★★★
- Crunchbang ★★★★★
- Manjaro ★★★★★
- Puppy ★★★★★
- Absolute ★★★★★

» Программа установки Absolute чисто базовая, а вот программа установки Elive способна привести новичков в восторг.

Интерфейс пользователя

Настраиваемый и привлекательный, но без алчности к ресурсам?

Интерфейс любого компьютерного устройства, от смартфона до традиционного настольного ПК, ищет собственный баланс противоречивых требований: минимума потребления ОЗУ и циклов CPU, что в нашем случае является обязательным; удобства в работе для непрофессионалов; возможности

индивидуальной настройки; и, наконец, приятной внешности (в каком бы смысле ни понимал приятность каждый конкретный пользователь).

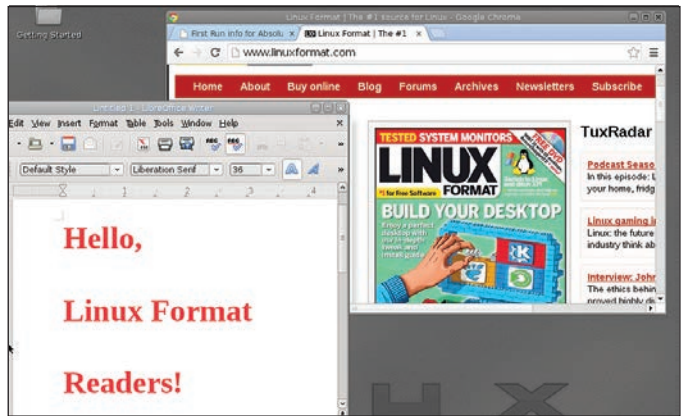
После тестирования мы остались довольны своей подборкой дистрибутивов. Дело в том, что каждый из них дает свой, причем достойный, ответ на вышеперечисленные требования. Одни

люди предпочтут простой, но безгранично настраиваемый визуально Elive, другие — богатую, но сравнительно жесткую структуру Absolute или Puppy, а кому-то импонирует бесхитростность Manjaro и Crunchbang. В любом случае, по вкусу что-то отыщется практически для любого пользователя, который во главу угла ставит экономию.

Absolute ★★★★★

Absolute использует оконный менеджер *IceWM*, обеспечивая вас быстрым и тактичным рабочим столом. Правый щелчок на главном окне по умолчанию запускает инструмент для смены обоев и разрешения экрана; можно настроить его на открытие системного меню. Менеджер файлов здесь *SpaceFM* — пусть это не самое последнее слово техники, но для решения базовых задач он вполне адекватен.

Из всех участников нашего Сравнения Absolute, вероятно, единственный, который напоминает Linux 15-летней давности. Но как раз поэтому он уже будет идеальным для многих пользователей, особенно если учесть, что это всего лишь первое и поверхностное впечатление: вы получаете современные браузер (*Chrome*) и офисный пакет (*Libre Office*), а ведь нередко именно это и есть главное, не так ли?



Crunchbang ★★★★★

При первом запуске Crunchbang весьма деловит и обходится без излишеств. Все, что вы видите — это внешне довольно скучный, но хорошо настроенный менеджер окон *OpenBox* с двумя рабочими областями, системным монитором и списком клавиш быстрого запуска. Однако сменить обои, если сплошной серый экран вас раздражает, легко и просто.

Правый щелчок на главном окне откроет меню с приложениями для всех задач рабочего стола. Данное меню плюс эмулятор терминала *Terminator* и менеджер файлов *Thunar* позволят вам эффективно работать и с клавиатурой, и с мышью.

Помимо всего прочего, Crunchbang, если можно так выразиться, является дистрибутивом, наиболее заинтересованным в работе с облаком обычными инструментами: системное меню включает пункты для запуска Google Docs и установки клиента Dropbox.

Документация

Просто ли изучить, как работают эти дистрибутивы?

Если вам нужен Linux только для запуска каких-то приложений на старом компьютере, количество страниц документации волнует вас довольно мало. Лучше проверьте, хороша ли документация у приложений, а не у дистрибутива, и ищите дистрибутив, который поддержит вас первые два-три часа работы, когда вам действительно нужно будет знакомиться с ним, и с его сообществом.

Обои по умолчанию Crunchbang — то, что должно быть во всех дистрибутивах для новичков: это шаргалка с основными

привязками к клавиатуре. Большинство материалов Debian относится также к Elive, так что документации здесь, быть может, даже слишком много.

Manjaro и Absolute в этой номинации лидеры. При первом входе в систему Manjaro приветствует вас окном, полным ссылок на документацию, онлайн-поддержку и описание имеющихся рабочих столов.

Значок Getting Started на рабочем столе Absolute загружает ориентированный на практику справочник из 14 пунктов, с разделами типа «Файлы мультимедиа»,

«Настройка беспроводной сети», добавление программ и драйверы Nvidia. Меню Documentation содержит ссылки на справочники по дистрибутиву и его основным пакетам. То же самое предлагает и Puppy.

Кстати о Puppy: учтите, что многие из его онлайн-руководств и справочников могут относиться только к определенной версии (прочитайте окно Choices, чтобы узнать, почему). И прежде чем пробовать работать с Puppy, спросите на www.murga-linux.com/puppy о самых лучших и самых новых ресурсах для своей версии.

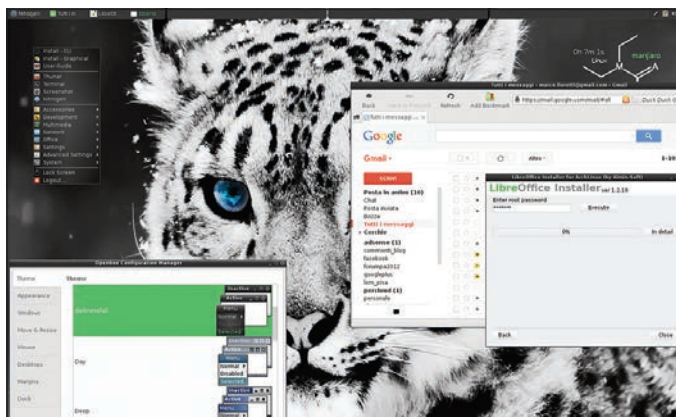
Вердикт

Absolute	★★★★★
Elive	★★★★★
Manjaro	★★★★★
Crunchbang	★★★★★
Puppy	★★★★★

» С документацией у наших дистрибутивов полный порядок. В Manjaro и Absolute она будет более доступна.

Elive ★★★★★

Elive — отличная демонстрация рабочего стола *Enlightenment*. В 1990-х, когда *Enlightenment* появился, многие (включая нас) по полу бы катались от смеха, скажи нам кто-нибудь, что он станет «нетребовательным к ресурсам». Ныне мы можем засвидетельствовать, что Elive дает оптимальный компромисс между визуальными эффектами и производительностью на оборудовании с ограничениями. Системный монитор, с его датчиками заряда батареи, температуры CPU и политики энергопотребления, не только привлекателен, но и удобен, и здесь достаточно приложений, включая инструмент синхронизации файлов *Unison*. Если у вас хватит терпения его настроить, то все будет выглядеть именно так, как вы хотите, от шрифтов в заголовках окон до комментариев-бирок. Однако при небольшом ОЗУ предварительно отключите все эффекты затухания и масштабирования.



Manjaro ★★★★★★

Если замечавшиеся пользователи вдруг перепутают Manjaro с Crunchbang, их можно понять: оба являются ответвлениями Arch Linux с приставленным *OpenBox*. Основное различие, помимо меньшего числа приложений, заключается, очевидно, в браузере по умолчанию. Manjaro использует *Midori*; его достаточно, чтобы запускать Gmail и аналогичные web-сервисы на базе *Ajax*, но видео с YouTube воспроизводить нельзя.

Зато Manjaro содержит инструменты, о которых вы, скорее всего, даже не слышали, от менеджера области обмена данными *ParcelLite* до браузеров *Avahi* для серверов SSH и VNC. Кроме того, работать с *OpenBox* в Manjaro легче, чем в Crunchbang, особенно если вы уже умеете с ним обращаться. Кроме программы настройки *Tint2*, системное меню содержит все возможные инструменты для настройки рабочих областей и динамического меню этого мощного менеджера окон.

Puppy ★★★★★★

Slacko 5.7, часть стаи Puppy, которую мы здесь рассматриваем, является на момент написания «рекомендуемой отправной площадкой для новичков» в своем семействе. Его пользовательский интерфейс — результат аккуратного смешения менеджера окон *JWM*, *ROX Filer* и множества больших и малых приложений, от вполне обиходных, например, *Firefox*, до весьма мало известных утилит. Подменю System даже содержит утилиту для форматирования флорпи-диска, вместе с менеджером разделов *GParted*.

Нам показалось не вполне понятной причина наличия трех отдельных подменю для приложений Document, Business и Personal, но они там тоже есть. Иными словами, *JWM* и *ROX* хорошо работают вместе, и *ROX* — это одно из тех приложений, которое должен по крайней мере разок попробовать каждый пользователь Linux: простое, быстрое и умное. «Компактный и эффективный» вовсе не означает «уродливый» или «скудный функциями».



Выбор

А если протестированная здесь версия — почти все, что вам нужно?

Работают ли все эти дистрибутивы по принципу «бери или проваливай»? Это зависит от того, что вы хотели бы изменить. Если вам нужен всего лишь альтернативный менеджер окон — никаких проблем. Absolute объявлен «совместимым по пакетам» с релизами Slackware того же номера, а у Crunchbang и Manjaro есть доступ к репозиториям Arch, и в этих пакетах хватит легко устанавливаемых альтернатив, чтобы вас осчастливить. То же относится и к Elive: он может использовать приложения из Debian

и также имеет режим «Experimental Mobile» для телефонов с сенсорным экраном, планшетов и подобных устройств.

У Manjaro три официальных варианта, все со значками Tulliana и Kfaenza: два для Xfce (полного или усеченного) и одна с KDE 4.12 (с дополнением KDE Connect, <http://community.kde.org/KDEConnect>), умышленно настроенным так, чтобы «предложить пользователям Windows новый дом».

Хотите еще? Попробуйте редакции Manjaro, созданные сообществом для других рабочих столов (Mate, Cinnamon, LXDE

и пр.), ответвление для нетбуков с оптимизированным под Atom ядром, или Manjaro-ISO, который облегчает создание клонов с фиксированным набором пакетов.

На закуску мы оставили самое интересное: Puppy. Woof «Puppy builder» (<http://bkhome.org/woof>) умеет компилировать Puppy из двоичных пакетов любого другого дистрибутива. Он уже создал Varebone Puppy, который, говорят, ладит даже с 200-МГц CPU и 64 МБ ОЗУ, и еще множество совершенно не похожих клонов — их так много, что здесь не перечислить.

Вердикт

- Puppy ★★★★★
- Manjaro ★★★★★
- Absolute ★★★★★
- Crunchbang ★★★★★
- Elive ★★★★★

» Хотите разнообразия — выберите Manjaro. Хотите создать нечто индивидуальное — обратитесь к Woof Puppy Builder.

Системное администрирование

От настройки оборудования до управления файлами, трудно ли это?

Если вы обладаете базовыми навыками администрирования Linux, и вам требуется многопользовательский компьютер с Linux для выполнения основных задач на рабочем столе... не ставьте Puppy. Это не злопыхательская критика, а предупреждение.

Puppy сделан очень хорошо, но скорее подходит для использования Live-версии с CD или устройства USB одиночными пользователями. Одни инструменты настройки Puppy (например, *Menu Manager* или *Application Chooser*) просты и достойно выглядят; другие, например, *Firewall configurator*, эффективны, но архаичны.

Но главная проблема в том, что для Puppy вообще-то не нужны многочисленные учетные записи обычных пользователей. По умолчанию вы root: это ваш компьютер, и вы всегда сможете делать все, что захотите. Браузеры и прочие приложения, которые должны засвечиваться онлайн, работают как пользовательская «точка входа» с домашней директорией, куда не разрешено писать и где привилегии ограничены. Когда «точки» недостаточно, вы можете стать fido, обычным пользователем, как в любом дистрибутиве.

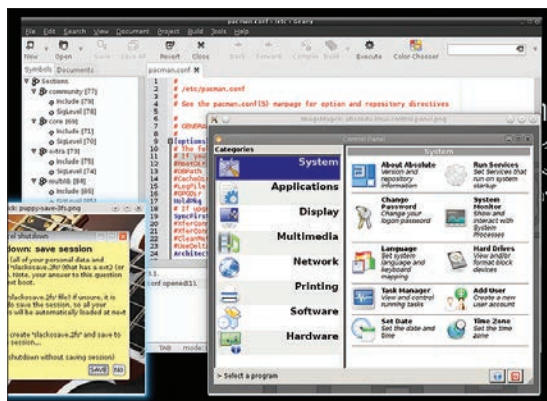
Кроме того, чтобы загрузиться и работать как можно быстрее, Puppy всегда остается целиком пребывать в оперативной памяти. Его ядро, библиотеки, бинарники, файлы настройки... все работает именно так, даже не делая попыток найти раздел swap. Файлы конечного пользователя, заодно с программами, установленными или обновленными после первой загрузки, идут в специальный файл с расширением `.3fs`, на системе-хосте или

внешнем устройстве хранения. Все ваши навыки администрирования в Puppy могут свестись к простому «всегда обновляй свои файлы `.3fs`!»

После Puppy, остальные четыре дистрибутива гораздо, гораздо более похожи на «обычный» Linux. В Absolute и Crunchbang для определенных задач может понадобиться командная строка, если вы не установите дополнительные пакеты. Например, в Absolute есть несложный графический интерфейс для добавления пользователей; но удалять их вам придется обычной командой `userdel`.

В Elive имеется симпатичная комбинация интерфейсов *Internet Configurator*, *Model Dialer* и *3G Mobile Phone*, но, как ни странно, не в отделении «GUI для системного администрирования». Есть риск так подсесть на эксперименты с притягательной панелью управления *Enlightenment*, что вы совершенно забросите все остальное. Работу в консоли Elive облегчает вам с помощью мощного эмулятора *Terminology*.

Crunchbang и Manjaro используют интерфейс настройки *Turbulence*, по крайней мере, в своих версиях с *OpenBox*. Manjaro также включает программы *MHWD* для установки неграфических драйверов. В остальном они аналогичны своему батюшке, Arch Linux.



Программа сохранения `.3fs` в Puppy, программа настройки Manjaro, Pacman и Absolute Control Panel.

Вердикт

- Crunchbang ★★★★★
- Elive ★★★★★
- Puppy ★★★★★
- Absolute ★★★★★
- Manjaro ★★★★★

» GUI-инструменты настройки есть во всех наших дистрибутивах (хотя, возможно, не те, что вы ожидали увидеть).

Установка и управление ПО

А если нужна программа, которая не установлена?

Как бы хорошо дистрибутив Linux ни смотрелся изначально, он абсолютно бесполезен, если не дает возможности установить новые программы или обновить уже имеющихся, не требуя от вас быть программистом и/или сражаться с адом зависимостей.

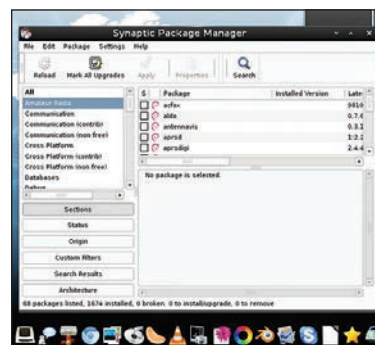
Elive в этом прощит всех: он использует тот же механизм, онлайн-репозитории и графические интерфейсы, что и Debian и его производные. В Crunchbang есть такие пункты меню, как *Install LibreOffice*. Щелчок по нему откроет терминал, где скрипт попросит у вас подтверждения, затем продолжит скачивание, скомпилирует исходники и установит результат.

В Absolute примитивное диалоговое окно спросит, например, хотите ли вы скачать пакеты *Multimedia*, то есть библиотеки

кодеков и тому подобное, не включенные по умолчанию. Другой инструмент перечислит установленные пакеты и предложит удалить их, либо установить новые, предварительно скачанные в формате TXZ.

Manjaro может найти и скачать полностью протестированные пакеты в онлайн-репозиториях типа AUR (Arch User Repository), через интерфейсы типа *Pamac*, который также способен компилировать и устанавливать архивы из исходника по двойному щелчку.

Puppy, повторимся, работает иначе — как на экране, так и за сценой. Каждая версия дает доступ к своей комбинации онлайн-репозиториям. Помимо этого, при первом запуске программы Puppy проверит наличие онлайн более новой версии



Elive использует интерфейс Synaptic, такой же, как в Debian, Ubuntu и других популярных дистрибутивах.

и предложит установить ее. Во всех случаях новое ПО устанавливается в файле `.3fs`, описанном в предыдущем разделе.

Вердикт

- Elive ★★★★★
- Crunchbang ★★★★★
- Puppy ★★★★★
- Absolute ★★★★★
- Manjaro ★★★★★

» Elive предлагает простейшее управление приложениями, посредством стандартных инструментов Debian.

Дистрибутивы-малоежки

Вердикт

Целью нашего Сравнения было помочь вам найти дистрибутив, быстро работающий даже на старых компьютерах и при этом удобный людям без опыта работы в Linux. Этот дистрибутив должен поддерживать и выполнение «офисных задач» (текстов, электронных таблиц и презентаций), и популярные web-сервисы, от домашних операций с банком до онлайн-видео, причем в идеале — посредством тех же приложений, что обычно используются в популярных дистрибутивах Linux, если даже не в Windows, или альтернатив со сравнимыми функциями и тем же дружелюбием к пользователю. Система должна удовлетворять этим требованиям сразу после установки или после установки пакетов из Интернета с примерно такими же возможностями, как в более популярных системах Linux. Ну и, по возможности, выглядеть приятно.

Pupru отчасти удовлетворяет этим требованиям: он быстр и очень компактен

(тестируемая версия весила всего 168 МБ). Pupru так хорошо спроектирован, что у нас велик был соблазн объявить его победителем. Однако не объявим, и вот причина: он не поддерживает многопользовательский режим, по крайней мере, не в том виде, который позволит новичкам в Linux с легкостью разделять один компьютер.

Absolute Linux отлично подойдет тем, кому нужен только офисный пакет и современный браузер, особенно если они предпочитают винтажные интерфейсы. Весьма голый дизайн по умолчанию Manjaro и Crunchbang может отпугнуть даже еще больше, чем архаичного вида значки и меню Pupru и Absolute. Однако эта проблема легко решается, и обе эти системы делают установку и обновление программ намного легче, чем две первые.

Онлайн-галереи экранных снимков *Enlightenment* показывают, как легко злоупотребить



его гибкостью, заделав его в такое барокко, что оно затруднит выполнение работы. А вот Elive отличается быстротой даже с анимацией по умолчанию, да и отключить ее очень легко. Настройка *Enlightenment* в Elive предлагает очень хороший баланс скорости, удобства и визуальной приятности. Помимо этого, управление приложениями в Elive осуществляется инструментами GUI, уже хорошо знакомыми многим линксоидам. Итак, объявляем победителем Elive, хоть он и имеет самый большой объем скачивания (около 2 ГБ).

» Благодаря Elive очень просто любить и использовать Linux, даже если все, что у вас есть — это пятилетний (а то и старше) компьютер.

«Elive предлагает хороший баланс скорости, удобства и визуальной приятности.»

I Elive ★★★★★☆
 Сайт: www.elivecd.org Лицензия: N/A Версия: 2.2.1 Alpha Hybrid
 » Быстрый, привлекательный и легко управляемый обычными инструментами Linux.

IV Puppy Linux ★★★★★☆
 Сайт: <http://puppylinux.org> Лицензия: N/A Версия: Slacko 5.7 NO-pae
 » Для очень старых компьютеров или компьютеров с одним пользователем.

II Crunchbang ★★★★★☆
 Сайт: <http://crunchbang.org> Лицензия: N/A Версия: 11-20130506
 » Здесь отличный менеджер окон и более гибкий браузер, чем в Manjaro.

V Absolute Linux ★★★★★☆
 Сайт: www.absolutelinux.org Лицензия: N/A Версия: 14.05 large
 » Идеален, если нужны только браузер и офисный пакет. В остальном ограничен.

III Manjaro ★★★★★☆
 Сайт: <http://manjaro.org> Лицензия: N/A Версия: 0.8.9 OpenBox
 » Почти не уступает Crunchbang, при большем числе инструментов настройки OpenBox.

Обратная связь
 Мы пропустили ваш любимый дистрибутив, потребляющий мало ресурсов? Сообщите нам свое мнение на lxfl.letters@futurenet.co.uk.

Рассмотрите также...

Наше Сравнение — не только для тех, кому нужно взбудить старое оборудование. Даже если вы довольны производительностью своего настольного Linux, почему бы не попробовать альтернативы, которые могут оказаться еще быстрее?

Стоит хотя бы раз попробовать Pupru Linux, чтобы осознать, какой может быть среда рабочего стола Linux. Правда, Pupru Linux живет в своем

мире, но у других дистрибутивов Сравнения есть одна общая черта: по большей части их скорость идет от простых, базовых версий их предшественников (Slackware, Arch Linux и Debian), а другие дистрибутивы добиваются этого, отказавшись от Gnome и KDE.

Итак, урок, который вы должны отсюда извлекать, таков: если ваш компьютер нестерпимо

медленный, не торопитесь менять дистрибутив. Лучше установите и настройте *OpenBox*, *Enlightenment* или *IceWM*! Если этого мало, отключите индексомеры и прочие ресурсоемкие демоны. Если мало и этого, попробуйте то производное своего дистрибутива, которое рекламируется как «нетребовательное к ресурсам». Ускорить Linux бывает намного проще, чем вы думали! **LXF**

Сравнение

»» Сравнить — так сравнивать! Подберем к экономичным дистрибутивам браузеры

Web-браузеры второго эшелона

Есть ли жизнь в Интернете без *Firefox* и *Chrome*? Ещё какая! Александр Толстой намерен извлечь на свет незаслуженно забытых героев.

Наша подборка

- » Konqueror
- » Midori
- » Opera
- » QupZilla
- » Rekonq
- » SeaMonkey Navigator

Сколько всего имеется web-браузеров, точно вам не скажет никто, но тех, что на слуху, около 80-ти. Статистика использования будет для вас, вероятно, предсказуемой. По всему миру, по данным W3schools, популярен *Chrome*, который всё никак не догонят *Firefox*, *Internet Explorer* и *Safari*. В мире Linux, как сообщает Webupd8.org, *Chrome* также лидирует, но *Firefox* уступает лишь немного и дышит ему в затылок. Остальные браузеры с большим отрывом отстают: получается, что корпорация Google и Mozilla Foundation поделили большинство Linux-пользователей, тогда как другие достойные проекты

незаслуженно лишились внимания. Здесь самое время рассказать о том самом критерии «второго эшелона», который мы применим в данном Сравнении.

С одной стороны, существует плеяда «лёгких» и минималистичных Linux-браузеров (*Dillo*, *Netsurf*), которые занимают свою нишу и ждут, когда вы поставите их на дедушкин ПК. С другой стороны, как грибы после дождя растут клоны *Chrome* и *Chromium*, которые неоригинальны. Тут и *Maxthon*, и новая *Opera 24*, и несколько других. С третьей стороны, мы встретим дюжину специализированных сборок на основе *Firefox* и того же *Chromium*, призванных защищать нашу приватность

и безопасность. Что остаётся? Оригинальные, самобытные, но при этом достаточно «взрослые» проекты. Да, они менее популярны, и у вас может возникнуть вопрос: зачем вообще нужно переходить на менее популярный браузер?

Причин на самом деле много. Скажем, вы не хотите доверять свои личные данные и предпочтения всемогущей Google, которая всё больше контролирует своих пользователей. Или же вы заметили, что на вашей конфигурации *Firefox 31* уже не так компактен и бодр, как первые версии «огнелиса». Наконец, само по себе наличие альтернативы — хороший повод исследовать мир браузеров чуть глубже.

Про наш тест...

Наши браузеры прошли огонь, воду и медные трубы... т.е. набор специальных тестов. А именно...

Соответствие стандартам

Типичные синтетические тесты на соответствие стандартам ACID3 и HTML5. Чем выше балл, тем лучше (в теории) браузер должен справляться с отображением сложных web-страниц.

Производительность под нагрузкой

Для стресс-теста изначально хотелось применить скрипт MemBench (<http://gregor-wagner.com/tmp/mem>), открывающий в браузере 150 наиболее популярных сайтов (по данным <http://mostpopularwebsites.net>). Но корректно он работает только для *Firefox* и *Chrome*, а в нашем исследовании с ходу заработал только для *Opera* и *SeaMonkey Navigator* (после легкой наладки — также и для *Konqueror*). И чтобы обеспечить всем

браузерам равные условия тестирования, пришлось использовать простую команду вида `browser $(cat urls.txt)`

где в файле `urls.txt` был список из 150 сайтов, позаимствованный у MemBench.

Такая нагрузка выявляет стратегию web-браузера по распределению памяти. При открытии 150 вкладок проверялась общая работоспособность, плавность прокрутки страниц и потребление ОЗУ. На десерт браузеру предлагалось открыть страницу, тяжело обремененную GIF-файлами; общий ее «вес» превышал 800 МБ. Насчет ОЗУ тут ничего необычного нет, но нагрузка на ЦП при рендеринге массы GIF-файлов многое говорит о стабильности браузера и качестве его кода.

Потребление памяти

Замер проводился на примере эталонной страницы (в качестве которой была выбрана yandex.ru).

Набор функциональных тестов

Была проверена работоспособность некоторых важных плагинов и расширений, таких как:

- » **AdBlock** Пожалуй, наиболее востребованная функция, для борьбы с назойливой рекламой.
- » **Adobe Flash Plugin** Умеет ли браузер находить установленный плагин в системе...
- » **WebGL** Современный способ реализации возможностей OpenGL ES 2.0 в языке JavaScript. Попробуем сыграть в *Angry Birds* прямо в браузере.
- » **Поддержка камеры и микрофона** Попробуем зайти на аналог ChatRoulette.

Каждый тест проводился 2 раза подряд на 64-битной Linux-машине в максимально изолированных условиях: без дополнительно запущенных приложений и без чего-либо интенсивного среди фоновых процессов. Для тестирования использовалась следующая конфигурация: Intel Core i3 2ГГц, 1,5 Гб ОЗУ, Intel HD Graphics.

Парад претендентов

Рассмотрим краткие биографии наших участников.



SeaMonkey Navigator

Воплощение классического *Netscape Navigator* и его наследник, данный проект был образован группой разработчиков, вышедших из Mozilla Foundation в 2005 году. *SeaMonkey Navigator* — потомок *Netscape Communicator* и *Mozilla Suite*, содержащий целый набор программ для работы в Интернете. Помимо классического web-браузера, основные компоненты пакета — почтовый клиент, редактор HTML, клиент RSS и коммуникатор *ChatZilla*. У *SeaMonkey Navigator* та же версия движка Gecko, что и у актуального *Firefox* от Mozilla, так что рендеринг сайтов у этих двух браузеров почти идентичен.

SeaMonkey Navigator многие ругают за «отсталый» дизайн, но для публики, которая нервно реагирует на очередное «упрощение» интерфейса, такой консерватизм будет очень кстати: традиционное размещение кнопок на панели навигации вкупе с широкими возможностями настройки — это настоящий восторг для любителей опрятного и чистого интерфейса.



Opera 12

Последняя версия на оригинальном движке Presto. Мы сознательно не берем новую *Opera 24* для разработчиков, т. к. это очередной клон *Chrome* с мелкими изменениями. Нас интересует «настоящая» *Opera* — своего рода «нож швейцарской армии» с невообразимо богатым набором функций. Помимо почтового клиента, здесь имеются менеджер загрузок с поддержкой Torrent и возобновления приостановленных загрузок, службы Opera Link (синхронизация настроек) и Opera Turbo (экономия трафика), поддержка жестов мыши и управления голосом... всего и не перечислишь. При обычном web-сёрфинге *Opera* радует функцией редактирования исходного кода страницы (результат вы увидите сразу), удобной Корзиной (для блокировки страниц или их элементов),

удобным выделением текста и много чем ещё. Чувствуется, что ещё недавно этот проприетарный продукт отчаянно пытался вернуть себе былую славу на настольных компьютерах — настолько качественно реализованы многие функции. Но сегодня нас будет интересовать, насколько этот браузер современен.



QupZilla

Этот быстрый и умелый браузер с Qt-интерфейсом существует с 2010 г. Как можно догадаться по названию, *QupZilla* неплохо вписывается в среду KDE, но, несмотря на отсылку к Mozilla, применяет для рендеринга движок WebKit, а не Gecko, как многие считают. *QupZilla* был написан на Python и привязан к PyQt к графической библиотеке Qt, но впоследствии переведен на C++. Ныне позиционируется как легковесный и быстрый браузер с малым потреблением памяти (да, шутка про наглуемую морду всё ещё актуальна). Кроме того, благодаря замечательной Qt, *QupZilla* одинаково опрятно выглядит в любом рабочем окружении. Но дело тут не только во внешнем виде: *QupZilla* способен интегрироваться с *gnome-keyring* в Gnome и с *KWallet* в KDE и содержит массу удобных мелочей, облегчающих работу с URL, скачивание файлов по FTP и многое другое.



Konqueror

Этот браузер по умолчанию в среде KDE (Plasma Desktop) и настоящий старожил в мире web-браузеров для Linux: *Konqueror* появился аж в 2000 г. (старше него только *Opera*). Изначально он использовал оригинальный движок KHTML, который в 2001 г. заимствовали инженеры Apple для написания WebKit. Поскольку обе технологии были открытыми, уже в 2005 году готовый WebKit пришёл обратно в KDE. С тех пор *Konqueror* остаётся редким браузером, допускающим переключение между KHTML

и WebKit почти на лету. С точки зрения совместимости с современными сайтами, преимуществом обладает, конечно же, WebKit. Однако браузер — это больше чем движок, и будет интересно поглядеть, как *Konqueror* проявит себя в деле.



Rekonq

Этот KDE-браузер на Webkit — ещё один легковесный, поджарый браузер, который представляет собой попытку осовременить стандартный инструмент просмотра web-страниц в KDE. В отличие от *Konqueror*, который плотно интегрирован в KDE и не распространяется отдельно, *Rekonq* позиционируется как обособленное приложение с более простым и минималистичным интерфейсом. Данный браузер умеет использовать уведомления KDE и функциональность KIO-модулей, и явно метит в стандартные браузеры этого рабочего стола. *Rekonq* по умолчанию используется в дистрибутивах Chakra и Kubuntu, однако к его функциональности ещё стоит присмотреться: редко когда минимализм не требовал жертв.



Midori

А вот вам минимализм в стиле Elementary. Стильный *Midori* известен не только пользователям дизайнерской ElementaryOS. Браузер входит в поставку рабочего окружения *Xfce*, а также включён по умолчанию в некоторые легковесные дистрибутивы, такие как SliTaz, Bodhi и Trisquel Mini. *Midori* — очередной приверженец Webkit с интерфейсом на GTK2, впрочем, совместимый и с GTK3. Среди поддерживаемых функций стоит отметить жесты мышью и «быстрый набор», явно позаимствованные из *Opera*, интеграция с ОС Maemo на мобильном устройстве, поддержка папок внутри коллекции закладок и прочее. По большей части, дополнительные функции реализованы в виде расширений на языке Vala.

Соответствие web-стандартам

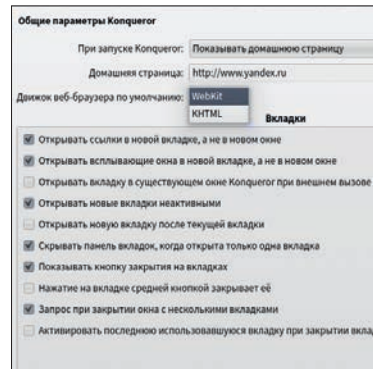
Именитые тесты, освященные временем.

Тест Acid3, представленный в 2008 г., оказался в своё время трудным испытанием для *Firefox* и *Safari*. В наши дни все современные браузеры уже научились набирать в нём 100 баллов из 100. Но и тут не обошлось без сюрпризов. Старичок *Konqueror* — единственный из всех участников — набрал всего 92 балла, причём смена движка с KHTML на Webkit ему ничем не помогла.

Ситуация с поддержкой HTML5 куда интереснее. Лидирует в рейтинге *SeaMonkey Navigator* с 467 баллами — по этому показателю он полностью соответствует свежему *Firefox*, откуда, видимо,

и заимствуется часть кодовой базы. Далее за ним следует *QupZilla* с 418 баллами, затем — с более-менее равномерными отрывами — *Opera* и *Midori*. Примечательно, что ещё совсем недавно Opera Software рекламировала свой продукт как вполне отвечающий стандартам WWW; теперь же несостоятельность этого утверждения стала наглядной.

Rekonq уверенно обходит *Konqueror* (259 баллов против 103), но не будем забывать о том, что всё относительно. Браузер из стандартной поставки Chakra и Kubuntu всё ещё критически отстаёт в деле поддержки HTML5.



» *Konqueror*: раз-два — и готово: движок переключается «на лету», весьма удобно!

Вердикт

- SeaMonkey ★★★★★
 - QupZilla ★★★★★
 - Midori ★★★★★
 - Opera ★★★★★
 - Rekonq ★★★★★
 - Konqueror (Webkit) ★★★★★
 - Konqueror (KHTML) ★★★★★
- » SeaMonkey и QupZilla возглавили данный рейтинг, заслужив за 400 баллов.

Много-много вкладок...

Как наши браузеры поведут себя под нагрузкой?

Чтобы данное задание смогло работать для *SeaMonkey Navigator*, пришлось вручную отключить блокировку всплывающих окон в настройках браузера и попросить его открывать ссылки не в новых окнах, а во вкладках одного окна. После этого *SeaMonkey Navigator* справился с задачей за 5:45 минут, заняв около 1 ГБ ОЗУ. По окончании загрузки данных на страницах переключение страниц и прокрутка шли достаточно быстро, хоть и не идеально плавно; нагрузка на ЦП составляла 30–40%. Однако работать с сайтами было вполне комфортно.

Opera 12 по умолчанию блокирует всплывающие окна, в каждом случае уведомляя. Чтобы задание автоматически открывало много вкладок, эта блокировка была вручную отключена в расширенных настройках. Первые пол-теста *Opera* вела себя бодро, ужимая панель вкладок в реальном времени. Но конец был печален: потратив 9:15 на открытие вкладок, браузер извел все 1,5 ГБ ОЗУ и залез в Swap. При нагрузке ЦП в 100% система зависла, вплоть до паралича мыши. Ещё 10 минут ожидания, и тест решено было прекратить. Неудачу усугубил ещё и непродуманный интерфейс *Opera*: при увеличении числа открытых вкладок их размер съеживался до нескольких пикселей, а прокрутки (кроме как с клавиатуры) не предусмотрено.

Konqueror с движком KHTML потерял управляемость уже после 20-й вкладки, став чемпионом по скорости расходования

памяти: уже через 2 минуты он обратился к подкачке, исчерпав ОЗУ. *Konqueror* не открыл и половины набора вкладок. Не помогло и переключение на Webkit, лишь продлив агонию на пару минут.

В *Midori* примерно за 8:30 минут удалось открыть все 150 вкладок, но работоспособными были лишь первые 50–60. Первые 5 минут браузер сохранял живучесть, позволяя переключаться между вкладками, но отзывчивость интерфейса падала на глазах. Ещё через несколько минут исчерпались ОЗУ и Swap, и система перестала отвечать на действия пользователя. Однако, в отличие от *Opera*, при своевременной реакции можно вовремя завершить процесс браузера ещё до того, как производительность упадёт до нуля.

Браузер *QupZilla* оказался единственным, у кого вообще отсутствуют средства открытия набора URL за один приём. *QupZilla* не воспринимает вызов `window.open` из скрипта MemBench и его аналогов, не берет более одного аргумента при запуске из командной строки и не имеет настроек, связанных с блокировкой всплывающих окон. Для проведения теста оставался единственный способ: сервис типа brief.ly. *QupZilla* показал себя намного лучше, чем *Konqueror*, *Opera* и *Midori*, но полностью выполнить задание всё же не смог: потребление памяти при открытии 100 вкладок (это ограничение у brief.ly) уже составляло 1,3 ГБ, браузер справился с заданием за 5 минут. Оставшиеся

50 вкладок открыть уже не удалось: отзывчивость браузер сохранял, но работать с ним было почти невозможно.

Rekonq тут же приготовил все 150 вкладок и стал уверенно загружать в них контент. Через пару минут начались просьбы остановить выполнение JavaScript на некоторых страницах, а также уведомления о просроченных SSL-сертификатах; все диалоги пришлось закрыть вручную. После множества предупреждений *Rekonq* сумел загрузить содержимое почти всех вкладок (на что ушло около 8 минут), а затем обрушился. Однако надо отдать ему должное: даже потратив 1,2 ГБ ОЗУ, браузер всё это время сохранял систему полностью отзывчивой и нисколько не притормаживал её. Если добавить чуть-чуть стабильности и корректную обработку уведомлений, *Rekonq* станет очень устойчивым вариантом для работы в условиях интенсивной нагрузки.

Однако нельзя однозначно утверждать, что браузеры, не осилившие открытие 150 вкладок, непригодны для подобных задач: вопрос скорее в эффективности расходования системных ресурсов. Так, в порядке эксперимента, *Opera* со скриптом MemBench была дополнительно проверена на более мощной машине с 8 ГБ ОЗУ, где задание успешно выполнилось — но за это пришлось заплатить более 4 ГБ использованной памяти. Едва ли такой результат служит рекламой «лёгкому и эффективному» web-браузеру.

Вердикт

- SeaMonkey ★★★★★
 - Opera ★★★★★
 - Rekonq ★★★★★
 - Midori ★★★★★
 - QupZilla ★★★★★
 - Konqueror ★★★★★
- » «Памяти много не бывает», — дружно сказали все, кроме SeaMonkey.

Скорость загрузки и кэширование

Длительное ожидание — это несносно.

Все шесть браузеров смогли загрузить гигабайтную страницу с GIF-изображениями. Замеры делались так: сначала браузер с «чистым» профилем (т.е. без настроек, истории, cookies и кэша) загружал страницу в первый раз — это т.н. «холодный старт». Для испытаний был специально организован быстрый интернет-канал с шейпером трафика на стороне маршрутизатора — это позволило исключить влияние самого канала на результаты теста. После успешной загрузки страницы она через некоторое время перезагружалась [reload]. Затем данное действие повторялось ещё раз. Смысл здесь в наблюдении за механизмом кэширования данных в браузере. Зачастую (см. таблицу на стр. 33), третий «заход» на страницу давался браузерам тяжелее второго. Наконец, тестировался «тёплый старт» — закрытие и повторное открытие браузера с последующей загрузкой той же страницы, которая уже была в кэше браузера. Посмотрим же на результаты.

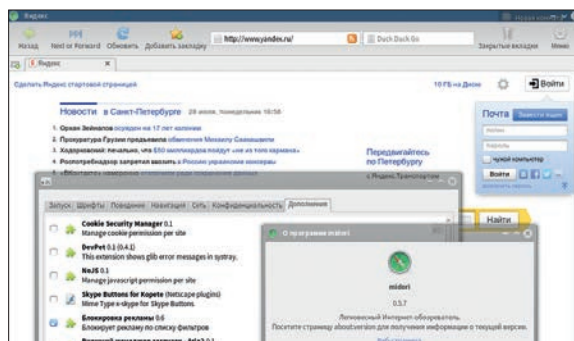
Абсолютным победителем по скорости стал браузер *QupZilla*, который не только хорошо разбирается в стандарте HTML5, но и уверенно загружает традиционные web-страницы с несложной разметкой. У *QupZilla* лишь немного возросло время на третьей попытке загрузить страницу,

после чего при «тёплом старте» браузер отстрелялся всего за 17 с.

Вторым по скорости «холодного» старта оказался *Midori*, справившись с заданием за 25 с. Однако последующие загрузки дались ему труднее. Загрузив страницу довольно быстро в первый раз, браузер уходил в глубокие раздумья на последующих попытках. Как ни странно, «тёплый старт» для *Midori* оказался медленнее холодного. Дальнейшее исследование выявило, что производительность *Midori* явно деградирует по мере разрастания кэша и профиля в целом.

SeaMonkey Navigator сначала долго раскачивался и загружал страницу дольше минуты. Зато дальнейшая работа с кэшем здесь организована куда грамотнее: повторные загрузки проходят на ура, а «тёплый старт» занял рекордные 15 с. *SeaMonkey Navigator* — отличный, быстрый браузер, особенно если сначала «познакомить» его со страницами, которые вы посещаете.

Достаточно бледно на фоне победителей выглядит *Opera 12.16*. Браузер демонстрирует средние значения по скорости загрузки, однако более любопытным оказался механизм слежения за активностью пользователя. Только что загруженная страница почти мгновенно может



» Особо быстро *Midori* работает вначале, пока кэш и история посещений невелики.

быть перезагружена. Но если пользователь становится на некоторое время неактивен, браузер при перезагрузке начинает «вытягивать» контент по новой. В целом чувствуется, что *Opera* работает не спеша.

Сзади всех плетутся *Konqueror* и *Rekonq*. Удивительно, но *Rekonq* полностью провалил возложенные на него ожидания и оказался медленнее, чем *Konqueror* с движком Webkit. Хотя и здесь следует оговориться: *Konqueror* на традиционном движке KHTML вообще не справился с задачей, за 2 минуты слопал всю память, а затем — весь раздел подкачки. Всё это время ядра ЦП были загружены почти на 100%.

Вердикт

QupZilla
★★★★★
Konqueror (Webkit)
★★★★★
Opera
★★★★★
SeaMonkey
★★★★★
Midori
★★★★★
Rekonq
★★★★★
Konqueror (KHTML)
★★★★★

» Пример *Konqueror* показал, как много зависит от движка.

Блокировка рекламы

Реклама отвлекает от контента. Надо бы пресечь рекламные паузы.

Расширение Adblock технически несложно: это набор фильтров, отсеивающих нежелательные баннеры и ролики. Тестовым объектом стал киносайт, найденный запросом «скачать фильмы бесплатно торрент» (eritern.com/films.html): он искрился «партнёрскими» баннерами и раздражал всплыванием окон.

У *SeaMonkey Navigator* система расширений и их список напоминают *Firefox*. Прямо в браузере мигмом находится несколько блокировщиков; из них самый популярный и качественный — Adblock Plus. Установив его, просто обновите страницу, не перезапуская браузер. Сработал Adblock Plus отлично: подавил абсолютно всю рекламу, без ущерба для основного контента.

Сходный результат дала *Opera*. Здесь масса отличных расширений, что иногда

извиняет огрехи, и Adblock Plus ставится через фирменный каталог и применяется сразу же. Рекламу как ветром сдуло!

У *QupZilla* каталога нет: проект еще юн. Adblock засунут в меню Инструменты. Для включения надо отметить его флажок; рядом — список фильтров из базы Easy List. Подавив рекламу, блокировщик от *QupZilla* не осилил всплывающее окно, мимикрирующее под уведомление Вконтакте, а вместо одного из баннеров зависла надпись «Загрузка...». Твёрдая четвёрка.

Konqueror позабавил контрастом работы при разных движках: корректно всё загрузить браузер смог только с WebKit. Применение его с KHTML само способно заблокировать часть рекламы. Но у него есть встроенный блокировщик, и пункт Включить фильтры уже по умолчанию отмечен

флажком; сами же фильтры неактивны. Надо вручную отметить нужные наборы фильтров из списка, где их более 30 (для разных стран), да еще флажок Скрыть фильтруемые изображения. Увы, даже со всеми наборами 60% рекламы уцелело. Убралось лишь несколько баннеров.

На диво хорошо блокируется реклама в *Rekonq*, хотя диалог настройки очень напоминает *Konqueror*! Опыт показал, что баннеры и всплывающие окна подавляет сочетание RuAdlist и EasyList, позволяя *Rekonq* смотреться выигрышнее *QupZilla*.

Встроенный в *Midori* блокировщик минималистичен, как и сам браузер: здесь лишь два списка от EasyList, но можно добавлять свои списки. Блокировщик оказался не хуже *QupZilla*: подавил всё, кроме вконтакте-подобного всплывающего окна.

Вердикт

Opera
★★★★★
Rekonq
★★★★★
SeaMonkey
★★★★★
Midori
★★★★★
QupZilla
★★★★★
Konqueror (KHTML)
★★★★★
Konqueror (Webkit)
★★★★★

» С несложным заданием справились все. Ну, почти все.

Поддержка Adobe Flash

Ох уж эти проприетарные технологии...

Сложно ожидать, что в наше время какой-либо web-браузер не сможет правильно отображать Flash-ролики. К сожалению, это представление оказывается романтическим.

В действительности, даже такая тривиальная задача под силу не всем. *Konqueror* по-разному некорректно пытался осилить случайный ролик с YouTube обоими своими движками, *Midori* не поддерживает «родную» 64-битную версию Adobe Flash — на 64-битных системах вам потребуется

делать обвязку из 32-битного плагина и *nspluginwrapper*, что не слишком удобно.

Наконец, *QupZilla* корректно воспроизводит один только звук, показывая вместо картинки чёрный прямоугольник (почти по Малевичу). *Rekonq* справился с YouTube в целом успешно, практически не уступая в проворности своим старшим товарищам по данному Сравнению.

Полноценную поддержку Adobe Flash де-факто имеют *SeaMonkey Navigator*, *Opera* и *Rekonq*.



» Каталог расширений Opera — настоящая жемчужина. Тут есть всё и даже больше...

Вердикт

- Opera ★★★★★
- SeaMonkey ★★★★★
- Rekonq ★★★★★
- Midori ★★★☆☆
- QupZilla ★★☆☆☆
- Konqueror ★☆☆☆☆

» Некоторым претендентам в пору надеяться на массовый HTML5...

Поддержка WebGL

Без игр жилось бы скучновато.

Поддержка технологии WebGL нужна современным браузерам для более зрелищной и динамичной работы с мультимедиа-данными — в первую очередь инди-играми вроде *Angry Birds*, которые в последнее время активно мигрируют из стана standalone-приложений в разряд браузерных игр. Преимущество WebGL перед тем же Flash очевидно: задействование аппаратного ускорения графики положительно сказывается как на 3D-возможностях игр, так и на энергоэффективности. Поддержка WebGL тем или иным web-браузером легко проверяется на тестовой страничке <http://freebrowsers.ru/technology/webgl>.

при этом результат не должен противоречить реальным «полевым испытаниям», для коих выбрано WebGL-приложение

<http://chrome.angrybirds.com/>. Результаты показаны в таблице ниже, и по ним видно, что успехом вновь могут похвастаться лишь *SeaMonkey Navigator* и *Opera*: в них WebGL-приложение не только корректно запускается, но и плавно работает. Остальные не справились ещё на этапе загрузки, показав капитулянтскую рекомендацию установить *Google Chrome*.

Браузер	Формальная поддержка	Работа WebGL-приложения
Konqueror+KHTML	✓	✗
Konqueror+Webkit	✓	✗
Midori	✓	✗
Opera	✓	✓
QupZilla	✓	✗
Rekonq	✓	✗
SeaMonkey Navigator	✓	✓

Вердикт

- Opera ★★★★★
- SeaMonkey ★★★★★
- Konqueror ★☆☆☆☆
- Midori ★☆☆☆☆
- QupZilla ★☆☆☆☆
- Rekonq ★☆☆☆☆

» Момент истины: новая технология доступна только в Opera и SeaMonkey.

Камера и микрофон

Лучше один раз увидеть... но и послушать тоже хочется.

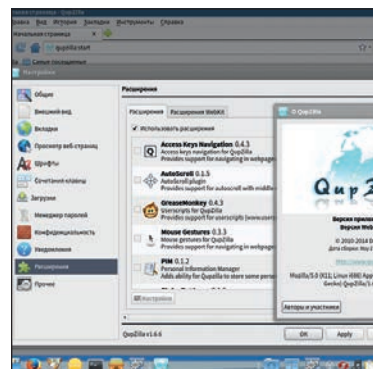
В нашем последнем тесте для проверки работы камеры и микрофона был использован популярный ресурс chattino.com, один из аналогов ChatRoulette. Несмотря на то, что интеграция браузера с камерой и микрофоном требует подтверждения через специальный диалог плагина Adobe Flash, результаты для наших подопытных браузеров вышли более обнадеживающими, чем в случае с Youtube.

Во время данного испытания вышло целых два сюрприза. Во-первых, *Opera*

отказалась работать с web-чатами вообще — показав на chattino.com и аналогичных ресурсах пустую страницу, что удивительно для браузера, в котором так хорошо работают WebGL и Flash.

Второй, уже приятный сюрприз — это совершенно неожиданный прорыв *Konqueror* с движком Webkit: браузер работает с чатом идеально, чего не скажешь о других технологиях.

» У *QupZilla* не так много расширений, зато они всегда находятся под рукой.



Вердикт

- Konqueror (Webkit) ★★★★★
- SeaMonkey ★★★★★
- Midori ★★★★★
- QupZilla ★★★★★
- Rekonq ★★★★★
- Opera ★☆☆☆☆
- Konqueror (KHTML) ★☆☆☆☆

» Стремительная контратака Konqueror... и неожиданный провал Opera.

Web-браузеры второго эшелона

Вердикт

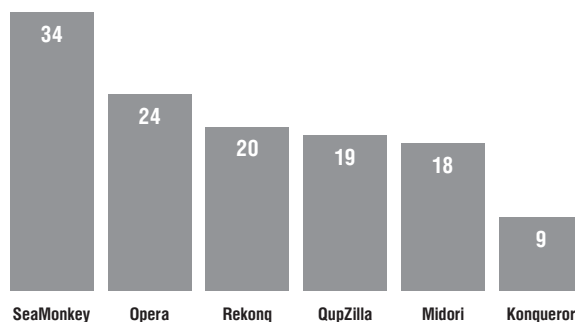
Однозначный победитель в нашем тесте — *SeaMonkey Navigator*. Это браузер старой школы, но под капотом у него новейшая версия Mozilla Gecko и отличная поддержка всех новых технологий. По сути, это отличная альтернатива *Firefox*, отличающаяся от своего именитого собрата меньшим потреблением памяти и отсутствием экспериментов в части интерфейса. Даже удивительно, что известность *SeaMonkey Navigator* так кардинально расходится с его качеством.

Второе место заслуженно досталось *Opera 12.16* на классическом движке Presto. Этот браузер уже не так быстр, как в былые времена, да и некоторые штуки он уже не всегда поддерживает, однако широкий набор функций и отсутствие критических недочётов позволяют и сегодня

использовать *Opera* в качестве основного браузера.

На третьем и четвертом месте расположились *QupZilla* и *Rekonq*. Увы, оба проекта так и не справились со своей задачей — предоставить быстрый и качественный браузер с Qt-интерфейсом. Их можно использовать при ограниченном круге задач, когда требуется корректная работа лишь нескольких функций. К тому же оба этих Webkit-браузера быстро теряют скорость в условиях интенсивной нагрузки, а это может быть важным для многих.

Приятный глазу *Midori* оказался на деле весьма невыразительным браузером. Минимализм по части функциональности, нестабильность работы и плохая поддержка мультимедиа явно не играют ему на руку.



Самые скромные возможности продемонстрировал *Konqueror*. Конечно, использование Webkit даёт ему весомое преимущество, но бороться с соперниками «завоевателю» всё равно не по зубам. Едва ли этот браузер ждёт большое будущее.

» Для наглядности мы подсчитали усредненные общие баллы наших браузеров.

I **SeaMonkey** ★★★★★
 Сайт: mozilla-russia.org Лицензия: MPL, GPL, LGPL Версия: 2.26.1
 » Похоже, «классике» жить ещё долго — SeaMonkey в этом убедил.

IV **Rekonq** ★★★★☆
 Сайт: rekonq.kde.org Лицензия: GPL Версия: 2.4.2
 » Неплохо подойдёт для не очень взыскательных пользователей.

II **Opera** ★★★★☆
 Сайт: www.opera.com Лицензия: Проприетарная Версия: 12.16
 » Даже спустя год, эта версия ещё хорошо работает.

V **Midori** ★★★★☆
 Сайт: www.midori-browser.org Лицензия: GPL Версия: 0.5.7
 » Стоит брать только за название и внешнюю миловидность.

III **QupZilla** ★★★★☆
 Сайт: www.qupzilla.com Лицензия: GPL Версия: 1.6.6
 » Юный, дерзкий, перспективный — оправдает ли он доверие?

VI **Konqueror** ★★★☆☆
 Сайт: www.konqueror.org Лицензия: GPL Версия: 4.13
 » Интересный и самобытный экспонат из музея web-браузеров.

Сводная таблица результатов тестирования

Браузер	Потребляемая память на yandex.ru, МБ	Результат Acid3	Результат HTML5	Скорость загрузки*, с
SeaMonkey Navigator	60	100/100	467/555	76/10/6/15
Opera 12	42	100/100	392/555	65/3/1/30
QupZilla	57	100/100	418/555	25/4/9/17
Konqueror (Webkit)	38	92/100	103/555	67/34/36/33
Konqueror (KHTML)	33	92/100	103/555	С задачей не справился
Rekonq	30	100/100	259/555	92/34/39/35
Midori	42	100/100	378/555	25/12/44/31

* Холодный старт/обновление/ещё обновление/тёплый старт страницы <http://gif.d3.ru/comments/474379>

КАК UBUNTU

Ломайте, настраивайте и отлаживайте Ubuntu, вместе с Маянком Шармой превращая его в дистрибутив мечты — как для новичка, так и для ветерана.



U buntu — вероятно, единственный дистрибутив, который в равной мере вызывает к себе как любовь, так и явную неприязнь. Одни его ненавидят за функции, влияющие на удобство использования; другие восхваляют за то, что он расширяет горизонты.

Самый последний релиз, 14.04, названный Trusty Tahr, это 20-й релиз 10-летнего дистрибутива. Это релиз с долгосрочной поддержкой [Long Term Support, LTS], поэтому в отличие от обычных релизов на полгода он будет поддерживаться и обновляться в течение пяти лет. Поскольку это релиз LTS, основное внимание во время разработки уделялось его работе и отладке, а не новым функциям. Именно поэтому он не слишком отличается внешне от предыдущего релиза (см. Обзоры LXF185, стр. 14, для более подробной информации).

Однако, в отличие от предыдущих релизов, 14.04 нацелен на то, чтобы привлечь больше внимания, чем когда-либо прежде. Это первый релиз популярного дистрибутива, появившийся в мире пост-Windows-XP. За десять лет своего существования Ubuntu достаточно созрел, чтобы позиционировать себя в качестве серьезного претендента на роль серьезной ОС рабочего стола. Подобно другим популярным ОС, Ubuntu готова к использованию немедленно, и при этом в ней достаточно настраиваемых элементов, чтобы пользователь получил то, что ему требуется. Благодаря куда более широкой экосистеме открытого кода, которая

обеспечивает большой выбор настроек и инструментов, дистрибутив предлагает очень гибкую среду, которую можно адаптировать к любой настольной машине.

В нашей статье мы поможем вам установить дистрибутив на ваш компьютер и затем познакомиться с разными возможностями настройки установленной системы в соответствии с вашими рабочими потребностями. Самый очевидный (и для многих — самый неприятный) аспект дистрибутива — рабочий стол Unity. Однако Unity развился в весьма зрелую среду рабочего стола, которая предлагает массу обходных путей своих странностей. Игры, которые являются самой явной Ахиллесовой пятой Linux, постепенно превращаются в его сильную сторону, и мы поможем вам задействовать все игровые возможности вашей машины в полную силу. Так что закатывайте рукава — и за дело.

«Ubuntu любят и ненавидят в равной мере, но он весьма гибок и адаптируем.»

нироваться себя в качестве серьезного претендента на роль серьезной ОС рабочего стола. Подобно другим популярным ОС, Ubuntu готова к использованию немедленно, и при этом в ней достаточно настраиваемых элементов, чтобы пользователь получил то, что ему требуется. Благодаря куда более широкой экосистеме открытого кода, которая

Установим Ubuntu

Начнем с отведения дистрибутиву кусочка вашего жесткого диска...

Для большинства пользователей путешествие в Linux начинается с программы установки. Именно поэтому очень мало компьютеров идут с предустановленным Linux. Но даже если такой найдется, велика вероятность, что через полгода вы удалите установленную версию ради нового релиза или другого дистрибутива.

Поскольку установщик является столь важной частью работы в Linux, основные дистрибутивы уделяют ему особое внимание. Технически все дистрибутивы одинаковы; отличают их друг от друга структура и способ использования. И Ubuntu обладает одним из лучших установщиков. Он отлично структурирован и достаточно гибок, чтобы устроить и новичков, и опытных пользователей Linux.

Дистрибутивы Linux идут в виде ISO-образов. Это архивные файлы оптического диска, которые можно записать на CD или DVD, в зависимости от их размера. (При записи диска выберите в своей программе прожига опцию записи ISO-образа). Еще лучше — создать инструментом вроде *Unetbootin* и *YUMI* загрузаемый диск USB с файлом ISO. Позднее можно будет переформатировать диск USB для постоянного использования.

Все наоборот

Создав загрузаемый диск с ISO-образом Ubuntu, перезагрузите свой компьютер, установив в него флэшку USB. Если он загружает уже установленную ОС, перезагрузите компьютер, зайдите в BIOS и настройте опцию загрузки с подключенного внешнего устройства или оптического диска. В каждом BIOS настройки для этого свои.

Устанавливая Ubuntu на ПК с Windows 8, отключите FastStartup в Control Panel в Windows.

Если у вас возникает ошибка подписи [signature error] при загрузке с устройства, содержащего ISO-образ Ubuntu, придется отключить SecureBoot в BIOS. При успехе загрузки Ubuntu предложит вам загрузиться в среду live или перейти прямо в установщик, но перед установкой следует ознакомиться с некоторыми техническими деталями.

Один из самых важных шагов при установке Linux — разбиение диска на разделы. В наши дни большинство установщиков, включая программу установки Ubuntu, могут сканировать жесткий диск и автоматически высвободить место для установки. Опционально вы можете вручную

вы можете указать установщику Ubuntu создать раздел для себя на этом свободном месте.

Затем установщик предложит вам выбрать для этого раздела файловую систему (ФС). В широком смысле, ФС управляет тем, как именно данные на диске хранятся и удаляются. Windows использует файловую систему NTFS, а Linux — ext (в настоящее время — до ext4).

Еще один термин, который нужно знать — загрузчик [bootloader]. Это небольшая программка, которая сообщает компьютеру, где найти разные операционные системы, установленные на жестком диске. Ubuntu, подобно большинству популярных настольных дистрибутивов, использует загрузчик *Grub 2*. Установщик спросит вас, куда ему установить загрузчик. По умолчанию *Grub 2* располагается в Master Boot

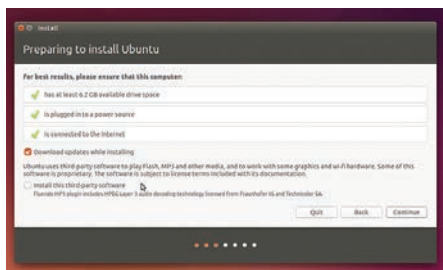
«Установщик достаточно гибок, чтобы устроить и новичков, и опытных.»

Record, помещенной в начале жесткого диска. Новым пользователям также нужно знать, как маркируются жесткие диски в Ubuntu (и в Linux в целом). В отличие от Windows, диски маркируются в соответствии с драйвером используемого ими устройства. Обычно жесткие диски используют драйвер массового запоминающего устройства, и имеют префикс **sd**. За префиксом следует буква для определения отдельных устройств; таким образом, **sda** — это первый жесткий диск. Далее идет цифра для определения отдельных разделов. На дисках со схемой разделов MBR первые и опциональные расширенные разделы нумеруются от 1 до 4, а логические разделы — от 5 и далее, вне зависимости от схемы бывших разделов.

Помимо установщика Ubuntu, вы можете использовать специальные инструменты для разбиения диска на разделы, например, *Gparted*, или даже создать раздел для Ubuntu из Windows при помощи инструмента Disk Management. Чтобы высвободить место для Linux, запустите инструмент, щелкните правой кнопкой по диску, на который вы хотите установить Ubuntu, и выберите опцию Shrink Volume. В появившемся диалоговом окне вы можете указать, насколько вы хотите сжать диск. Существующий раздел будет сжат, и на диске создастся свободное место. Затем

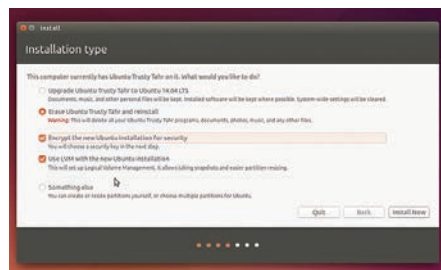
Record, помещенной в начале жесткого диска. Новым пользователям также нужно знать, как маркируются жесткие диски в Ubuntu (и в Linux в целом). В отличие от Windows, диски маркируются в соответствии с драйвером используемого ими устройства. Обычно жесткие диски используют драйвер массового запоминающего устройства, и имеют префикс **sd**. За префиксом следует буква для определения отдельных устройств; таким образом, **sda** — это первый жесткий диск. Далее идет цифра для определения отдельных разделов. На дисках со схемой разделов MBR первые и опциональные расширенные разделы нумеруются от 1 до 4, а логические разделы — от 5 и далее, вне зависимости от схемы бывших разделов.

Устанавливаем Ubuntu



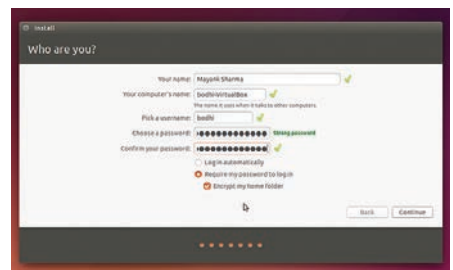
1 Подготовка к установке

Когда вы запустите установщик Ubuntu, тот, помимо всего прочего, запустит таблицу контрольных проверок, с целью убедиться, что у вас достаточно места для установки Ubuntu. Вы увидите две кнопки-флажка. Первая будет находить все имеющиеся обновления в Интернете, а другая установит плагин, необходимый для воспроизведения MP3-контента.



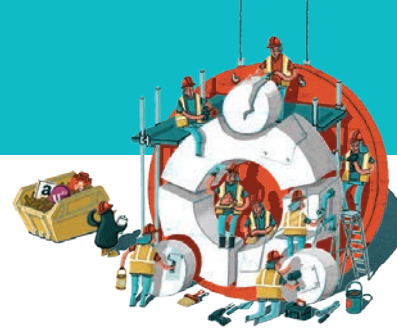
2 Схема разбиения на разделы

Разбиение на разделы — очень важный шаг. Возможности здесь будут зависеть от состояния жесткого диска компьютера. Пользователи Windows получат опцию дать возможность программе установки перераспределить объемы разделов и выделить место под Ubuntu. Есть и опция для продвинутых пользователей, предпочитающих разбиение диска вручную.



3 Логин и прочие детали

Это вся информация, которая нужна установщику для начала установки Ubuntu. Пока файлы копируются на диск в фоновом режиме, установщик спросит у вас о вашем местоположении и часовом поясе, а также о раскладке клавиатуры. На последнем экране вам будет предложено ввести выбранный логин и пароль.



Общие отладки Unity

Уберем самые общие проблемы стандартного настольного Ubuntu.

Разработчики Ubuntu предлагают дистрибутив, полезный для большинства пользователей. И хотя вы вполне можете начать работать с дистрибутивом, не меняя настроек по умолчанию, со временем вы почувствуете необходимость изменить систему так, чтобы она больше соответствовала вашим личным потребностям. **» Добавьте значки в программу запуска приложений** Программа запуска приложений [Launcher] Ubuntu является важнейшим компонентом, обеспечивающим удобство в использовании. Помимо того, что она размещает в себе значки для наиболее часто используемых приложений, она также отслеживает открытые окна. Однако место для значков в программе запуска ограничено. Дополнительные значки внизу впоследствии сжимаются, и чтобы толком их разглядеть, придется навести на них мышью. Со временем это начинает раздражать, особенно если вы управляете множеством приложений и окон.

Для решения этой проблемы вы можете изменить размер значков в программе запуска, чтобы вместить как можно большее их количество

в ограниченное пространство. Размер значка по умолчанию — 48 пикселей. В Ubuntu до версии 13.10 можно снизить его до 32, а самый последний релиз позволяет довести эту цифру до 16.

Чтобы настроить размер значков в программе настройки, перейдите в System Settings и щелкните по Appearance в разделе Personal. Во вкладке Look используйте бегунок рядом с Launcher Icon Size, чтобы настроить размер значка.

» Автоматически скрывать программу запуска На более маленьких экранах и при большем разрешении программа запуска приложений занимает слишком много места на экране. Одним из способов устранить эту проблему будет автоматически скрывать программу запуска, если она не используется. Для этого перейдите в System Settings > Appearance и переключитесь на вкладку Behavior. Здесь вы найдете опцию Auto-Hide the Launcher; установите ее в On.

Теперь программа запуска будет появляться только когда вы ее вызовете. Чтобы увидеть ее, переместите мышью в левую часть экрана. Чтобы настроить яркость отображения, вернитесь

во вкладку Behavior в разделе Appearance и используйте бегунок Reveal sensitivity.

» Показывать программу запуска Кроме опции Reveal sensitivity, у вас имеется возможность настроить и некоторые другие параметры для мыши, чтобы лучше контролировать поведение программы запуска. Для этого понадобится *CompizConfig Settings*

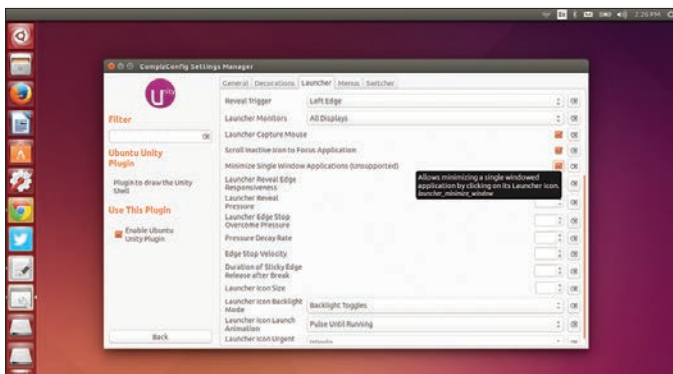
Manager (CCSM). Запустите приложение и выберите Desktop в левой панели, затем щелкните по опции Ubuntu Unity Plugin. Перейдите во вкладку Launcher и настройте параметры Launcher Reveal Pressure и Launcher Reveal Edge Responsiveness, чтобы контролировать вид отображения программы запуска приложений.

» Щелкните для минимизации По умолчанию, если вы щелкнете по значку открытого приложения в используемой на данный момент программе запуска, ничего не произойдет. Однако в Ubuntu 14.04 имеется скрытая функция, которая, если ее включить, позволяет минимизировать открытые окна щелчком по их значкам.

Чтобы включить эту функцию, воспользуйтесь приложением *CCSM*. Запустите приложение, выберите в левой панели Desktop, перейдите в раздел Ubuntu Unity Plugin и переключитесь на вкладку Launcher. Здесь щелкните по кнопке-флажку рядом с опцией Minimize Single Window Applications (Unsupported). Надо сделать оговорку — эта функция работает только с приложениями одного окна. Если у вас в приложении открыто несколько окон, то вместо нее щелчок по значку запустит Spread view [развернуть].

Настройка горячего угла

При разумном использовании, горячие углы могут способствовать вашей работе, позволяя быстрее и проще управляться на рабочем столе. Рабочий стол Unity предлагает восемь горячих углов, и их поведение настраивается в приложении *Unity Tweak Tool*. Запустите приложение и щелкните по разделу Hotcorners в разделе Window Manager. Здесь переключите Hotcorners с Off на On. Теперь вы можете настроить поведение восьми горячих углов, выбрав одно из предустановленных действий с помощью выпадающего меню рядом с каждым горячим углом.



» Вы можете использовать CCSM, чтобы активировать функции Compiz, в том числе пульсирующие окна и рабочий стол в виде куба.

Обязательные инструменты настройки Ubuntu

Хотя вы можете настроить разные аспекты рабочего стола с помощью встроенных инструментов Ubuntu, возможности индивидуальной настройки в дистрибутиве ограничены. Если настройка вам действительно нужна, установите ряд дополнительных инструментов.

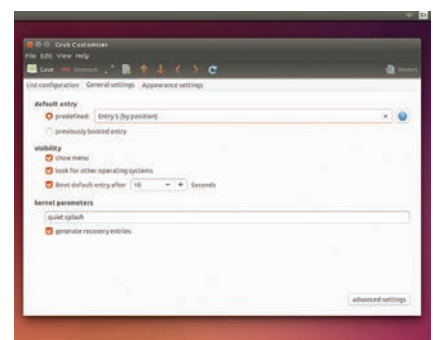
» Unity — это плагин менеджера окон *Compiz*. Чтобы тщательно настроить разные аспекты *Compiz*, вам нужно приложение *CompizConfig Settings Manager (CCSM)*. Для установки *CCSM* перейдите в терминал и введите `sudo apt-get install compizconfig-settings-manager`.

» Используйте *Unity Tweak Tool* для индивидуальной настройки многих опций настройки рабочего стола и системы. Этот инструмент также доступен

в официальных репозиториях; установите его с помощью `sudo apt-get install unity-tweak-tool`.

» QLE Quick List Editor может создавать и редактировать Quicklists для значков в программе запуска. Для его установки сперва добавьте его PPA командой `sudo add-apt-repository ppa:vlijm/qle`, затем обновите репозиторий — `sudo apt-get update`, и, наконец, установите инструмент: `sudo apt-get install qle`.

» Наконец, установите приложение *Grub Customizer* для индивидуальной настройки разных аспектов загрузчика *Grub2*. И снова, сначала надо добавить PPA — `sudo add-apt-repository ppa:danielrichter2007/grub-customizer`, затем освежить репозиторий — `sudo apt-get update`, и установить инструмент с помощью `sudo apt-get install grub-customizer`.



» С помощью *Grub Customizer* можно даже настроить внешний вид экрана *Grub*.

Отключить размытие фона

Размытие фона [Background Blur] в Unity выглядит мило, но обременяет процессор, приводя к тому, что рабочий стол тормозит при выводе Dash. Чтоб не тратить ресурсы на данный эффект, можно отключить эту функцию. Откройте *Unity Tweak Tool* и щелкните по настройке Launcher setting в разделе Unity. Здесь перейдите на вкладку Search и переведите переключатель рядом с Background Blur в положение Off. Если же вам эта функция симпатична, переключитесь на тип Static blur, который требует намного меньше ресурсов, чем Active blur, применяемый по умолчанию.

Отключите Global Menu

Глобальное меню Unity по душе не всем. В Ubuntu 14.04 есть опция отключения глобального меню и интеграции меню в сами окна. Для переключения в локальное меню перейдите в System Settings и щелкните по Appearance в разделе Personal. Здесь перейдите во вкладку Behavior, и в настройке Show the Menus for a Window [Показать меню для окна] щелкните по радиокнопке рядом с In the Window's Title Bar [В заголовке окна].

Сдвиньте кнопки

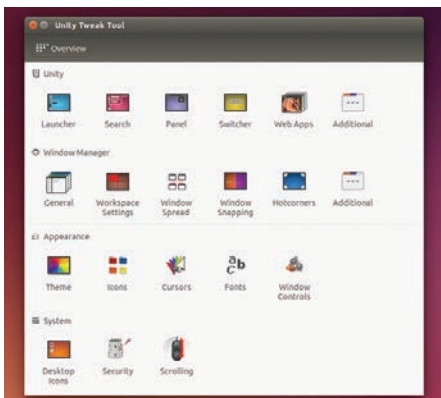
Еще одна особенность Unity, к которой, видимо, придется привыкать — размещение кнопок управления окнами (Свернуть, Развернуть и Закрывать) слева. Как вернуть кнопки на привычное место справа? Запустите *Unity Tweak Tool* и выберите опцию Window Control в разделе Appearance. В разделе Layout измените Alignment с Left на Right.

Уберите наложение полос прокрутки

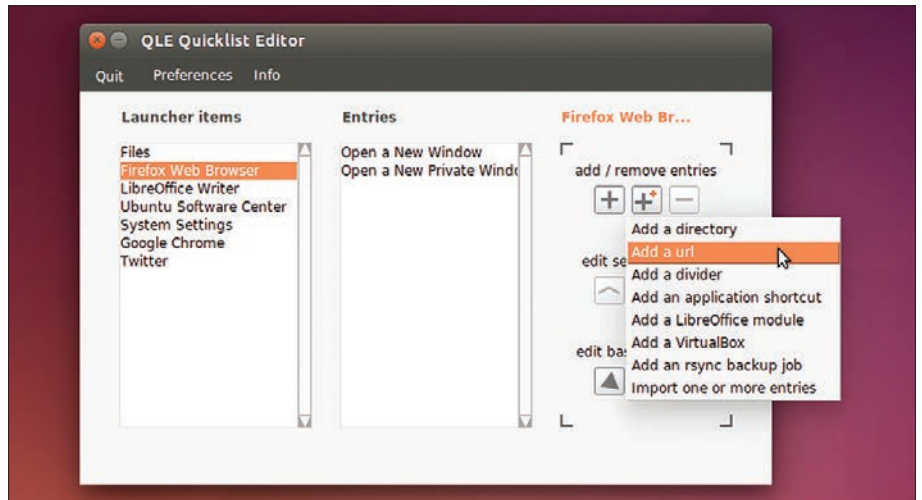
Еще одна настройка по умолчанию, способная вызвать изжогу — функция наложения полос прокрутки, которая скрывает кнопку прокрутки, пока вы не наведете на нее мышью. Для возврата к традиционной полосе прокрутки запустите *Unity Tweak Tool*. Затем щелкните по кнопке Scrolling в разделе System. Здесь можно переключить опцию Legacy, убрав наложение полос прокрутки.

Настройте Quicklists

Quicklists — это наборы наиболее часто используемых задач, которые можно открыть, щелкнув



► *Unity Tweak Tool* — универсальный инструмент настройки рабочего стола Unity.



► *Quicklist Editor* довольно интуитивен, да еще и снабжен подробным Руководством пользователя.

правой кнопкой по значку приложения в программе запуска. Quicklists есть не у всех приложений. Желая добавить Quicklist в приложение, где его нет, воспользуйтесь *QLE Unity Quicklist Editor*. Это богатое функциями приложение поможет создать индивидуально настраиваемые программы запуска приложений Unity с Quicklists и отредактировать готовые Quicklists, изменив порядок, и переименовать действие или вообще его сменить. Приложение также позволяет добавлять новые предустановленные Quicklists для ряда наиболее общих приложений, типа *LibreOffice*, *VirtualBox* и пр.

«Размытие фона в Unity может привести к тому, что рабочий стол тормозит.»

При запуске приложение просканирует текущую программу запуска и отобразит все ее пункты вместе с их Quicklist. Теперь можно добавить запись в имеющийся пункт, введя свою команду или выбрав одну из готовых настроек. Можно, скажем, добавить запись для запуска *Firefox* в безопасном режиме [Safe Mode] командой `firefox -safe-mode`. А запись для запуска *Firefox* на конкретном сайте в новом окне добавляется через предустановленную опцию, вводом в нее требуемого URL.

Убрать подсказки покупок

Все вводимое в Dash возвращает результаты из онлайн-магазинов вроде Amazon. За эту функцию отвечает компонент *Shopping Lens*. Вам не надо, чтобы ваши запросы попадали к третьим сторонам в Интернете? Отключите эту функцию.

Чтобы выборочно исключить из результатов некоторые сервисы, щелкните по значку Dash Home в программе запуска и по Filter Results рядом с окном поиска. Затем снимите выделение с категорий или источников, которые хотите убрать из результатов поиска, скажем, Amazon и eBay.

Для полного отключения этой функции перейдите в System Settings и в разделе Personal выберите Security & Privacy. Теперь переключитесь

во вкладку Search и отключите опцию, отмеченную как When Searching in the Dash: Include Online Search Results.

Скатать окна

Популярной опцией у тех, кто работает с несколькими окнами, является возможность скатать окно вместо того, чтобы его минимизировать. Самое популярное действие для выполнения этой задачи — двойной щелчок по строке заголовка окна. Однако по умолчанию это действие будет максимизировать или убирать максимизацию окна. Чтобы

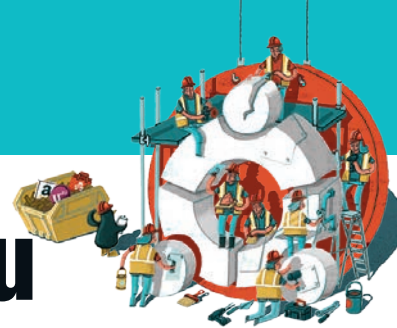
изменить результат этого действия, перейдите в *Unity Tweak Tool*. В разделе Window Manager щелкните по кнопке Additional. Здесь выведете выпадающее

меню рядом с Double click в разделе Titlebar Actions и настройте его на Toggle Shade.

Не ведите истории

По умолчанию Dash показывает все недавно открытые файлы вместе с другими файлами, обнаруженными в вашей системе. Если вы разделяете компьютер с другими людьми, вы вряд ли захотите, чтобы они видели, какие файлы вы недавно открывали. Чтобы Ubuntu не отслеживал некоторые приложения, перейдите в System Settings и в разделе Personal выберите Security & Privacy. Теперь перейдите на вкладку Files & Applications, где вы можете выборочно отключать отслеживание определенных категорий приложений, таких, как Music, Pictures, Videos, Presentation, и т. д.

Желая исключить содержание определенных папок, щелкните по значку + под разделом Exclude, и укажите на папку, которую хотите исключить. Подобным же образом можно исключить определенные приложения. Вы также можете использовать кнопку Clear Usage Data, чтобы удалить имеющиеся записи о недавно открытых файлах и приложениях. Чтобы совсем отключить эту функцию, просто переключите настройку Record File and Application Usage на Off.



Индивидуальный Ubuntu

Настройте функции Ubuntu, чтобы он стал совсем своим.

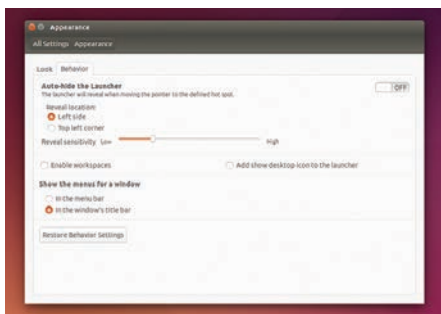
Unity — всего один компонент рабочего стола Ubuntu, хотя и самый броский. Есть и другие аспекты, играющие важную роль в обеспечении полноценной работы Ubuntu. Подобно Unity, большинство других компонентов используют консервативные предварительные настройки, большинству пользователей подходящие; но их можно изменить, чтобы расширить свой опыт работы.

Ставим ПО быстрее

У Ubuntu есть зеркала по всему миру, и для лучшей производительности следует использовать зеркало, расположенное ближе всего к вам. По умолчанию Ubuntu автоматически выберет наилучшее зеркало. Однако на всякий случай запустите инструмент *Software & Updates* и во вкладке Ubuntu Software выведите выпадающее меню Download from и выберите опцию Other. В окне, которое должно открыться, щелкните по кнопке Select Best Server. Ubuntu запустит серию тестов на всех зеркалах скачивания и выберет наилучшее для вас.

Если вы устанавливаете пакеты из командной строки, надо также заменить приложение командной строки *apt-get* скриптом-надстройкой *apt-fast*. Этот скрипт повышает производительность, скачивая пакеты с разных соединений одновременно. Вы можете установить скрипт через PPA:

```
$ sudo add-apt-repository ppa:apt-fast/stable
$ sudo apt-get update
$ sudo apt-get install apt-fast
```



► Ubuntu 14.04 ввел мелкие, но полезные изменения в инструмент System Settings.

После установки скрипта придется также выбрать, хотите ли вы использовать *apt-get* или приложение *aptitude*, и хотите ли вы использовать менеджер загрузок *axel* или *aria2c*. Мы предлагаем остановиться на последнем, поскольку он умеет возобновлять скачивания. Затем все, что вам нужно сделать для работы скрипта — это изменить *apt-get* на *aptfast* при установке пакетов, по типу: `sudo apt-fast install firefox`.

Заменим Software Center

Хотя *Ubuntu Software Center* разработан как исключительно дружелюбный к пользователю, на некоторых компьютерах он работает не очень хорошо, да и не слишком интуитивен для установки компонентов помимо приложений — например, библиотек. *Synaptic* — это графический интерфейс для утилиты командной строки *apt-get*, и один из самых популярных менеджеров пакетов. Он доступен в официальных репозиториях Ubuntu и устанавливается всего лишь командой `sudo apt-get install synaptic`.

Используем Google Drive

Запустив Ubuntu 14.04, Canonical закрыл свой сервис облачного хранения Ubuntu One. Однако альтернативы есть: например, Google Drive, предлагающий аж 15 ГБ бесплатного хранилища. Хотя для этого сервиса нет официального клиента Linux, вы можете использовать инструмент *google-drive-ocamlfuse*. Это файловая система FUSE, написанная на OCaml, и она позволяет монтировать Google Drive в Linux.

Чтобы установить этот инструмент, сначала включите его PPA:

```
$ sudo add-apt-repository ppa:alessandrostrada/ppa
$ sudo apt-get update
$ sudo apt-get install google-drive-ocamlfuse
```

После установки запустите инструмент из командной строки, чтобы авторизовать его в Google: `google-drive-ocamlfuse`.

Когда вы введете эту команду, она откроет новую вкладку в вашем web-браузере по умолчанию и предложит вам разрешить инструменту доступ к вашему Google Drive. Щелкните по кнопке Accept,



► Перейдите в профиль Powersave, чтобы заряд батарей вашего ноутбука держался дольше.

затем войдите в свою учетную запись Gmail, когда вам предложат это сделать, и нажмите на кнопку Allow. Перед тем, как покинуть web-браузер, подождите несколько секунд, пока инструмент получит нужный ему код авторизации. Теперь у вас все готово к монтированию Google Drive. Создайте точку монтирования в своей домашней директории, например, `mkdir ~/gdrive`, и затем монтируйте его с помощью инструмента командой `google-drive-ocamlfuse ~/gdrive`.

Установим новую тему

Вы можете установить новую тему, изменив внешний вид рабочего стола Unity. Некоторые темы для Ubuntu доступны через их собственные PPA, например, проект Numix (`ppa:numix/ppa`) и Moka (`ppa:snwh/moka-gtktheme-daily`). Однако большинство доступны в виде сжатых архивов. Вы можете найти несколько тем в этом формате на <http://gnome-look.org>.

Чтобы установить тему, сначала создайте папку `.themes` в своей домашней директории с помощью `mkdir ~/.themes`. Теперь распакуйте скачанные темы в эту папку. Чтобы активировать темы, запустите *Unity Tweak Tool* и нажмите на кнопку Themes в разделе Appearance. Скачанные вами темы будут перечислены во вкладке Themes. Щелкните по теме, чтобы немедленно активировать ее. Хотя многие темы содержат собственные значки, курсоры и шрифты, вы можете изменять эти отдельные элементы, переходя в другие вкладки в разделе Appearance.

Что делать после установки Ubuntu

Если вы собираетесь использовать Ubuntu в качестве постоянной настольной ОС, вам стоит установить дополнительные пакеты из *Ubuntu Software Center*, или через командную строку: `sudo apt-get install ubuntu-restricted-extras`.

Установятся несколько пакетов, которые Ubuntu не может законным образом предлагать вместе с дистрибутивом. На 32-битных дистрибутивах

заодно установится плагин Flash; пользователям 64-битной версии нужно для воспроизведения контента Flash установить пакет *flashplugininstaller*.

Ubuntu включает в свои репозитории версию Open Source Java под названием OpenJDK. Ее вы можете установить с помощью `sudo apt-get install openjdk-7-jre`, а соответствующий плагин браузера IcedTea — через `sudo apt-get install icedtea-7-plugin`.

Можете также установить проприетарный Oracle Java через скрипт. Сначала вставьте строку `deb http://www.ubuntu.com/ubuntu/pkgs deb all в файл /etc/apt/sources.list`. Затем импортируйте GPG-ключ зеркала с помощью `sudo apt-key adv --keyserver pgp.mit.edu --recv-keys 5CB26B26`. Теперь обновите репозиторий с помощью `sudo apt-get update` и, наконец, установите пакет командой `sudo apt-get install update-sun-jre`.

Управляем запуском

Если вы постоянно пользуетесь какими-то приложениями, вы можете сэкономить свое время, запуская их при загрузке автоматически. Или, возможно, вы захотите, чтобы эта задача выполнялась в фоновом режиме. Например, если вы используете инструмент *google-drive-ocamlfuse*, то вам нужно, чтобы он автоматически монтировал ваш Google Drive, когда вы загружаете Ubuntu.

Чтобы сделать это, выведите Dash и введите и запустите приложение *Startup Applications*. Щелкните по кнопке Add в приложении, которая откроет окно, чтобы определить приложение или задачу, которую вы хотите выполнить. Самое важное поле — Command, где вы определяете команду для исполнения. Щелкните по кнопке Browse и перейдите в директорию, в которой содержатся все ваши приложения (*/usr/bin*), и выберите приложение в списке.

Теперь введите имя и, опционально, комментарий для описания приложения. Выбранное приложение теперь будет перечислено в Additional startup programs. Некоторым приложениям, например, инструменту *google-drive-ocamlfuse*, также нужны добавочные параметры для монтирования диска. Их можно ввести вручную в поле Command, например: */usr/bin/google-drive-ocamlfuse ~/gdrive*.

Запуск приложений быстрее

Демон *Preload* — умный инструмент, который работает в фоновом режиме и отслеживает наиболее часто используемые приложения. Затем он использует эту информацию и загружает часто используемые приложения в память перед тем, как они вам понадобятся, значительно ускоряя время запуска. Все, что вам нужно для использования этого приложения — скачать его из официальных репозиториях по `sudo apt-get install preload`. После этого демон приступит к работе, и вы заметите, насколько быстрее загружаются ваши приложения, когда он выяснит частоту их использования.

Предпросмотр файлов

Sushi позволяет вам просматривать файлы в менеджере файлов *Nautilus*, не открывая их. Чтобы установить это удобное небольшое приложение, запустите терминал и введите `sudo apt-get install gnome-sushi`. Затем запустите менеджер файлов, выберите любой файл и нажмите на пробел, чтобы предварительно просмотреть файл. *Sushi* может предварительно просматривать текст, PDF, аудио- и видеофайлы. Для закрытия предпросмотра нажмите на пробел еще раз.

Комбинации клавиш

Лучшее в рабочем столе Ubuntu то, что его можно полностью контролировать с клавиатуры. Чтобы просмотреть преднастроенные сочетания клавиш для часто выполняемых действий, нажмите и удерживайте клавишу Super (она же Windows).

Желая отредактировать имеющиеся горячие клавиши для предлагаемых действий или задать



➤ **Нажмите и удержите клавишу Super (Windows), чтобы отобразить горячие клавиши по умолчанию для популярных действий. При желании отредактируйте их или добавьте другие в System Settings.**

собственные, выведите System Settings, щелкните по кнопке Keyboard в разделе Hardware и перейдите во вкладку Shortcuts. Имеющиеся горячие клавиши перечислены там по категориям в разных разделах в зависимости от их функций, например, Launcher, Screenshots, Navigation, System и т. д. Щелкните по категории, чтобы вывести все содержащиеся в ней горячие клавиши. Чтобы отредактировать одну из них, щелкните по ней, а затем нажмите на клавишу или комбинацию клавиш, которые вы хотите использовать для этого действия.

изменять тактовую частоту CPU своего компьютера, выбрав или перечисленную величину, или одну из оптимизированных опций, например, Performance, Powersave и т. д.

Настроим загрузчик

По умолчанию Ubuntu настраивает загрузчик на автоматическую загрузку дистрибутива через 10 секунд. Есть ряд причин, по которым вы можете решить изменить это поведение загрузчика. Если, например, у вас на компьютере несколько операционных систем, вы, возможно, решите их реорганизовать и изменить операционную систему по умолчанию, или захотите увеличить или уменьшить время перед автоматической загрузкой операционной системы.

Приложение *Grub Customizer* является отличным инструментом для графического изменения загрузчика. Чтобы настроить загрузку операционной системы по умолчанию, запустите инструмент и перейдите во вкладку General Settings. Выберите кнопку-переключатель Predefined в разделе Default Entry и используйте выпадающее меню, чтобы выбрать ОС по умолчанию. Если вы включили радиокнопку Previously Booted Entry вместо использования предварительной настройки по умолчанию, *Grub* по умолчанию загрузит последнюю загруженную ОС. Вы также можете прямо на этой странице настроить время ожидания загрузки ОС по умолчанию. Внеся все изменения, щелкните по кнопке Save, и приложения запишет все изменения в загрузчик *Grub*.

Чистим меню загрузки

Каждый раз, когда Ubuntu обновляется до нового ядра, он добавляет его в загрузчик, но предыдущее ядро не удаляет. Если новое ядро работает нормально, можно использовать инструмент *Grub Customizer* для очистки меню загрузки и удалить более старые записи. Запустите инструмент и перейдите во вкладку List Configuration, где перечислены все записи, появляющиеся на экране загрузки. Выделите запись, которую хотите удалить, и выберите опцию Remove в меню правой кнопки.

«Можете установить новую тему и изменить вид рабочего стола Unity.»

Продлим жизнь батарей

Одним из недостатков мощных многоядерных ноутбуков является то, что они опустошают заряд батарей намного резвее, чем их более медленные собратья. Есть пара инструментов, которые вы можете использовать, чтобы продуманно управлять батареями своего ноутбука.

Первый — это *TLP*, инструмент командной строки для тонкого управления питанием. Лучшее в этом инструменте то, что он применяет самые распространенные технологии управления питанием без привлечения пользователя. Просто установите инструмент через его PPA и запустите его:

```
$ sudo add-apt-repository ppa:linrunner/tlp
```

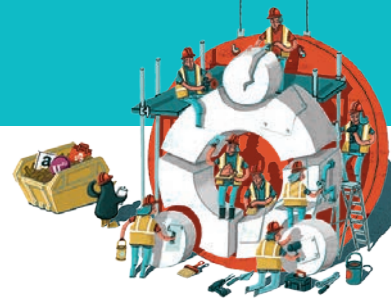
```
$ sudo apt-get update
```

```
$ sudo apt-get install tlp tlp-rdw
```

```
$ sudo tlp start
```

При следующих загрузках инструмент будет грузиться автоматически и творить для вас чудеса.

Другой инструмент — *Cpufreq Indicator*; это апплет, который можно использовать для ручного изменения частоты CPU компьютера прямо на лету. Установите его с помощью `sudo apt-get install indicator-cpufreq` и затем перезапустите компьютер, чтобы его активировать. Когда он начнет работать, вы сможете с его помощью вручную



Игры в Ubuntu

Превратите свою рабочую станцию в классную игровую установку.

Играть в Linux можно точно так же, как и на любой другой платформе, но следует проверить, есть ли у вас нужные драйверы для видеокарты вашего компьютера. В зависимости от возраста вашего оборудования и выбранной вами игры вам, возможно, придется смириться с проприетарными драйверами от производителя видеокарты вместо драйверов с открытым кодом, старательно написанных энтузиастами.

У большинства видеокарт есть драйверы с открытым кодом. Бывает, что для старых видеокарт отсутствуют проприетарные драйверы, например, для серии AMD Radeon 8xxx. Основным недостатком драйверов с открытым кодом тот, что они не всегда предоставляют полную поддержку всех функций видеокарты. Исключение из этого правила — Intel, выпускающая свои драйверы под лицензией открытого кода. Драйвер с открытым кодом от AMD называется Radeon, а его эквивалент от Nvidia именуется Nouveau.

Перед установкой драйвера для видеокарты выясните ее производителя и модель, командой **lspci | grep VGA**. Больше подробностей, например, про скорость и мощность, выяснит **sudo lshw -C video**. Какой бы ни была ваша видеокарта, Ubuntu предложит вам самый подходящий драйвер с открытым кодом, когда вы загрузите дистрибутив. Однако этот драйвер может быть не самым свежим, поскольку Ubuntu требуется время на включение драйверов в свои пакеты и предоставление их через свои репозитории. Популярен репозиторий Oibaf — он предлагает пользователям самые

последние драйверы. Помимо самых свежих драйверов с открытым кодом (**xf86-video-***), из этого репозитория также можно установить Mesa, Gallium3D и прочие компоненты с открытым кодом, относящиеся к X.Org.

Используйте следующие команды для добавления и включения репозитория Oibaf и обновления драйверов с открытым кодом для вашего видеоборудования:

```
$ sudo add-apt-repository ppa:oibaf/graphics-drivers
$ sudo apt-get update
$ sudo apt-get dist-upgrade
```

Для большинства видеокарт драйверы с открытым кодом прекрасно справляются с показом фильмов и управлением окнами. Некоторые из них даже обеспечат вам достойную частоту кадров для несложных игр. Но серьезным геймерам с оборудованием от Nvidia или AMD определенно нужен проприетарный драйвер.

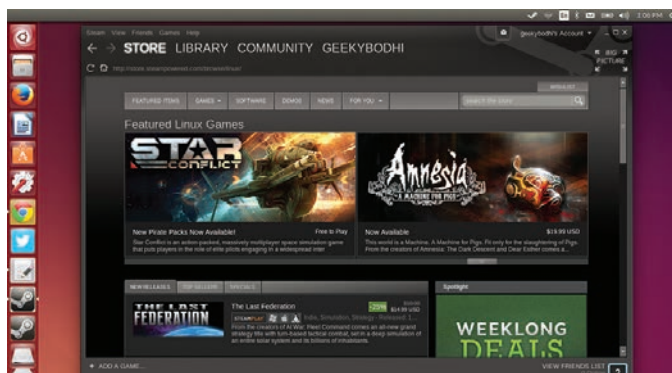
Чтобы найти свежайшие стабильные драйверы Nvidia, введите команды активации PPA X-Swat:

```
$ sudo add-apt-repository
ppa:ubuntu-x-swat/x-updates
$ sudo apt-get update
$ sudo apt-get install nvidia-current
```

Единственная оговорка с этим PPA — его придется удалить перед обновлением вашего Ubuntu, а в обновленной системе активировать снова. Чтобы удалить PPA, установите инструмент **PPA-Purge** с помощью **sudo apt-get install ppa-purge**, и затем удалите PPA X-Swat командой

```
$ sudo ppa-purge ppa:ubuntu-x-swat/x-updates
```

Решив вернуть драйвер с открытым кодом Nouveau, сначала удалите проприетарный — **sudo apt-get purge nvidia**, а потом установите драйвер с открытым кодом — **sudo apt-get install nouveau**.



В некотором смысле, наличие Steam в Linux стало переломным пунктом для игровых возможностей в Linux.

Для карт AMD наилучший выбор проприетарных драйверов предлагает сайт AMD (<http://support.amd.com/en-us/download>). Через выпадающее меню введите подробную информацию по вашей видеокарте, затем скачайте предложенный драйвер. Распакуйте его, чтобы увидеть скрипт **.run**. Перед установкой драйвера убедитесь, что вы установили его зависимости, таким образом:

```
$ sudo apt-get install dh-make dh-modaliases
execstack libc6-i386 lib32gcc1
```

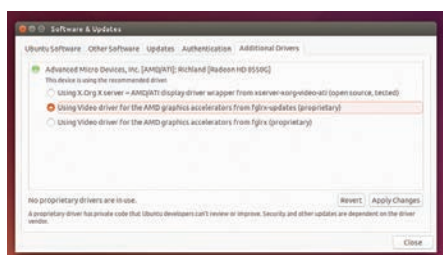
Теперь запустите скрипт:

```
$ sudo sh ./amd-driver-installer-13.35.1005-x86.
x86_64.run
```

Запустится **Catalyst**, графическая программа установки проприетарных драйверов AMD, которая также установит программу управления GPU — **Catalyst Control Center**. Найдя предыдущую версию проприетарного драйвера, установщик предложит вам сперва удалить ее, сделав так:

```
$ sudo apt-get remove --purge fgfrx fgfrx_* fgfrx-
amdccc* fgfrx-dev*
```

По окончании работы установщика перейдите в терминал и введите **/usr/bin/aticonfig--initial** для настройки драйвера. Теперь перезапустите Ubuntu, чтобы его включить. Учтите, что этот драйвер не обновляется автоматически, но в **Catalyst Control Center** вы найдете опцию, которая будет периодически проверять наличие обновлений.



Серьезным геймерам с оборудованием Nvidia и AMD нужны проприетарные драйверы.

Игровые возможности в Ubuntu

Вы только начинаете играть в Linux? Эта платформа предлагает целый ряд возможностей для игр. Начните со знакомства с **Ubuntu Software Center**, где есть несколько популярных «родных» игр Linux, таких, как **Battle of Wesnoth**, **OAD**, **Nexuiz**, **Cube 2: Sauerbraten**, и др.

Еще один источник «родных» игр под Linux — **Humble Bundle**. Помимо схемы «плати сколько хочешь», основная фишка bundles в том, что все

игры являются кросс-платформенными и работают в Windows, Mac OS X и Linux. На сегодняшний день имеется более дюжины «вязанок»-bundles с более чем сотней игр.

Вы также можете играть в игры Windows на Linux, благодаря уровню совместимости **Wine** и интерфейсу **PlayOnLinux**. В графическом приложении имеется длинный список игр (и приложений) Windows, устанавливаемых одним щелчком, при наличии

у вас их файлов **.exe** или **setup**. Вы можете найти **PlayOnLinux** в Ubuntu Software Center.

Затем есть **Desura**, менее известная платформа дистрибуции игр; она предлагает игры для Linux, в основном созданные Indie-разработчиками. И, наконец, есть официальный клиент Steam Linux, устанавливаемый через **Ubuntu Software Center**. Клиент Linux ничем не отличается от клиента Windows, и сейчас поддерживает более 500 наименований.

Прыжок налево

Попробуйте новый рабочий стол или вообще иной дистрибутив.

Несмотря на всяческие отладки и настройки, которые мы вам показали в данной статье, некоторым будет, вероятно, проще и продуктивнее переключиться на другой рабочий стол. Ubuntu предоставляет возможность удалить рабочий стол Unity, заменив его другим.

Таким образом вы сможете и дальше использовать лучшие функции настольного Ubuntu, но с комфортным для вас интерфейсом.

Кроме того, Ubuntu был специально разработан для того, чтобы им пользовались как можно больше людей. Немалое количество разработчиков

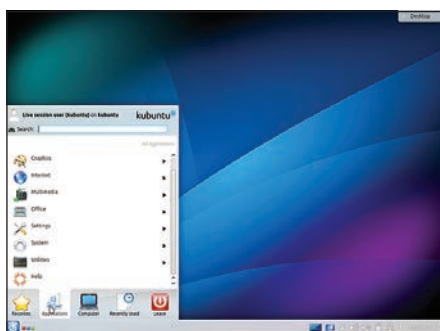
воспользовались его гибкостью для создания индивидуальных ответвлений, подходящих для конкретных случаев. Эти дистрибутивы дают вам преимущества куда более широкой экосистемы Ubuntu, будучи в то же время «заточены» под определенную среду. **LXF**

Альтернативные рабочие столы



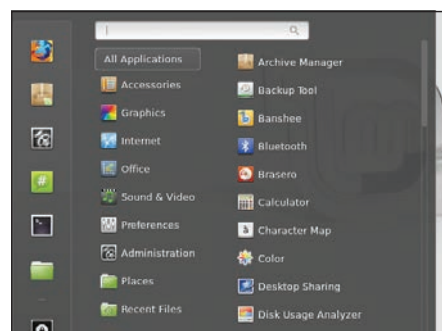
Рабочий стол GNOME

Рабочий стол GNOME 3 более или менее похож на рабочий стол Unity. Он также содержит некоторые похожие компоненты. Например, GNOME Activities Overview похож на Unity Dash, и включает подобную программу запуска панель Favourites, которую можно использовать для размещения наиболее часто используемых приложений. Одна из лучших функций — среда Extensions, способная полностью изменить рабочий стол. Простейший способ установить рабочий стол GNOME на Ubuntu — использовать официальный дистрибутив Ubuntu GNOME.



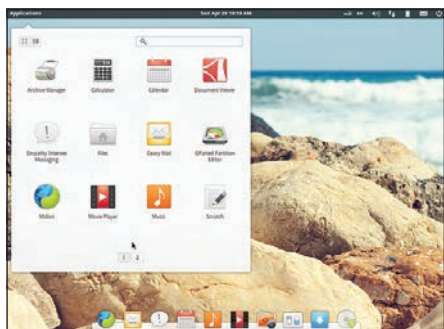
Рабочий стол KDE

KDE — одна из двух сред рабочего стола, которая покажется знакомой пользователям, пришедшим из ОС не-Linux. Структура и поведение KDE, а также программа запуска приложений *Kickoff* напоминают Windows. Однако свою популярность рабочий стол KDE заслужил благодаря неограниченным возможностям индивидуальной настройки. Вы можете установить рабочий стол KDE на уже существующую систему Ubuntu, но простейшей возможностью будет дистрибутив Kubuntu. Дистрибутив Netrunner идет с красиво модифицированным рабочим столом KDE.



Рабочий стол Cinnamon

Вам нужен рабочий стол, соответствующий классическому представлению? Тогда обратите внимание на Cinnamon. У него есть панель внизу экрана, но ее можно расширить до привычной текстовой категоризированной структуры традиционного меню Applications с функциями удобства, взятыми из других сред. Cinnamon имеется в официальных репозиториях Ubuntu. На нем не основано ни одного официального дистрибутива, но по умолчанию Cinnamon включен в популярный дистрибутив Linux Mint, базирующийся на Ubuntu.



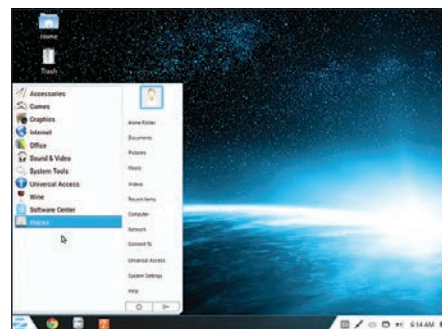
Elementary OS

Один из красивейших дистрибутивов на базе Ubuntu, Elementary OS, предлагает массу инструментов и приложений индивидуальной настройки. У него похожий на Mac OS X интерфейс с Dock внизу. Проект появился как набор тем для основной установки Ubuntu, а потом перерос в полноценный дистрибутив. Он содержит фотоменеджер *Shotwell*, IM *Empathy*, видеоплеер *Totem* и *Gparted*, а также индивидуальные приложения, в том числе текстовый редактор *Scratch*, почтовик *Geary* и медиа-плеер *Music*.



Bodhi Linux

Если ваше оборудование не тянет Ubuntu, взгляните на Bodhi Linux. У него минимум системных требований и легковесный, но привлекательный рабочий стол *Enlightenment*. Предлагается ряд профилей, от *Vare* и *Laptop* до *Fancy*, и каждый оптимизирован под разные типы оборудования. Выбор приложений здесь невелик, но через его инновационную систему управления пакетами можно придать дистрибутиву любой желаемый вид. Или берите легковесный дистрибутив *Lubuntu*, с большей готовностью заработать сразу.



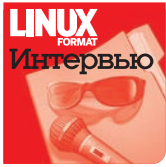
Zorin OS

Этот дистрибутив целился в пользователей Windows. В нем модифицированный рабочий стол GNOME, отлично справляющийся с задачей мимикрировать под рабочий стол Windows 7, с приложением *Look Changer*, способным одним щелчком изменять интерфейс так, чтобы он напоминал другие ОС. У Zorin OS несколько редакций: *Core*, *Gaming* и *Educational* скачиваются свободно, а для редакции *Ultimate* вам придется пожертвовать не менее €9,99 (но вы получите так же техническую поддержку).

Билл Куинн

Мастер Linux

Миссия Билла Куинна из LPI — распространение обучения Linux по всей Великобритании. Нейл Мор выясняет, как это делается...



Linux Professional Institute был создан в 1999 году, в Канаде. И теперь за его плечами уже 15 лет стабильного роста, с учётом впечатляющего всплеска популярности их сертификатов LPIC в последние годы.

Мы побеседовали с главой британского отдела LPI, Биллом Куинном (Bill Quinn), о роли института в мировом Linux-сообществе, о его вкладе в британское образование и о том, как LPI удаётся не терять актуальность в быстро меняющемся мире FOSS.

Linux Format: Расскажите о себе.

Билл Куинн: В сфере ИТ я работаю 15 лет, из них 9 — в компании, являющейся «золотым» партнёром CISCO, затем в другой, сотрудничавшей с Red Hat, где и начался мой путь в Open Source. Там я провёл около 5 лет, потом лет 5 работал по контрактам на просторах Open Source и, наконец, в конце 2012 основал свою компанию.

LXF: Как вы связаны с LPI?

БК: Принцип работы LPI состоит в том, что сама корпорация находится в Канаде и назначает компании-представителей в разных частях света. Мы стали таковой по Великобритании и Ирландии в январе 2013. Предыдущие 2 компании, где я работал, также были партнёрами LPI. Так что в общей сложности я работаю с ними пять или шесть лет.

LXF: В регионе только один представитель?

БК: Совершенно верно. В Великобритании и Ирландии это я. Моя роль весьма проста: нужно набирать партнёров по обучению, таких как Verhoef Training, где мы сейчас находимся, партнёров по науке, таких как Гринвичский университет или Университет Бирмингема, и готовить сертифицированных специалистов Linux. Отчасти посредством приобщения к Linux, отчасти — к Open Source. Ведь многие даже не представляют масштаб возможностей Open Source и ширину его охвата.

LXF: LPI ведь существует не так давно. Его основали в 1999-м?

БК: Да, и с тех пор он стал номером один в подготовке сертифицированных Linux-специалистов, выпустив около 370 000 человек. [LXF: Последние 5–6 лет это число росло взрывными темпами.] Он всегда держал планку, но даже сейчас 2013 год остаётся рекордным для LPI. Никогда прежде не было такого количества экзаменов, такого количества новых партнёров по обучению, и этот рост продолжается и в нынешнем году.

LXF: Это было ожидаемо?

БК: На мой взгляд, это была смесь ожиданий и приятного сюрприза. Естественно, я наблюдал за ситуацией после банковского кризиса, ставшего поворотным пунктом для данной отрасли, поскольку тогда спрос на ИТ-специалистов впервые исчез или, по крайней мере, упал



из-за сокращений. Именно в этот период люди начали понимать: «Работодатель должен считать меня ценным работником, мне нужно продемонстрировать, на что я способен. Значит, пора вкладывать деньги в обучение, нужно сдать какой-нибудь экзамен, что подтвердит мою квалификацию». Тогда и стала востребована сертификация.

Ещё один фактор — рост числа услуг. Речь идёт и о больших хранилищах данных, но они, как правило, предоставляются вместе с исходными продуктами. И вот возьмите, например, Ирландию: в планах по кадровой политике на 2013 год утверждается, что страна должна остаться технической столицей Европы, [а для этого] нужны специалисты Linux.

LXF: Удаётся ли Ирландии сохранить рабочие места для технарей?

БК: Ирландия — в некотором роде моя боль. Уровень безработицы там около 18%, в некоторых регионах — до 27%. Местное правительство очень много сделало для привлечения в Ирландию [технических] компаний. Таких как Google, Microsoft, Red Hat, LinkedIn, Twitter — список можно продолжить. К сожалению, наличием квалифицированной рабочей силы они не озаботились, и довольно многим компаниям пришлось против воли властей везти в Ирландию своих специалистов. Помоему, абсолютно нелепо, когда создали рабочие места — и их оказалось некому занять.

LXF: Что же стало толчком к созданию LPI в 1999-м?

БК: Формальное признание компаниями того, что им необходима помощь и поддержка со стороны,

и эту функцию должен выполнить эталонный стандарт. Потому целью LPI стала формальная организация обучения, результаты которого соответствовали бы запросам отрасли, и его обеспечение.

LXF: Изменилось ли что-нибудь за эти годы?

БК: В каком-то смысле — ничего! LPI до сих пор верен своим убеждениям, они по-прежнему поддерживают контакт с сообществом, интересуясь: «Правильно ли мы действуем?» и обсуждая это. Я думаю, самым большим достижением является LPI Essentials. Он появился около двух лет

О РОЛИ ОБРАЗОВАНИЯ

«Нелепо, что в Ирландии создали рабочие места — и их оказалось некому занять.»

назад (в 2012 г.), и опять же по запросу публики. У нас уже были программы сертификации LPI 1, 2 и 3, но по отзывам, LPI 1 была слишком сложной для тех, кому нужен был быстрый старт. Так и появилась спецификация Linux Essentials. Одна из главных наших целей — сделать её доступной с 16 лет, чтобы она была ориентирована на школы, чтобы люди раньше приходили в профессию — и при этом весь процесс подготовки был абсолютно прозрачным. Мы связались с научной средой, со специалистами, сказали: «Есть вот такая идея, это реально?» Затем они помогли нам её оформить — большинство вопросов, включённых в Linux Essentials, составлены учёными и людьми из отрасли. LPI проходит строгий контроль. Вопросы проверяются: «Верна ли грамматика, правильно ли задан вопрос, нет ли двусмысленности, »

можно ли его улучшить, не слишком ли сложен, не слишком ли прост?»

LXF: Как именно вы работаете с сообществом, чтобы гарантировать обеспечение их потребностей?

БК: LPI очень открыт и прозрачен. У него есть свой wiki на wiki.lpi.org, где публикуются все критерии и сроки сдачи. Многие подписаны на наши рассылки, и если есть какие-то изменения в учебной программе... Мы вежливо об этом уведомляем, однако народ в Linux темпераментный — ругаться они не будут, но выскажут претензии по существу. Поэтому мы работаем с рассылкой, я много занимаюсь маркетингом, и когда в LPIC-1 вносятся изменения, я уведомляю об этом людей, чтобы они могли поучаствовать в этом процессе.

LXF: Обращаясь к структуре, LPI 3 появилась последней. Высок ли на неё спрос?

БК: Программа всегда задумывалась как последовательная, чтобы люди могли идти по нарастающей. Так, LPIC-1 и LPIC-2 обеспечивают базовую подготовку, а в LPIC-3 есть очень много специализированного: например, высокая доступность или виртуализация. Отвечая на ваш вопрос — да, она пользуется спросом. Её рекламируют, некоторые компании включают её в список своих требований. Некоторые используют как сравнительный тест. Например, в Netgear сейчас тренируют своих инженеров на LPIC-1. В общем, да, в отрасли её признают.

LXF: Ходят слухи, будто LPI вот-вот договорится с правительством, чтобы добавить её в учебный план?

БК: Если коротко, то да. За последние несколько месяцев я провёл несколько встреч в Парламенте

и в Палате лордов. Этот процесс требует длительных переговоров и создания единого образовательного пространства, каким в Англии является QCF [Qualifications and Credit Framework — прим. пер.]. Прежде чем LPI включают в учебный процесс, его должен признать QCF, над чем мы сейчас и работаем. Между тем, ряд университетов и колледжей организует свои курсы LPI, используя наши цели и собственные методы обучения. Мы также общались со школами, использующими на уроках информатики Raspberry Pi, и выяснили, что у них активно действует кружок на базе Linux Essentials.

LXF: Добавить Raspberry Pi в учебный план, на ключевые этапы, подготовить учителей — это большой труд.

БК: Это сложный процесс. Его можно разделить на три стадии: нужен учебный план, соответствующий QCF, органы, уполномоченные присваивать квалификации уровня GCSE или A Level [британские экзамены, дающие право на получение аттестата и поступление в вуз, — прим. пер.], и квалифицированные преподаватели. Работа колоссальная.

LXF: То есть, возможно, появится GCSE по Linux?

БК: На самом деле, я бы очень удивился, если бы в обозримом будущем появилось нечто под названием GCSE Linux. Скорее уж по Программированию или Поддержке серверов. Эти электронные навыки очень востребованы. Ранее любая программа сертификации утверждалась QCF. Что существенно затруднило работу с электронными навыками, поскольку были отдельные стандарты для Windows 2003, 2008, 2010, 2012, для LPI и так далее. То есть для них требуется более гибкая система QCF.

LXF: В Microsoft, я полагаю, были бы не в восторге от появления курса обучения, всецело посвященного Linux? Но раз правительство идёт навстречу открытым стандартам...

БК: Не только правительство. На мой взгляд, деловая сфера тоже. И причина тому — новый кризис. Бюджеты урезали повсеместно. Кое-где ни о каком развитии и речи не было: средства, выделенные на ИТ, шли на оплату электричества. Это стало очень важным катализатором. Неприятно так говорить, но они вынуждены были обратиться к Open Source, кроме самых консервативных, кто говорил: «Нет уж, Open Source — это вне нашей зоны комфорта».

LXF: Помнится, в начале 2000-х Microsoft активно нападали на Linux, утверждая, что его использование увеличивает расходы. Выходит, боялись они не зря, но похоже, всё идёт к тому, что Linux уже занимает ведущее место?

БК: Скорее, одно из ведущих. Мне кажется, что Microsoft уже просто не справляется с такой широкой аудиторией и таким обилием сфер. Обратите внимание хотя бы на прорыв Apple. Посмотрите на то, что делает Google — они же очень серьёзно наступают на территорию Microsoft. Так что теснит его не только Linux.

LXF: Последние 5 лет мы наблюдаем тотальную миграцию с настольных систем на облачные, где правит исключительно Linux.

БК: Мне как-то понравилась ваша шутка о том, что каждый год, обычно в районе января, кто-нибудь объявляет его «годом настольного Linux», и это никогда не сбывается. Но последние пару лет, в особенности на смартфонах, люди используют облачные приложения, и конечно пользователю совершенно всё равно, какая у него операционная система, пока ему хватает её возможностей. Он и понятия не имеет, что пользуется Linux.

LXF: Не совсем в тему, но, возможно, как раз в этом плане правильно действует Canonical. Они стремятся сделать единое ядро Linux для всех платформ. Чтобы облачные приложения можно было бы точно так же использовать и вне Сети, независимо от устройства.

БК: Вероятно, ту же стратегию выбрали и Microsoft, запуская Windows 8, именно поэтому у неё такой интерфейс. На ноутбуке ли, на настольном ПК или на телефоне — рабочее окружение одно и то же; цель была как раз в унификации. Но, по моему, они недоучили: интерфейс на телефоне или планшете — совсем другое дело, чем на обычном ПК или на ноутбуке. Используются они совершенно по-разному. Похоже, что в Canonical это как раз поняли.

LXF: Вы упомянули о привлечении в ИТ женской аудитории. Как вы это реализуете относительно LPIC?

БК: Мы нашли одну женщину-парламентария из сферы ИТ, и на данный момент активно стараемся сформировать группу, в которую помимо LPI вошли бы другие специалисты по сертификации, такие как Microsoft и CISCO. Ведь проблемы у них общие, равно как и регистрационные органы, поскольку они все относятся к образованию. При наилучшем раскладе, эта система должна работать со школы и выше. Сейчас, при поддержке этого парламентария, мы формируем рабочую группу и надеемся, что вскоре приступим к делу. Поживём — увидим.

LXF: Я читал, что соотношение девочек и мальчиков в области науки, инженерии и ИТ стремительно сокращается от выпускных до вступительных экзаменов и получения учёных степеней — примерно до 10%.

БК: Мне кажется, есть две вещи, которые следуют изменить. Во-первых, стереотипы, связанные со сферой ИТ. Бытует представление, что там работают мужчины, и все они чокнутые, невымытые, живут в грязи среди завалов коробок от пиццы, и прочее. На самом же деле эта сфера ничем не отличается от других — здесь тоже всё зависит от профессиональных качеств. И появляются свои примеры для подражания. Речь не только о президенте Yahoo [Марисса Майер [Marissa Meyer], — прим. пер.] — это, пожалуй, слишком высокий уровень. Но, как видите, человек моего возраста, работая в этой отрасли, вполне может получать достойную зарплату, и это реально для всех.



Кроме того, надо сделать учебный план интереснее. Я считаю, что Майкл Гоув [Michael Gove, министр образования Великобритании — прим. пер.] сейчас начал очень большое дело. Ведь не только женщинам, но и мужчинам надоело учить только работе с офисными приложениями, а в ICT только это и предполагалось. Идея же Майкла Гоува в том, что информатика должна готовить к поступлению в университет и давать практические навыки. В хороших школах она выведена на межпредметный уровень. В одной из таких, на уроке информатики, в рамках проекта на базе Raspberry Pi ученики запустили в воздух метеозонд, способный снимать показания и фотографировать. А значит, проект был также и по географии, а полученные фото могли использоваться на рисовании. Проект один — предметов несколько.

LXF: Такого рода практические проекты помогают увлечь учеников.

БК: И в этом тоже была проблема обучения чисто приложениям. Дети далеко не всегда понимают, зачем нужны электронные таблицы [LXF: А кто понимает?!] И скольким из класса в 40 человек действительно придётся регулярно их использовать? А вот если вы их научите захватывать данные, чтобы потом работать с ними хоть в электронных таблицах, хоть в базах данных, осваивать приложение им будет гораздо интереснее.

LXF: Я хотел бы вновь вернуться к *Minecraft*.

Как только девочки или мальчики узнают, что посредством программирования им можно управлять, их это сразу цепляет.

БК: Вы слышали о *CoderDoJo*? Это проект из Ирландии. Автор идеи — Билл Ляо [Bill Liao]. Проект открытый, так что не нужно покупать дорогое ПО и прочее. Изначально планировалось, что это будет внеклассный кружок программирования

О ТОМ, КАК ВОВЛЕЧЬ УЧЕНИКОВ

«Следует изменить стереотипы, и надо сделать учебный план интереснее.»

для детей, только вместо преподавания — взаимообучение. То есть учитель в классе присутствует, но только чтобы подать пример, по принципу: «Эй, поглядите, какую крутую штуковину я сделал!», чтобы пробудить в других энтузиазм. Проект пользуется невероятным успехом.

LXF: Одна из главных заслуг Raspberry Pi Foundation в том, что они продвигают в учебный план не столько Pi, сколько Linux, чтобы учителя могли действительно давать детям эти знания.

БК: Raspberry Pi — это невероятное изобретение. При такой цене оно весьма по карману школам. Не устаю об этом говорить. Устройство за £30 — это же практически даром, по цене учебника. И решает множество проблем, которые школы испытывают с ИТ. Но самое главное — что учителя



сами начинают изучать Linux и применять его. Это очень неплохо.

LXF: Стало быть, Linux Essentials создавался с прицелом на школы?

БК: Изначально LPI Linux Essentials был ориентирован на аудиторию 16+, поскольку мы очень рассчитываем на школы и старшеклассников, но мы также прекрасно знаем, что его изучают и дети гораздо младше. В первую очередь он задумывался как вводный курс свободного ПО — для тех, кому никогда не приходилось иметь с ним дело, там есть своя специфика, и очень важно иметь представление о лицензионной политике и разнице между открытыми и закрытыми лицензиями. А затем это переходит к освоению кодовой строки, что становится опорной точкой.

LXF: Значит, читателю LXF со знанием Linux, желающему работать в этой сфере, нужен LPI?

БК: Не помешает! Во-первых, я бы посоветовал оценить уровень своих знаний. Один из самых ненавистных мне вопросов — «Я работаю с Linux два года. Как вы думаете, я сдам LPIC-1?» Это же зависит от того, чем и как вы занимались! Поэтому я всегда рекомендую людям заглянуть на wiki.lpi.org, изучить критерии и честно оценить свои знания, задавшись вопросом: «Моё ли это?» Люди действительно попадают впросак, особенно те, кто трудится в определенной области и выполняет одну и ту же рутинную работу изо дня в день,

не зная ничего помимо этого или заблуждаясь, что знают. Они приходят к нам и думают, что сдадут, но увы. Так что проверьте себя по wiki, или пойдите учиться, либо займитесь этим самостоятельно — решать вам. Существуют также университеты и колледжи, где есть такие программы. Но в конце концов сдайте экзамен и получите бумагу, подтверждающую вашу квалификацию.

LXF: А как университеты и колледжи взаимодействуют с LPI?

БК: Есть две совершенно разных модели: высшее образование и дополнительное образование. Колледжи в основном предлагают краткосрочные курсы, скажем, 10-часовые, три раза в неделю по Linux Essentials, или же диплом по поддержке серверов, где Linux отведена только часть. В университетах подход будет иной: там будет специальность Информатика или Защита информации, и обучение должно будет соответствовать запросам отрасли. По окончании студент получает степень, или может пойти дальше и получить LPIC.

LXF: Хороший подход.

БК: Об этом активно говорят в Университете Бирмингема: получение степени — это лишь начало. Это очень конкурентный рынок, и нужно делать всё, чтобы вас хотели принять на работу, так что профессиональная сертификация параллельно с получением образования — это очень эффективно. Интересно и то, что мало кто за пределами Великобритании знает, насколько престижна степень того или иного университета. Наличие сертификата, общепризнанного в данной отрасли, служит хорошим её подтверждением — показателем уровня знаний, который понятен всем. **LXF**



Нетребовательные к ресурсам

Ощущаете некую раздутость? Джонни Бидвелл накрывает шведский стол легковесных приложений, которые не сожрут всю вашу память.

Мы рассказали о ряде дистрибутивов-легковесов (стр. 22) и экономных веб-браузерах (стр. 28), но многие мелкие приложения тоже достойны внимания. Помимо огромного числа прочих прелестей Linux, есть ряд небольших, но отлично продуманных программ, которые делают все, что нужно, не таща за собой целый воз зависимостей и не сжирая ваше ОЗУ.

Если вы серьезно обеспокоены потреблением ресурсов, подумайте о применении консольных приложений. Сначала это пугает, но стоит вам освоить необходимый минимум волшебных клавиш и отказаться от подхода «щелчков и перетаскиваний», как вы с изумлением обнаружите, насколько эффективнее станет ваша работа.

Более того, работая именно так, вы и выглядеть будете куда круче. Но если вы по-прежнему не хотите идти таким путем, есть смысл рассмотреть небольшие графические приложения.

Менеджеры файлов

Файлы играют важную роль, а значит, важен функциональный менеджер файлов. В этом свете наше первое предложение покажется вам довольно противоречивым: не думайте о нем. Или, точнее, станьте самому себе менеджером файлов. Освоив всего несколько приемов shell-fu [аллюзия на кун-фу, — прим. пер.], вы вполне справитесь с рядом балетных па и систематических переименований, а автозаполнение поможет обойти

проблему набора длинных имен файлов. Доктор Браун и Нейл Ботвик ранее приводили замечательные примеры *mv*, *find*, *for*, *grep*, *xargs* и т.д., так что стоит взглянуть в предыдущие номера LXF, и ваш скромный автор, жалкий дилетант по сравнению с этими титанами, отправляет вас к их работам в данной области.

Если вы жаждете обзавестись целенаправленным файловым менеджером, но вас устроит менеджер командной строки, первым делом стоит обратить внимание на *Midnight Commander* — ох, простите, *GNU Midnight Commander*. Это Ортодоксальный менеджер файлов [Orthodox File Manager, OFM], предлагающий двухпанельный вид с терминалом команд в стиле классического *Norton*

Философия Suckless и KISS

Сильно огорченные кажущимся неотвратимым раздуванием основных инструментов и библиотек, сообщество хакеров основало <http://suckless.org>. Этот коллектив стремится создавать программы на принципах простоты, ясности и умеренности — «не такие упырские», по их собственным словам [software that sucks less — отсюда и название, — прим. пер.]. Проект нацелен на более передовых конечных пользователей, поскольку именно эта группа сильнее всего страдает от вышеназванных

симптомов. На сегодняшний день их работа включает — но этим не ограничивается — следующее: менеджер окон *dwm*, система меню *dmenu*, браузер *Surf* и *st*, простой терминал. На их сайте вы найдете список «крутых» программ — некоторые из них попали в нашу статью. Кроме того, вы найдете список программ и допущений, сделанных этими программами, которые считаются «упырскими».

Более общая идеология KISS [Keep It Simple, Stupid — Будь проще, дурень], весьма часто цити-

руемая в wiki Arch Linux, но подходящая ко множеству ситуаций в реальной жизни — стремиться к простоте и элегантности через минимализм, избегая ненужной сложности. В терминах программ это обращение к этосу UNIX: небольшие инструменты, делающие что-то одно, но хорошо. Можно увидеть, что они охватывают те приложения (в частности, графические), которые избегают чересчур раздутых интерфейсов, ненужных функций и myriad зависимостей.

Commander. Он довольно интуитивен: Tab переключает с одной панели на другую, а действия, совершаемые с помощью функциональных клавиш, показаны внизу; очень удобно. Вы даже можете использовать мышь и открывать файлы в их родных приложениях (через *xdg-open*), если запустите его в сессии X. В *Midnight Commander* также имеется родной просмотрщик и редактор (для hex и ascii) — он умеет открывать архивы, просматривать сайты FTP и окна с распределенным доступом, и в нем очень симпатичные темы. Если вам нужно нечто более продвинутое, поинтересуйтесь *vifm*, который использует «vi-подобные привязки клавиш». Тут добавить нечего.

Заглянув в HotPicks LXF185 на стр. 100, вы найдете более подробное описание могучего *Midnight Commander*.

Переходя к представителям графических менеджеров, мы начнем со *SpaceFM*. Он предлагает интерфейс со вкладками и множеством панелей, плюс собственную виртуальную файловую систему, и перестраивается так, что его и не узнаешь. Его можно скомпилировать для GTK+ 2 или 3; он отлично ладит с программой монтирования устройств *Udevil* с малым количеством зависимостей, и вам не придется задавать себе вопрос: «Это что, 2004 год?», когда вам предложат ввести пароль root, чтобы смонтировать USB-брелок. *SpaceFM* умеет даже управлять значками на рабочем столе, что вам особенно понравится, если вы используете отдельный менеджер окон, или оставит равнодушным, если вы считаете значки на рабочем столе реликтом 1995 года.

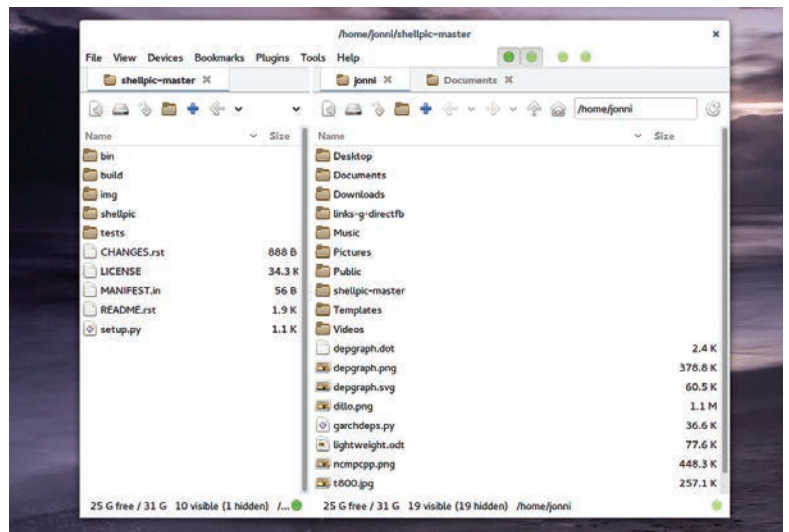
SpaceFM отделился от почтенного *PcManFM* еще во времена HAL. Последний является менеджером файлов по умолчанию в Raspbian и LXDE, и его основное отличие — использование Gnome VFS (и, следовательно, среды *Udisks2*) для монтирования дисков, совместно используемых ресурсов и чего душе угодно. Если вы не против установки ряда библиотек Gnome 2 и вам нужен OFM, то *Gnome Commander*, вероятно, будет весьма достойным кандидатом. Если вы — фанат KDE, попробуйте *Krusader*.

Эмуляторы терминала/мультиплексоры

Если вы используете некую среду рабочего стола, то есть вероятность, что вы вполне довольны тамашней программой терминала. Но есть и альтернативы, и весьма неплохие.

Особо отметим *rxvt-unicode*, который создан на почтенном *Rxvt* и обладает такими функциями, как поддержка Unicode, прозрачность (псевдо и настоящая) и поддержка шрифтов *Xft*. Он поддерживает расширения Perl, способные обеспечить кликабельные URL'ы и интерфейс со вкладками, хотя все это основано на GTK+ 2. Довольно нестандартно то, что он создан на модели клиент-сервер, и как только загрузится демон, окна терминала открываются быстро и не злоупотребляют памятью.

Желая добиться истинного минимализма, используйте *st* (*Simple Terminal*) с <http://suckless.org>, однако некоторым сюрпризом для вас может оказаться отсутствие буфера прокрутки



► *SpaceFM*, легковесный менеджер файлов, работающий в Gnome 3.

и необходимость редактировать файл, чтобы заставить клавишу Del работать в *Bash*.

Далее, у нас есть *Lilyterm*, *Sakura* и *Evilvte* на базе GTK. *Lilyterm* лучше всего работает с GTK+ 2, *Sakura* — только с 3, а *Evilvte* может работать и с 2, и с 3. У всех трех очень мало зависимостей сверх *VTE*, достойной библиотеки Gnome, и они предлагают вкладки, прозрачность и поддержку Unicode.

Очень популярны также терминалы, которые выпадают с верхней части экрана в стиле *Quake*. Среди них *Yakuake* для KDE и *Guake*, которому мало что требуется помимо *VTE* и привязок Python для GTK+. Есть еще *Tilda* (клавиша тильды в *Quake* вызывала терминал), которой не нужны эти привязки. Наличие горячей клавиши для терминала весьма удобно, но бывает нужно и несколько окон, а тогда это перестает быть полноценным решением. Мы также должны упомянуть *Terminator* («роботизированное будущее терминалов»), поскольку он позволяет размещать сессии мозаично в стиле произвольной конфигурации.

Помните, что у вас также есть множество бесплатных терминалов консоли (доступных через клавиши Ctrl+Alt+Fn). Если вы прикипели к консоли, можете использовать их как своего рода примитивные вкладки. Или измените свою игру, перейдя на мультиплексор терминала, типа *tmux* или *screen*. В основном эти существа используются для того, чтобы удаленные программы работали после выхода,



«Если вы стремитесь к минимализму, стоит использовать st.»

»

но они могут делать намного больше — множественные сессии, буферы прокрутки, копирование и вставка, неудобные клавиши быстрого запуска... список можно продолжить.

Web-браузеры

В зависимости от ваших действий, работа в консоли может стать разумным решением. Например, вам надо быстренько заглянуть в Wikipedia или прошерстить документацию Gentoo, чтобы выяснить, какие флаги USE нужны для работы вашего менеджера входа. В подобных случаях вам достаточно Lynx или Links (в текстовом режиме). Однако не все страницы подвергаются рендерингу хорошо: Lynx не поддерживает CSS или JavaScript; а Links хоть и поддерживает JavaScript, но в текстовом режиме от этого проку мало. Однако у Links имеется графический режим, с использованием буфера кадров [framebuffer]. В этом режиме фреймы, таблицы и изображения поддерживаются, но не обольщайтесь — с Flash он не справится.

Самый легкий браузер для X — Dillo, основанный на ftk; он умудряется втиснуть в свой крошечный объем базовую поддержку CSS и отображение вкладок. Отсутствие в нем поддержки Javascript и ведения журнала можно считать функциями безопасности. Cookies поддерживаются, но по умолчанию отключены. То же касается базовых транзакций SSL, хотя пользователей предупреждают, что данная поддержка скорее на стадии

«Работа в консоли может стать разумным решением.»

альфа. Luakit (GTK+) и dwb используют движок Webkit для обеспечения соответствующего стандарта рендеринга при исключительно аскетичном интерфейсе. Оба активно используют клавиатурные комбинации и хорошо расширяемы. Среди более привычных браузеров на Webkit — Midori (GTK+) и QupZilla (Qt).

Почтовые клиенты

Mutt — командный клиент электронной почты, появившийся в середине 1990-х. Он поддерживает IMAP, SMTP, SSL, GPG и сторонние плагины. Конечно же, вам придется выучить некоторые клавиши быстрого запуска, и если вы хотите проверять свою почту офлайн, настройте offlineimap (IMAP) или getmail (POP). Mutt использует внешний текстовый редактор по вашему выбору, и вы можете создать нечто вроде адресной книги, используя алиасы, или применить специальное приложение Abook. В 1991 году вышел клиент электронной почты Pine, но исходный

код был доступен только для UNIX. Pine был прост в освоении и шел в комплекте с простым, но мощным текстовым редактором Pico. Его клон — редактор GNU Nano, который все мы знаем и любим. Выходящий под лицензией Apache Alpine (Alternatively Licensed Pine) появился в 2007 году, и хотя он больше не разрабатывается активно, проект под названием Re-alpine пытался его продолжить. Последний релиз Re-alpine вышел в середине 2013 г.

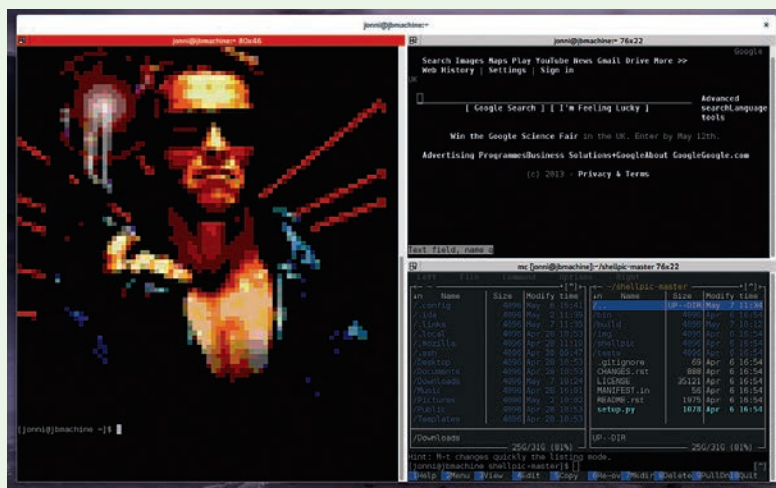
На графическом фронте у нас есть Sylpheed и его чуть более функциональное ответвление Claws Mail; оба используют GTK+ 2 и поддерживают всякие плагины. Sylpheed делает практически все, что вам нужно, и ничего сверху. А вот Claws Mail предлагает поддержку фильтрации спама и GPG и проводит рендеринг почты HTML через Webkit. Если вам нравится Qt и IMAP, стоит взглянуть на Trojita, который поддерживает почту HTML (через Webkit) и создан для максимально бережного обращения с ресурсами.

Музыкальные плееры

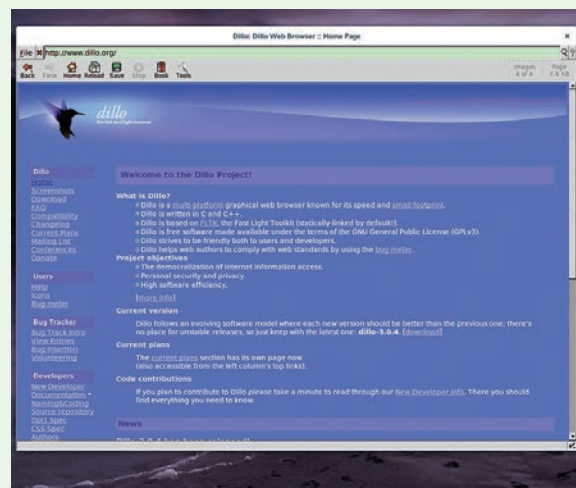
MPD (Music Player Daemon) — превосходная программа, которая ведет базу данных вашей музыкальной коллекции и воспроизводит треки в ответ на команды из (возможно, удаленной) клиентской программы, которых немало. MPD невероятно легковесный, и это означает, что его можно использовать с клиентом консоли на базе Ncurses — например, Ncurses (или еще более непонятно именуемый ncstpp) — так, что ваш CPU его и не заметит. MPD не скуден функциями: он воспроизводит все, что понимает библиотека Ffmpeg (то есть вообще все), поддерживает плей-листы Spotify и SoundCloud и имеет все виды плагинов. Если вам нужен простой музыкальный плеер Ncurses, то первыми кандидатами должны быть Cmus или Herrie.

Есть и графические клиенты для MPD, например, Xfmcp для XFCE и Gimmix, однако для большинства других, GTK+, разработка, похоже, застыла. Что касается Qt — все, что нам удалось найти, это Cantata. По части отдельных приложений есть отлично названный Deadbeef [англ. «мертвая говядина»], который использует GTK+ 2 или 3, и, собственно, все, но зато поддерживает все возможные форматы, редактирование тэгов и плавное воспроизведение без заминок, и имеет множество плагинов.

Немного тяжелее, но все равно далекий от ожирения, мощный Audacious, который можно сделать похожим на Winamp/ XMMS. Аналогично, для Qt обратите внимание на Qmmp. Кстати об XMMS: XMMS2 — проект клиент-сервер, цель которого — продолжить дело своего популярнейшего предшественника. Клиенты на любой вкус найдутся здесь: <https://xmms2.org/wiki/Clientlist>.



➤ Terminator, в котором работают shellpic, links и mc. Работая таким же образом, вы могли бы добиться намного большего в губернаторах, Арнольд.



➤ Браузер Dillo отлично справляется с рендерингом <http://dillo.org>, УММВ работает для других сайтов.

GTK+ и Qt: антагонисты рабочего стола Linux

GTK+ и Qt — кросс-платформенные инструментарию виджетов. Они обеспечивают однородный внешний вид и восприятие приложений в отношении меню, диалоговых окон и т. п. GTK+ используется в Gnome, а LXDE, Xfce и Qt применяются в KDE и Razor Qt. Если вам нужен действительно минимальный рабочий стол, то в идеале вы будете устанавливать только приложения с общим инструментариом.

С одной стороны, GTK+ легче, чем Qt, но ситуация осложняется тем фактом, что некоторые

приложения предпочитают версию 2, а некоторые — 3. Далее, есть множество тем GTK+, и мало какие из них хорошо работают с обеими версиями, отсюда и избыток сообщений на форумах типа «мои приложения GTK{2,3} выглядят ужасно». Некоторые приложения работают с обеими версиями, и тогда дистрибутивы стараются включить в пакеты их воплощение GTK+ 3. Нужна версия GTK+ 2? Ее компиляция, как правило, не доставляет трудностей, если вы малость поредактируете

настройки скриптов. Для пользователей Arch Linux в Arch User Repository есть ряд уже готовых пакетов *-gtk2, что позволит вам сэкономить 100 МБ на GTK3 со товарищи.

Если вы не против разношерстной среды с GTK+ и Qt, вы добьетесь разумной степени однородности, применив какой-нибудь QGtkStyle, чтобы Qt стал походить на вашу тему GTK+, или движок GTK-Qt, чтобы сделать то же самое, но в противоположном направлении.

Просмотрщики изображений и редакторы

Изображения можно просматривать в консоли: для этого нужен всего лишь Shellpic от Ларса Солберга [Lars Solberg]. Он написан на Python и зависит только от библиотеки Pillow, и использует управляющие коды ANSI для отображения картинок в таком количестве цветов, которое поддерживает ваш терминал. Если вы работаете вне X-сервера, но у вас включен KMS (или вы иным способом желаете поиграть с такими вещами, как ivesaft, чтобы получить высокое разрешение буфера кадров), можете использовать fbi (framebuffer image viewer) для отображения картинок в чуть более правдоподобной манере.

Вы также можете обрабатывать изображения командами convert и mogrify из ImageMagick. Если у вас есть графическая среда, то программа отображения последнего очень легковесная, хотя и управляется исключительно из командной строки. То же можно сказать о популярной программе feh. Обе могут работать с несколькими файлами и использоваться для настройки фона рабочего стола при наличии отдельного менеджера окон. Если вам нужен более традиционный просмотрщик изображений, обратите внимание на Geeqie — простую программу GTK+ 2, способную работать с данными EXIF и изображениями .raw. Если вам также нужно редактировать изображения, хорошие варианты — Mirage (построенный на PyGTK) и Converseen (Qt4).

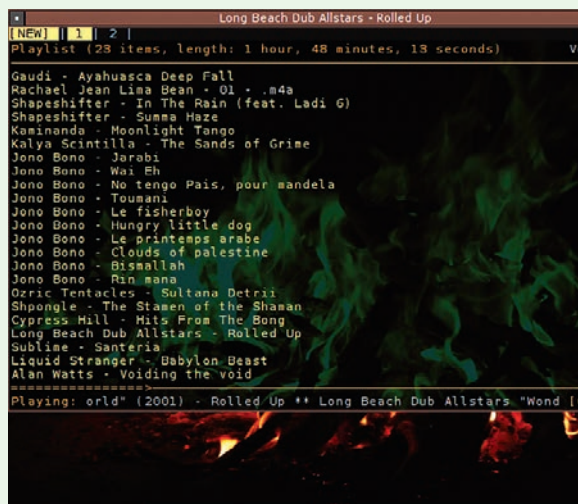
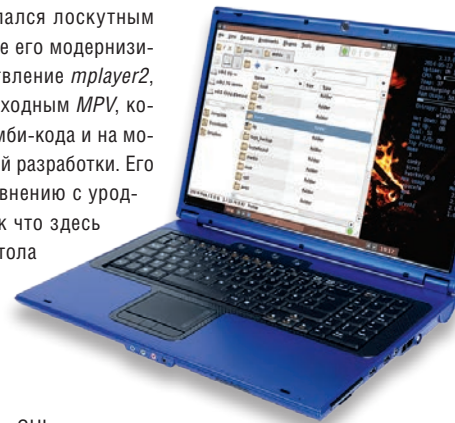
Было бы упущением не упомянуть здесь непоколебимый MPlayer — он воспроизведет практически все, что вы в него ни кинете, и делает это в устройстве буфера кадров или в виде ascii art), и выполнит практически любые ваши запросы по кодировке с помощью mencoder. Грустно то, что несмотря на по-прежнему

активную разработку кода, с годами он сделался лоскутным и напоминает перемешанную лапшу. В попытке его модернизировать создали оригинально названное ответвление mplayer2, хотя, похоже, этот проект затмевается превосходным MPV, который избавился от большей части старого зомби-кода и на момент написания находится в стадии интенсивной разработки. Его внутренний GUI действует безотказно (по сравнению с уродцем, которым был исходный GUI MPlayer), так что здесь нет необходимости в интерфейсах рабочего стола вроде Gnome-mplayer и smplayer (Qt), которые многие полагают неизбежными для его предшественника.

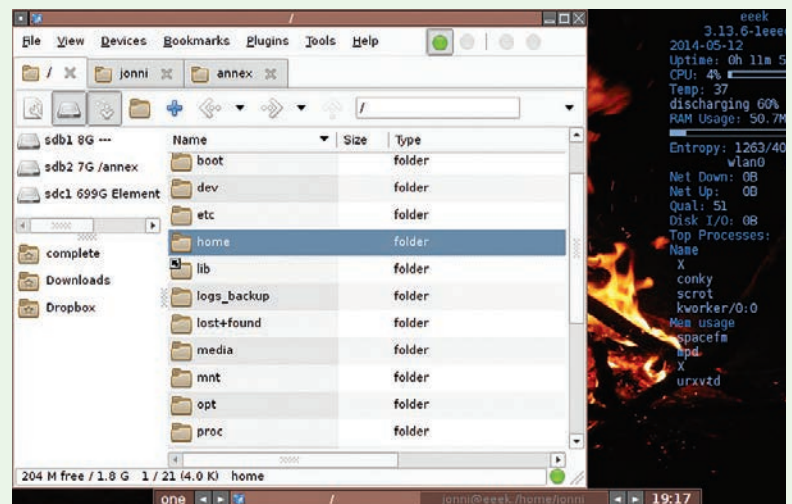
Медиа-плееры

VLC остается все таким же надежным, хотя его GUI использует Qt4. Иными словами, у него манеры всех остальных интерфейсов (ncurses, web, remote-control), и он может выводить результат в буфер кадров или в виде ascii art. Snappy — плеер GTK+ 3 среднего веса, построенный на средах GStreamer и Clutter, но имеет и ряд других зависимостей Gnome, а выглядит очень симпатично.

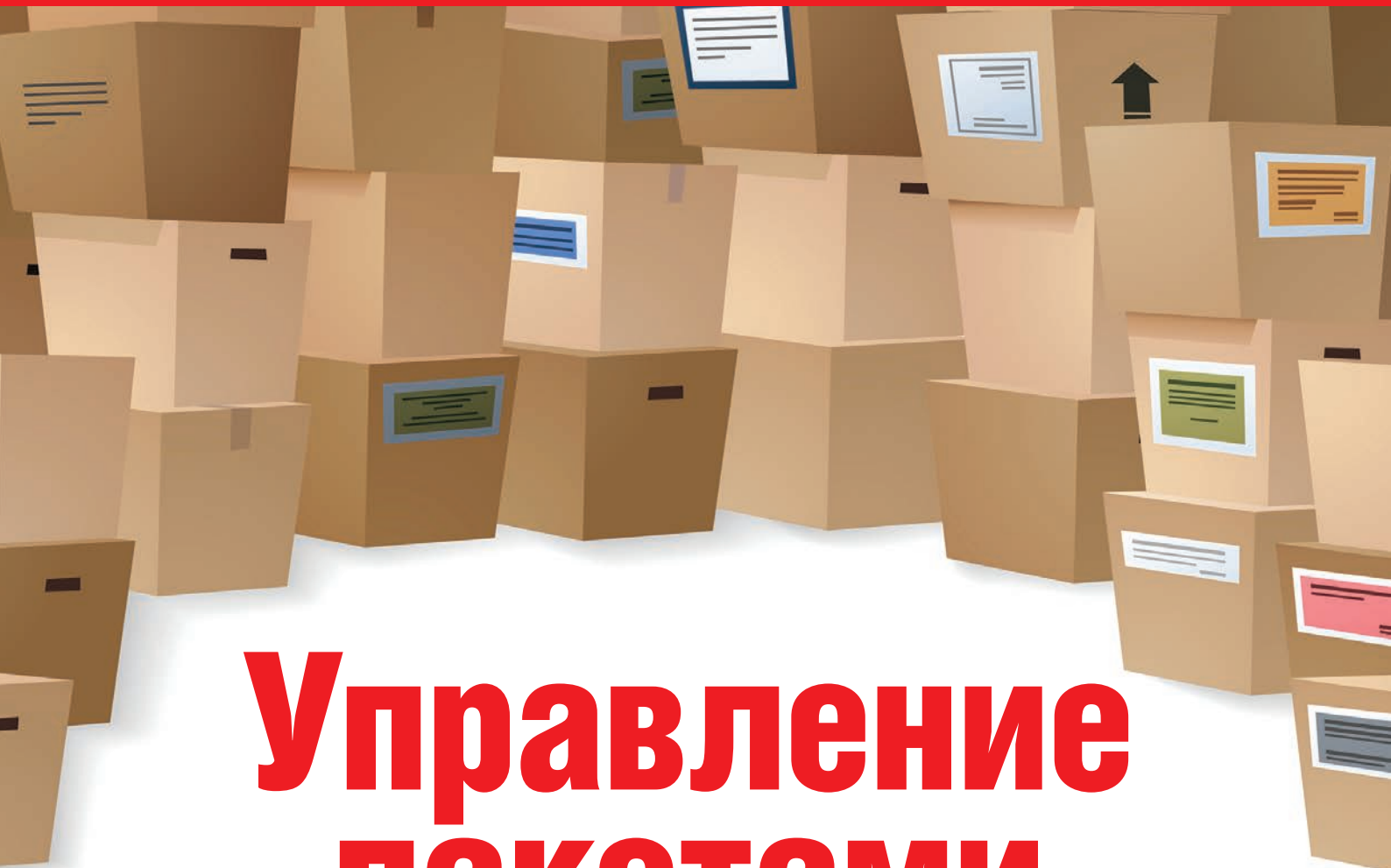
И это завершает наш обзор небольших приложений; надеемся, вы смогли найти для себя нечто легковесное взамен чего-то громоздкого, и выцарапали себе часть драгоценных ресурсов. Даже если у вас мощный компьютер, вы многого добьетесь, познакомившись с инструментами, о которых мы рассказали, или с иными инструментами того же сорта. Они помогут повысить эффективность работы, покажут вам новые способы выполнения задач и натренируют вашу память через необходимость запоминания горячих клавиш. LXF



» Ncstrapp и Conky, работающие на Fluxbox. Гламурно.



» SpaceFM, скомпилированный для GTK2+ с тремя вкладками, ест всего 15 МБ ОЗУ.



Управление пакетами

И как же оно работает?

Другие ОС предлагают вам скачивать и запускать непроверяемые бинарники, но **Джонни Бидвелл** считает эту практику ненавистной — Linux куда лучше.

Помимо всех прочих прелестей вашего любимого дистрибутива Linux, одной из самых важных, безусловно, является менеджер пакетов. Этот парень отвечает за столько всего, что трудно даже представить себе тот колоссальный объем работы, молча выполняемой им за сценой: он обновляет списки пакетов, разрешает зависимости, решает, в каком порядке эти зависимости устанавливать, находит и проверяет контрольные суммы пакетов и их аутентичность, проверяет, созданы ли они в правильном месте, осуществляет разного рода задачи пост-установки и, наконец, фиксирует успешность или неудачу всех этих усилий. И все это лишь потому, что вам приспичило установить KDE. Итак, посмотрим поближе.

Начнем со вводной номенклатуры. Вы, возможно, уже знаете некоторые из этих слов. Если

так, хорошо, а если нет — поясним некоторые ключевые слова. Во-первых, *пакет* — это контейнер для набора файлов вместе с инструкциями, сообщающими, что именно надо делать с этими файлами, чтобы они оказались полезными. В пакете могут быть некоторые *зависимости* — то есть другие пакеты, необходимые для обеспечения его

«Трудно представить колоссальный объем работы, выполняемый за сценой.»

работы. Могут также быть пакеты, улучшающие функциональность, но не обязательные; они называются *опциональными зависимостями*.

Чтобы пойти дальше, необходимо представить себе граф зависимостей. Эта структура содержит все зависимости нашего желаемого пакета,

и все зависимости тех, и т.д. Отсюда нам надо рассчитать упорядоченный список, без дублирования, чтобы ни один пакет не был установлен раньше своих зависимостей, и для решения этой задачи нам понадобятся математические расчеты. OpenSUSE и новый движок *Hawkey* в Fedora предлагают самое блестящее решение этой головоломки, обращаясь к Булеву алгоритму удовлетворительности. В общем и целом проблема в том виде, в котором она поставлена, именуется «топологической сортировкой», и решается только в том случае, когда нет циклических зависимостей, чего, при условии, что все остальное нормальное, не должно быть.

Не способствует ясности и то, что есть еще и виртуальные пакеты, не содержащие ничего осязаемого. Пример — виртуальный редактор пакетов Debian: он таинственным образом возникает

загвоздками являются флаги USE и CFLAGS), и *Portage* стоически продирается сквозь прихотливую паутину зависимостей времени компиляции или выполнения. При наличии обновления файла настройки *Portage* предложит вам запустить *etc-update*, позволяющий использовать старый, новый или сочетание обоих.

Менеджер пакетов Arch воплощает этический принцип Arch «Быть проще [Keep it simple]». Основные пакеты распределены по трем репозиториям (Core, Extra and Community) и имеют вид сжатых tar-архивов дерева директории пакетов и некоторых метаданных. Подпись пакетов появилась только в 2011 году; история той говорильной драмы лучше всего описана Toofishes (<http://bit.ly/Archsigning>), но все это делается очень стабильно и удобно с помощью PGP.

Простота *pacman* подразумевает, что определенные вещи не идеальны (и по сути невозможны) — например, установка более старой версии пакета, если у вас нет соответствующего tar-архива, потребует обращения к сайту Arch Rollback Machine, и в принципе не является хорошей идеей. А процесс обновления ядра, который обязан быть очень тонким и осторожным, вызывает немалое раздражение из-за своего довольно

медлительного и болезненного подхода. Изредка archlinux.org размещает уведомление «Требуется вмешательство пользователя [User intervention required]» — когда ожидается особо рискованное обновление. Обычно там содержатся вполне простые инструкции и предостережения, но если вы их проигнорируете и принудите (--force) транзакцию, есть вероятность повредить что-то критически важное.

Все делается командой *pacman*, в которой имеется шесть режимов:

- » **-S** для синхронизации и установки из репозитория Arch.
- » **-R** для удаления.
- » **-Q** для запросов.
- » **-D** для работы с базами данных.
- » **-T** для тестирования зависимостей.
- » **-U** для обновления файлов вне репозитория.

Если для обновления пакетов необходимо обновление файла настройки, то он устанавливается добавлением *.pacnew* к имени файла. Вашей обязанностью как ответственного пользователя будет перенести все изменения в файле *.pacnew* в свою текущую настройку.

Пакеты Debian имеют расширение *.deb* и являются архивами *.ar*, состоящими из трех файлов:

debian-binary, *control.tar.gz*, *data.tar.gz*. Первый содержит информацию о версии для двоичных файлов формата *.deb*, а для любых невинтажных пакетов просто содержит текст “2.0”.

Второй несколько важнее, поскольку содержит все пред- и пост-установочные скрипты, метаданные пакетов (архитектура, версия, ответственный, а также зависимости и предложенные/рекомендованные пакеты) и файлы настройки (если вы обновляете пакет, то они применяются так, чтобы сохранить все локальные изменения, которые вы могли сделать). И, наконец, tar-архив данных содержит все файлы, которые пакет собирается скинуть в вашу файловую систему root.

А в самом сердце системы управления пакетами Debian находится инструмент под названием *dpkg*. Он довольно прост: он может установить, запросить или удалить пакет Debian, и, следовательно, рассмотреть зависимости для определенного пакета, но никак не поможет вам в групповом их скачивании или установке. Скачивать файлы пакетов, сообщать вам о пакетах, которые вам больше не нужны, и заниматься отслеживанием версий скорее будет *apt* (*advanced packaging tool*). *apt* выступает в роли интерфейса для *dpkg*, и, в свою очередь, работает движком для разных графических инструментов, типа *Synaptic*. Наиболее часто наблюдаемые части пакета *apt* — команды *apt-get* для установки (удаления) и *apt-cache* для запросов.

Повседневные задачи

А теперь давайте перечислим самые часто встречающиеся задачи. Удобно использовать опции *Portage* *--pretend* или *--dry-run* от *apt-get*, которые делают одно и то же — то есть ничего — чтобы узнать, какое разорение учинит команда, если ее запустить без подстраховки. В *pacman* нет такой опции, но ему, по крайней мере, хватает ума потребовать подтверждения перед тем, как что-то сделать. Неблагоприятные могут отключить эту опцию с помощью *--noconfirm*.

Установка самой последней доступной версии пакета:

```
emerge packagename
pacman -S packagename
apt-get install packagename
```

Обновление системного списка имеющихся пакетов:

```
emerge --sync # (or eix-sync)
pacman -Sy
apt-get update
```

Обновление всех пакетов до последней доступной версии:

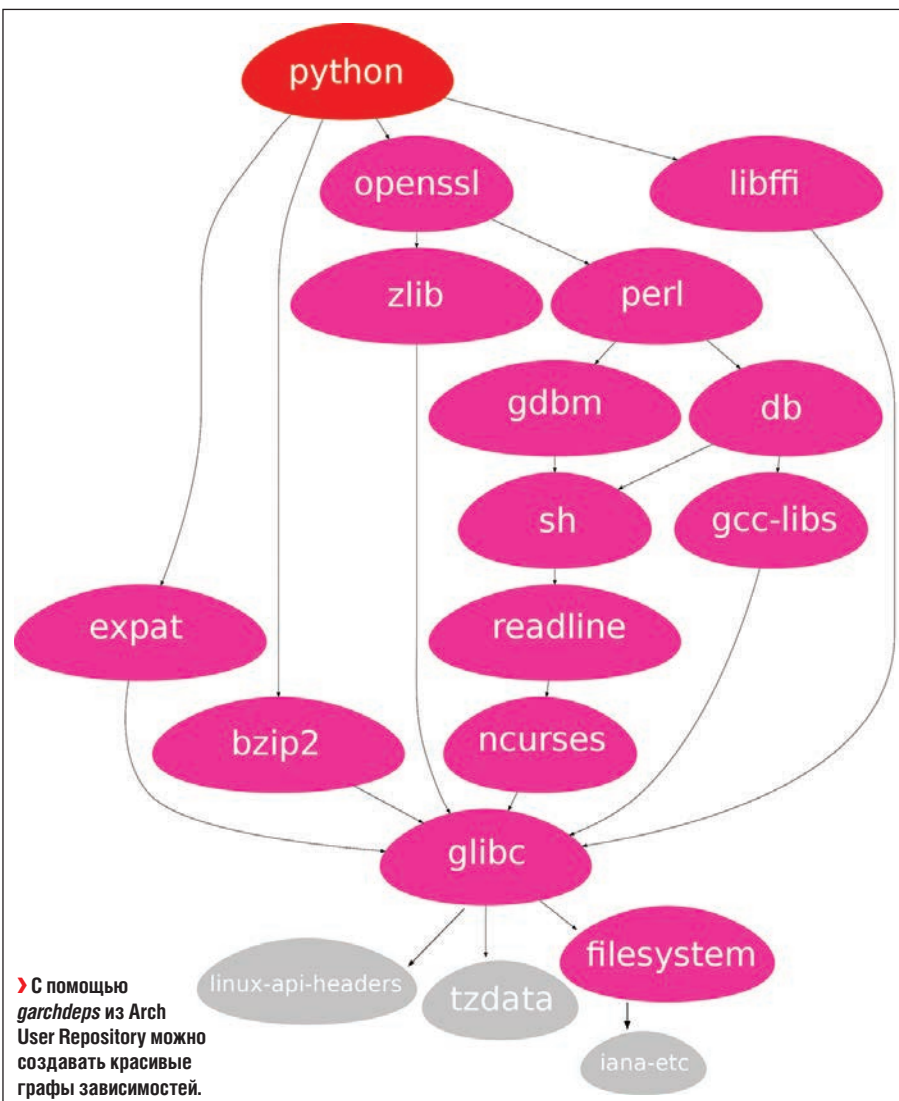
```
emerge -Duv @world
```

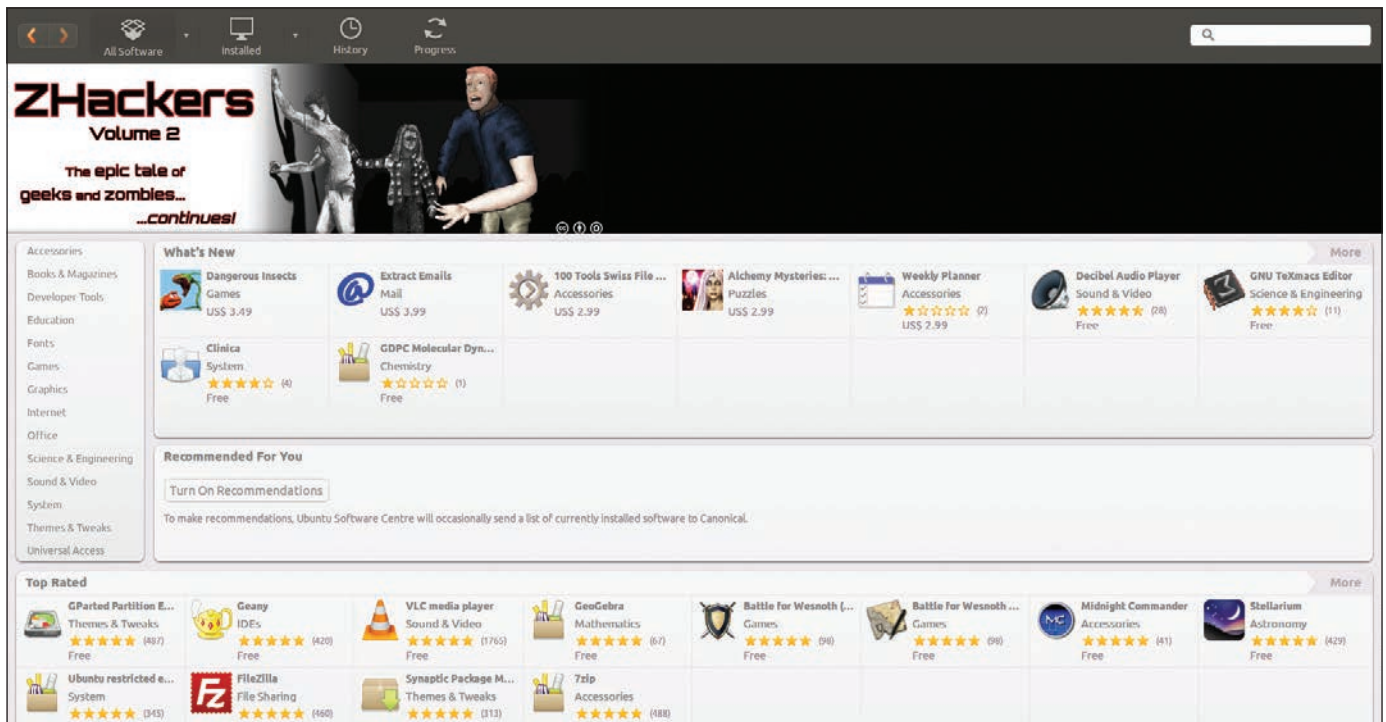
Опция **-D** (*--deep*) тоже обновляет зависимости, **-v** (*--verbose*) предоставляет больше информации о флагах USE. Вы также можете использовать опцию **-N** (*--newuse*), чтобы заново установить пакеты, где изменились флаги USE.

```
pacman -Su
```

Используйте *pacman -Syu* для комбинированного update/upgrade (это рекомендуемая процедура).

```
apt-get upgrade
```





► И умники, и зомби скоро будут в равной мере использовать магазины приложений, подобные Software-Centre.

Для более деструктивного обновления, обычно нужного исключительно для Unstable/Testing, используйте `apt-get dist-upgrade`.

Поиск пакета:

```
emerge --search имяпакета # (это медленно, и да поможет вам еix)
```

```
распап -Ss имяпакета # (или -Qs для поиска в установленных пакетах)
```

```
apt-cache search имяпакета
```

Удаление пакета:

```
emerge -C имяпакета
```

```
apt-get remove имяпакета
```

или `apt-get purge` для ликвидации файлов настроек.

```
распап -R имяпакета
```

Используйте `распап -Rs` также для рекурсивного удаления зависимостей.

Полное обновление пакетов совершенно не является проблемой в Debian и очень редко может стать таковой в Arch, но в Gentoo ситуация несколько иная. Железное правило Gentoo — при большом обновлении что-нибудь обязательно да пойдет не так. Вину нередко можно свалить на использование нестабильных версий пакетов и неофициальных накладок, но каждый пользователь Gentoo рано или поздно столкнется с подобной проблемой. Иной раз флаг USE пакета A требует установки определенной версии пакета B, тогда как пакет C настаивает на необходимости совершенно другой версии. Обычно эту проблему обходят, маскируя или демаскируя разные версии A, B или C либо изменяя проблемный флаг USE; что зачастую приводит к возникновению разного рода конфликтов. Иногда препятствия намного сложнее, и хотя *Portage*, спасибо ему, прилагает максимум усилий, чтобы сообщить вам, где именно возникают проблемы,

к сожалению, не всегда понятно, как их решить. Обычно на форумах полным-полно страдающих от разных недугов подобного рода, и рано или поздно кто-то умудряется найти решение; или проблема устраняется в новом релизе.

Packagekit

Проект *Packagekit* является уровнем абстракции, разработанным, чтобы включить независимый от дистрибутива подход к управлению пакетами. Он предназначен для реализации механизма «запусти-и-забудь» для всех ваших требований по установке программ. Новое приложение *Gnome Software* — пример интерфейса *packagekit*, как и *Apper* в KDE, как и инструмент командной строки *pkcon*. Интерфейс обращается к демону *packagekitd*, а тот обращается к движку каждого дистрибутива. И все это идет через Dbus, так что если пользователь запускает длительную транзакцию пакетов, а затем выходит из системы, или сессия X дает сбой, транзакция продолжится без проблем. Движки существуют у большинства дистрибутивов, но не все считаются стабильными/удобными. Полнофункциональная матрица доступна на www.packagekit.org/pk-matrix.html. Например, нужно переписать всю среду *Packagekit* для здоровой поддержки *Portage*, а *Pacman* во время перемещения пакетов часто требует вмешательства (знающего) пользователя, что нарушает закон Хьюси [Hughesie] насчет неинтерактивности транзакций. Однако *Packagekit* исключительно действенный для Debian, Fedora и openSUSE, и весьма вероятно, что пользователи

этих дистрибутивов смогут решать все свои дела, связанные с пакетами, через *Packagekit*, не контактируя с родными инструментами своего дистрибутива вообще.

Appstream — это схема XML, поддерживаемая основными производителями Linux, благодаря предоставлению экранных снимков пакетов и рейтингов, а также стандартизованных баз данных для хранения данных репозитория. Конечно их цель — обеспечение всех основных мелочей, чтобы любой, у кого появится такое желание и возможности, смог создать независимый

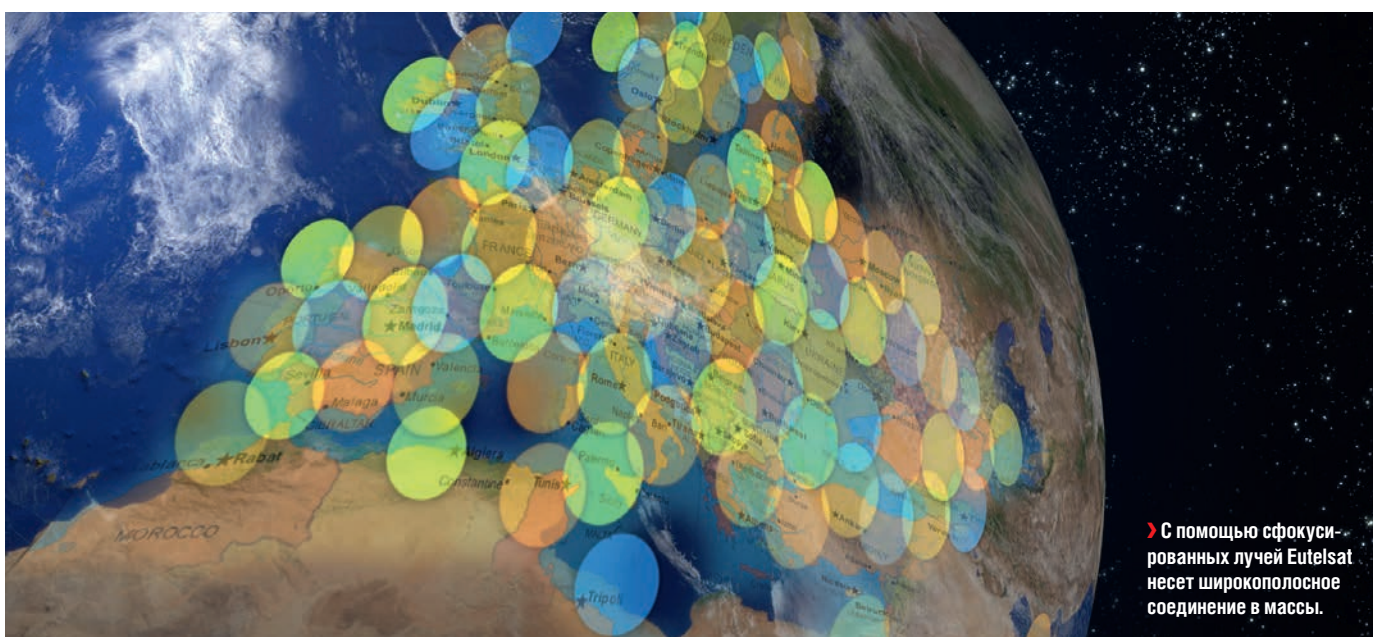
«Железное правило Gentoo — при обновлении что-нибудь обязательно пойдет не так.»

от дистрибутива Software Centre, или позволено ли назвать это App Store.

Подобного типа конвергенция может показаться анафемой Linux — в том смысле, что нам нравится обладать множеством разных возможностей что-то сделать и учить множество команд, чтобы сделать одно и то же, но разными способами. Однако ни в коем случае не стоит рассматривать это как попытку гомогенизировать весь процесс управления пакетами. Проверенные и перепроверенные инструменты каждого дистрибутива никуда не делись. Конечно же, они будут развиваться, чтобы лучше соответствовать этим новым моделям, но исключительно к пользе данного дистрибутива. *Packagekit* со товарищи предоставят средства обеспечения столь необходимой стабильности при повседневном управлении пакетами с рабочего стола. LXF

Скоростное, через спутник

Может ли один спутник обеспечить высокоскоростным широкополосным соединением всю Европу? **Нейл Мор** устал в небо в поисках истины...



► С помощью сфокусированных лучей Eutelsat несет широкополосное соединение в массы.

Все мы в наши дни рассчитываем на быстрый доступ к Интернету; и вот вы переехали, а вам говорят: кроме линии ADSL 2 Мб/с, ничего больше нет! Ужас. Хотите — верьте, хотите — нет, но именно с этой ситуацией сталкивается около 3,3 миллиона семей Великобритании.

И чтобы нарваться на медленное соединение, вам даже не надо попадать в такие отдаленные места, как Шетландские острова. На столь густонаселенных территориях, как Милтон-Кейнс и Слау, можно встретить соединение 2 Мб/с, которым пользуется около 20% семей. Порядка четверти из них проживает в центральном Уэльсе, и большая часть Северной Ирландии тоже страдает от столь низкой скорости передачи.

В этих местностях даже клиенты-везунчики не получают ничего сверх жалких 5 Мб/с. Проблема технологии ADSL заключается в расстоянии от центрального пункта — чем больше это расстояние, тем сильнее спадает сигнал и меньше

скорость передачи. Окажитесь на расстоянии более трех миль — и можете благополучно распрощаться с любым соединением ADSL.

Изначально пределом ADSL было 8 Мб/с, но широко используемый ныне ADSL2+ поднял эту планку до 24 Мб/с — однако при удаленности в 2,5 мили скорость падает до 4 Мб/с независимо от применяемой технологии ADSL. Итак, если только вы не проживаете в радиусе двух миль

«Лучи ка-диапазона дают прием на 20 Мб/с и передачу на 6 Мб/с по всей Британии.»

[~3,2 км] от центральной телефонной станции, то вы будете страдать от буквально ползающего интернет-соединения. Но помощь уже близка...

Спутники на подъеме

Используйте спутниковое соединение. От ледяных просторов Финляндии до жарких берегов Испании с помощью одного-единственного спутника одна

компания может обеспечить своих клиентов соединением 90 Гб/с по всему континенту. Так как же это все работает?

Домашний спутниковый интернет — отнюдь не новинка; в начале 2000-х потребительские продукты предлагали ускоренный сервис, только для закачек, с передачей через стандартный телефонный модем. Точно так же, как антенна спутникового телевидения получает цифровой видеопоток настроенного канала, эту технологию можно использовать для передачи цифровых данных для работы в Сети, скачивания и т.п.

Однако все это было во время оно, а технология не стоит на месте. Недавно мы поговорили с Eutelsat, одним из ведущих мировых операторов спутниковой связи, с 29 спутниками, обеспечивающими ТВ и коммуникационные сервисы по всему миру. В конце 2010 года он запустил Ka-Sat 9A, бывший на тот момент крупнейшим специализированным спутником с высокой пропускной способностью [High Throughput Satellite, HTS] в мире. Созданный в Великобритании и имеющий

Широкополосный интернет через спутник

вес 6,1 т, это был первый спутник, оптимизированный для обеспечения высокоскоростных широкополосных сервисов.

Использование лучей ка-диапазона позволяет ему предложить скорость приема 20 Мб/с и передачи 6 Мб/с всем и каждому, кто проживает в радиусе действия его 82 сфокусированных лучей — без снижения из-за географического положения внутри зоны действия луча. Расположенный на геостационарной орбите 22 200 миль в космосе, спутник размещается в фиксированной позиции над Европой.

Лучи, фокус на меня!

С этой высокой выгодной позиции он в состоянии перекрыть своими сфокусированными лучами большую часть Европы. Каждый сфокусированный луч покрывает территорию диаметром около 300 миль; сюда входит охват целиком Великобритании и Ирландии. Фактически, на этих прекрасных островах сфокусировано семь лучей, и это означает, что абсолютно все, от островов Силли до одиноких буровых установок в Северном море, могут наслаждаться быстрым доступом к Интернету.

Система сфокусированных лучей обеспечивает одинаковые радиочастоты вплоть до 20-кратной ретрансляции без потерь — каждый сфокусированный луч может работать с полосой почти в 1,1 Гб/с при общей пропускной способности спутника в 90 Гб/с. Интересно, что при запуске спутник выдавал скорость 70 Гб/с со скоростью в нисходящем направлении при 10 Мб/с в оригинальной спецификации — но когда систему настроили после ввода в действие, эта цифра увеличилась до общей скорости 90 Гб/с и скорости в нисходящем направлении 20 Мб/с.

Система радиопередачи в ка-диапазоне работает на 30 ГГц в восходящем и 20 ГГц в нисходящем направлении, и является ключом к повышению пропускной способности — более старые спутниковые системы должны были покрывать куда большую площадь пятна. Новые, узкие сфокусированные лучи увеличивают производительность целым рядом способов, в том числе посредством многократного повторного использования выделенных частот — это позволяет им достичь намного большей мощности, чем обычным спутникам

с широкими лучами. Узкие лучи обеспечивают более высокую и более сфокусированную мощность и в 35 раз увеличенную чувствительность к приему по сравнению с пропускной способностью стандартного спутника диапазона ku.

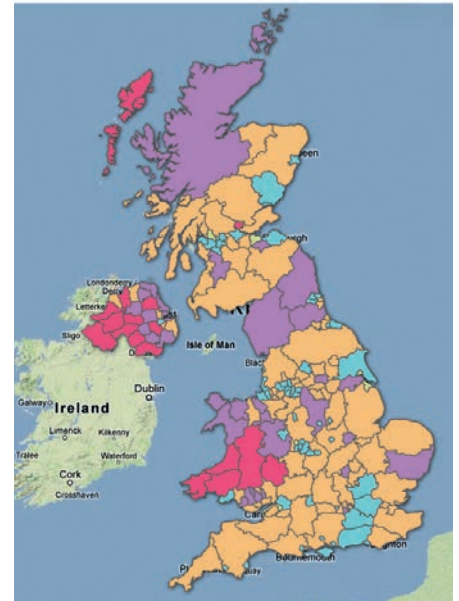
Независимый от погоды

С потребительской стороны уравнения находится 77-сантиметровая спутниковая тарелка со спутниковым модемом, демодулирующим/модулирующим информацию сети TCP/IP в соответствующий радиосигнал, который может принимать и отправлять тарелка. Модем Tooway предлагает обычный Ethernet-выход, что способствует интеграции с существующей домашней сетью или созданию новой сети вокруг нее. Eutelsat работает с восемью специализированными базовыми станциями, известными как шлюзы, куда спутник направляет соответствующий сигнал, а шлюз — действует в качестве ISP — перенаправляет сигналы в более крупную сеть Интернет и из нее.

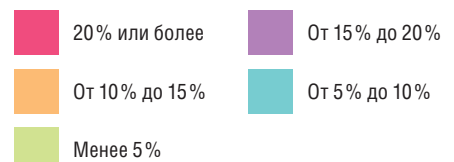
Интересно, что современная система ка-диапазона меньше страдает от неблагоприятных погодных условий. Пользователи Sky и Freesat часто обнаруживают, что проливные дожди влияют на ТВ-сигналы со спутника или полностью блокируют их — эффект, известный как «замирание при дожде». При развертывании системы ка-диапазона ожидалось, что она справится с условиями сильного дождя — так и произошло: согласно Tooway, при выходе на полную работоспособность он не будет подвержен подобному воздействию, в основном благодаря управлению уровнем мощности и некоторому снижению скорости восходящего потока для сохранения непрерывности работы канала связи.

Проблемы с ожиданием

Единственной слабостью спутникового интернета остается период ожидания: ликвидировать его невозможно. Поскольку спутник находится на расстоянии 22 236 миль, радиосигналу, перемещающемуся со скоростью света, требуется 250 мс, чтобы добраться до него. Обычно период ожидания колеблется в пределах 600–800 мс, делая эту систему неприемлемой для игр — даже у телефонного модема задержка составляет 120 мс.



Семьи, получающие ADSL менее 2 Мб/с



Потенциально спутниковый интернет может предложить меньший период ожидания, размещая спутники на более низкой орбите: 125 мс на 5000 миль, или 10 мс на 620 милях. Но на столь низких орбитах спутники не остаются в фиксированной позиции, и для поддержания соединения требуется сеть спутников. Это может вызвать дополнительные сложности в антеннах приемников, которым придется сбалансировать скорость передачи информации с удобством приема.

Несмотря на это, игры остаются единственным сервисом, который невозможен для спутникового интернета. Благодаря кэшу просмотр сайтов может быть таким же быстрым, как и при использовании наземной линии, а Skype и голосовые сервисы работают вполне нормально. LXF

Спутниковый интернет Tooway

Мы беседовали со Стивом Петри (Steve Petrie), коммерческим директором Бристанского филиала Tooway, о том, что их новый сервис может предложить жителям Великобритании, страдающим без широкополосного интернета. Клиентский сервис Tooway предлагает четыре пакета. Пакет Extra, стоимостью £ 79,50 в месяц, предлагает нисходящее соединение при 22 Мб/с, и восходящее — при 6 Мб/с, с предоставлением 100 Гб в месяц предоплаченного трафика (но любое ночное использование — с 0 до 6 утра — не входит в эту квоту).

Пакет Tooway 25 стоимостью £ 39,95 в месяц предлагает такую же скорость и до 25 Гб ежемесячного трафика плюс неограниченный трафик ночью. Tooway 10 предлагает 10 Гб трафика за £ 24,95 в месяц. Базовый пакет Tooway 2, за £ 15, предлагает 2 Гб трафика с нисходящей скоростью 5 Мб/с для нетребовательных пользователей.

Созданный компанией Astrium в Великобритании, Ka-Sat — спутник коммерческой связи весом 6,1 тонны, первый в Европе специально разработанный для обеспече-

ния широкополосного цифрового соединения с высокой пропускной способностью. Он был запущен 27 декабря 2010 года ракетой «Протон» с разгонным блоком «Бриз-М» и был введен в эксплуатацию в конце января 2011 года.

Основанный на платформе Eurostar E3000, разработанной Astrium, Ka-Sat остается самым передовым по своей конструкции и возможностям, и прогнозируемый срок его эксплуатации — 15 лет. Более подробную информацию вы найдете на www.tooway.com.





Ностальгия по приставкам

Петр Семилетов воскрешает устройства, милые сердцу геймера, на системе, милой сердцу линуксоида.

После развала Советского Союза на его просторах в 1990-е годы популярностью пользовались две игровые приставки: 8-битная «Денди» (российский клон японской NES — Nintendo Entertainment System) и 16-битная Sega Megadrive. «Восьмибиток» было еще великое множество, под диковинными именами вроде «Сюбор» или “Zhilitong”. Игровые картриджи продавались в особых отделах универмагов, в памятных коммерческих ларьках со жвачками и кассетам, на радиорынках и просто барахолках. Родилась целая субкультура людей, которые читали журналы о «Денди», помещали в газетах объявления про обмен картриджами, ездили за ними к черту на кулички и смотрели телевизионные передачи с обзорами игр.

В отличие от современных пресных и простых в прохождении игр, там был задор, соперничество человека и машины. Помер персонаж — играй заново, ведь сохранение не поддерживалось. И сидели до глубокой ночи, проходя уровень за уровнем да сражаясь с огромными боссами. Завершение последнего уровня считалось чуть ли не подвигом — можно было позвонить товарищу и похвастаться.

Наступление XXI века, кажется, вытеснило и эти старые добрые времена, и связанные с ними игровые консоли. Но буквально сразу с угасанием выпуска самих приставок... Хотя погодите, китайцы же штампуют до сих пор и NES, и SEGA Megadrive, и я покупаю к ним современные же

картриджи — даже *Angry Birds* вон портировали! Так вот, с угасанием официального, скажем так, производства упомянутых консолей появились их эмуляторы и так называемые «ромы [ROM, Read Only Memory — прошивки игр]» — образы картриджами, сохраненные в виде файлов. В эмуляторе вы запускаете такой ром-файл и играете на компьютере.

Долгое время в Linux было на чем поиграть в любимые приставочные игры. Если какая-то не шла в одном эмуляторе, ее поддер-

«В отличие от пресных в прохождении игр, там был задор.»

живал другой. Я был спокоен, что мои игры всегда под рукой. Для эмуляции SEGA я пользовался *Gens* (www.gens.me) и его вариантом *GensGS* (segaretro.org/Gens/GS). Это один из самых развитых эмуляторов SEGA Megadrive, выдающий графику и звук, мало чем отличающиеся от аппаратного подлинника.

С установкой 64-битной системы Mageia ни тот, ни другой эмулятор собрать из исходника не удалось — 64-битная архитектура не поддерживается. В Сети описаны мучительные пляски с бубнами, чтобы запустить *Gens* на 64-битном дистрибутиве, но игры должны приносить радость, а не способствовать умственному напряжению

и физическому истощению. Поэтому я решил поискать другие эмуляторы. Оказалось, есть такие. Большей частью тоже 32-битные, да еще только в двоичном формате.

Экзамен на совместимость и открытость выдержал только *Dgen* (dgen.sourceforge.net). Раньше я не обращал на него внимания, потому что пользовался *Gens*.

Корни *Dgen* уходят в глубокое прошлое, еще в 1990-е, когда, собственно, еще на ходу было множество «железных» консолей SEGA Megadrive.

С тех пор исходный код эмулятора неоднократно переписывался и был портирован на мультимедийную библиотеку SDL, плюс в недавнее время переработан для поддержки 64-битных систем, чему я несказанно рад. По сути, *Dgen* теперь — единственный эмулятор под Linux, который нормально работает на 64-битных дистрибутивах. Заслоненный *Gens*, годы пробивший у него в тени, *Dgen* может не входить в состав репозитория вашего дистрибутива, но прекрасно устанавливается из исходника.

У *Dgen* нет графического интерфейса, и файлы-ромы надо передавать в качестве параметра командной строки:

```
dgen имя_файла
```

Эмулятор настраивается через файл конфигурации `.dgen/dgenrc`, правкой его в обычном текстовом редакторе. Чтобы изменения вступили в силу, файл надо сохранить, а эмулятор перезапустить.

Первым делом настроим джойпад. Можно играть и без него, просто с клавиатуры — в таком случае функцию клавиши Start выполняет Enter, а клавиши Mode — Backspace. Курсорные же клавиши работают как крестовина джойстика, а для прыжков, стрельбы и прочего используется буквенное ASDQWE. Но если у вас есть джойпад, то почему и не подключить его?

Чтобы названия нажимаемых клавиш отображались в нижнем левом углу эмулятора, в файле настроек надо установить в значении true параметр `bool_buttons`, вот так:

```
bool_buttons = true
```

Запускаем эмулятор, нажимаем кнопки на джойстике и попутно правим файл `dgenrc`. Там есть строки, например, такого вида: `joy_pad1_a = "joystick0-button0"`

Это значит, что в эмулируемом сеговском джойпаде его кнопке A ставится в соответствие кнопка настоящего джойпада под названием "joystick0-button0". Итак, вы узнаете реальные названия кнопок прямо из окна эмулятора и вписываете их в файл настроек.

Скажем, `joy_pad1_mode` служит для назначения кнопки Mode, а `joy_pad1_start` — кнопки Start. Опции `joy_pad1_up` (down — вниз, left — влево, right — вправо) — это настройка осей крестовины джойстика. Для настроек второго джойстика — то же самое, только вместо `joy_pad1` в префиксе параметра используется `joy_pad2`.

Аппаратной кнопке сброса Reset соответствует компьютерный Tab. Другие клавиши, которые вы часто будете нажимать — цифровые, от 0 до 9. Ими выбирается слот для быстрого сохранения состояния игры. Таким образом, есть 10 слотов под каждую игру. Чтобы выполнить сохранение, нажмите цифру, а потом F2. Или просто F2, если хотите перезаписать текущее сохранение. А для загрузки — выбираем слот, жмем F3.

Alt+Return переключает эмулятор из оконного режима в полноэкранный и наоборот. Для того, чтобы полноэкранный режим был при запуске, служит параметр `bool_fullscreen = true`, однако у меня он почему-то не работает, хотя по Alt+Return переключение происходит отлично.

По умолчанию игра отображается в небольшом окошке, но его размером можно управлять через параметры:

```
int_width = ширина в пикселях
```

```
int_height = высота в пикселях
```

Картинка автоматически подгоняется под этот размер. Если соотношения сторон выглядит неестественно при использовании вывода графики через `OpenGL`, то поможет установка в true параметра `bool_opengl_aspect`.

Отрисовка осуществляется в двух режимах — с `OpenGL` и без него. При `OpenGL` нагрузка идет на видеокарту, а центральный процессор разгружается. Это если у вас всё в порядке с драйверами видеокарты и поддерживается аппаратное ускорение. Далее, в `OpenGL` по умолчанию включена линейная фильтрация текстур, что несколько размывает картинку. Я предпочитаю четкое изображение, сходное с приставочно-телевизионным, поэтому отключаю фильтрацию, указав следующее значение параметра:

```
bool_opengl_linear = false
```

К графическим опциям относится также выбор телевизионной системы — PAL или NTSC. По умолчанию выбрана последняя, и под нее задана частота обновления экрана — 60 герц. Для выбора PAL в файле настройки надо изменить значения двух параметров, а именно:

```
int_hz = 50
```

```
bool_pal = true
```

В первой строке мы устанавливаем частоту в 50 Гц, а во второй — делаем PAL режимом по умолчанию. Режимы переключаются также из командной строки запуска эмулятора, клавишами -P (для PAL) и -N (для NTSC). Большинство ром-файлов не требуют



► Dgen с запущенной игрой *Captain Havoc* (1994 г.).

с этим возни. А иногда и возня не помогает — я так не добился запуска PAL-версии некоторых игр.

Улучшаем звук. Вообще говоря, *Dgen* звучит лучше, чем *Gens*, но совершенству нет предела. Параметр `int_soundrate` ставим равным 44100 и наслаждаемся чистым звуком.

Некоторые хакнутые игры имеют неправильную контрольную сумму и не хотят запускаться. Чтобы исправить сумму, нажмите клавишу F1.

DGen умеет записывать «демо» вашей игры, чтобы потом вы смогли, не прилагая усилий, насладиться просмотром прохождения или показать его друзьям. Рассадите зрителей перед монитором и устройте им многочасовой показ. Но вначале надо записать прохождение. Демо-файлы сохраняются в каталоге `.dgen/demos`. Запускаем игру следующим образом:

```
dgen -d имя-демо-файла.demo имя-ром-файла
```

Например:

```
dgen -d sonik.demo sonik.smd
```

А чтобы воспроизвести демо-файл в том же каталоге, где у вас игра, а не демо, даем команду

```
dgen -D sonik.demo sonik.smd
```

То бишь для записи — маленькая буква **d** в параметре командной строки, а для воспроизведения — большая **D**.

Вот и весь краткий обзор возможностей *Dgen*, а подробности описаны в документации эмулятора. Добавлю, что *Dgen* умеет открывать ром-файлы в форматах BIN и SMD, а также ром-файлы, сжатые архиваторами `ZIP`, `gz` и так далее. Чего не хватает в эмуляторе, так это вывода игровой сессии в видеофайл (чтоб закинуть на Youtube) или звуковой файл (чтоб потом в плеере слушать музыку из игры). Ну да это с тем или иным успехом осуществляется сторонними программами.

Даже если *Gens* станет компилироваться под 64-битные дистрибутивы, я вряд ли теперь покину *Dgen* — он удобен в использовании и дает ощущение «приставочности». Это важно, потому что каким бы ни был технически навороченным эмулятор, если вы не получаете от него ощущений, близких к реальной консоли, вы теряете то, зачем вообще играете в старые игры — возможность душой и телом перенестись в прошлое.

К сожалению, не все эмуляторы обеспечивают это. Возьмем к примеру — а заодно перейдем к обзору эмуляторов NES, или Денди — эмулятор *FCEUX* (<http://www.fceux.com>). Он многолик и портирован почти на все мыслимые платформы, включая мобильные. Отлично работает на 64-битных системах. Оснащен громадным количеством настроек, обладает удобным графическим интерфейсом, основанным по вашему выбору (при компиляции) на `GTK2` или `GTK3`. Вывод графики — через



OpenGL и без него. Идут работы по переносу кода с библиотеки SDL1 на SDL2. Сложно найти более популярный и активно развивающийся эмулятор.

Но... когда вы играете в нем под Linux, не ждите плавной прокрутки. Особенно это ощутимо в бродилках. Ваш персонаж идет, местность сдвигается в сторону, но происходит это с ощутимыми и постоянными рывками. И что толку в совместимости FCEUX со всеми играми, в десятках полезных функций, если игровой процесс можно выдержать от силы несколько минут? Ну, ежели, конечно, не привыкнуть или не знать, как игра должна идти на самом деле.

К счастью, существует один из старейших эмуляторов NES — *Nestopia*, причем в нескольких ипостасях. *Nestopia* дает плавную прокрутку. Оригинальная *Nestopia* больше не разрабатывается, последняя ее версия вышла в 2008 году, а исходник заточен под сборку в среде Windows с использованием компилятора от Microsoft. Однако уже долгое время *Nestopia* успешно работает и в Linux. Чтобы установить эмулятор из исходника, необходимо следующее.

Вначале отправимся на официальный сайт *Nestopia* (nestopia.sourceforge.net) и скачаем оттуда архив с исходником — **Nestopia v1.40 Source Code**. Распакуем его куда-нибудь в отдельный каталог. Теперь нам нужна надстройка над этим исходником, позволяющая компилировать его в Linux при помощи GCC. Надстройку мы берем на странице Linux-порта *Nestopia* (http://rbelmont.mameworld.info/?page_id=200). Этот исходник надо распаковать поверх оригинального, и в том же каталоге дать из консоли команду **make**. Если у вас есть всё необходимое для сборки — заголовочные файлы *GTK2*, *OpenGL*, *GLU*, *SDL* — то начнется сборка, и вы получите бинарник под именем **nst**. Цель сборки **make install** не работает — бинарник надо скопировать от имени root вручную в видимый системе каталог, например, в **/usr/local/bin**.

При запуске появляется окошко со вкладками. Со вкладки Emulator можно загрузить ром-файл (кнопка Открыть) и после этого запустить игру (кнопка Play Game). Именно так: русская локализация выполнена лишь частично.

На вкладке Video — настройки видео. Я играю через рендер OpenGL, выставив Scaler в None, а параметр Scale равным 4 — при запуске игры это дает довольно большое игровое окно. Впрочем, можно переключиться в полноэкранный режим (Alt+Enter) или включать его постоянно, поставив галочку на Full screen.

▶ FCEUX с игрой Jackie Chan (1990 г.)



«К счастью, существует один из старейших эмуляторов, Nestopia.»

Вкладка Input служит для настройки устройств ввода, в частности, клавиатуры и джойпада, а вкладка Audio — опций звука. Сложно советовать что-то конкретное, ибо в зависимости от конфигурации вашей системы вы можете получить, скажем, плохой звук или графику там, где у меня всё идет гладко и звучит отлично.

Например, на вкладке Audio из списка выбирается звуковая подсистема — SDL, OSS или ALSA. С SDL игры у меня тормозят, с ALSA звука вообще нет, а вот с OSS работает на ура, хотя OSS в современных системах реализована через эмуляцию внутри ALSA! Исходя из своего опыта, дам, конечно, совет: выставить звуковую систему OSS и частоту оцифровки 44100 Гц, чтобы получить нормальный звук и правильную скорость игрового процесса. Но как это сработает у вас, я не знаю.

Я бы и дальше рассказывал про *Nestopia* в ее варианте, где надо сначала оригинал скачать, потом надстройку, но хочу уделить внимание дальнейшему развитию *Nestopia*, более совместимому с играми, более продвинутому. Это *Nestopia Undead Edition* (0ldsk00l.ca/nestopia/).

Первый и большой плюс *Nestopia UE* — исходник уже изначально совместим с Linux. Более того, в *Nestopia UE* упор делается именно на Linux и *BSD, хотя компиляция под Windows тоже доступна. Для сборки вам понадобятся *SDL 1.2* и *GTK+3*. *OpenGL* поддерживается, однако не является необходимостью. Сборка делается очень просто:

```
make
make install
```

Интерфейс у *Nestopia UE* более привычный, чем у «подлинника» — часть функций разбросана по меню, а для настроек предназначено отдельное окно. Во время написания статьи разработчик *Nestopia UE* готовит новую версию, поэтому если я буду говорить о недоработках, шероховатостях, возможно, в ней они будут исправлены.

Недоработки есть, и они мешают. Речь идет о быстром сохранении (F5) и загрузке (F8) состояния игры. Быстрое сохранение вообще не работает. А быстрая загрузка делает то же, что обычное сохранение. Обычные сохранение и загрузка имеются — это клавиши F3 и F4, они нареканий не вызывают. Вообще документированные привязки клавиш зачастую не работают, что, впрочем, не мешает вызывать те же функции через меню. Я пытался разобраться в исходнике, и кажется, что дело в запутанном коде передачи клавиатурных сообщений от SDL к GTK, или наоборот, я в подробности не вникал.

Другие ошибки. Сначала уж о плохом, а потом о хорошем. Настройка джойпада через графический интерфейс страдает тем, что в файл настроек запись новых значений идет без перезаписи предыдущих значений. Это вызывает полную неразбериху — вроде настроили, а работает не так. Лечение таково: открыть в текстовом редакторе файл **./nestopia/nstcontrols** и править его вручную. Там опции манипуляторов раскиданы по двум игровым портам. P1 — игрок 1, основной джойпад, и P2 — игрок 2, джойпад 2. Дам примеры из этого файла и поясню, как что трактовать.

- ▶ P1UP_UP — за положением крестовины «вверх» джойпада 1 закреплена курсорная клавиша клавиатуры «вверх».
- ▶ P1UP_JOA1MINUS — то же, но назначается «крестовина вверх» реального джойпада.

Отвлечемся: вот почему глючит настройка игровых контроллеров. Изначально задумано, что одному, скажем так, положению или нажатию кнопки может быть назначено несколько внешних сигналов — от клавиатуры и, кроме того, от джойпада. Это вызывает в коде некоторые ошибки, приводящие к наслоению старых настроек именно джойпада на новые.

Как же бороться и где получить хитромудрые идентификаторы клавиш джойпада? А очень просто. Закрываем файл настроек контроллеров. Идем в *Nestopia UE*, через графический интерфейс настраиваем джойпад, хотя и кажется, что безуспешно. Затем закрываем эмулятор, снова открываем файл настроек управления, и — удаляем вручную старые настройки джойпада, но оставляем новые!

Теперь о хорошем. *Nestopia UE* меньше нагружает процессор, чем обычная *Nestopia*. Есть интересная функция записи «фильмов» — это не видеофайлы, а фактически наборы команд, которые потом могут быть воспроизведены. Используйте *File > Record movie* для начала записи фильма, *File > Stop movie* для остановки записи или воспроизведения, и *File > Load movie* для выбора файла и последующего воспроизведения фильма.

Nestopia UE, в отличие от простой *Nestopia*, более совместима с играми. Например, есть такая пиратская игра *Aladdin*. Многие думают, что она официально выпущена компанией Virgin; на самом деле это умельцы-пираты забражали порт на собственном движке от *Somari* (NES-вариант ёжика Соника, но с Марио в главной роли), и не каждый эмулятор его поддерживает. И вот на обычной *Nestopia* в него не получится играть, а в *Nestopia UE* — можно!

Совместимость игр и эмулятора зачастую определяется такой штукой, как маппер [mapper]. Если эмулятор поддерживает конкретный маппер, игра работает; нет — ошибка! Мапперы в игровых картриджах — это чипы, которые, упрощенно говоря, кроме предоставления доступа к памяти картриджа расширяют аппаратные возможности консоли, служа посредниками между нею и железом картриджа (где может быть дополнительная, непредусмотренная в момент выпуска консоли аппаратная часть).

По адресу <http://wiki.nesdev.com/w/index.php/Mapper> находится описание мапперов. Какой неподъемный труд проведен создателями эмуляторов, чтобы обеспечить возможность загрузки ваших любимых старых игр!

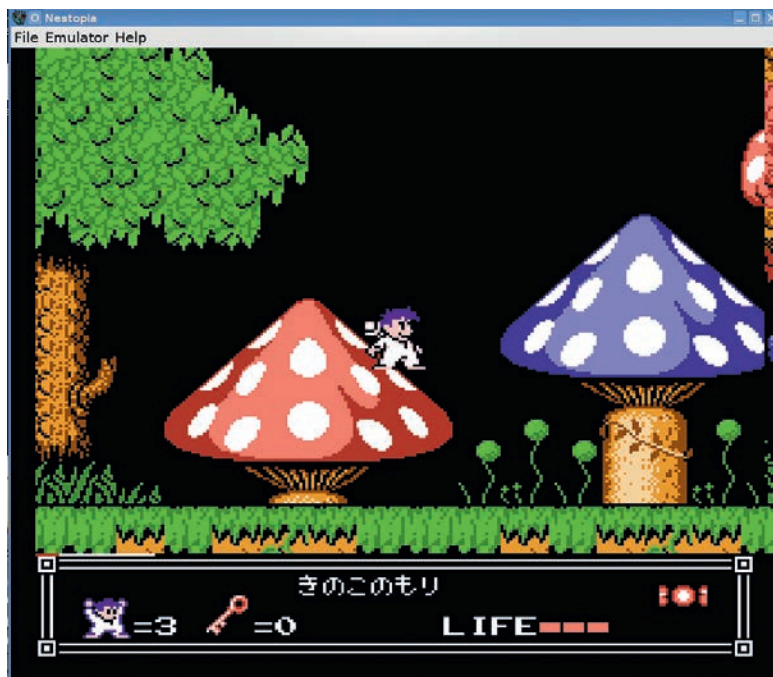
Некоторые картриджи, обычно с ролевыми играми (*Kirby's Adventure*, *Startropics*, *Zelda 1*, *Zelda 2*), даже содержали в себе батарейки для поддержки памяти, в которой сохранялись состояния игр.

Стандартный картридж NES может хранить в себе 16 или 32 килобайта игрового кода и 8 килобайт тайлов (англ. tile — плитка; это кусочки графики, из которых составляется картинка). Можно расценивать тайлы как двумерные текстуры. Если графики или кода больше, нужен маппер, чтобы обеспечить консоли доступ к этой дополнительной памяти картриджа, где находится «большой» код и графика.

Настройки в *Nestopia UE* большей частью те же, что в оригинальной *Nestopia*, и хранятся в том же файле, хотя есть и ряд отличий. Например, в *Nestopia UE* в *Configuration > Video* есть опция *TV Aspect ratio* — советую ее включить, чтобы изображение не было квадратным, с искаженными соотношениями сторон. При запуске обеих версий *Nestopia* те настройки, которые не поддерживаются «конкурентной», сбрасываются в значения по умолчанию. Поэтому лучше остановиться на каком-то одном эмуляторе.

Опция, разделяемая обеими — *Default system* [Система по умолчанию] со списком доступных для эмуляции систем: *Dendy*, *Famicom*, *NES (PAL)*, *NES (NTSC)*. Ведь NES выпускалась как *Famicom* в Японии, как NES в США, ну а про *Dendy* вы сами знаете. Разница между режимами — в эмулируемой скорости процессора и графического чипа (PPU — *Picture Processing Unit*), обработке видео и звука. Если конкретная игра в эмуляторе работает странно — например, тормозит, или звук запинаяется — попробуйте выбрать другую «систему по умолчанию».

В Git-хранилище *Nestopia UE* (github.com/rdanbrook/nestopia) доступна разрабатываемая версия эмулятора, я ее пробовал — пока играть на ней неудобно, очень сырая. Не зря разработчик



Nestopia UE
с игрой *Nemo the Pajama Hero*.

пока не выложил новый официальный релиз. Но можно, с поправкой на предварительность данной версии, поговорить о том, чем он нас порадует в будущем.

Вывод графики портирован на современную версию библиотеки SDL — вторую. Визуально всё осталось как прежде, но добавилась возможность регулировать, как на телевизоре, контраст, оттенок и тому подобное. Появилась и куча разных опций: включение и выключение вертикальной синхронизации, настройки палитры.

Переписан вывод звука — теперь вместо внутреннего для *Nestopia* кода поддержки OSS и ALSA используется библиотека-прослойка *libao*. В моем случае, при текущем состоянии исходника, звук воспроизводится хорошо, только если выбрать частоту оцифровки 48000 Гц. Если к *libao* приложить руки, то возможен вывод звука не только на звуковую карту, но и в WAV-файлы.

Прежние версии *Nestopia* не эмулировали, в отличие от *FCEUX*, турбо-кнопки джойпада. В новой версии это есть, но пока отсутствуют настройки. Стал поддерживаться световой пистолет, а в ядре — дополнительные мапперы.

Грядущая *Nestopia UE* может не показывать графический интерфейс, если в командной строке не передать ром-файл. С одной стороны, это странно — по привычке хочется запустить эмулятор и в нем через диалог Открыть выбрать нужный файл. С другой стороны, в последнее время в GTK диалог выбора файла претерпел столь много порицаемые пользователями изменения, что попробуем обойтись без него. Хотя диалог никуда не денется, вызывается через *Файл > Открыть*. Подождем обновления эмулятора, посмотрим, как будет в окончательном варианте.

На этом обзор *Nestopia* и ее форков завершу, да и вся статья подходит к концу. За бортом осталось еще несколько эмуляторов с открытым кодом, о которых я не говорил по разным причинам. Например, в *Mednafen* — плавная, качественная графика, однако он в упор не видит подключенный через USB джойпад. Другие эмуляторы вроде и под GPL, но компилируются только в Windows, а зачем мне Windows?

Таким образом, сейчас реально играть на 64-битных дистрибутивах Linux в «сеговские» игры только на *Dgen*, а на любых дистрибутивах в «восьмибитные» игры — на *Nestopia UE*. И если *Dgen* находится в состоянии поддержания на плаву, то *Nestopia UE* активно развивается, и по обилию функций скоро сравняется с *FCEUX*, уже обгоняя его по качеству игрового процесса. LXF



Что за штука... GoboLinux

Мэтт Хэнсон испытывает дистрибутив, не похожий на другие.

В Так-так... стало быть, GoboLinux — очередной дистрибутив на базе Debian, или какой-нибудь другой банальной версии Linux?

О Вы даже и не представляете, насколько вы далеки от истины. Хотя на первый взгляд GoboLinux и может показаться обычным дистрибутивом — вероятно, этому способствует вид довольно популярного рабочего стола *Enlightenment*. Однако за знакомым интерфейсом скрывается уникальная система.

В О, вы меня заинтриговали! Что же особенного в GoboLinux?

О Ну, GoboLinux — это модульный дистрибутив, имеющий ряд ключевых отличий от других вариантов Linux. В первую очередь это альтернативная файловая структура, основанная на более ранних операционных системах, таких как NeXTStep и BeOS.

В Альтернативная файловая структура? Что это значит?

О Может быть, на бумаге это и не самая удивительная в мире функция, но по сути она кардинально влияет на принцип работы GoboLinux. В стандартных файловых системах Unix, применяемых большинством дистрибутивов Linux, при установке какой-либо программы некоторые её файлы помещаются в поддиректорию `usr/bin`, а остальные — в папку `/etc`. А в GoboLinux принцип другой. Когда вы устанавливаете программу, у неё появляется собственная поддиректория в дереве `/programs` — то есть если вы установили VLC, она появится в `/programs/VLC` — и все файлы будут храниться там по версиям. Это означает, что параллельно могут устанавливаться и поддерживаться несколько версий одной и той же программы.

В ОК, какие ещё плюсы у этой альтернативной файловой системы?

О Плюсов у более структурированной и в некотором отношении более прозрачной файловой системы может быть множество. К примеру, раз все программы установлены в одной и той же

директории `/programs/`, то в неё можно зайти командой `cd`, а затем, набрав `ls`, просмотреть все установленные в вашей системе программы.

В Значит ли это, что из-за иной схемы расположения файлов у меня не получится установить мои любимые программы?

О Хотя файловая структура GoboLinux существенно отличается от привычной Unix'овской, запустить программу, ориентированную на этот стандарт, проще простого. Это осуществляется путём сопоставления традиционных путей, по которым установлены разные программные компоненты с их местонахождением в файловой системе GoboLinux — благодаря чему его совместимость даже лучше, чем у других дистрибутивов.

В Звучит неплохо — но откуда взялась эта альтернативная файловая система?

О Идея такой необычной файловой системы возникла, когда разработчик GoboLinux, Хишам Мухаммад [Hisham Muhammad], был ещё

студентом. Не имея доступа к записи в стандартной директории Unix на машинах, которыми он пользовался, Хишам создавал собственные внутри \$HOME. Поскольку обновлять приходилось из исходников — ведь он не мог использовать менеджер пакетов — решением стало расположить каждую программу в своей директории. Со временем переменные окружения росли, поэтому начали создаваться централизованные директории с символическими ссылками для каждого класса файлов, а также сценарии оболочки для работы с ними. Эта файловая система оказалась столь удобной в использовании, что когда Хишам и его друг Андре Деч [Andre Detsch] приступили к созданию улучшенной системы Linux From Scratch, они включили туда и эту новую систему структуры файлов; так и появился GoboLinux.

В Минуточку. GoboLinux... Кажется, когда-то давно я уже слышал это название?

О Верно. GoboLinux существует с 2003 года, а смутность ваших воспоминаний связана с тем, что некоторое время не было слышно об обновлениях — однако теперь всё иначе.

В Что значит — не было слышно об обновлениях?

О Ну, GoboLinux версии 015 вышел в прошлом месяце, но предыдущее обновление (014) имело место в конце 2007-го, так что разработка велась темпами, скажем так, оползания ледника.

В Да уж, отнюдь не торопись. Чем же занималась команда все эти годы между релизами?

О Шестилетний промежуток между релизами не означает, что команда бездействовала. Помимо выпуска промежуточных обновлений и исправлений ошибок (таких как 014.01), команда занималась обновлением рецептов компиляции.

В Рецепты компиляции — это нечто новое... что это такое?

О С помощью инструмента *Compile* пользователи GoboLinux могут скачивать tar-архивы с исходниками, чтобы затем распаковать и установить их, используя простой скрипт, именуемый рецептом. Хотя компиляция программ из исходников в Linux — дело привычное, в силу специфики GoboLinux (и в особенности его уникальной структуры каталогов) многие программы по умолчанию не предполагают такой конфигурации, для чего и нужны *Compile* и рецепты.

В Значит, эти рецепты Gobo Linux — нечто вроде .deb или .rpm?

О Вроде того. Большинство программ изначально представляют собой сжатые файлы .tar.gz, набитые битами и блобами — это, кстати, вполне приемлемые технические термины — из которых собираются исходники. Системы

управления пакетами затем компонуют, настраивают и устанавливают эти пакеты для создания и запуска программы на вашем дистрибутиве.

В основе *Compile* в GoboLinux лежит идея о том, что сформированный по всем требованиям GNU .tar.gz делает программу доступной на любом дистрибутиве Linux; тогда как если сжатый файл упакован в разные пакеты под разные дистрибутивы, это приводит к тому, что некоторые называют «адом зависимостей» — когда всё усложняется. Существуют такие проекты, как *GNU Autoconf* и *GNU Libtool*, способные обеспечить совместимость программ с GoboLinux или любым другим дистрибутивом без специальных пакетов. Так что в GoboLinux, на самом деле, нет своего Deb-стандарта. Пока растёт поддержка *GNU Autoconf*, команда GoboLinux выбрала *Compile* и рецепты,

Чтобы создать совершенно новый рецепт для программы, у которой такового ещё нет, вам понадобится команда **MakeRecipe**, а также web-адрес файла **tar.gz** с исходным кодом и файлами, чтобы скачать и создать его автоматически.

Если у программы, которую вы хотите установить, уже есть рецепт, но для старой версии, воспользуйтесь командой **NewVersion**, чтобы его обновить.

В ОК, вот я создал свой собственный рецепт, прямо как шеф-повар. А дальше что?

О Теперь поделитесь своим открытием со всем миром! У GoboLinux — огромная библиотека рецептов для программ и инструментов всех мастей, во многом благодаря преданному сообществу людей, создающих свои собственные рецепты

«За знакомым интерфейсом скрывается уникальный дистрибутив Linux.»

в основе которых лежат идеи *GNU Autoconf* и *GNU Libtool*. Рецепты отличаются от RPM, Deb или ebuild тем, что являются не скриптами, а файлами, описывающими процесс компиляции программы, чтобы по возможности упростить этот процесс.

В А где я могу найти список этих ценных рецептов?

О На сайте GoboLinux, страница <http://recipes.gobolinux.org/ru/>. Здесь же вы можете скачать последние релизы (они упорядочены по дате: самые новые — в начале списка), или воспользоваться текстовым полем для поиска отдельных программ или рецептов. Каждый рецепт сопровождается информацией о его размере и создателе, а также аннотацией. Скачивать рецепты можно там же, или обратиться на сайт программы за дополнительной информацией.

В Хорошо, допустим, я нашёл нужный рецепт. Что дальше?

О Поскольку *Compile* входит в состав GoboLinux, начиная с версии 011, вам нужно просто открыть терминал и набрать *Compile* и имя программы, например, **Compile Audacious**. Если всё хорошо, то больше от вас ничего и не требуется — *Compile* найдёт рецепт для программы, имя которой вы ввели, скачает файлы, распакует их и установит её для вас.

В А могу я создавать собственные рецепты? Это сложно?

О Конечно, можете, ведь и *Compile*, и рецепты созданы для обеспечения максимальной простоты.

и готовых ими делиться. Если и у вас появится такое желание, то упакуйте свой рецепт командой **GenRecipeStore**, укажите своё имя в файле **Compile.conf**, чтобы отметить на сайте проекта, и добавьте файл в рассылку gobolinux-recipes.

В Значит ли это, что арсенал программ GoboLinux не меньше, чем у других крупных дистрибутивов?

О Как взглянуть. Из коробки, в GoboLinux нет такого обилия программ, как в более популярных Ubuntu или Mint. Но, как уже отмечалось ранее, многие программы становятся доступными в GoboLinux благодаря рецептам; их список легко скачать, и он быстро растёт. Это также означает, что при готовности немного потрудиться, даже если какой-то программы там нет, вы можете создать к ней свой собственный рецепт.

В ОК, наверное, я решусь попробовать GoboLinux. Откуда можно его скачать?

О С официального сайта www.gobolinux.org можно скачать ISO-образ последнего релиза. Если вы не уверены, понабравшись ли вам GoboLinux, попробуйте его на live CD. В последней версии GoboLinux есть графический установщик, значительно упрощающий весь процесс.

На данный момент GoboLinux существует в сборке для систем класса i686 — так что если у вас Pentium IV или выше, всё должно пройти нормально. Минимальный объём ОЗУ — 128 МБ, но мы бы посоветовали иметь не меньше 1 ГБ для лучшей производительности. За более подробной информацией об уникальных особенностях GoboLinux стоит обратиться на официальный сайт. **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.

Эзотерическое системное администрирование из причудливых заворотов кишок серверной

Нездоровое привыкание

Действие короткого рассказа Айзека Азимова [Isaac Asimov] «Чувство власти [The Feeling of Power]», который был написан еще в 1958 г., происходит во времена, когда люди стали настолько полагаться на компьютеры, что разучились считать. (Прочтите рассказ на <http://bit.ly/1mXM4Ax>). Как и любая хорошая фантастика, с течением времени она становится все реальнее.

Недавно мы покупали новый ковер. Оценивая сопутствующие затраты, продавец пощелкал кнопками на калькуляторе и сообщил, что нам нужно 2000 квадратных метров. Но квадратный корень из 2000 — м-м, ага, квадратный корень из 20, м-м, между 4 и 5, значит, квадратный корень из 2000 будет где-то между 40 и 50. Но наша гостиная никак не 40 м в длину! Смысл этой истории в том, что молодой продавец и не думал, что ошибся — он не имел представления о «порядке величины». Некоторые из вас помнят, как продавцы считали сдачу в уме. Сегодня это делает за них кассовый аппарат.

Я не ворчу, о нет. Раньше я сам помнил телефонные номера двух десятков друзей, а теперь достаю их из телефона. А многие из нас разучились читать географические карты, полностью положившись на содействие спутниковой навигации.

Скоро гаджеты вроде очков Google станут повсеместными, и люди разучатся помнить факты. Думаете, это вред ли? Возможно. Но мне уже приходит на ум рассказик...

chris.linuxformat@gmail.com



Сердце еще кровоточит

Ошибка исправлена, но в мире еще полно сертификатов, которые могут взломать.

Если в последние несколько месяцев вы жили не в пещере, то точно слышали об ошибке Heartbleed. (Или прочтите статью ArsTechnica на <http://bit.ly/stillheartbleeding>). Насколько я знаю, это единственная уязвимость с собственным логотипом.

Ладно, новость несвежа, но интересно, как по-разному на нее отреагировали. Брюс Шнайер [Bruce Schneier], известный эксперт по безопасности, пишет: «Это катастрофа. По шкале от 1 до 10 я бы поставил ей 11». Но хотя эта уязвимость существовала в OpenSSL два года, никто понятия не имеет, много ли народу знали о ней или (что важнее) ею воспользовались, и были ли вообще такие люди. Вера в честность спецслужб ныне переживает не лучшие времена, и кое-кто считает, что они знали об этой уязвимости с самого начала.

Большинство сайтов, где были уязвимые версии OpenSSL, обновили ее, но проблема этим не исчерпывается: злоумышленник, который благодаря этой уязвимости получил, скажем, доступ к секретному ключу, может по-прежнему пользоваться им и цифровым сертификатом сайта, создавая видимость того, что сайт по-прежнему

использует защищенное соединение. Многие сайты сгенерировали новые пары ключей и обновили сертификаты безопасности. Некоторые центры сертификации позволили клиентам сделать это бесплатно. Для других Heartbleed, должно быть, стал золотым дном.

Но выпуск нового сертификата не мешает злоумышленнику выдавать свой сайт за настоящий, пользуясь старым сертификатом и ключом, пока срок действия сертификата не истечет. Чтобы этому помешать, старый сертификат нужно отозвать. Отзыв — это объявление сертификата недействительным до истечения срока действия. И это непросто, поскольку инфраструктура сертификатов не рассчитана на большие объемы отзывов. По оценке Netcraft, на 16 апреля из 500 000 или большего количества сертификатов, требовавших замены или отзыва, было обработано всего 80 000. А по мнению некоторых экспертов, отзыв сертификата — совершенно неэффективная мера.

Тем не менее, я не бросил все и не побежал менять все пароли. Мне следует взять себя в руки и жить дальше. Но вообще-то в пещере по нынешним временам гораздо безопаснее.

Был ли сертификат отозван?

Проверить, был ли отозван действительный с виду сертификат, для браузера не так-то просто. Можно проверять по списку отзывов, который должен часто обновляться — хотя современные браузеры норовят не делать этого по умолчанию. Вместо этого

они могут уточнить, был ли сертификат отозван, обратившись к центру сертификации по OCSP (Online Certificate Status Protocol — онлайн-протокол статуса сертификата). К сожалению, ни одно из этих решений толком не масштабируется.



Интеграция Linux и Windows

Отринув свои планы мирового господства, добрый Доктор размышляет, сумеют ли сработаться Linux и Windows...

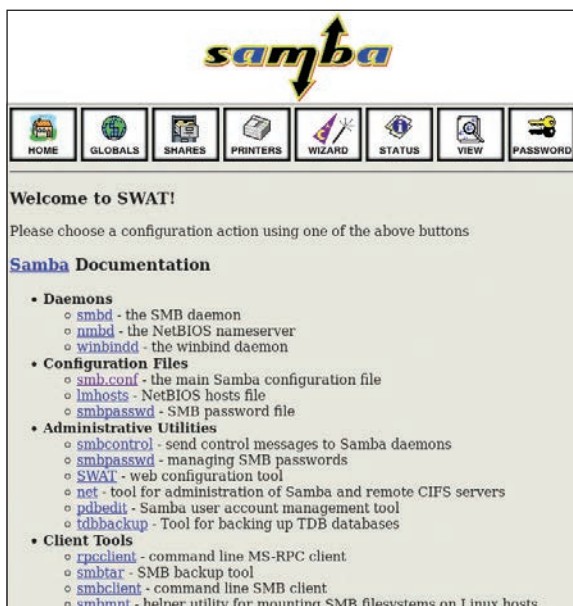
Мечта системного администратора — чтобы у всех пользователей в его компании была одна и та же версия одной и той же операционной системы (желательно Linux) — не сбывается; по крайней мере, на этой неделе. По мне, оно и к лучшему: я предпочитаю некоторое разнообразие видов.

Тем не менее, многие компании внедряют Linux (и серверный, и настольный) в инфраструктуры, которые раньше были вотчиной Windows, и следовательно, этим системам нужно общаться друг с другом. Не стоит и говорить, что в вопросе совместимости Linux продвинулся гораздо ближе к Windows, чем Windows к Linux.

Поэтому в ближайшие несколько месяцев я хочу поговорить об утилитах, которые поддерживают интеграцию Linux и Windows. Немалую роль здесь играет *Samba*, но есть и другие заслуживающие внимания решения, например, *Wine*. На данном уроке мы поговорим об обмене файлами.

Урок истории

Как вы, наверное, знаете, и в Linux, и в Windows есть свои протоколы для доступа к удаленным файловым системам. В Linux есть NFS (Network File System — сетевая файловая система), разработанная в Sun Microsystems. В Windows изначально использовался протокол под названием SMB (сокращение от “Server Message Block” — Блоки сообщений сервера), предшествовавший принятию в Microsoft сетей TCP/IP и вместо них использовавшийся сетевой API под названием NetBIOS, который, в свою очередь, работал по протоколу под названием NetBIOS Frames (NBF). У каждого компьютера было простое текстовое название — его имя для NetBIOS — и оно изначально использовалось для прямой идентификации компьютера. Цифровых адресов, аналогичных IP-адресам, не было.



» Домашняя страница web-утилиты администрирования Samba (SWAT). Здесь можно найти map-страницы и варианты окон с параметрами.

Где узнать больше?

Имеется три книги о Samba 3: “A Samba 3 HOWTO guide [Практическое руководство по Samba 3]”, “Samba 3 by Example [Samba 3 в примерах]” и “Developers’ Guide [Руководство для разработчиков]”. Полные PDF-версии этих книг включены в пакет *samba-docs* в репозиториях CentOS. Их также можно найти в пакете с документацией (зайдите на <http://www.samba.org/>

[samba/docs](#) и щелкните по ссылке Daily Docs Build); этот архив также содержит все map-страницы.

Книг по Samba 4 пока вышло немного, но Марсело Леаль [Marcelo Leal] только что (в апреле 2014 г.) опубликовал свою — “Implementing Samba 4 [Реализуем Samba 4]”. И в следующем месяце я про нее напишу!

SMB с тех пор получил значительное развитие, и в 1996 году Microsoft дала ему новое имя — CIFS, Common Internet File System [общая система для доступа к файлам в Интернете]. Затем на замену прежнему немаршрутизируемому протоколу NBF (который всегда предназначался только для малых сетей) пришел TCP/IP, породив слой под названием NBT (NetBIOS поверх TCP). Этот слой принес с собой необходимость наличия метода разрешения имен (преобразования имен NetBIOS в IP-адреса), и Microsoft разработала три решения. Первое — просто разослать широковещательный запрос («У кого из вас имя NetBIOS — VENUS?»). Второе — централизованный сервис WINS (Windows Internet Name Service — сервис имен Интернета Windows), сервис, который регистрирует компьютеры NetBIOS и отвечает на запросы имен NetBIOS. Третьим решением стал локальный файл *lmhosts*, аналогичный */etc/hosts* в Linux.

Более поздние реализации CIFS работают прямо поверх TCP/IP, а старый механизм разрешения имен NetBIOS уступил дорогу DNS.

Протоколы Linux и Windows, понятное дело, друг с другом не работают. Поэтому еще в 1992 году Эндрю Триджелл [Andrew Tridgell] начал работать над открытой реализацией протоколов SMB под названием Samba. Самая важная функция Samba — предоставить клиентам Windows доступ к разделяемым файлам и принтерам. Более свежие версии также могут подключаться к домену Windows, выступать в качестве контроллера домена и интегрироваться с Active Directory.

Следуйте за мной

Если вы захотите следовать за мной, вам понадобится стандартная установка CentOS 6.5. А если захотите проверить взаимодействие на практике, вам также понадобится Windows в качестве клиента — я воспользовался Windows 7.

В установке CentOS 6.5 по умолчанию есть несколько компонентов Samba, в том числе утилиты командной строки, такие как *findsmb*, *smbclient* и *smbprint*, вместе с *winbind* и сопутствующими библиотеками. Подробнее о них позже. Для установки серверных компонентов Samba и документации я скомандовал

```
yum update -y
yum install samba samba-doc
```

В пакете Samba есть два основных демона: *nmbd* и *smbd*. Пакет *samba-doc* включает все map-страницы, а также полные PDF-версии трех книг (см. врезку «Где узнать больше?» вверху). Обратите внимание, что все это относится к Samba 3 (3.6.9, чтобы

быть точным); по умолчанию Samba 4 в CentOS еще нет, и установка мне не удалась, хотя, честно сказать, я не особенно старался.

Если вы, подобно мне, любите поскорее добиваться чего-то работающего, то, установив все компоненты, обратитесь к файлу настройки `/etc/samba/smb.conf`. Большая часть вашего взаимодействия с Samba будет связана с этим файлом. В CentOS он довольно велик (288 строк), но большая его часть — полезные комментарии и закомментированные примеры. Я решил переименовать этот файл и начать с нуля:

```
cd /etc/samba
mv smb.conf smb.conf.original
```

Вот моя минимальная рабочая версия файла `smb.conf`:

```
[global]
    workgroup = WESTWICK
    netbios name = LXF
    security = user
    name resolve order = wins bcast

[docs]
    comment = documentation
    path = /export/docs
    read only = yes
```

Тем, кто помнит старые INI-файлы Windows, синтаксис покажется знакомым. Как вы, видимо, догадались, в разделе `[global]` задаются настройки для сервера в целом. Рабочая группа (`workgroup`) представляет собой именованную группу компьютеров (обычно около десятка), находящихся в одной локальной сети. О параметре `security` мы поговорим через месяц. В параметре `name resolve order` задаются механизмы, применяемые для разрешения имен NetBIOS; в моем случае надо было сделать так, чтобы клиентские программы вроде `smbclient` не пытались использовать DNS для разрешения имен.

В разделе `[docs]` задается общий ресурс; `docs` — не ключевое слово, а имя ресурса. Я сам его придумал. Path, естественно, содержит каталог файловой системы сервера, которому соответствует общий ресурс.

Изменив файл `smb.conf`, стоит запустить программку `testparm`, проверяющую синтаксис файла настройки. С ее помощью удобно

находить опечатки. Например, если у меня в файле будет вот такая опечатка:

```
security = user
to testparm сообщит:
Unknown parameter encountered: "security"
```

Теперь создадим несколько тестовых файлов и поместим их туда, где они должны быть согласно файлу настройки:

```
# mkdir -p /export/docs
# echo «how to bake bread» > /export/docs/bread
# echo «how to scramble eggs» > /export/docs/eggs
```

Запускаем главный двигатель

Наконец, можно запустить демоны `smbd` и `nmbd`. Демон `smbd` выступает в качестве файлового сервера, а `nmbd` предоставляет сервисы просмотра сети и разрешения имен.

```
service smb start
service nmb start
```

Также стоит сделать так, чтобы эти серверы запускались при загрузке системы:

```
# chkconfig smb on
# chkconfig nmb on
```

Прежде чем двигаться дальше, надо создать учетную запись пользователя и пароль, через которые клиенты смогут подключаться к общему ресурсу. Это занятие в мире Linux/Samba/Windows наводит на мысль о банке с червями (я попробую ее приоткрыть — по крайней мере, чуть-чуть — в следующем месяце). Пока просто отметим, что Samba позволяет создавать пользователей (и задавать им пароли), актуальных в мире Windows, программой `smbpasswd`. Так, команда

```
# smbpasswd -a chris
```

создаст учетную запись Samba для пользователя `chris` и задаст его пароль. Учтите, что для этого пользователь `chris` уже должен иметь обычную учетную запись в Linux, например, в `/etc/passwd`.

Проверка из Linux

Мне уже не терпится загрузить Windows и насладиться своим новым ресурсом. Но в Linux есть клиентские утилиты, с помощью которых можно предварительно убедиться, что все работает. Команда `findsmb` выведет список всех компьютеров локальной сети, отвечающих на запросы SMB:

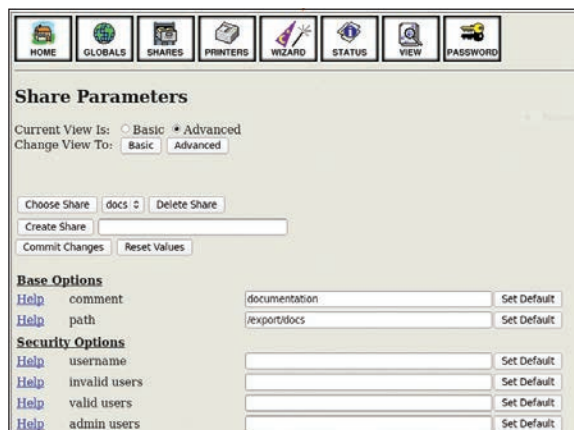
```
# findsmb
*=DMB
+=LMB
IP ADDR      NETBIOS NAME
WORKGROUP/OS/VERSION
-----
192.168.1.80  LXF
+[WESTWICK] [Unix] [Samba 3.6.9-168.el6_5]
```

Затем с помощью `smbclient` можно вывести список доступных ресурсов нашего сервера:

```
$ smbclient -L LXF -U chris
Enter chris's password:
Domain=[WESTWICK] OS=[Unix] Server=[Samba 3.6.9-168.el6_5]
Sharename    Type      Comment
-----
docs         Disk     documentation
IPC$         IPC      IPC Service (Samba 3.6.9-168.el6_5)
Server       Comment
-----
LXF          Samba 3.6.9-168.el6_5
Workgroup    Master
-----
WESTWICK    LXF
```

Таинственный `IPC$ share` — это скрытый ресурс, используемый для межпроцессного взаимодействия.

Окно настройки ресурса SWAT выглядит приятнее редактирования `smb.conf` вручную, но понять, что происходит, в нем не легче.



Почему Samba?

Samba появилась в 1991 году, когда Эндрю Триджелл разбирал протокол SMB с помощью перехватчика пакетов. (Надо отметить, что вся разработка Samba — это результат мучительного разбора протоколов, а не уютных встреч тет-а-тет с ребятами из Microsoft.) Легенда гласит, что название

Samba было найдено в ходе поиска словарных слов, содержащих "SMB", следующей командой:

```
grep -i '^s.*m.*b' /usr/share/dict/words
```

хотя у меня при ее запуске получилось 98 слов, в том числе Стромболи [Stromboli] и «негодяй» [scumbag].

Пока все хорошо. Но настоящей проверкой будет отправить-ся в Windows (в данном случае Windows 7), запустить проводник и убедиться, что в нем есть наш общий ресурс. У меня с разной степенью успеха получалось сделать так, чтобы компьютер внезапно появлялся в дереве сети Проводника, а иногда для этого приходилось явно вводить имя ресурса (в данном случае `\\LXF\docs`) в адресной строке Проводника (см. небольшой экранный снимок Windows на рисунке справа).

О клиенте

Linux также может выступать в качестве клиента, обращаясь к файлам на общем ресурсе, расположенном на компьютере с Windows. Для этого не придется запускать серверы Samba: вам понадобятся только программа `smbmount` и другие клиентские утилиты Samba, которые могут быть уже установлены по умолчанию. Для проверки я разрешил общий доступ к файлам на компьютере с Windows 7 и создал общий ресурс `windowphotos` для своей библиотеки фотографий. Имя NetBIOS этого компьютера — HP250.

В Linux для просмотра общих ресурсов на сервере Windows мы можем воспользоваться командой `smbclient -L`, как в примере выше. Повторяться не буду: все это выглядит более или менее как в прошлый раз. Взамен попробуем подключиться к ресурсу, вывести его содержимое и загрузить файл с него. Если вы когда-либо пользовались клиентом FTP в командной строке, следующий вывод покажется знакомым:

```
# smbclient //HP250/windowphotos -U chris
Enter chris's password:
Domain=[HP250] OS=[Windows 7 Professional 7601 Service Pack 1] Server=[Windows 7 Professional 6.1]
smb: \> dir
2012-08-15 Germany D 0 Sat Oct 26 19:56:37 2013
desktop.ini AHS 504 Mon Oct 28 06:04:23 2013
DSC08116.JPG A 3314307 Wed Oct 23 17:06:05 2013
DSC08119.JPG A 3204112 Wed Oct 23 17:05:53 2013
florida2013 D 0 Wed Jan 1 20:46:02 2014
smb: \> get DSC08116.JPG
getting file \DSC08116.JPG of size 3314307 as DSC08116.JPG
smb: \> quit
```

Конечно, все это довольно неуклюже; гораздо лучше смонтировать ресурс напрямую в файловую систему Linux и обращаться к нему через путь, как и к любому другому файлу:

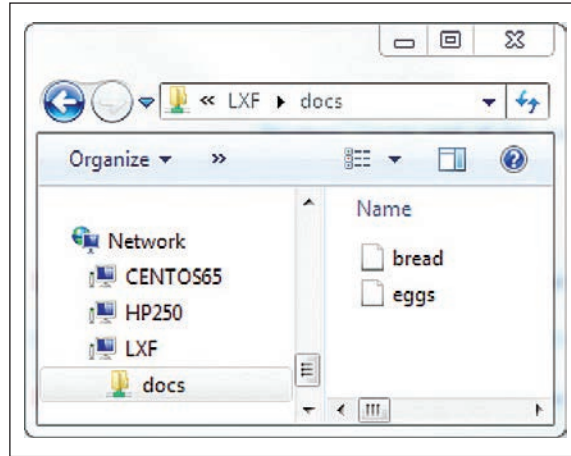
```
# mount -t cifs -o user=chris,password=xxxxxx //HP250/windowphotos /mnt
```

Теперь мы можем выводить список содержимого каталога и обращаться к файлам так, как будто они являются частью локальной файловой системы.

```
# ls -l /mnt
total 191609
drwxr-xr-x. 1 root root 655360 Oct 26 19:56 2012-08-15 Germany
-rwxr-xr-x. 1 root root 504 Oct 28 06:04 desktop.ini
-rwxr-xr-x. 1 root root 3314307 Oct 23 2013 DSC08116.JPG
```

Дополнительный слой

Если сравнить общий доступ к файлам по CIFS и NFS, то в CIFS есть слой именования, который в NFS отсутствует. Смонтировав ресурс NFS `venus:export/chris` (ну, «ресурс» — понятие не из терминологии UNIX, но уж оставим его), мы увидим путь в файловой системе, каким он на самом деле является на сервере. С другой стороны, если смонтировать ресурс CIFS `//LXF/docs`, то я не узнаю, какой путь на сервере ему соответствует.



► Взаимодействие в действии: ресурс Samba в Проводнике Windows.

```
-rwxr-xr-x. 1 root root 3204112 Oct 23 2013 DSC08119.JPG
drwxr-xr-x. 1 root root 262144 Jan 1 20:46 florida2013
```

Монтированию ресурса есть альтернатива — некоторые графические файловые браузеры в Linux имеют возможность подключения к ресурсам Windows и отображения их содержимого.

Команда SWAT

Если не хотите много набирать, воспользуйтесь отличной web-утилитой SWAT (Samba Web Administration Tool). Для этого потребуется еще кое-что настроить:

```
yum install samba-swat
```

При установке также скорее всего придется установить `xinetd`, в качестве зависимости. Это «суперсервер», который запускает SWAT, и нам нужно приказывать ему слушать запросы на подключение. Для этого откройте файл `/etc/xinetd.d/swat` и измените строку

```
disable = yes
на
disable = no
```

Желая обращаться к SWAT из web-браузера на другом компьютере, найдите строку

```
only_from = 127.0.0.1
```

и либо закоментируйте ее, либо измените так, чтобы она отражала диапазон локальных IP-адресов, например:

```
only_from = 192.168.1.0/24
```

Выполнив эти изменения, перезапустите `xinetd`:

```
# service xinetd restart
```

Теперь SWAT должен слушать порт 901, и если вы зайдете на `http://127.0.0.1:901` и войдете от имени `root`, то увидите домашнюю страницу SWAT. Там есть ссылки на map-страницы (см. рис. на стр. XX) и кнопки для доступа к отдельным окнам настройки.

На другом экранном снимке (см. рис. на стр. 64) показан фрагмент окна настройки ресурса. Что мне нравится в SWAT — параметры, имеющие ограниченный набор значений, отображаются в виде выпадающих списков, и невозможно указать некорректное значение. Еще мне нравятся ссылки Help, при нажатии на которые открываются соответствующие разделы map-страницы `smb.conf`. Учтите, что при сохранении файла настройки из SWAT затрутятся все комментарии, включая закоментированные примеры из оригинальной версии файла. Поэтому сначала создайте копию файла.

SWAT — удобная утилита, но графический интерфейс не освобождает вас от необходимости представлять, что означают все эти параметры, которые вы изменяете. И это нелегко.

Через месяц мы займемся созданием учетных записей пользователей в Samba и посмотрим, как работает аутентификация. Возможно, вернуть всех червей в банку не получится. Увидимся! **LXF**

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



А если у вас Android, подпишитесь
на Linux Format через Zinio!

 zinio™
Доступно в Google Play



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Запрещение сайтов — зло

Заведу будильник, встану спозаранку, сяду за компьютер, подключусь к Сети, нацеплю я совесть мехом наизнанку и пойду учить людей, как себя вести!

leo.aha.ru

Про закон о блокировке злобных сайтов Роскомнадзором, наверно, не слышал разве что тот, кто не пользуется Интернетом. А теперь — жёсткая сказка о том, как он работает в реальности: жила-была Вики по имени sportswiki.ru, и были у неё пользователи, её наполнявшие. Варились эти пользователи в своём соку, никого особо учить жить не пробовали, а просто размещали на своих страничках значимую с их точки зрения информацию по различным аспектам спорта. Разной степени вредности медикаменты тоже не были оставлены в стороне, так как их действие на организм очевидно влияет в том числе и на олимпийские успехи. И попался пользователям под прицел один склизкий производитель БАДов. БАДы — это такие штуки, которым разные деятели любят приписывать волшебные свойства и страшно обижаются, если им не верят, а самое главное — не дают им денег. Написали пользователи статью про производителя и подробно разобрали его. Что сделал этот производитель? Правильно: написал кляузу, и sportswiki.ru занесли в реестр.

Итак, будьте готовы, что отныне, если вы боретесь с мошенниками просветительскими методами, эти мошенники вашу площадку просто прикроют, благо закон это позволяет и разбираться никто особо не будет. E.M.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Работать в терминале 68

Нейл Ботвик представляет начинающим линуксоидам ряд важных программ и инструментов, доступных из терминала.



Мониторить систему 70

Приглядывайте за здоровьем своей системы, пользуясь *GkrellM*. **Ричард Смедли** покажет, как установить и настроить этот удобный инструмент мониторинга.



Шифровать диск 74

В заботе о сохранности важных данных, **Нейл Ботвик** советует зашифровать жесткий диск, спрятав его от любопытных глаз с помощью *dm-crypt*.



Строить облака 76

Джонни Браун автоматизирует сборку *OpenStack*, а заодно исследует свойства его свежей версии, *IceHouse*.



Вещать музыку 80

Превратите старый ПК — или даже Raspberry Pi — в сервер потокового вещания, при содействии **Маянка Шармы** и *Mopidy*.

Код в учебниках

Строки исходного кода помещаются в цветные плашки. Если строка кода не умещается в колонке, ее остаток переходит на следующую строку, в той же плашке:

```
procedure TfrmTextEditor.  
  mniWordWrapClick  
A плашки разделены зазорами:  
begin  
  mniWordWrap.Checked := false
```

АКАДЕМИЯ КОДИНГА



И новичкам, и гуру!
Всегда полезно будет познать нечто доселе неведомое

Python

Изучите объектно-ориентированный Python и создайте пушку в *Minecraft* с **Джонни Бидвеллом** с. 84

Octave

Для **Терри Дьюэлла** и *Octave 3.8* не существует слишком больших или слишком сложных численных задач с. 88

Erlang

Собравшись с духом, **Андрей Ушаков** возобновляет тестирование программ — правильными методами с. 92

Терминал: Про основы

Из сотен команд терминала, **Нейл Ботвик** описывает самые полезные.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ничем не расскажет, где находится центральный сервер.

В этой серии мы рассматривали разные shell-команды, но каждый раз в контексте выполнения конкретной задачи. Пришла пора поговорить о некоторых командах общего назначения. В терминале их тысячи, от простых до сложных, но для начала достаточно нескольких. Здесь мы рассмотрим ключевые рабочие команды, дав каждой краткое описание. И, как всегда, более подробную информацию об их применении можно найти на map-страницах. Вывод многих из них способен превысить размеры дисплея терминала, поэтому стоит просматривать его через *less*, как мы описывали в прошлом выпуске (Map-страницы: Помощь близка, стр. 64, **LXF185**).

Центральное место в работе с терминалом занимают операции с файлами — создание, удаление, просмотр и прочее. Вот основные команды для этого.

» **ls** Отображает содержимое текущего или заданного каталога.

» **ls -l** Как и **ls**, но дает больше информации о каждом элементе. Чтобы узнать размеры файлов в читаемом формате, добавьте **-h**:
`ls -lh MyPhotos`

» **rm** Удаляет файл. Используйте параметр **-i** для подтверждения перед каждым удалением или **-f** для быстрого удаления. С ключом **-r** удаляются каталоги и их содержимое.

» **rmdir** Удаляет каталог (который должен быть пустым).

» **df** Показывает объём свободного дискового пространства на всех файловых системах или только тех, что заданы в командной строке.

» **du** Показывает объём дискового пространства, занимаемого отдельными файлами или каталогами.

```
df -h /home
```

```
du -sh /home/user/*
```

» **file** Определяет тип файла. В отличие от Windows, где это делается по расширению, данная команда действительно исследует содержимое файла.

» **find** Поиск файлов в текущем или указанном каталоге по определенным критериям. Например, вы можете найти все электронные таблицы *LibreOffice* с помощью

```
find Documents -name '*.ods'
```

» **locate** Тоже поиск, но с помощью гораздо более быстрой системы. База данных *locate* автоматически обновляется каждый день с помощью *updatedb*, и затем *locate* ищет по ней. Быстро, но без учёта самых недавних изменений.

Обработка текста

Текстовые файлы окружают нас повсюду, от электронных писем до файлов настройки, и существует множество команд для работы с ними. Отредактировать текстовый файл можно несколькими способами; самые распространённые — *Emacs* и *Vi*. Но если вам нужно всего лишь немного подправить файл настройки, оба будут перебором — лучше попробуйте *nano*:

```
nano -w somefile.txt
```

Параметр **-w** отключает перенос слов, который при редактировании файлов настройки вам, естественно, не нужен. Основные команды показаны внизу, в строке состояния — например, жмите **Ctrl+X**, чтобы сохранить файл и выйти.

Предполагается, что вы знаете, какой именно файл вам нужен; ну, а если вы знаете, что ищете, но забыли имя файла? Тогда воспользуйтесь *grep*. Это поиск текстовых файлов по строке или регулярному выражению.

```
grep sometext *.txt
```

переберет все TXT-файлы в текущем каталоге и покажет строки, содержащие искомый текст в каждом из них, а также имя соответствующего файла.

Вы можете даже искать по всем каталогам, используя **-r** (или **--recursive**):

```
grep -r -l sometext somedir
```

Как получить помощь

Командная строка поначалу многих отпугивает, но есть куча справочного материала, нужно только знать, где искать. Большинство команд имеют параметр **--help**, поясняющий все прочие параметры. Основным источником информации являются страницы *man* и *info*. Чтобы узнать всё о параметрах программы и её функциях, запустите

```
man programme
```

Map-страницы включают пронумерованные разделы. Для пользователя, самые важные — это

» **1** Пользовательские команды

» **5** Форматы файлов и условные обозначения

» **8** Средства администрирования системы

Если вы не укажете номер, *man* выберет первый, чего обычно достаточно. Но *man*-страницы посвящены не только программам, но и файлам настройки. Например, для управления паролями используется команда **passwd**, а информация хранится в файле **/etc/passwd**, поэтому вы можете использовать

```
man passwd
```

```
man 1 passwd
```

```
man 5 passwd
```

Первые две расскажут вам о **passwd**, а третья объяснит содержание и формат файла **/etc/passwd**.

Man отображает всю информацию в формате одной страницы; *info* представляют собой гипертекст страниц, связанных ссылками, в едином файле. Как правило, они содержат более детальную информацию, но с ними непросто разобраться — попробуйте **info info**, чтобы узнать, как ими пользоваться. Зачастую проще прибегнуть к поисковой системе и найти интернет-версию *info*-страниц, где то же самое представлено в более привычном формате HTML.

Команды, параметры и аргументы

Вам будут часто встречаться упоминания об аргументах и параметрах программ, но что это на самом деле? Параметры и аргументы определяют конкретные действия программы. Проще говоря, аргументы говорят команде, что нужно сделать, а параметры — как именно, хотя граница между этими понятиями немного размыта. Возьмём для примера команду **ls**: она показывает содержимое каталога. Без уточнения параметров и аргументов, это будет вывод текущего каталога в стандартном формате:

```
ls
Desktop Downloads Music Public Videos
Documents examples.desktop Pictures Templates
```

Если вы хотите просмотреть другой каталог, укажите его как аргумент:

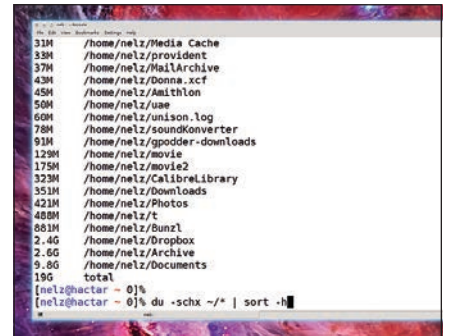
```
ls Pictures
или
ls Desktop Downloads
```

Аргументы представляют собой просто имена того, что вы хотите отобразить, а параметры преобразуются дефисом, чтобы они распознавались. Согласно стандартам GNU, поддерживаемым и многими другими, параметры могут иметь короткую

и длинную форму. Короткий параметр содержит дефис и одну букву: например, в **ls -l** указывается представить полный вывод, с детализацией по каждому файлу. Длинные параметры — это двойной дефис плюс слово: так, **ls --reverse**, как вывевает из названия, представит вывод в обратном порядке. **ls -r** делает то же самое, но это менее наглядно. Многие параметры, как и этот, существуют в длинном и в коротком варианте, но поскольку букв в алфавите всего 26, менее популярные параметры доступны только в длинном. Короткие параметры проще набрать, а длинные понятнее. Сравните

```
ls -l --reverse --time
с
ls -l -r -t
или даже
ls -lrt
```

Все три дают подробный вывод в обратном порядке по времени/дате. Обратите внимание на то, как несколько коротких параметров совмещаются под одним дефисом. И раз уж мы говорим об **ls**, уместно вспомнить о так называемых «скрытых» файлах. В Linux любые файлы или директории,

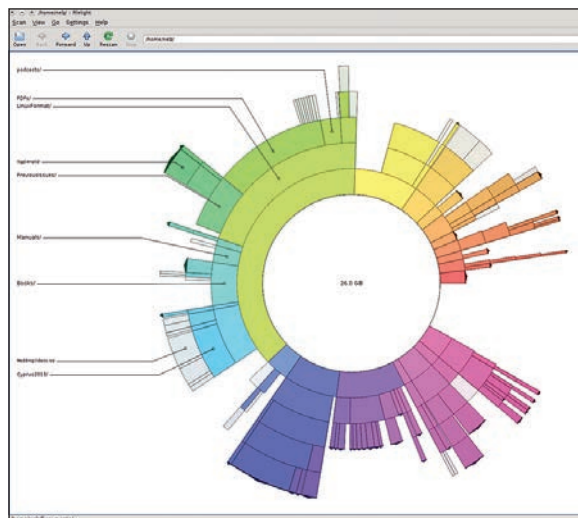


Обработав вывод **du** с помощью **sort** и добавив аргументы к обеим командам, мы увидим, какие каталоги занимают больше всего места.

имена которых начинаются с точки, считаются скрытыми и по умолчанию не отображаются в выводе **ls** в большинстве файловых менеджеров. Как правило, это файлы настройки, только захламляющие ваш дисплей — но если вы хотите их просмотреть, просто добавьте к **ls** параметр **-A**.

Будьте осторожны с поиском по большим деревьям каталогов: он бывает медленным и возвращает странные результаты от любых нетекстовых файлов. Параметр **-l** велит **grep** пропускать такие двоичные файлы.

Текст также является предпочтительным способом передачи данных между многими программами через каналы [pipes], которые мы рассматривали ранее. Иногда требуется сделать это напрямую из одной программы — в другую, но иногда их необходимо предварительно изменить. Вы можете перенести текст в файл, отредактировать его, а затем отправить следующей программе, или же передать его через канал, изменив на ходу. **Nano** редактирует файлы в интерактивном режиме, **grep** автоматически их находит, так что нам нужна только программа для автоматического редактирования; она называется **sed** (Stream Editor).



» **Filelight** в KDE (на рис.) и **GDU** в Gnome — графические альтернативы команды **du**, визуально представляющие занятость диска.

Sed захватывает текстовый поток из файла или из канала и выполняет необходимые преобразования. Чаще всего это удаление или замена. Обычно **sed** направляет вывод в **stdout**, но это можно изменить с помощью параметра **-i**:

```
sed -i 's/oldtext/newtext/g somefile.txt
sed -i '/oldtext/d' somefile.txt
```

Второй пример удаляет все строки, содержащие "oldtext". Ещё одна полезная программа — **awk**; она пригодится для печати отдельных элементов из текстового файла или потока.

```
awk '{print $1}' somefile.txt
cat *.txt | awk '/^Hello/ {print $2}'
```

Первый пример печатает первое слово из каждой строки файла. Второй берет содержимое всех файлов с расширением **.txt**, фильтрует строки, начинающиеся с "Hello" (строка между слэшами — это шаблон), а затем печатает второе слово из каждого совпадения.

Сеть

Думая о сетевом программном обеспечении, мы обычно имеем в виду большие графические приложения, вроде **Chromium** и **Thunderbird**, но существует и множество программ командной строки для создания, тестирования и использования вашей сети или интернет-подключения.

» **ping** Псылает небольшой пакет на удаленный сервер и фиксирует время ожидания, что позволяет проверить доступ к сайту и работу сети.

```
ping -c 5 www.google.com
```

» **wget** Загружает файлы. Единственный аргумент, который нужно указать — это URL. Хотя у команды огромный набор возможностей, вам они, как правило, не понадобятся.

» **hostname** Показывает имя компьютера, или его IP, если указать параметр **-i**.

» **lynx** Текстовый web-браузер. Пусть он и не столь нагляден, как **Chromium** или **Firefox**, но его стоит иметь в виду, на случай, если у вас вдруг возникнут проблемы с графикой. **LS23**

» Месяц назад **LXF185** Все, что вы боялись спросить про Tar.

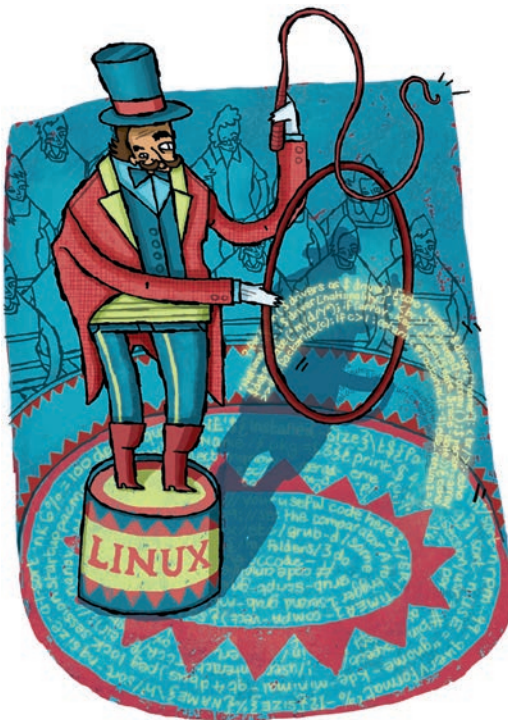
GkrellM: ПК в мониторе

Арсенал мониторов *GKrellM* на вашем рабочем столе сообщит вам обо всём, что происходит в вашей системе, сулит **Ричард Смедли**.



Наш эксперт

Ричард Смедли
Ричард занимается мониторингом систем более 30 лет, с тех пор как забыл термометр на раскалённом Sinclair ZX81.



G *KrellM* — прекрасный системный монитор, простой в настройке и радующий глаз, причем неприязнительный в плане ресурсов. Неудивительно, что Маяк Шарма определил его ближайшим соперником *Conky* по отслеживанию работы ПК, в нашем Сравнении средств мониторинга системы два номера назад (**LXF184**); более того, *GKrellM* — единственное приложение, протестированное также для мониторинга удалённых машин.

Об этом свойстве следовало бы трубить во все концы, а не скромно замечать где-то в конце tap-страницы: «Возможен сбор данных с сервера *gkrellmd*, установленного на удалённой машине». Также там упоминается использование ключа **-s** для сбора данных «с сервера *gkrellmd*». Мы расскажем вам подробнее об этом, а также о работе с *GKrellM* на локальной машине.

Это не мониторинг в промышленных масштабах — и слава Богу; оставим нудную и изнурительную настройку *Icinga* или *Zabbix* тем, кому это реально нужно, для огромных сетей машин перво-степенной важности. *GKrellM* предлагает хороший комплект параметров по умолчанию, с которыми легко разобраться; ряд неплохих плагинов для расширения функционала; и простой удалённый мониторинг с *gkrellmd*. Мы расскажем вам обо всех трёх эпостасях *GKrellM*.

Как отмечалось в Сравнении, разрабатывается *GKrellM* не слишком активно — но отчасти это обусловлено стабильностью базового функционала проекта. А расширить его можно плагинами. Ещё один плюс замедленной разработки в том, что последняя версия легко доступна в диспетчере пакетов вашего дистрибутива.

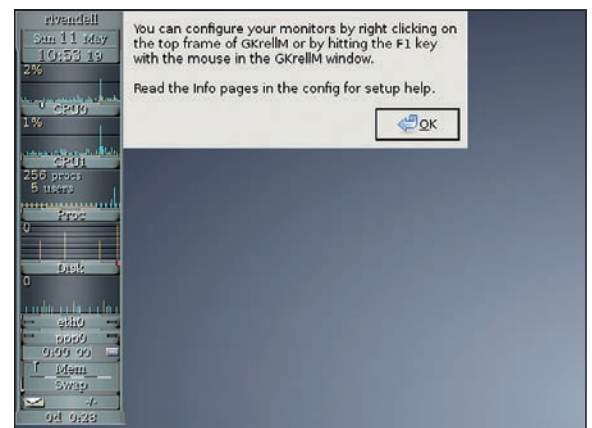
Если же вам понадобится собрать его из исходников, последний релиз находится на <http://billw2.github.io/gkrellm/gkrellm-2.3.5.tar.bz2>.

Число зависимостей минимально — *gtk 2.0*, *gdk 2.0*, *glib 2.0* — а для серверной версии только *glib 2.0* или *glib 1.2*. Желая следить за ходом разработки, клонируйте репозиторий *git* на [git://git.srctbx.net/gkrellm](http://git.srctbx.net/gkrellm).

После установки и запуска *GKrellM* должна появиться директория `~/gkrellm2/` — где будет файл настройки и куда будут сохраняться плагины и темы. В `/usr/share/doc/` также попадёт кое-какая документация по многочисленным компонентам *GKrellM*. Текстовые файлы, которые вы там найдёте, будут по большей части излишни — вся настройка *GKrellM* ведётся через GUI.

С чего начать

Во-первых, запустим приложение и посмотрим на него — ваш диспетчер пакетов должен был добавить соответствующую строку в меню дистрибутива, и для загрузки с настройками по умолчанию этого достаточно. Как видно на экранных снимках, перед вами аккуратный столбик мониторов, с довольно симпатичной темой по умолчанию, сообщающий вам всё, от нагрузки процессора до уровня заряда аккумулятора.



» Именно эту тему по умолчанию вы увидите на *GKrellM* после первой загрузки, а также предложение настроить программу и почитать полезную документацию.

Щелкнув правой кнопкой по имени системы вверху дисплея, вы откроете меню настройки с опциями, определяющими поведение и параметры *GkrellM*, а также любого компонента, счётчика или диаграммы — в плане свойств отображения или полного отказа от него. Некоторые опции Формата строки дают возможность очень гибкого выбора параметров отображения, а вкладки справки помогут взять от *GkrellM* всё самое лучшее. Правым щелчком по отдельному счётчику или диаграмме приведет вас в соответствующий раздел меню настроек.

Если предполагается, что этот ящик с секретами будет мирно сидеть в углу вашего монитора, вероятно, вы захотите, чтобы он вписывался в оформление вашего рабочего стола.

По умолчанию, никаких изысков в менеджере окон нет, поскольку у *GkrellM* свой стиль, но если вы хотите единообразия с другими установленными приложениями, начните с ключа **-wm**. Добавьте ключ геометрии **-g** — и сможете задать расположение *GkrellM* на экране (хотя любой оконный менеджер позволит перетаскивать его по Alt+щелчок, как любое другое оконное приложение); чтобы поместить его около верхнего правого угла, при экране 1280×1024 наберите

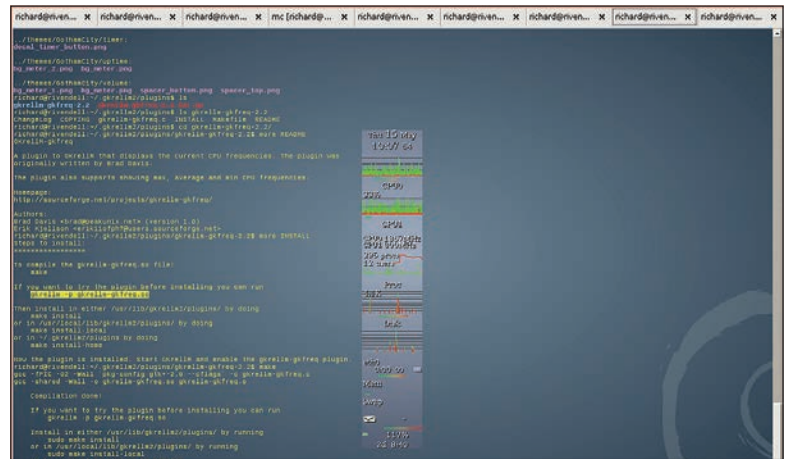
```
gkrellm -wm -g +1170+30
```

Как вам будет угодно

Но каким бы ни было понятие о красоте в вашем оконном менеджере, с темами *GkrellM* вы можете настроить каждый пиксель дисплея. Загляните на www.muhi.net/gkrellm — там их полно. Найдите парочку, которые вы хотите загрузить прямо сейчас, а затем распакуйте в `~/gkrellm2/themes`. На многопользовательской машине их можно установить в `/usr/local/share/gkrellm2/themes` или даже `/usr/share/gkrellm2/themes`.

Для совместимости с темами *GTK* можно также установить их в `~/themes/THEME_NAME/gkrellm2` или `/usr/share/themes/THEME_NAME/gkrellm2`. Как однажды сказал Эндрю Таненбаум [Andrew Tanenbaum]: «Стандарты хороши тем, что их много, и есть из чего выбирать».

Где бы ни находилась ваша тема (*GkrellM*, запущенный с ключом **-t**, захватит её в любом месте, куда вы укажете), меню настройки позволит вам выбрать особо понравившуюся, а также размер шрифта, подходящий для вашего экрана. Вы сможете



► Рис. 1. Чтобы установить плагины, нужно просто положить их куда следует.

переключаться между установленными темами, просто наводя мышью на *GkrellM* и нажимая PageUp/PageDn.

Приятно, что всё происходит без задержек, как и с большинством других операций *GkrellM*, даже на наших древних тестовых ПК. Объединив всё в один лёгковесный процесс, автор *GkrellM*, Билл Уилсон [Bill Wilson], дал нам одно из самых быстрых приложений, способных менять свой облик.

Низкое потребление системных ресурсов, в сочетании с простотой смены оформлений, объясняет популярность *GkrellM* среди тех, кто заботится о состоянии своего рабочего стола больше всего — опытных пользователей с глубоко настраиваемыми, минималистичными оконными менеджерами.

Ведь за пределами Gnome, KDE и Unity — целый мир: от клонов NeXTStep до наследников Plan9 и мозаичных [tiling] дисплеев, управляемых без помощи мыши. Для пользователей альтернативных оконных менеджеров в *GkrellM* FAQ перечислены способы его добавления в запуск самых разных систем, какие вы, возможно, сроду не пробовали.

Пользователи таких оконных менеджеров, как правило, заботятся о том, как отображается каждый пиксель — и часто предпочитают более старое оборудование, где каждый цикл процессора и КБ памяти идёт в счёт. Всё это не касается вашего 4-ядерного



Темы «сделай сам»

Большое количество доступных тем говорит о том, что их добавление несложно и хорошо документировано. Руководство можно найти на <http://billw2.github.io/gkrellm/Themes.html>.

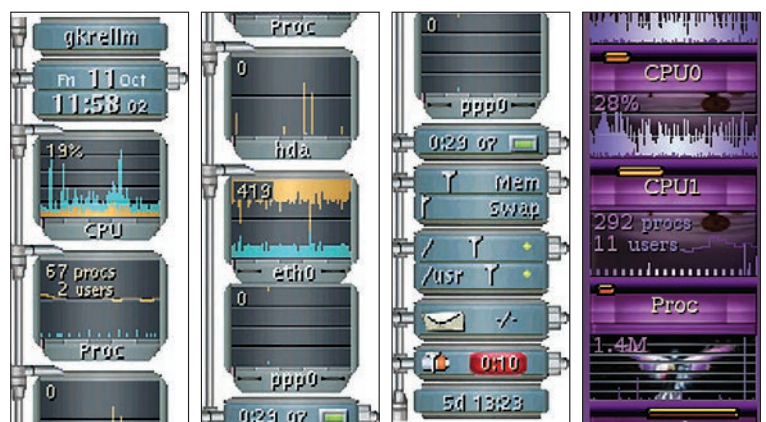
По существу, компоненты *GkrellM* — это либо счётчики с простой панелью, либо более красочные графики, отображаемые отдельно. Добавьте управление рамкой, фоновые рисунки и каждым компонентом графика или счётчика, и вот вам полный контроль.

Вы даже можете визуально сгруппировать мониторы с помощью верхних и нижних разделителей:

-----	spacer_top
	bg_meter

	bg_meter
-----	spacer_bottom

Левый ряд — без разделителей, правый — с разделителями. Хотя броское, хоть ненавязчивое оформление — решать вам.



► Аппаратная тема (первые три фото) ставит системные мониторы на (виртуальные) аппаратные стойки. В теме GothamCity оранжевые счетчики CPU, пульсирующие на фиолетовом фоне, действуют гипнотически.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Ultra-HD ноутбука с 8 ГБ ОЗУ? Допустим; но вам всё равно пригодится гибкость *GkrellM*.

В некоторых комбинациях цвета и размера диаграмм, движение счётчиков процессора странно заворачивает — если вы вдруг обнаружите, что уже давно следите за ними не отвлекаясь, возможно, лучше перейти на менее отвлекающую тему. Или запустить *GkrellM* в другом окне рабочего стола и просто периодически проверять важные для себя показатели. Именно поэтому большинство доступных тем черные или серые: это ненавязчиво.

Плагины

Конечно, если среди доступных тем не нашлось подходящей, вы всегда можете создать собственную — это не так сложно, благодаря логичной структуре и хорошей документации (см. Темы «сделай сам», предыдущая страница). Но темы — это только внешняя сторона, а для многих пользователей гибкость и эффективность программы определяются, прежде всего, широкой функциональностью.

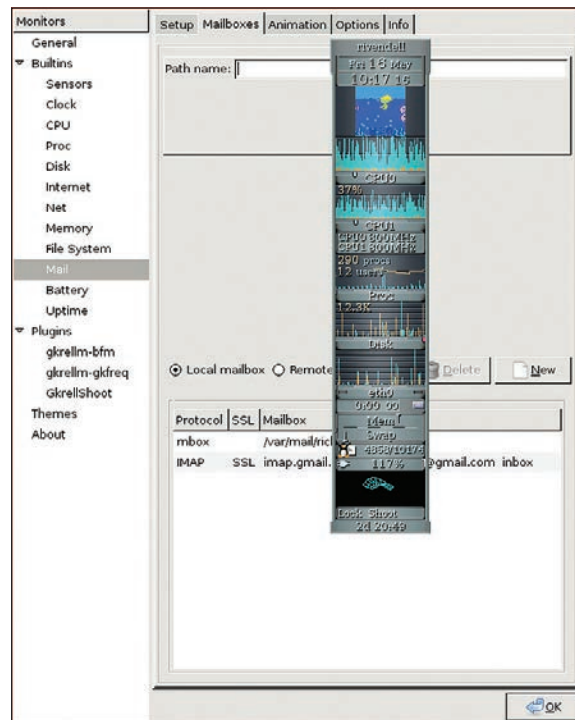
К сожалению, при работе с последней версией *GkrellM* на новых процессорах и дистрибутивах выясняется, что некоторые плагины, представленные на сайте *GkrellM*, прихрамывают. Некоторые из них уже не поддерживаются — хотя, поскольку это открытый код, если они вам нужны, то обновить их можно. На самом деле, обновление небольшой программы для работы с более поздними библиотеками — это отличный опыт. Более скорые плоды, пожалуй, принесет поиск на GitHub, показывающий плагины, разработанные относительно недавно.

Один из таких полезных модулей — *GkrellM-gkfreq* Брэда Дэвиса [Brad Davis], который отображает частоту процессора; в наши дни энергосберегающих машин с ним очень удобно наблюдать, как меняется скорость при разных нагрузках, а также при питании от сети и от аккумулятора. Мы скачали плагин с сайта Sourceforge, распаковали его и запустили скрипт *make*. Посмотреть, как работает плагин, можно и без установки — просто запустите его с ключом `-p`, вот так:

```
GkrellM -p GkrellM-gkfreq.so
```

но если эффективность и стабильность плагина вас устраивают, `sudo make install` позволит вам не повторять этого снова.

Как видно на рис. 1 с предыдущей страницы, активированные плагины имеют собственное окно настройки в меню конфигурации, для некоторых — например, для *GkrellShoot*, плагина для создания снимков экрана — есть также много полезной документации во вкладках графического интерфейса. Многие плагины



➤ Укажите *GkrellM* локальный или удалённый почтовый ящик, чтобы посмотреть, есть ли новые сообщения — на самом деле, до #inboxzero вам далеко.

выполняют функции, относящиеся не только к системному мониторингу, такие как интерфейсы к медиа-плеерам, содержание буфера обмена или считывание местной погоды.

Если вы в ладах с C, взгляните на ясный и хорошо документированный код *GkrellM-gkfreq* — он гораздо больше расскажет вам о том, что нужно знать, чтобы написать плагин *GkrellM*; однако, как и с темами, для этого есть и подробное руководство — <http://bit.ly/LXFgkrellm>. Если бы в мире свободного и открытого ПО эта документация была оценена по достоинству, *GkrellM* бы несомненно прославился.

По проводам

Теперь, когда *GkrellM* настроен именно под ваш ПК, пришло время для расширения по сети. *GkrellM* может работать в «режиме клиента», слушая демон-процесс *GkrellM* на другой удалённой машине — с помощью пакета *gkrellmd*, который собирает те же данные и отправляет их по сети. Как и основные пакеты *GkrellM*, *gkrellmd* будет в вашем менеджере пакетов.

Установив *gkrellmd*, откройте `/etc/gkrellmd.conf` в своём любимом текстовом редакторе и найдите этот блок:

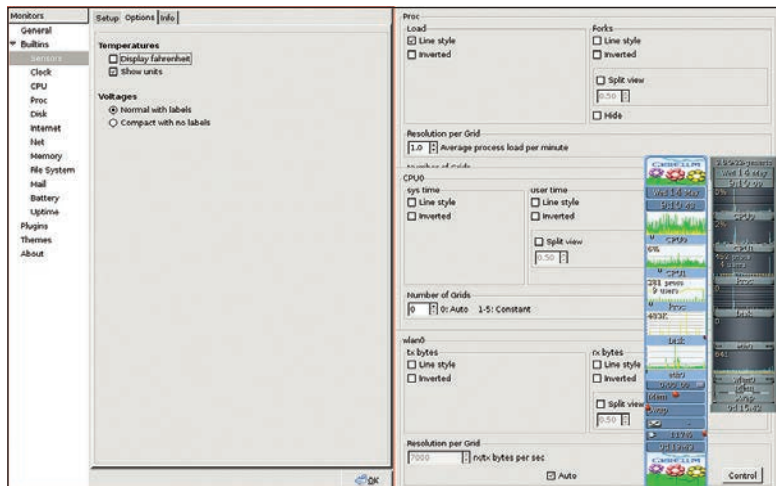
```
#allow-host localhost
#allow-host 127.0.0.1
#allow-host ::ffff:127.0.0.1
#allow-host 192.168.0.*
```

Возможно, вы обнаружите, что в настройках вашего дистрибутива некоторые из этих строк не закомментированы знаком # — в случае Ubuntu и Debian, это три строки localhost; если вы работаете с исходниками, то закомментированы все. Раскомментируйте записи. Если ваша локальная сеть расположена не в подсети 192.168.0.0/24, настройте её правильно. Вы также можете установить другой номер порта, отличный от того, что стоит по умолчанию — 19150 — в особенности, если у вас более одного *gkrellmd* в сети.



Щелкнув на удалённом почтовом ящике и введя детали IMAP в меню конфигурации, нажмите не кнопку ОК внизу страницы, а кнопку Создать, чтобы добавить новую учётную запись. Иначе не видать вам прыжков пингвинчика, сообщающего о полученном письме.

➤ Для мониторинга удалённого ПК (справа) или просто для красоты (слева), *GkrellM* всё равно вам пригодится.



➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Рецепт сети

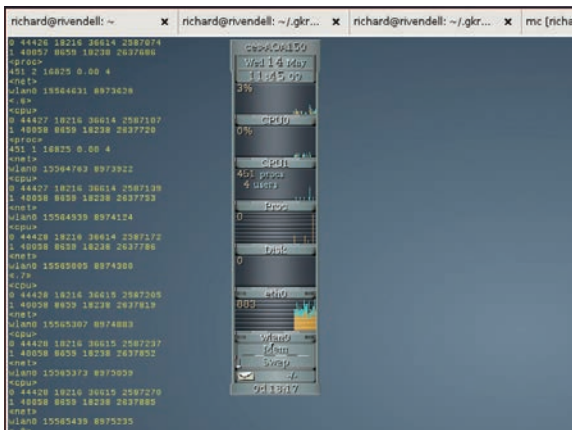
Если вы убеждены в эффективности тотальной автоматизации, и даже используете *Chef* на в домашней сети, то на <https://github.com/locomote-cookbooks/gkrellmd> есть рецепты — так уж называются эти заклинания для обработки программ — для *GkrellMd Chef*.

Там же вы найдёте всё о настройке порта и разрешённых хостах для вашей сети.

Chef использует предметно-ориентированный язык (DSL) на основе Ruby для написания «рецептов» конфигурации системы для установки и настройки ПО на ваших серверах (а поскольку в мире Linux между ними разницы нет, также и на вашем настольном ПК). Для большинства домашних сетей это перебор, но если вы много занимаетесь отладкой, тестированием, разработкой, и вам часто

приходится переустанавливать одни и те же пакеты, *Chef* — или другой открытый инструмент управления конфигурацией — может в перспективе сэкономить вам кучу времени.

Сборник рецептов *GkrellM* тестирован на CentOS и RHEL 6.3, но его автор, LocomoteChris, считает, что он «должен работать на Ubuntu и Debian». Детали — на <http://community.opscode.com/cookbooks/gkrellmd>.



► *GkrellM* позаботится о красивом представлении данных, или можно захватить текстовый поток из других средств мониторинга, написав подходящий скрипт.

Вы можете поместить данные о разрешенном хосте в аргументы командной строки, чтобы проверить различные сценарии, например, так:

```
nohup gkrellmd -u 2 -P 27770 -m 1 -a 127.0.0.1 -a 192.168.0.7 &
```

Это разрешит соединение локального хоста и 192.168.0.7 через порт 27770. Параметр **-u** определяет, сколько раз в секунду эти данные будут обновляться, а **-m** — максимальное число одновременных клиентов. Его стоит указать — вы ведь не хотите, чтобы в вашем контроле использования системных ресурсов зияла брешь; в `/etc/gkrellmd.conf` это директива `max-clients`.

Ваш установщик должен правильно разместить скрипт запуска, чтобы работа демона восстанавливалась после перезагрузки. Команду запуска можно изменить там, но куда разумнее настроить ее через `/etc/gkrellmd.conf`.

Вы можете поставить *GkrellM* на машины с Windows, если таковые имеются в вашей сети, чтобы приглядывать и за ними. Обратите внимание, что данные об обмене, использовании памяти и другие атрибуты могут прочитываться иначе, а некоторые настройки будут полностью игнорированы — к примеру, не будет `~/mail mbox` для почтовых уведомлений.

В `/etc/gkrellmd.conf` перечислены все исключения Windows по каждой директиве.

Теперь, где бы вы ни были в вашей сети, вы можете следить за всеми компьютерами, на которых установлен *gkrellmd*. Простейший способ подключить *GkrellM* к нужной машине —

```
gkrellm -s 192.168.0.6 -P 27770
```

Какой бы порт вы ни использовали, при наличии сурового брандмауэра вам нужно будет открыть его для TCP-трафика. Настроив локальный и удаленный мониторинг, можете поэкспериментировать с расположением *GkrellM* на экране, воспользовавшись

```
gkrellm -s 192.168.0.6 -P 27770 -g +1060+30 &
```

```
gkrellm -g +1170+30 &
```

Вы можете сохранить эти команды в файле `~/xinitrc` — или куда укажет ваш оконный менеджер — чтобы *GkrellM* запускался автоматически при входе в систему.

Текстовые данные

GkrellM получает по сети данные в открытом текстовом формате — это основное средство Linux-коммуникации. В чем вы можете убедиться, запустив клиент в режиме отладки:

```
# gkrellm -s 192.168.0.6 -P 27770 -d 0x1000
```

Другие параметры отладки можно увидеть, запустив

```
gkrellm --help
```

Взглянуть на вывод данных с сервера может быть полезно не только для отладки, но и для использования данных *gkrellmd* другими средствами мониторинга.

Безопасное соединение

Если свободное перемещение этих данных по вашей сети и через Wi-Fi в незашифрованном виде вас беспокоит, пора освоить туннелирование! Одна из главных особенностей SSH в способности передавать часть трафика через закрытый-от-прослушивания туннель с переадресацией портов.

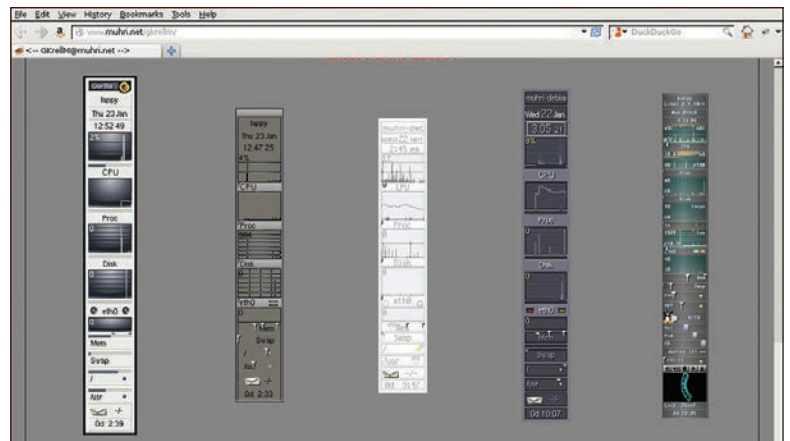
Просто выберите неиспользуемый порт — здесь это 19143; а 19150 является портом *GkrellM* по умолчанию, если вы не поменяете конфигурацию.

```
ssh -f -N -L 19143:localhost:19150 remote-box.com
```

```
gkrellm -s localhost -P 19143 &
```

Если такое решение вам не подходит, есть также набор классов PHP5 для работы с *gkrellmd*: <http://php-gkrellm.sourceforge.net/> — хотя он не особо активно разрабатывается, так что, возможно, придется над ним поколдовать, чтобы настроить свой web-интерфейс для работы с данными *GkrellMd*.

При его неприязнательности к системным ресурсам и невероятно тонкой настройке при простоте установки, мы уверены, что вы полюбите *GkrellM*, так же, как и мы. LXF



► Можно скачать множество скинов для оформления каждого пикселя окна *GkrellM*.



Если вы используете SSH-туннелирование впервые, разобраться в этой теме вам поможет руководство с <http://bit.ly/LXFssh>. Это также отличный способ защиты почты и другого трафика при использовании незащищенного Wi-Fi.

Dm-crypt: Диск под шифром

Хотите спрятать свои данные от посторонних глаз? **Нейл Ботвик** обучает приемам шифрования жёсткого диска.



Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни о чем не расскажет, где находится центральный сервер.



За последний год тема безопасности интернет-данных всплывала часто: сначала в связи с откровениями Сноудена, потом — с уязвимостью в OpenSSL (Heartbleed bug). Незадолго до последней была обнаружена и менее заметная ошибка в GnuTLS. В результате мы стали уделять больше внимания безопасности удалённых данных; но как насчёт тех, что хранятся у нас на диске? Ранее мы рассматривали *TrueCrypt*, которая отлично подходит для шифрования съемных носителей, в особенности потому, что доступна и для Windows, и для Mac, но при необходимости вы можете зашифровать и всю свою домашнюю папку, или даже весь диск. Речь не только о том, как защититься от взломщиков-хакеров: что если ноутбук с вашей личной, деловой или другой конфиденциальной информацией будет потерян или украден?

Ядро Linux поддерживает два популярных типа шифрования, *dm-crypt* и *ecryptfs*. Второй — это зашифрованная файловая система, являющаяся «надстройкой» над стандартной. Смонтировав «нижнюю» файловую систему, вы увидите все файлы, но их содержание, как и имена, будут представлены в зашифрованном виде. Работает она на уровне каталога, так что прочие каталоги на той же файловой системе можно оставить в открытом виде или зашифровать отдельно. Этот метод, среди прочих, используется в Ubuntu, для шифрования домашних папок пользователей.

Другой метод, который нам хотелось бы рассмотреть подробнее, называется *dm-crypt*, и он работает на более низком уровне, обеспечивая шифрование блочного устройства, на котором расположена файловая система. Недавний тест, проведённый Phoronix, показал более высокую производительность в случае, когда диск был полностью зашифрован *dm-crypt*, чем при использовании *ecryptfs* на домашних каталогах.

Стек блочных устройств

Прежде чем рассматривать шифрование, важно осознать, как работают блочные устройства. Они являются системными интерфейсами для хранения данных, например, */dev/sda1*. Внутри блочного устройства находится аппаратный драйвер, например, SATA, и собственно оборудование. Затем операционная система взаимодействует с блочным устройством для создания на нем файловой системы.

Блочное устройство обычно рассматриваются именно в этом качестве, хотя у них есть и другие функции. В частности, подобное устройство может быть интерфейсом для ряда других блочных устройств — они могут составлять стек. И вы такое уже делали: у вас есть файловая система на */dev/sda1* (раздел диска), а это блочное устройство ссылается на */dev/sda* (весь диск).

Такие технологии, как RAID и LVM (Logical Volume Management), также представляют собой стеки блочных устройств. У вас может

```
luksKillSlot <device> <key slot> - wipes key with number <key slot> from LUKS device
luksUUID <device> - print UUID of LUKS device
isLuks <device> - tests <device> for LUKS partition header
luksDump <device> - dump LUKS partition information
tcryptDump <device> - dump TCRYPT device information
luksSuspend <device> - Suspend LUKS device and wipe key (all IOs are frozen).
luksResume <device> - Resume suspended LUKS device.
luksHeaderBackup <device> - Backup LUKS device header and keystlots
luksHeaderRestore <device> - Restore LUKS device header and keystlots

You can also use old <action> syntax aliases:
open: create (plainOpen), luksOpen, loopaesOpen, tcryptOpen
close: remove (plainClose), luksClose, loopaesClose, tcryptClose

<name> is the device to create under /dev/mapper
<device> is the encrypted device
<key slot> is the LUKS key slot number to modify
<key file> optional key file for the new key for luksAddKey action

Default compiled-in key and passphrase parameters:
Maximum keyfile size: 8192kB, Maximum interactive passphrase length 512 (characters)
Default PBKDF2 iteration time for LUKS: 1000 (ms)

Default compiled-in device cipher parameters:
loop-AES: aes, Key 256 bits
plain: aes-cbc-essiv:sha256, Key: 256 bits, Password hashing: ripemd160
LUKS1: aes-xts-plain64, Key: 256 bits, LUKS header hashing: sha1, RNG: /dev/random

[nelz@hactar ~] 0%
```

» Запустив `cryptsetup --help`, вы увидите список не только доступных команд, но также хэшей и шифров.

быть LVM поверх массива RAID, который, в свою очередь, также расположен на блочных устройствах отдельных дисков или их разделов. Шифрование всего устройства с помощью *dm-crypt* работает следующим образом: на базе вашего носителя информации создаётся блочное устройство, которое шифрует данные при сохранении и дешифрует при чтении. Затем вы монтируете стандартную файловую систему поверх зашифрованного устройства, и она функционирует так же, как и на обычном разделе диска.

Многие дистрибутивы можно установить на зашифрованный диск, но мы рассмотрим непосредственно создание и работу устройств *dm-crypt*, не касаясь чёрной магии, которую творит установщик. *Dm-crypt* использует подсистему ядра для отображения устройств [device mapper] для управления блочными устройствами криптографическими функциями ядра в целях шифрования. Всё делается за счёт ядра, но на уровне пользователя нам необходимо программное обеспечение для создания устройств *dm-crypt* и управления ими; таким стандартным инструментом выступает *cryptsetup*. Вероятно, он уже установлен в вашем дистрибутиве; а если нет, то уж точно будет в основных репозиториях.

Шифрование

Cryptsetup может создавать два типа зашифрованных устройств: простые *dm-crypt* и LUKS. Если вы знаете, как пользоваться простыми *dm-crypt*, то вы знаете о шифровании больше, чем мы сможем здесь рассказать; поэтому мы остановимся только на LUKS, которые в большинстве случаев являются лучшим вариантом. Экспериментировать с файловыми системами, шифрованными или нет, значит в процессе обучения рисковать данными на диске. Во всех примерах мы используем */dev/sdb*, который, как мы полагаем, является внешним или иным запасным устройством (имеющим резервную копию) — но не пытайтесь ничего менять на своём системном диске, пока не разберётесь как следует. Все эти команды должны быть запущены с правами администратора, поэтому войдите в терминал от имени *root* через *su*, или перед каждой командой добавляйте *sudo*.

Для начала создадим зашифрованное устройство:

```
cryptsetup luksFormat /dev/sdb1
```

После запроса на ввод пароля создастся шифрованный раздел на */dev/sdb1*. Вы можете открыть зашифрованное устройство с помощью

```
cryptsetup luksOpen /dev/sdb1 name
```

Вновь последует запрос пароля, а затем в */dev/mapper* создастся устройство с именем, заданным в командной строке. Далее */dev/mapper/name* можно использовать как любое блочное устройство:

```
mkfs.ext4 /dev/mapper/name
```

```
mount /dev/mapper/name /mnt/encrypted
```

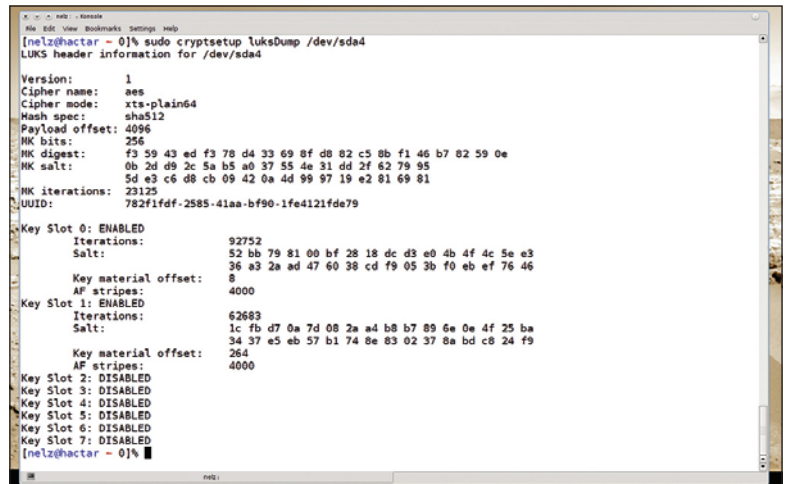
Для паролей действуют обычные правила: они должны быть длинными и вариативными, чтобы их было трудно угадать, но легко запомнить. Утратив пароль, вы теряете содержимое устройства.

Безопасное хранение ключей

Устройство, зашифрованное LUKS, содержит восемь ключевых слотов. Ключи — то же, что и пароли, и их можно назначить несколько на одно устройство, что актуально при наличии нескольких систем, когда вы хотите иметь мастер-пароль, который знаете только вы. При использовании *LuksFormat* пароль хранится в слоте 0, затем вы можете добавить ещё, используя

```
cryptsetup luksAddKey /dev/sdb1
```

Вам предложат ввести текущий пароль, а затем новый. Ключом может также быть не фраза, а содержимое файла; файл может содержать все что угодно, но обычно используют случайные данные:



```
dd if=/dev/urandom of=/path/to/keyfile bs=1k count=4
```

```
chmod 0400 /path/to/keyfile
```

```
cryptsetup luksAddKey /dev/sdb1 /path/to/keyfile
```

```
cryptsetup luksOpen --key-file /path/to/keyfile /dev/sdb1 name
```

Разумеется, ключевые файлы должны быть сохранены надёжно, на незашифрованном устройстве, и доступ к ним должен иметь только администратор. Из личного опыта: даже если том защищён ключевым файлом, лучше всё равно установить сложный пароль, записанный в надёжном месте, на случай, если файл окажется повреждён или недоступен по другим причинам. Ключи также можно менять или удалять, командами *luksChangeKey* и *luksRemoveKey*.

Дополнительные опции

До сих пор мы разбирались с вариантами шифрования по умолчанию, но *cryptsetup* включает также опции *--hash* и *--cipher*. Первая определяет способ хэширования паролей, а вторая задаёт метод шифрования. Значений по умолчанию, как правило, более чем достаточно, а все доступные опции можно просмотреть с помощью

```
cryptsetup --help
```

Эти параметры необходимы только с *LuksFormat*. При создании защищённого устройства *cryptsetup* автоматически использует правильные настройки для его открытия. Лучше всего придерживаться популярных шифров и хэшей, если только у вас нет более веской причины выбрать что-то другое. У методов, используемых реже, могут быть скрытые недостатки, просто потому, что они менее протестированы, что и произошло недавно с реализацией хэша Whirlpool в библиотеке *libcgrypt*, используемой *cryptsetup*. При внесении исправлений пострадали те системы, где уже использовались дефектные хэши.

Ещё одна причина придерживаться общепринятых методов — портативность. Для внутреннего диска это не важно, но если вы хотите использовать шифрованный диск на другой системе, там тоже должны быть установлены те же хэши и шифры. **LXF**

► По *cryptsetup luksDump* можно получить информацию о зашифрованном разделе LUKS. Там же находятся команды резервного копирования и восстановления, чтобы продублировать информацию LUKS.

LUKS

Linux Unified Key Setup был создан ради предоставления стандартного, кросс-платформенного (несмотря на название) формата хранения зашифрованных данных на дисках. Он касается не методов шифрования, а способа хранения информации о них. Он также является более надёжным

способом хранения ключей или паролей, так как метод с *dm-crypt* подвержен взлому. Поскольку LUKS кросс-платформенный, доступ к зашифрованным устройствам можно получить и из Windows, с помощью *FreeOTFE* (<http://sourceforge.net/projects/freetotfe.mirror>).

OpenStack: Создаем облако

Джолион Браун демонстрирует автоматическую сборку *OpenStack* и рассматривает продвинутые возможности последней версии, *Icehouse*.



Наш эксперт

Джолион Браун

К ужасу жены, Джолион ушел со своей стабильной и уважаемой работы и начал карьеру фрилансера со специализацией на *DevOps* и *OpenStack*.



В прошлом номере *Linux Format* мы познакомились с *DevStack* и создали с его помощью один экземпляр *OpenStack*. Хотя *DevStack* — отличное начало, оно не отражает процесса построения облака *OpenStack* в реальном мире. Те, кто пользовался руководствами по установке с <https://docs.openstack.org>,

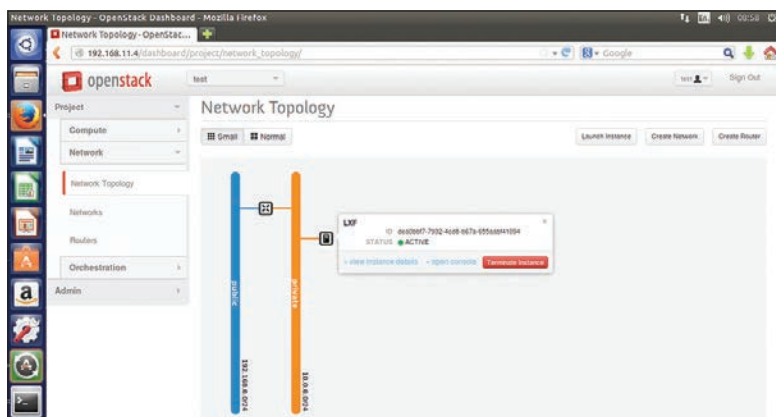
знают, что при ручной настройке *OpenStack* надо вручную менять огромное количество конфигурационных файлов. К счастью для нас, в экосистеме *OpenStack* предпринята масса усилий по автоматизации таких задач. Экономика также требует, чтобы облачные провайдеры предоставляли свои ресурсы быстро. Точно так же, как пользователи хотят быстро получать ресурсы по запросу (и быстро избавляться от них), провайдеры хотят быстро разворачивать новые ресурсы в дата-центрах для своих клиентов, с минимальным вмешательством или вообще без него.

Поэтому на сей раз мы в автоматизированном режиме создадим облако большего размера, а затем рассмотрим некоторые другие компоненты *OpenStack*, способные вас заинтересовать. К счастью, только что вышла последняя версия *OpenStack* — *Icehouse* — и это прекрасная возможность на нее взглянуть. (См. врезку «Что нового в *Icehouse*?», стр. 79.) Однако, будущие администраторы облака, будьте осторожны! *Icehouse* — сравнительно новая программа, и в ней почти наверняка будут ошибки (мы и сами с ними столкнулись при написании этой статьи). Но, с другой стороны, нет лучшего способа познакомиться со сложной программой, чем скомпилировать ее и решая проблемы по мере их появления.

Подготовка хоста

Для решения этой задачи создадим с помощью *Vagrant* несколько виртуальных узлов, с которыми можно поэкспериментировать. На нашем уроке предполагается, что у нас настольная установка с достаточным объемом памяти (мы выделим виртуальным машинам 6 ГБ). Мы также предполагаем, что читателям знаком процесс сборки такой установки, и не будем описывать его подробно, чтобы сэкономить место. Мы также предполагаем, что у этой системы есть доступ к Интернету. Наконец, мы предполагаем, что сеть, к которой подключен настольный компьютер — обычная домашняя сеть, в которой роутер или ADSL-модем выступает в качестве шлюза по умолчанию, перенаправляющего DNS-сервера и сервера DHCP для сети /24, а его IP-адрес — 192.168.0.1.

Выбор виртуализированной среды, выполняющейся на одном физическом компьютере, определяется необходимостью использования нашим облаком нескольких сетей. Виртуализация всего на системе с достаточным набором ресурсов позволяет легко создавать виртуальные машины и, надеюсь, открывает наш урок для тех читателей, кто предпочел бы не ползать по дому в поисках переключателей. Затем мы воспользуемся *Puppet* для разворачивания наших гостевых систем в облаке *OpenStack*, воспользовавшись бесплатными модулями, которые Puppetlabs (компания-разработчик *Puppet*) создала совместно с сообществом *OpenStack*.



» Топология сети с программным роутером и нашей новой запущенной системой.

Для гостевых систем мы воспользовались образом 64-битной Ubuntu 14.04 (загрузив его с ubuntu.com). На самом деле дистрибутив здесь не играет особой роли — лишь бы поддерживал *Vagrant* и *Virtualbox*; но с 64-битным дистрибутивом все будет проще. Настроив и запустив настольную систему, первым делом установите *Virtualbox* в качестве платформы виртуализации. Нам также понадобятся *git*, некоторые служебные файлы Ruby и пакет *build-essentials*; можете установить их сейчас. Для этого достаточно открыть терминал в настольной системе и выполнить команду

```
sudo apt-get update && sudo apt-get install virtualbox git ruby-dev build-essential —y
```

Вместе с этими пакетами могут устанавливаться дополнительные зависимости. Теперь установим *Vagrant*. Рекомендуем воспользоваться преимуществами последней версии, загрузив ее с www.vagrantup.com (Ubuntu здесь немного отстает). Опять же, в терминале выполните следующие команды:

```
wget https://dl.bintray.com/mitchellh/vagrant/vagrant_1.6.2_x86_64.deb
sudo dpkg -i vagrant_1.6.2_x86_64.deb
```

Либо загрузите файл **.deb** через браузер с сайта www.vagrantup.com/downloads.html.

Vagrant — прекрасный выбор для быстрого развертывания сред разработки (см. врезку «Коротко о *Vagrant*» на следующей странице). В облаке *OpenStack* существует масса способов развертывания узлов, и выбор того или иного способа может перерасти в спор фанатиков. Зато *Vagrant* нет равных по простоте и скорости работы. Изменив один файл (называемый *Vagrantfile*), мы сможем задать узлы и сделать их доступными для использования с операционной системой практически мгновенно. По умолчанию в качестве «провайдера», на котором будут запускаться эти узлы, в *Vagrant* используется *Virtualbox*, но также можно воспользоваться *VMWare*, *Hyper-V* (!) и другими программами, включая внешние облачные провайдеры. Но для наших задач отлично сойдет *Virtualbox*.

Наша следующая задача — определить базовый образ, который мы будем использовать в проекте. В *Vagrant* для быстрого клонирования виртуальных машин применяются базовые образы («компьютеры» на языке *Vagrant*). Это экономит массу времени, затрачиваемого на определение и построение виртуальной машины с нуля. На нашем уроке мы построим гостевые системы на базе «компьютера» CentOS, предоставляемого Puppetlabs. Для этого просто добавим «компьютер» в локальное окружение *Vagrant*, выполнив команду

```
vagrant box add puppetlabs/centos-6.5-64-puppet
```

и выбрав из представленных опций *Virtualbox*. На это обычно требуется некоторое время.

Vagrant загрузит для нас требуемый образ (он хранится в `~/vagrant.d`, если кому интересно), который будет использоваться неоднократно. В нем уже установлен *Puppet*, что очень удобно, но позже мы его обновим.

Puppet, OpenStack и Vagrant

В модулях *Puppet* для *OpenStack* есть несколько примеров конфигураций *OpenStack*, дополненных несколькими готовыми файлами настройки для *Puppet*. Здесь мы адаптируем пример 'multinode' для использования с *Vagrant* и *Virtualbox*. Для этого потребуется загрузить несколько файлов с сайта **LXF** (<http://bit.ly/LXFicescript>). Загрузите файл `lxf-icehouse.tar.gz`. Поместите его в домашний каталог, а затем распакуйте командой

```
tar -zxvf lxf-icehouse.tar.gz
```

Вы увидите, как в новом подкаталоге **icehouse** появятся несколько новых файлов. Войдите в него командой `cd` и взгляните

на эти файлы. Важнее всего для нас сам файл *Vagrantfile*: он содержит определения используемых нами виртуальных систем. Вот конфигурация для узла 'control':

```
config.vm.define «control» do |control|
  control.vm.box = "puppetlabs/centos-6.5-64-puppet"
  control.vm.hostname = "control.localdomain"
  control.vm.provision :shell, :path => "bootstrap.sh"
  control.vm.network "private_network", ip: "192.168.11.4"
  control.vm.network "private_network", ip: "192.168.0.44"
  control.vm.network "private_network", ip: "172.16.33.4"
  control.vm.network "private_network", ip: "172.16.44.4"
  control.vm.provider "virtualbox" do |vbl|
    vbl.customize ["modifyvm", :id, "--memory", "2048"]
  end
end
```

Возможно, вы распознали синтаксис Ruby; файл *Vagrantfile* написан на Ruby DSL (domain specific language — предметно-ориентированный язык). Этот фрагмент представляет собой блок кода Ruby, настраивающий новую виртуальную машину: он сообщает *Vagrant*, какой базовый образ использовать (наш импортированный образ centos), задает имя хоста (в этом примере — шаблонное имя, необходимое для *Puppet*), указывает запускаемый скрипт и затем задает четыре используемых интерфейса (для API, публики, управления и трафика данных соответственно). Наконец, здесь мы выделяем немного дополнительной памяти контроллеру, так как он запускает множество сервисов *OpenStack*.

Что касается других файлов, то `openstack.yaml` содержит настройки, которые *Puppet* использует для наполнения облака во время сборки; `puppet-bootstrap.sh` и `bootstrap.sh` — очень простые скрипты, которые выполняют обновление Puppet и проверяют, что параметры хостов в файлах в демонстрационном облаке идентичны (для *Vagrant* существуют плагины, которые сделают это за нас, но во время тестирования они выдавали ошибки). Файл `site.pp` назначает каждому узлу роль — согласно ей *Puppet* решат, как настроить узел.

Вы видите, что в некоторых из этих файлов используются ссылки на сеть 192.168.0. Этот адрес следует заменить на свой.

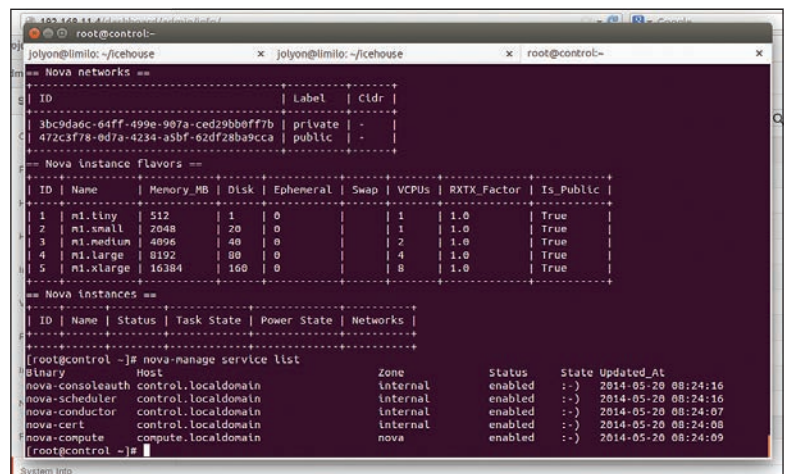
Закончив, приступите к запуску среды, выполнив следующую команду:

```
vagrant up puppet
```

Vagrant запустит виртуальную машину, установит сервер *Puppet* (и обновит агент), затем установит нужные нам модули *OpenStack*. Это потребует времени, так что проявите терпение. Далее »

Скорая помощь

HashiCorp (разработчик *Vagrant*) поддерживает удобный репозиторий образов, известный как *Vagrant Cloud* (www.vagrantcloud.com). Сотрудники *Vagrant Cloud* загружают в репозиторий настроенные образы с различными пакетами программ. Эти образы доступны для использования всем желающим.



» Если новые сервисы настроены и запущены, *OpenStack* поприветствует вас улыбкой :-)

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



На странице puppetlabs.com/learn можно найти несколько исчерпывающих руководств и документацию по *Puppet*, которые расскажут, что это такое и как этим пользоваться.

нужно запустить контроллер *OpenStack*, причем перед другими узлами, поскольку те зависят от его сервисов. Выполните команду

```
vagrant up control
```

Эта установка проще, и должна завершиться быстрее, чем установка узла *Puppet*. Прежде чем сервер *Puppet* сможет управлять узлом, следует подписать сертификат агента. При первом запуске агент *Puppet* отправляет запрос серверу, который можно подтвердить, выполнив следующую команду:

```
vagrant ssh puppet -c "sudo puppet cert list"
```

Вы должны увидеть запрос узла управления, ожидающий подтверждения. Его можно обработать командой

```
vagrant ssh puppet -c "sudo puppet cert sign --all"
```

Puppet подтвердит, что подписывает сертификат, и удалит его из очереди. Теперь можно приступить к установке контроллера. Это сделает еще одна команда *Vagrant* — запустите *Puppet* вручную:

```
vagrant ssh control -c "sudo puppet agent -t"
```

На экране должны появиться сообщения от агента *Puppet*. Если вы видите сообщение о том, что запуск уже производится, не пугайтесь — это агент *Puppet* берет дело в свои руки. Мы можем отслеживать его запуск в гостевой системе:

```
vagrant ssh control
```

```
sudo tail -f /var/log/messages
```

(Для выхода из команды *tail* просто нажмите **Ctrl+C**.) Учтите, что первоначальное развертывание — процесс длительный. Разумеется, длительность зависит от мощности настольного ПК, но нередко будет не менее десяти минут (а то и больше). Это может показаться чрезмерным, но лучше подождать десять минут, чем часами редактировать файлы настройки.

В конце концов вы будете вознаграждены сообщением "Finished catalog run [Запуск каталога завершен]". Если все хорошо, запускайте три оставшихся узла:

```
vagrant up storage && vagrant up network && vagrant up compute
```

Эти узлы тоже отправят запросы на сертификаты. Удовлетворить их можно, снова выполнив команду **puppet cert sign** (как и прежде). Команды агентов *Puppet* можно запускать одновременно в отдельных окнах терминала/вкладках (только не забудьте запускать их в каталоге **icehouse**):

```
vagrant ssh storage -c "sudo puppet agent -t"
```

```
vagrant ssh network -c "sudo puppet agent -t"
```

```
vagrant ssh compute -c "sudo puppet agent -t"
```

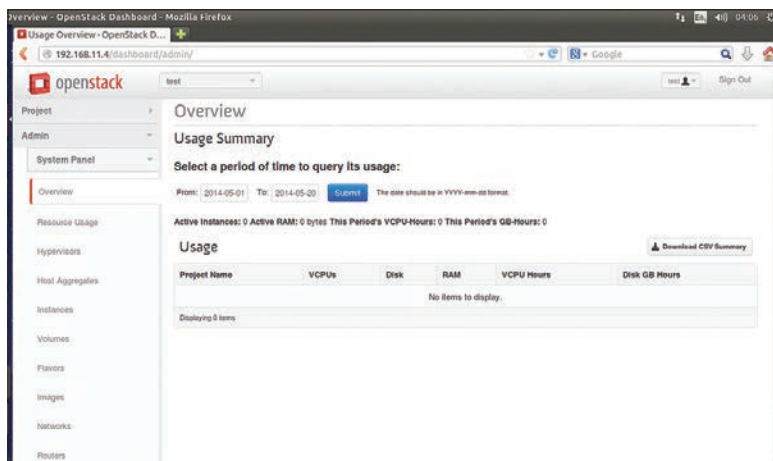
К счастью, эти запросы не уйдут за пределы узла **control**. Запрос для узла **compute**, скорее всего, будет выполняться дольше всех. Во время тестирования у нас появились ошибка во время первоначального запуска узла **network**: «Выполнение команды "/usr/bin/neutron router-gateway-set test public" не удалось [Execution of "/usr/bin/neutron router-gateway-set test public" failed]» (вместе с некоторыми другими зависимостями). Об этом не беспокойтесь — просто еще раз выполните команду **sudo puppet agent -t**, и при последующих запусках это сообщение должно исчезнуть. Когда все эти команды завершатся, наше новое облако на *Icehouse* готово и запущено! Теперь можно зайти на панель управления *OpenStack* через браузер. Введите адрес <http://192.168.11.4>.

Исследование Icehouse

В соответствии с содержимым файла **openstack.yaml** мы можем зайти в **web**-интерфейс как пользователь **test** с паролем **abc123**. Откроется довольно приятная глазу панель управления *Icehouse*. Первым делом проверим, что узлы можно запустить.

Выберите в выпадающем списке слева **Project** [Проект], затем **Access & Security** [Доступ и безопасность]. Напротив группы безопасности по умолчанию выберите **Manage Rules** [Управление правилами]. Добавьте правило, выбрав **SSH** из выпадающего списка. Нажмите **Add** [Добавить] для подтверждения. Теперь снова зайдите в **Access & Security**, но на сей раз перейдите на вкладку **Key Pairs** [Пары ключей]. Нажмите **Create Key Pair** [Создать пару ключей], назовите ее "LXF", набрав это в поле **Key Pair Name** [Имя пары ключей], и затем нажмите **Create Key Pair**. После этого должно открыться окно сохранения ключей. Сохраните их на настольный компьютер. Теперь перейдите на вкладку **Floating IPs** [Плавающие IP]. Выберите **Allocate IP to Project** [Выделить проекту IP-адрес]; выбирать адреса можно только из публичного пула. Затем нажмите **Allocate IP** [Выделить IP]. Вы должны получить IP-адрес **192.168.0.101** (он зависит от вашей локальной сети).

➤ В окне системной информации можно увидеть, какие сервисы включены в нашем облаке.



Коротко о Vagrant

Vagrant — средство для создания легких, портативных и воспроизводимых сред разработки. Создав всего один файл (как описано на нашем уроке), настроить и запустить виртуальную машину можно за несколько минут. *Vagrant* также гарантирует, что конфигурация будет воспроизводимой, т.е. дубликатами системы можно обмениваться, просто пересылая текстовые файлы, используемые программой (чем мы на нашем уроке и занимаемся). Наряду с простыми скриптами оболочки, которыми

мы пользуемся, в *Vagrant* есть встроенные возможности *Puppet* и *Chef* (и другие), благодаря которым *Puppet* способен развернуть свой *Puppet*...

По принятому соглашению, в «компьютерах» *Vagrant* (особенно в импортированных из *Vagrant Cloud*) используется фиктивная пара ключей **ssh**; это позволяет подключиться к гостевой системе без указания пароля командой **vagrant ssh <vm>**. По тому же соглашению, пароль пользователя в *Vagrant* обычно "vagrant". Довольно часто, хотя

и не всегда, такой же пароль имеет **root**. Одна из приятных возможностей — после настройки и запуска гостевой системы очень легко синхронизировать каталоги между основной системой *Vagrant* и гостевыми системами. По умолчанию каталог с *Vagrantfile* доступен гостевому компьютеру по пути **/vagrant** (им мы пользуемся на нашем уроке).

Здесь мы можем лишь слегка коснуться полезных возможностей программы. Чтобы узнать больше, зайдите на <http://docs.vagrantup.com>.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.


```

jolyon@limilo: ~/icehouse
AULT/rpc_backend]/ensure) created
May 20 01:33:40 localhost puppet-agent[8700]: (/Stage[main]/Heat/Heat_config[key
stone_auth_token/auth_uri]/ensure) created
May 20 01:33:40 localhost puppet-agent[8700]: (/Stage[main]/Heat/Heat_config[DEF
AULT/rabbit_port]/ensure) created
May 20 01:33:40 localhost puppet-agent[8700]: (/Stage[main]/Heat::Api/Heat_conf
ig[heat_api/workers]/ensure) created
May 20 01:33:40 localhost puppet-agent[8700]: (/Stage[main]/Heat::Api_cfn/Heat_c
onfig[heat_api_cfn/bind_host]/ensure) created
May 20 01:33:40 localhost puppet-agent[8700]: (/Stage[main]/Heat/Heat_config[ec2
auth_token/auth_uri]/ensure) created
May 20 01:33:41 localhost puppet-agent[8700]: (/Stage[main]/Heat/Heat_config[key
stone_auth_token/auth_protocol]/ensure) created
May 20 01:33:41 localhost puppet-agent[8700]: (/Stage[main]/Heat::Engine/Heat_co
nfig[DEFAULT/auth_encryption_key]/ensure) created
May 20 01:33:41 localhost puppet-agent[8700]: (/Stage[main]/Heat/Heat_config[DEF
AULT/rabbit_userid]/ensure) created
May 20 01:33:41 localhost puppet-agent[8700]: (/Stage[main]/Heat::Api_cfn/Heat_c
onfig[heat_api_cfn/workers]/ensure) created
May 20 01:33:41 localhost puppet-agent[8700]: (/Stage[main]/Neutron::Server/Neut
ron_config[database/idle_timeout]/ensure) created
May 20 01:33:41 localhost puppet-agent[8700]: (/Stage[main]/Cinder::Api/Cinder_a
pi_paste_ini[filter:auth_token/admin_password]/ensure) created

```

» Созерцание мелькающих логов установок — занятие скучноватое, но представьте, что вам пришлось бы настраивать все это вручную.

Теперь можно запустить наш образ с помощью меню Images [Образы] слева. Попробуем сделать это с тестовым образом Cirros. Нажмите Launch [Запустить] (назовите систему как хотите) и подтвердите, что пара ключей LXF доступна, нажав на вкладку Access & Security. Справа от этой вкладки находится вкладка Networking [Сеть]. Выберите частную сеть, нажав на синий плюсик. Наконец, нажмите кнопку Launch [Запустить]. После того как браузер закончит работу перейдите в меню Instances [Системы] слева и выберите Associate Floating IP [Назначить плавающий IP]. Наш адрес .101 должен быть доступен в выпадающем меню. Выберите его и нажмите Associate [Назначить].

Теперь можно подключаться к новой системе из сеанса терминала на рабочем столе по SSH, воспользовавшись загруженной ранее парой ключей LXF. Сначала измените права доступа к ключам, а затем воспользуйтесь ими следующим образом:

```
chmod 600 ~/Downloads/lxf.pem
```

```
ssh cirros@192.168.0.101 -i ~/Downloads/lxf.pem
```

Взгляните на вкладку Network Topology [Топология сети] слева (под вкладкой Network [Сеть]). Здесь мы видим нашу новую систему, подключенную к частной сети, с программным роутером, который связывает эту сеть с нашей публичной. При необходимости легко добавить другие сети или роутеры. Этим роутером

управляет *Neutron*, сетевой компонент *OpenStack*. *Neutron* позволяет создавать продвинутые виртуальные сетевые топологии с такими сервисами, как брендмауэры, балансировщики нагрузки и виртуальные частные сети.

Теперь у нас есть воспроизводимое облако *OpenStack*, и почему бы не поэкспериментировать с различными сетевыми плагинами и сервисами? Превосходная документация *OpenStack* будет отличной стартовой площадкой. Другая интересная задача — включение Swift, сервиса хранения объектов. Это делается в файлах настройки *Puppet*, которые теперь есть на вашем настольном ПК, а руководства и документация имеются на <https://github.com/puppetlabs/puppetlabs-openstack>.

В нашем облаке также установлен механизм управления. Наконец, реальные возможности *OpenStack* (как и со многим другим, что есть в Linux) проистекают из командной строки и вызовов API, доступных системным администраторам и конечным пользователем. Перейдите на узел control командой **vagrant ssh**, переключитесь на root и откройте файл **openrc** в его домашнем каталоге, чтобы можно было воспользоваться командами **openstack-status** и **nova-manage service list** (команд на самом деле намного больше). Не забывайте: с *Vagrant* и *Puppet* вы сможете уничтожить все и воссоздать это снова. Удачи! **LXF**

Скорая помощь

Если вы хотите выключить свою систему, но не собираетесь воспроизводить их с нуля, воспользуйтесь командой **vagrant halt** «*сия узла*». Для перезапуска системы воспользуйтесь командой **vagrant up**.

Что нового в Icehouse?

OpenStack Icehouse, девятая версия проекта, появилась в середине апреля, как раз для этой статьи. Цикл разработки *Icehouse* имел четкую цель в сосредоточении на конечных пользователях платформы. Это означает повышение объема и качества тестирования в целях стабильности, а также несколько ключевых проектов для облачных операторов. В этой задаче приняли участие 1202 пользователя (цифра впечатляет), которые создали почти 350 новых возможностей и устранили 3000 ошибок. Несколько ведущих компаний, участвовавших в разработке *OpenStack*,

восславили качества *Icehouse* версии Enterprise (вероятно, с оглядкой на прибыли, которые возраснут с ростом количества предприятий, использующих платформу). Вот несколько самых привлекательных возможностей:

» В Nova, компоненте Compute *OpenStack*, появились скользящие обновления (переход с *Havana* на *Icehouse* происходит гораздо быстрее, чем раньше).

» Keystone (Identity Management) продвинулся вперед, добавив «интегрированные» данные аутентификации. Это позволяет управлять как

публичными, так и частными ресурсами, используя одни и те же данные, и это большой шаг по направлению к подлинно гибридным облакам *OpenStack* (организация может захотеть, чтобы часть элементов облака хранились в ее собственном дата-центре, а часть — где-то еще, но иметь возможность централизованно управлять всеми компонентами облака).

» Trove, сервис базы данных, полностью интегрирован в систему *OpenStack*, чтобы конечные пользователи могли быстро и легко разворачивать экземпляры базы данных.

Mopidy: Вещаем музыку потоком

Маянк Шарма поясняет, как превратить старый компьютер — или Raspberry Pi — в сервер потокового воспроизведения музыки с поддержкой Spotify.



Наш эксперт

Маянк Шарма любит отвечать на вопросы о Raspberry Pi и был пишущим редактором [Linux.com](http://linux.com).

Можно ли найти более захватывающее применение для старого компьютера, чем трансляция с его помощью музыки в свои динамики Hi-Fi? Немного повозившись, можно соорудить из него сервер для потоковой трансляции музыки — как локальной, так и размещенной на сервисах потоковой трансляции в Интернете, вроде Spotify и Google Play. И управлять своим изобретением вы сможете с другого компьютера или даже со своего смартфона, если он будет подключен к вашей локальной сети.

Душой нашей схемы будет музыкальный сервер *Mopidy*. Он умеет проигрывать музыку из разнообразных источников, и вы сможете транслировать музыку с интернет-радиостанций и сервисов вроде Spotify и SoundCloud так же легко, как с локального жесткого диска.

У *Mopidy* нет графического клиента, а поскольку мы будем управлять воспроизведением с удаленного компьютера, *Mopidy* можно установить на чистый Ubuntu Server, чтобы не подвергать престарелый компьютер ненужному стрессу.

Установка Mopidy

Для начала назначим компьютеру, с которого будет транслироваться музыка, фиксированный IP-адрес. Сделать это удобнее всего из web-интерфейса администрирования роутера. Для целей нашего урока предположим, что наш сервер имеет адрес 192.168.1.100.

Теперь зайдите на сервер и введите следующую команду:

```
wget -q -O - http://apt.mopidy.com/mopidy.gpg | sudo apt-key add
```

Она добавит ключ аутентификации для репозитория. Загрузив ключи, можно добавить репозиторий пакетов *Mopidy* следующей командой:

```
sudo wget -q -O /etc/apt/sources.list.d/mopidy.list  
http://apt.mopidy.com/mopidy.list
```

Добавив репозиторий, обновите менеджер пакетов командой `sudo apt-get update` и загрузите сервер командой `sudo apt-get install mopidy`.

После загрузки и установки сервера его можно запустить, просто набрав `mopidy` в командной строке. Сервер создаст в вашем домашнем каталоге файл настройки `~/.config/mopidy/mopidy.conf`.

Вам будет нужно изменить этот файл согласно своей системе. Перед изменением файла убедитесь, что сервер *Mopidy* не запущен. Остановить сервер можно, нажав `Ctrl+C` в терминале.

Настройка сервера

У *Mopidy* есть масса параметров, которые можно менять, и в файле настройки по умолчанию перечислены некоторые из наиболее полезных. Функциональность *Mopidy* доступна через расширения. По умолчанию включены четыре основных.

Для воспроизведения музыки с локального диска используется расширение "Local". Чтобы указать диск, откройте файл настройки в текстовом редакторе и раскомментируйте строки в разделе [Local], убрав # в начале строк. В параметре `media_dir` укажите каталог с музыкой, например, `media_dir = /home/bodhi/Music`.

Теперь сохраните файл и попросите *Mopidy* просканировать все треки в указанном каталоге, командой `mopidy local scan`. Запускайте данную команду при каждом добавлении новых треков в этот каталог, иначе *Mopidy* не сможет проиграть их.

Для управления сервером с удаленных компьютеров и устройств также понадобится включить еще несколько расширений.

Что такое MPD?

В сервере музыки *Mopidy* используется собственная реализация сервера *MPD*. Но изначальные возможности *MPD* гораздо шире, чем у *Mopidy*. *MPD* — сокращение от "Music Player Daemon [Демон музыкального проигрывателя]". В отличие от обычных музыкальных проигрывателей, в *MPD* используется клиент-серверная модель. Разделение проигрывателя на два компонента имеет несколько преимуществ. Благодаря ему *MPD* не только потребляет меньше системных ресурсов, но и приобретает замечательные новые возможности, такие как удаленное воспроизведение

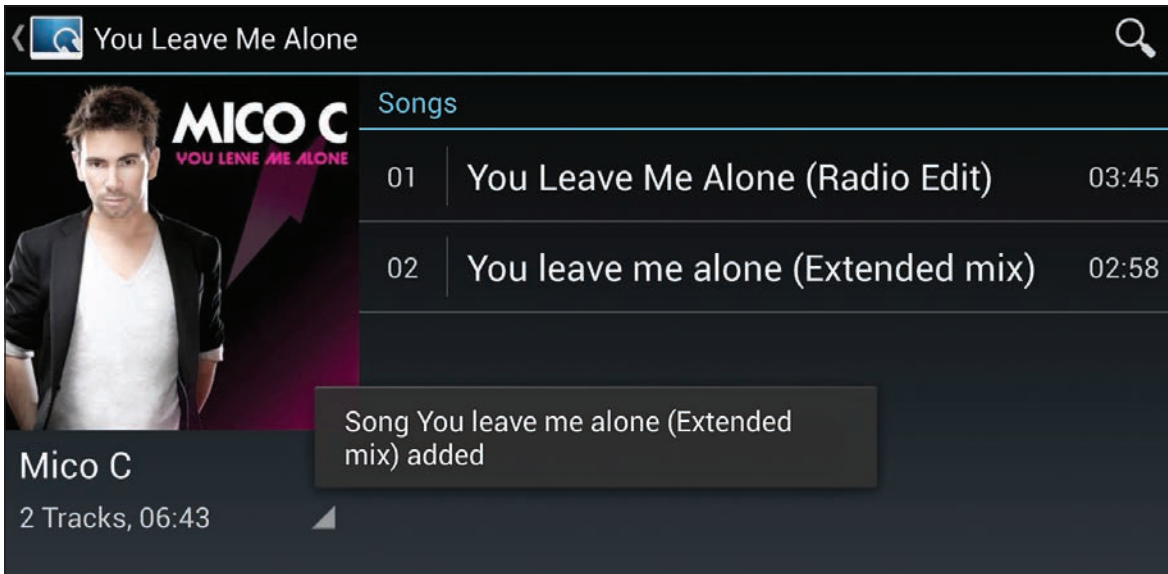
и управление и возможность использовать различные интерфейсы.

MPD умеет проигрывать файлы множества форматов, включая Ogg Vorbis, FLAC, MP3 и другие форматы, поддерживаемые библиотекой *FFmpeg*. Он также воспроизводит HTTP-потоки Ogg и MP3, считывает и кэширует метаданные, обладает «родной» поддержкой Zeroconf и содержит встроенный потоковый сервер HTTP.

Поэтому *MPD* идеально подходит для работы на серверах малой мощности без монитора, а его клиент можно запустить на любом компьютере

сети. Реализация *MPD* в *Mopidy* унаследовала многие возможности исходного сервера *MPD*. Например, его можно удаленно управлять с любого клиента *MPD* и искать музыку в своей коллекции.

Но некоторые возможности в этой реализации *MPD* недоступны. Самая заметная из них — невозможность потоковой трансляции музыки с сервера клиенту. Это означает, что с помощью клиентов *MPD* можно только управлять сервером *Mopidy*. Также в этой реализации на данный момент нельзя менять сохраненные плей-листы; но данная возможность будет доступна в грядущей версии.



» **MPDroid** — чудесное приложение: оно поддерживает все функции сервера MPD и прекрасно ладит с сервером Mopidy.



Компьютер, который будет использоваться в качестве сервера музыки, не обязан иметь особые системные ресурсы. Подойдет любой старый компьютер, который потянет Ubuntu Server.

В Mopidy используется собственная реализация популярного сервера MPD. Благодаря ей можно управлять Mopidy с любого клиента MPD.

Чтобы клиенты MPD могли подключиться к Mopidy, откройте файл настройки Mopidy в текстовом редакторе и раскомментируйте строки в разделе [mpd]. Переменная hostname должна содержать IP-адрес, откуда сервер будет принимать соединения. Так как мы хотим, чтобы сервер принимал соединения с любого устройства сети, замените адрес двойным двоеточием: `hostname =::`.

Также нужно включить расширение HTTP, которое позволяет управлять Mopidy из любого браузера. Как и ранее, раскомментируйте все строки в разделе [http] и измените hostname так, чтобы сервер слушал все соединения из локальной сети. В переменной `static_dir` можно указать расположение клиента (см. руководство на следующей странице).

Вот и все. Теперь запустите сервер командой `mopidy`. Введя IP-адрес сервера в адресную строку браузера на любом компьютере сети, вы увидите web-страницу HTTP-сервера Mopidy по умолчанию.

Сервер настроен; но для удаленного управления и воспроизведения музыки из интернет-источников его нужно нарастить другими компонентами и расширениями. Чтобы к серверу можно было подключиться с удаленного компьютера, на сервере надо установить web-клиент. Для управления музыкой на сервере также можно установить клиент MPD на удаленном компьютере или на мобильном устройстве с ОС Android. Чтобы узнать, как это сделать, воспользуйтесь руководством на следующей странице.

По окончании настройки сервер сможет проигрывать музыку с локального диска или из интернет-сервиса Spotify, и вы сможете управлять им со смартфона из любой точки дома.

»

Потоковое воспроизведение музыки со Spotify

```

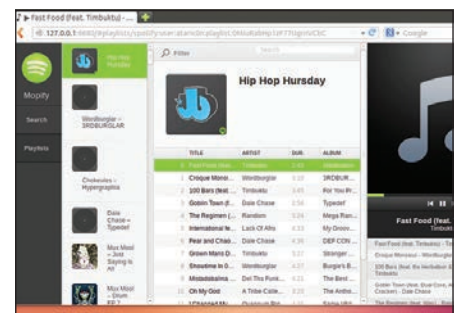
bodhi@bodhi-Aspire-573E:~$ sudo apt-get install mopidy-spotify
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libspotify12 python-spotify python-support
Suggested packages:
  python-spotify-dbg python-spotify-doc
The following NEW packages will be installed:
  libspotify12 mopidy-spotify python-spotify python-support
Need to get 1,469 kB/1,495 kB of archives.
After this operation, 2,647 kB of additional disk space will be used.
Do you want to continue [Y/n]?
  
```

```

[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtp
# rtsp

[spotify]
username = geekybodhi
password = omlp1ntextf4ssw0rd

[soundcloud]
enabled = true
auth_token = 1-35204-22335109-979f766d9d95d9
explorer = electronic/Ambient, pop/New Wave, rock/Indie
  
```



1 Установка

Как и для воспроизведения локальной музыки, для воспроизведения музыки со Spotify у Mopidy тоже есть соответствующее расширение. Если вы следуйте нашему руководству и добавили репозитории mopidy, можно установить его командой `sudo apt-get install mopidy-spotify`. Но предварительно убедитесь, что у вас есть премиум-аккаунт на сервисе Spotify — с бесплатными учетными записями расширение работать не будет.

2 Настройка Mopidy

Если вы создавали учетную запись Spotify с Facebook, вам понадобится пароль на устройстве: зайдите на www.spotify.com/account/set-devicepassword и следуйте инструкциям для получения числового имени пользователя и пароля. Теперь остановите сервер Mopidy, если он запущен, и откройте файл настройки Mopidy в текстовом редакторе. Создайте новый раздел [spotify] и введите логин и пароль для доступа к Spotify.

3 Воспроизводим!

Вот и все. Теперь запустите сервер, который проверит логин и пароль и подключится к сервису Spotify. Для потокового воспроизведения с сервера Spotify просто подключитесь к серверу Mopidy с клиента, умеющего считывать потоки Spotify — например, с клиента MPDroid для Android. Если захотите воспользоваться браузером, установите клиент Mopify (github.com/dirkgroenen/Mopify), следуя руководству на следующей странице.

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

Расширения Mopidy, без которых нельзя обойтись

```

bodhi@bodhi-Aspire-5738: ~/config/mopidy
[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtmp
# rtmps
# rtsp

[gmusic]
username = geekybodhi
password = yetanotherplaintextpassword

[soundcloud]
enabled = true
auth_token = 1-35204-22335169-979f766d9d495d9
explore = electronic/Ambient, pop/New Wave, rock/Indie
(END)
    
```

```

bodhi@bodhi-Aspire-5738: ~/config/mopidy
[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtmp
# rtmps
# rtsp

[gmusic]
username = geekybodhi
password = yetanotherplaintextpassword

[soundcloud]
enabled = true
auth_token = 1-35204-22335169-979f766d9d495d9
explore = electronic/Ambient, pop/New Wave, rock/Indie
(END)
    
```

1 Google Play Music

Mopidy может воспроизводить музыку и с сервиса Google Play Music. Для установки этого расширения сначала установите *Pip*, систему управления пакетами Python, командой `sudo apt-get install python-pip`, а затем с ее помощью установите расширение, командой `sudo pip install mopidy-gmusic`. Затем остановите сервер *Mopidy* и измените его файл настройки, добавив раздел `[gmusic]` со своим пользователем и паролем в Google Play.

2 SoundCloud

Наряду с Google Play Music и Spotify, *Mopidy* может загружать треки и с сервиса SoundCloud. Установить расширение можно с помощью системы управления пакетами *Pip* командой `sudo pip install mopidysoundcloud`. После этого зайдите на www.mopidy.com/authenticate и получите токен аутентификации SoundCloud. Потом добавьте его в параметр `auth_token` раздела `[soundcloud]` в файле настройки *Mopidy*.

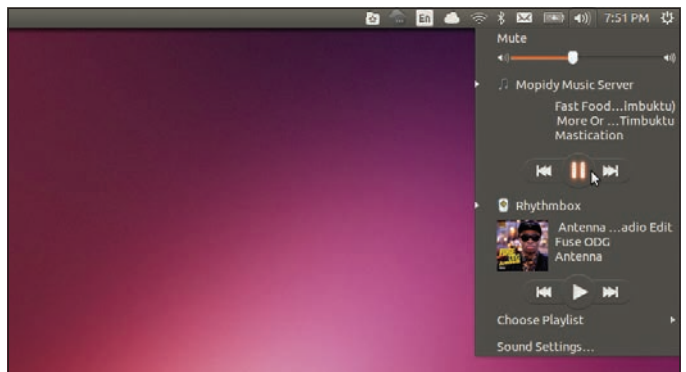
```

bodhi@bodhi-Aspire-5738: ~/config/mopidy
[stream]
#enabled = true
#protocols =
# file
# http
# https
# mms
# rtmp
# rtmps
# rtsp

[gmusic]
username = geekybodhi
password = yetanotherplaintextpassword

[scrobbler]
username = geekybodhi
password = @n6133t

[soundcloud]
enabled = true
auth_token = 1-35204-22335169-979f766d9d495d9
explore = electronic/Ambient, pop/New Wave, rock/Indie
(END)
    
```



3 Скробблер Last.fm

Раз уж вы слушаете много музыки, умно будет следить за тем, что вы слушаете, с помощью скробблера. *Mopidy* может отправлять все прослушанные треки сервису Last.fm. Загрузите расширение командой `sudo pip install mopidy-scrobbler`. Вам также нужно зарегистрировать бесплатную учетную запись на сервисе и изменить файл настройки *Mopidy*, добавив раздел `[scrobbler]` с логином и паролем учетной записи на Last.fm.

4 Интеграция в Ubuntu

Если у вас Ubuntu, то вы сможете управлять сервером *Mopidy* в том числе и из звукового меню Ubuntu. Просто установите расширение *Mopidy* для управления сервером через интерфейс MPRIS D-BUS командой `sudo pip install mopidy-mpiris`. В отличие от других расширений, никаких изменений в настройке *Mopidy* в этом случае вносить не потребуется. Установив его, вы сможете управлять воспроизведением из звукового меню Ubuntu, а также из *Rhythmbox*.

Mopidy на Raspberry Pi

Если у вас есть Raspberry Pi, его можно использовать в качестве музыкального сервера. *Mopidy* работает на архитектурах ARM — как на `hard-float`, так и на `soft-float`, и его можно установить из дистрибутива Raspbian.

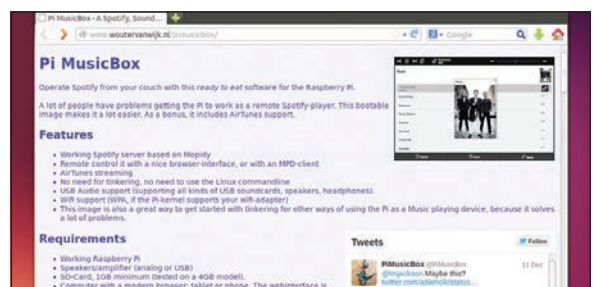
Перед установкой *Mopidy* загрузите в ядро модуль IPv6 командой `sudo modprobe ipv6`, а затем выполните команду `echo ipv6 | sudo tee -a /etc/modules`, чтобы модуль загружался при включении системы.

Также по умолчанию *Mopidy* будет выводить звук с HDMI-порта Raspberry Pi.

Если вы подключили динамики к порту AUX, придется попросить его перенаправить звук командой `sudo amixer cset numid=3 1`.

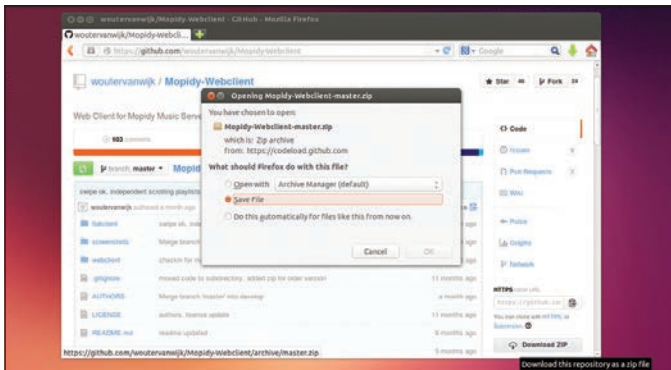
Протестируйте звуковой выход командой `aplay /usr/share/sounds/alsa/Front_Center.wav`. Если звук слышен, добавьте команду `amixer cset numid=3 1` в файл `/etc/rc.local`, чтобы сделать изменения постоянными.

На этом настройка Raspberry Pi закончена. Воспользуйтесь руководством по установке и настройке сервера *Mopidy*.



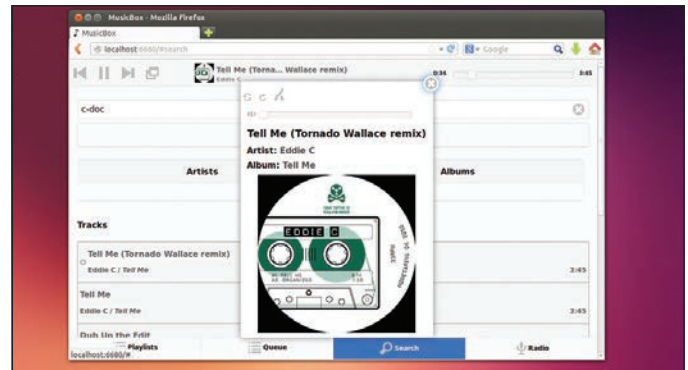
Вместо настройки *Mopidy* воспользуйтесь дистрибутивом *Pi MusicBox* для Raspberry Pi.

Добавляем клиентов



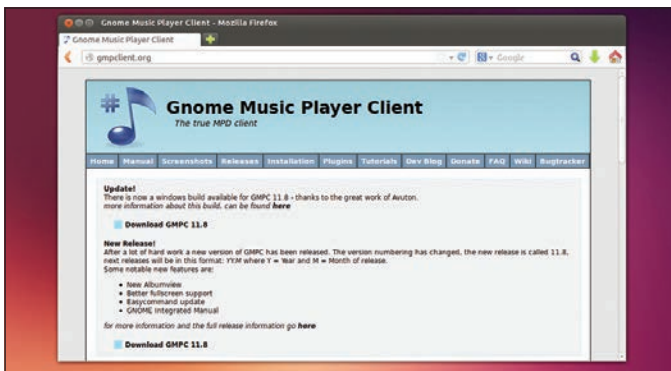
1 Загрузка web-клиента

С web-клиентом *Mopidy* можно управлять музыкальным сервером с любого компьютера сети. Доступных клиентов несколько. Я предпочитаю *mopidy-webclient*, который можно загрузить с <http://github.com/woutervanwijk/Mopidy-Webclient>. После окончания загрузки распакуйте архив и переместите созданный каталог *webclient/* в каталог */opt*.



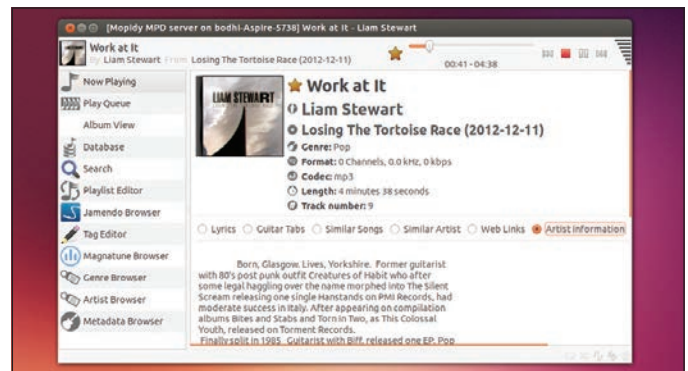
2 Управление из браузера

Теперь откройте файл настройки *Mopidy*, найдите раздел `[http]` и в параметре `static_dir` укажите каталог web-клиента в */opt*. Затем запустите сервер *Mopidy* и введите его IP-адрес в браузере на любом компьютере локальной сети. Вместо страницы web-сервера *Mopidy* по умолчанию вы увидите интерфейс web-клиента *Mopidy* для управления воспроизведением.



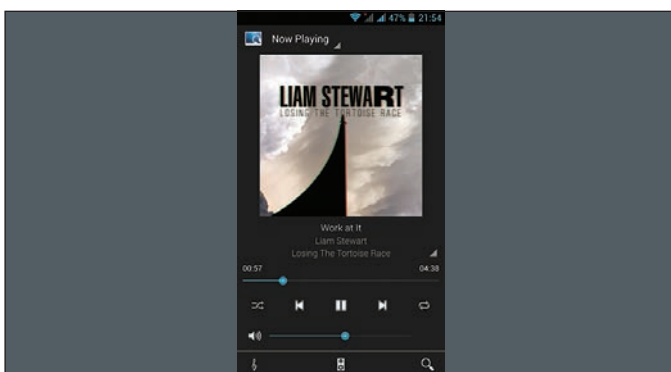
3 Загрузка настольного клиента

Если вы хотите управлять *Mopidy* с рабочего стола, воспользуйтесь одним из настольных клиентов. Я предпочитаю *Gnome Music Player Client*. Несмотря на название, он также доступен в Windows и Mac OS X. Для установки *GMPC* сначала добавьте его репозиторий командой `sudo add-apt-repository ppa:gmpc-trunk/gmpc-stable` и затем обновите менеджер пакетов командой `sudo apt-get update`.



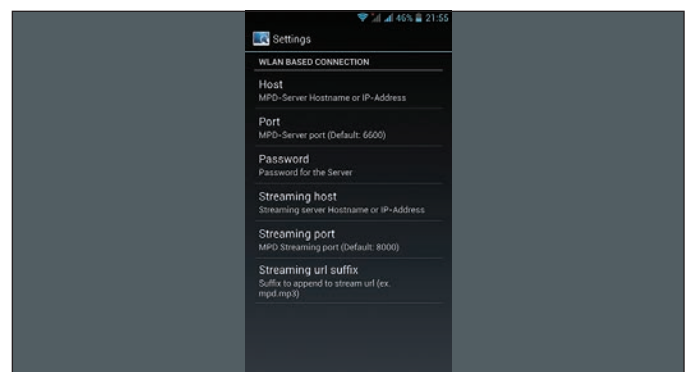
4 Управление с рабочего стола

Теперь установите *GMPC* и его плагины командой `sudo apt-get install gmpc gmpc-plugins`. При первом запуске приложения запустится мастер подключения к *Mopidy*. Благодаря *Zerogonf* ваш сервер будет обнаружен автоматически (перед запуском *GMPC* убедитесь, что сервер запущен), и вам останется только выбрать его из выпадающего списка Profile [Профиль] и нажать кнопку Connect [Подключиться].



5 Загрузка клиента для Android

Для управления сервером *Mopidy* с устройства Android вам понадобится клиент *MPD* для него. В Google Play Store таких клиентов несколько. Как и прежде, можете выбрать любой, но я предпочитаю *MPDroid*. Приложение бесплатно и имеет интерфейс как для телефона, так и для планшета. В *MPDroid* можно искать музыку в своей коллекции и сортировать ее по исполнителям, альбомам, жанрам и т. д.




6 Управление с мобильного устройства

При первом запуске приложения вас попросят его настроить. Сначала выберите подходящее подключение WLAN и свою беспроводную сеть. В следующем окне щелкните на параметр Host и введите IP-адрес музыкального сервера *Mopidy*. Теперь возвращайтесь в главное окно. Отсюда вы сможете просматривать музыку на удаленном сервере и управлять ею. **LXF**

Python: Наладим себе пушечку

Изучите объектно-ориентированный Python, а главное — заодно с **Джонни Бидвеллом**, искателем приключений в *Minecraft: Pi Edition*, что-нибудь взорвите.



Наш эксперт

Джонни Бидвеллу хотелось бы побольше блочных вещей; и провалился эта чертова вселенная из кварков.



как то, из чего состоят все стандартные языки программирования (массивы, типы данных, функции и все остальное — это объекты различных типов). Так что скорее всего вы уже сталкивались с объектами, да только не знали об этом. У каждого объекта есть несколько связанных с ним методов: например, целые числа можно складывать, вычитать, делить и умножать; строки — сравнивать и объединять; списки — резать, укорачивать и дополнять, ну и т. д. Все эти методы появляются у нас сами собой, когда мы создаем свой объект. Поэтому в объектно-ориентированном программировании мы создаем модель с описанием методов существующих объектов. Эта модель называется классом, и в Python методы класса определяются в виде функций. В коде Мартина первый класс, который нам встречается (строка 33), называется **MinecraftDrawing**. Этот класс довольно длинный, состоит из шести методов и способен нарисовать все графические примитивы, какие только можно пожелать — точки, линии, сферы и грани.

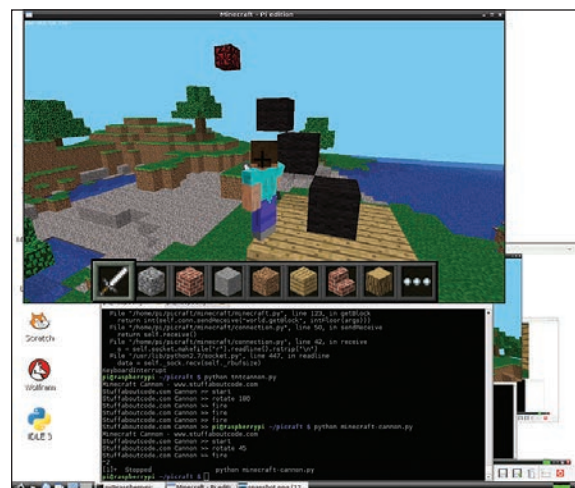
Попользуемся объектами

Вы можете поинтересоваться, зачем нужно переводить эти графические компоненты в объекты вместо того, чтобы просто разместить все функции рисования в одном модуле. Вопрос полностью оправдан — поместить все функции в один модуль вполне возможно; но внимательно посмотрев на блок класса, вы увидите, что многие методы разделяют общую переменную **self.mc**, определенную в специальном методе **__init__()**. И если вы захотели бы извлечь все эти методы в независимые функции, многим из них потребовалось бы передавать добавочный параметр **mc**.

Месяц назад ваш блоколюбивый автор повеселился, построив и разрушив дом, пока вы, верный и прилежный читатель, знакомились с API *Minecraft*, с алгоритмом Брезенхема [Bresenham] для рисования линий, состоящих из блоков, и с тем, меняя различные биты **blockType**, можно создавать прекрасные двери и горящую взрывчатку TNT. Говорите, у вас нет последнего номера **LXF**? Марш на страницу 108! А на нашей прогулке мы получим несколько не менее ценных уроков и продолжим разрушать, на сей раз с помощью управляемой пушки, код которой принадлежит Мартину О'Хэнлону [Martin O'Hanlon] (stuffaboutcode.com). Весь этот код есть на диске, и для его запуска выполните простую команду **python minecraft-cannon.py** в каталоге с кодом.

Всего в коде около 400 строк, и мы не будем рассказывать обо всем — только о самом вкусном. Данный проект можно считать отличным введением в объектно-ориентированное программирование, о котором сперва и поговорим.

На языке программистов объект — это экземпляр класса. Это определение по меньшей мере неудовлетворительное, а по большей — бессмысленное. Вместо этого представьте себе объект



➤ Стив бесечно созерцает, как рушатся обломки обсидиана на остропрометливо подставившееся дерево.

Скорая помощь

Попробуйте поэкспериментировать со скоростью и радиусом взрыва в методе **cannon.fire()** в строке 376 файла **minecraftcannon.py**.

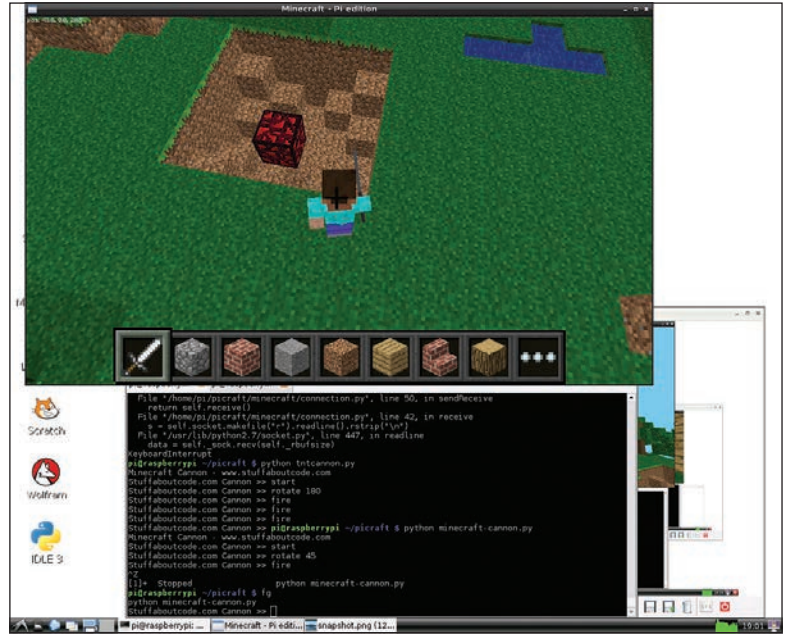
С одной переменной это не страшно, но легко представить, как ситуация может ухудшиться. Объектно-ориентированное программирование помогает нам разумно группировать переменные и функции (или данные и поведение). Вы не могли не заметить повторение ключевого слова **self** в каждом классе. Оно передается в качестве первого аргумента метода и означает, что данный метод унаследует все свойства класса — переменные **self.***, определенные в методе **__init__()**.

В тройке оставшихся классов модуля (**MinecraftBullet**, **MinecraftCannon** и **CannonCommands**) определено несколько свойств, что более чем оправдывает их выделение в классы, и они ссылаются друг на друга в логичной манере, как и полагается.

Пушкой управляет простой командный интерпретатор, предоставляемый модулем **cmd**. Этот модуль позволяет вызывать функции с аргументами, оборачивая их в «директивные функции». Вот хороший пример, когда Python превращает в детскую забаву то, что иначе было бы трудоемким и скучным. Все, что здесь нужно — класс, который наследует **cmd.Cmd** и содержит все команды, которые должен понимать интерпретатор. Команды эти определяются как функции с именами вида **do_***() — так, например, функция, описывающая команду **exit**, будет зваться **do_exit()**. Любая функция, которая возвращает значение, прервет цикл работы интерпретатора, поэтому все функции, кроме **do_exit()** и **do_EOF()** (вызываемой при нажатии Ctrl+D), ничего не возвращают. Поскольку мы унаследовались от **cmd.Cmd**, нужно вручную вызвать его метод **__init__()** для запуска интерпретатора, и здесь мы также задаем приветственное сообщение и строку приглашения. В итоге класс **CannonCommands** начинается так:

```
class CannonCommands(cmd.Cmd):
    def __init__(self):
        cmd.Cmd.__init__(self)
        self.prompt = "Stuffaboutcode.com Cannon >> "
        self.intro = "Minecraft Cannon - www.stuffaboutcode.com"
```

Сделав все это, мы получим полностью функциональную командную строку с историей, редактированием команд и даже дополнением по Tab. Более того, у нас есть даже команда **help**, которая выводит описание для функций **do_***. «Очаровательно! А где же пушка?» — спрашиваете вы, и я слышу, как ваша нетерпеливость берет верх над энтузиазмом. Чтобы пушка появилась, достаточно вызвать команду **start**, которая создает объект **mc** (помещенный в пространство имен **self**, так как доступ к нему есть из всего класса) и создает объект **MinecraftCannon** на расстоянии трех блоков от текущего положения игрока. Сама пушка довольно проста: **drawCannon()** рисует деревянную станину размером 3×3 блока, а **drawGun()** — ствол из пяти блоков темного



дерева. Две функции необходимы потому, что мы можем менять горизонтальную и вертикальную наводку ствола, для чего потребуются перерисовка пушки. Для этого, соответственно, вызываются команды интерпретатора **rotate** и **tilt**, которые в свою очередь вызывают методы **setDirection()** и **setAngle()** класса пушки. Координаты дула пушки вычисляются по точке, находящейся в центре сферы соответствующего размера, находящейся у точки запала пушки. Подробности описаны во врезке «Сферическая тригонометрия» на обороте — не читайте ее, если воспоминания о тригонометрии пробуждают фантомную боль травм юности.

При горизонтальной или вертикальной наводке пушки вызывается метод **clearGun()**, который рисует блоки из воздуха поверх ствола. После этого вычисляется новая конечная точка ствола, как описано во врезке, и ствол прорисовывается методом **mcDrawing.drawLine()**. В последней функции вызывается функция **getLine()** (самая длинная функция в модуле), представляющая собой реализацию алгоритма Брезенхема, о котором мы рассказывали в прошлом номере. (Если номер вы упустили, не волнуйтесь, а если волнуетесь, добудьте экземпляр, как описано выше.)

Теперь переходим к главному — к выстрелу. Для этого инициализируем объект **MinecraftBullet** со скоростью 1 (он перемещается со скоростью 1 блок за такт, или одну сотую секунды) и радиусом

» Стив хочет, чтобы его снаряды были чуть более разрушительными.

»

Мои первые объекты

В качестве мягкого введения в объектно-ориентированное программирование рассмотрим простое приложение — библиотеку музыки. В нем нашими объектами будут сама библиотека и наши любимые треки (будь то какой-нибудь воодушевляющий Брамс или свежие израильские психоделические гимны), и у нас будет метод добавления треков.

```
class library:
    def __init__(self):
        self.lib = []
    def add(self, trackobj):
        self.lib += [trackobj]
```

```
class track:
    def __init__(self, artist, title):
        self.artist = artist
        self.title = title
    Здесь наша библиотека использует стандартные методы работы со списками, и мы создадим ее и полным следующим образом:
    >>> mylib = library()
    >>> mylib.add(track("Tom Lehrer", "Poisoning pigeons in the park"))
    >>> mylib.add(track("Bill Bailey", "Beautiful ladies in danger"))
```

Негусто, но вы поняли идею. Можно (и нужно) добавить еще один специальный метод **__repr__()**, который возвращает способ отображения этих объектов, так как представление по умолчанию не слишком понятно:

```
>>> mylib.lib[0]
<__main__.track object at 0x7f12eaad9908>
```

И в класс трека можно добавить следующие строки:

```
def __repr__(self):
    return("<Track> %s,%s" % (self.artist, self.title))
```

В каждом случае этот метод даст вам чуть более информативное описание.

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

взрыва 3 (когда объект с чем-то сталкивается, в этом месте образуется пустая сфера с радиусом 3). Сам снаряд — просто блок раскаленного вулканического стекла-обсидиана, поэтому рисовать и стирать его нетрудно — взгляните на однострочные методы `draw()` и `clear()`. Нам нужно рассчитывать скорости в трех измерениях; для этого используется тригонометрия, похожая на ту, что используется при прорисовке ствола пушки. После этого мы заходим в цикл `while` и один раз за такт вызываем функцию `bullet.update`, которая будет возвращать `True`, пока не произойдет столкновение. Скорость в отрицательной части оси Y (из-за силы тяжести) линейно возрастает с течением времени, как можно заключить из строки 250:

```
self.yVelocity = self.yStartVelocity + self.gravity * self.ticks
```

тогда как по осям X и Z она остается постоянной, поскольку трение можно не учитывать. Скорость измеряется в блоках

за такт, и новое положение снаряда вычисляется посредством сложения каждого компонента предыдущего положения снаряда с соответствующим компонентом скорости (строка 253) и последующим округлением полученных координат до целочисленных значений (строка 258). Если снаряд движется медленно, то после округления координат он может остаться на прежнем месте; тогда перерисовывать его не нужно.

Мы проверяем, что снаряд переместился, с помощью оператора условия

```
if matchVec3(newDrawPos, self.drawPos) == False:
```

и затем выполняем перерисовку. Перерисовка включает простую проверку того, что в новом блоке, в который попал снаряд, ничего нет:

```
if self.mc.getBlock(newDrawPos.x, newDrawPos.y,
newDrawPos.z) == block.AIR:
```

Если это так, мы удаляем старый блок вулканического стекла, обновляем положение перерисовки и перерисовываем его:

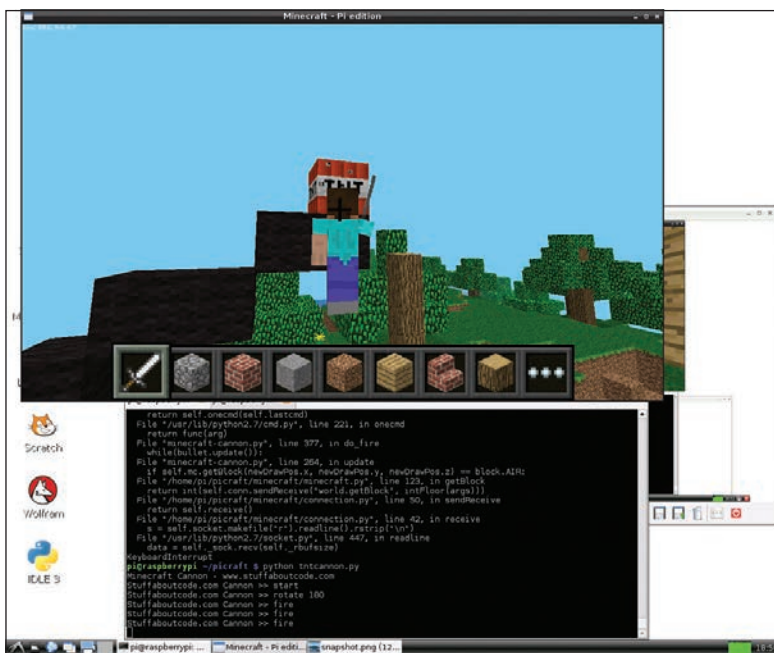
```
self.clear()
self.drawPos = minecraft.Vec3(newDrawPos.x,
newDrawPos.y, newDrawPos.z)
self.draw()
```

Если нет, мы создаем кратер и изменяем `movedBullet` на `False`, чтобы выйти из цикла обновления:

```
self.mcDrawing.drawSphere(newDrawPos, self.blastRadius,
block.AIR)
movedBullet = False
```

Взрывы помощнее

Помните, как в прошлом номере мы взрывали все с помощью цепочки из взрывчатки? Теперь мы улучшим нашу пушку, используя похожие принципы. Так как детонировать взрывчатку можно лишь ударив ее или взорвав другой блок взрывчатки рядом с ней, нам придется инициировать взрыв вручную, поэтому команда `fire` теперь будет работать иначе. В частности, сейчас мы добавим блок взрывчатки в конец ствола, и бедному Стиву останется только поджечь фитиль и убежать. Нам нужно следить за этим блоком и незамедлительно отреагировать, когда снаряд ударит по нему — для этого придется точно отслеживать время. Как раз перед взрывом блока мы подсунем еще один блок взрывчатки прямо перед ним, и т.д. — получив иллюзию движущейся



➤ Этот нужно поджечь вручную, но Стиву все равно — раз уж он безумный.

Сферическая тригонометрия

По азимутальному (горизонтальному) углу «фи» (ϕ) и углу вертикальной наводки «тэта» (θ), точка на сфере с радиусом 1, расположенной в начале координат, вычисляется по схеме, показанной на рисунке справа.

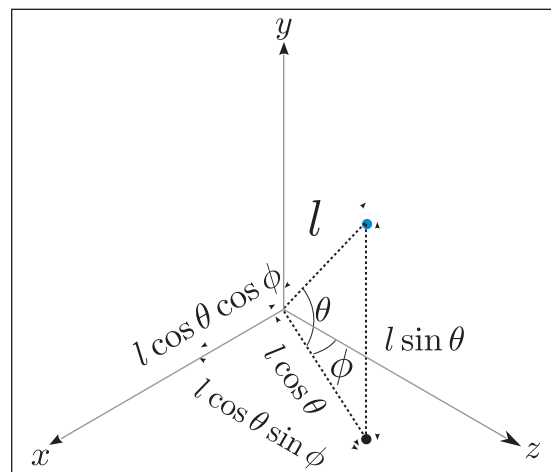
Здесь синяя точка — это точка на сфере, а черная точка — ее проекция на плоскость X–Z. Из-за того, каким образом определяются углы, координата y рассчитывается по чуть более скучной формуле, чем остальные. Мы можем обозреть это вычисление в функции `findPointOnSphere()` (строка 24):

```
def findPointOnSphere(cx, cy, cz, radius, phi,
theta):
    x = cx + radius * math.cos(math.
radians(theta)) * math.cos(math.radians(phi))
```

```
z = cz + radius * math.cos(math.
radians(theta)) * math.sin(math.radians(phi))
y = cy + radius * math.sin(math.
radians(theta))
```

Углы, являющиеся аргументами тригонометрических функций, следует преобразовать в радианы (предполагается, что команды `rotate` и `tilt` принимают значение углов в градусах). Возможно, вы помните, что в 180 градусах ровно половина радиан, и к нашему счастью, углы совершенно не зависят от осей X, Y и Z.

➤ Когда я включил в статью этот рисунок, в Башнях LXF почему-то никто меня не зауважал.



➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

и разлетающейся взрывчатки, крушащей все вокруг. Код этой части упражнения находится в файле `tntcannon.py`.

Когда Стив ударит по взрывчатке и та заискрит, метод `getBlock()` фактически определяет ее как воздух, и это удобный сигнал к подготовке следующего блока в цепочке. Мы ожидаем этого, застав дыхание, в следующем цикле, в котором `pass` — это стандартная команда Python для «ничего не делать»:

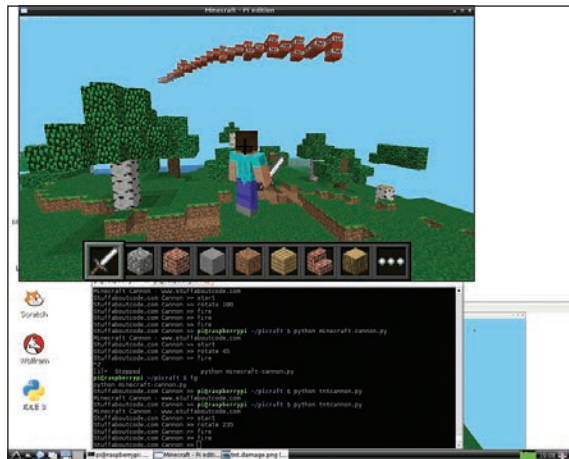
```
while self.mc.getBlock(xt, yt, zt) == 46:
    pass
```

Начиная с момента воспламенения взрывчатки, у нас есть четыре секунды, чтобы спрятаться в безопасное место, и вовремя добавить еще один блок взрывчатки. Первый взрыв вызовет цепную реакцию, и эмпирические исследования показывают, что реакция распространяется со скоростью примерно 1 блок за каждые 0,3 с, так что мы будем добавлять новую взрывчатку по этому воображаемому метроному. Траекторию потребуется немного изменить, чтобы на пути не было дубликатов (некоторое время назад мы отметили, что они могут появиться из-за округления координат — взгляните!), иначе все шоу будет испорчено.

Результат всего этого таков, что на самом деле у нас нет никакого контроля над скоростью нашего снаряда: он будет двигаться со скоростью около четырех блоков в секунду. В направлении Y мы хотим сохранить нашу модель гравитации; воспользуемся приемом из прошлой статьи и нарисуем вертикальный столбик из взрывчатки, если возникнет существенное движение по вертикали. Так по меньшей мере траектория будет более или менее точной, даже если снаряд не ускоряется, и кроме того, удлинится и сократится при вертикальном перемещении снаряда.

Метод `fire()` полностью приведен ниже:

```
def fire(self, velocity, blastRadius):
    xt, yt, zt = findPointOnSphere(self.baseOfGun.x, self.
    baseOfGun.y, self.baseOfGun.z, self.lenghtOfGun, self.direction,
    self.angle)
    #draw the TNT trigger
    self.mcDrawing.drawPoint3d(xt, yt, zt, 46, 1)
    #support so that it don't fall when hit
    self.mcDrawing.drawPoint3d(xt,yt - 1, zt, block.WOOL.id, 15)
    # wait patiently for trigger
    while self.mc.getBlock(xt, yt, zt) == 46:
        pass
    time.sleep(3.6)
    startPos = minecraft.Vec3(xt,yt,zt)
    tntBullet = MinecraftTNTBullet(self.mc,startPos,self.
    direction,self.angle,1)
    while not tntBullet.update():
        time.sleep(0.3)
```



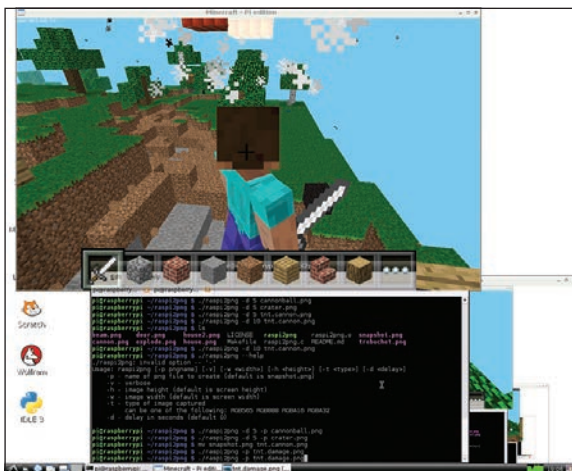
► Да, это не всегда работает верно. Тем, кто сумеет это исправить, я разрешу покрасить мой забор.

Почти так же, как и ранее в этой статье, у нас есть метод `update()` для нашего объекта, который вычислит новое положение и будет возвращать `False`, пока мы не столкнемся с чем-то. Он выглядит так:

```
def update(self):
    self.yVelocity += self.gravity
    oldPos = minecraft.Vec3(int(round(self.Posx)),int(round(self.
    Posy)),int(round(self.Posz)))
    self.Posx += self.xVelocity
    self.Posy += self.yVelocity
    self.Posz += self.zVelocity
    newPos = minecraft.Vec3(int(round(self.Posx)),int(round(self.
    Posy)),int(round(self.Posz)))
    if newPos != oldPos:
        if self.mc.getBlock(newPos.x,newPos.y,newPos.z) != block.
        AIR.id:
            return True
        if abs(newPos.x) > 128 or newPos.y > 128 or abs(newPos.z) >
        128 or newPos.y < -10:
            # off the edge of the world or deep underground
            return True
        self.mc.setBlocks(newPos.x,newPos.y,newPos.z,newPos.x,old
        Pos.y,newPos.z,46,1)
        height = newPos.y
    return False
```

Так как взрывчатка достаточно разрушительна, возможно, наш снаряд перед окончательной остановкой пересечет значительную часть пейзажа, но мы ограничиваем разрушения, заставляя метод `update()` вернуть `True`, если снаряд опустится ниже `-10`. Мы также делаем это, если снаряд покидает границы мира *Minecraft* — т.е. если любая из его координат превышает `128`. Когда огонь наконец затихнет, мы перерисовываем нашу пушку, поскольку она немного пострадала при выстреле, и затем возвращаемся в командную строку, чтобы Стив мог еще пострелять по каким-нибудь невинным блокам.

К сожалению, благодаря неизмеримым силам, возникающим при одновременном взрыве такого количества блоков взрывчатки, во всем этом есть элемент случайности. В результате блоки могут разлетаться во всех направлениях или детонировать в неподходящий момент, что приведет к прерыванию цепочки взрывов и к тому, что половина параболы из блоков взрывчатки останется висеть в воздухе. Время задержки указывается в строках `276` и `280`, и оно было выбрано довольно случайным образом; изменив его, вы, возможно, сможете улучшить ситуацию. С другой стороны, никакое оружие не совершенно, и такая ненадежность — только отражение сей неизбежной истины. Удачи вам с экспериментами, и, надеюсь, мы скоро снова встретимся с Мартином О'Хэнлоном! **LXF**



► Даже безумный Стив преклоняется перед мощью нового оружия.

Octave: Труды в математике

Терри Дьюэлл представляет Octave 3.8 и знакомит вас с жизнью численных расчетов и анализа данных.



Наш эксперт

Терри Дьюэлл — инженер-консультант, который пользуется Unix и Linux двадцать пять лет. Он обширно использует Octave в своей работе и считает, что Linux и Octave просто шикарны.

Если вы подобны мне и хотите решать численные проблемы или анализировать данные, это делается массой способов. Например, можно писать программы на любом из известных языков программирования. Именно для этой цели был создан и обширно используется Fortran. Python, один из современных языков, породил для этой задачи несколько утилит и приложений, таких как *NumPy* и *SciPy*. Можно было бы воспользоваться MATLAB, коммерческим языком и средой численных расчетов высокого уровня, но это недешево... или GNU Octave — открытым языком высокого уровня, основное назначение которого — численные расчеты. GNU Octave очень похож на MATLAB, да и большая часть его кода совместима с MATLAB: разработчики рассматривают несовместимость с MATLAB как ошибку. Octave обладает возможностями решения линейных и нелинейных численных задач и обширными возможностями манипулирования данными и их визуализацией.

С выходом Octave 3.8.0 в декабре 2013 года у языка появился графический интерфейс пользователя, который пока считается экспериментальным. Преимущество Octave перед другими языками, такими как Fortran, C и C++, в том, что большая часть функций, которые вам могут понадобиться, уже включены или готовы к установке, и это интерпретируемая среда, благодаря чему вы сосредотачиваетесь на том, что вы делаете, а не на том, как это делается.

Octave 3.8 уже должна быть в репозиториях вашего дистрибутива Linux, а если еще нет, то появится очень скоро. Если вам нужно собрать программу из исходников, то исходники вместе с инструкциями по сборке есть на DVD **LXF182** (для загрузки свежей версии также см. ссылки во врезке «Ресурсы» на стр. 89).

Octave обычно запускается в терминале, и мы начнем именно с него, так как познакомиться с основами определенно стоит. Создайте рабочий каталог своего проекта и откройте терминал в этом каталоге. Теперь наберите в командной строке **octave** для запуска Octave. Octave представится, затем откроется командная строка 'octave:1>'.
 В зависимости от вашей деятельности в Octave, может потребоваться установка нескольких дополнительных пакетов Octave-Forge (см. врезку «Инструментарий Octave» на стр. 91 внизу).

Векторы и матрицы

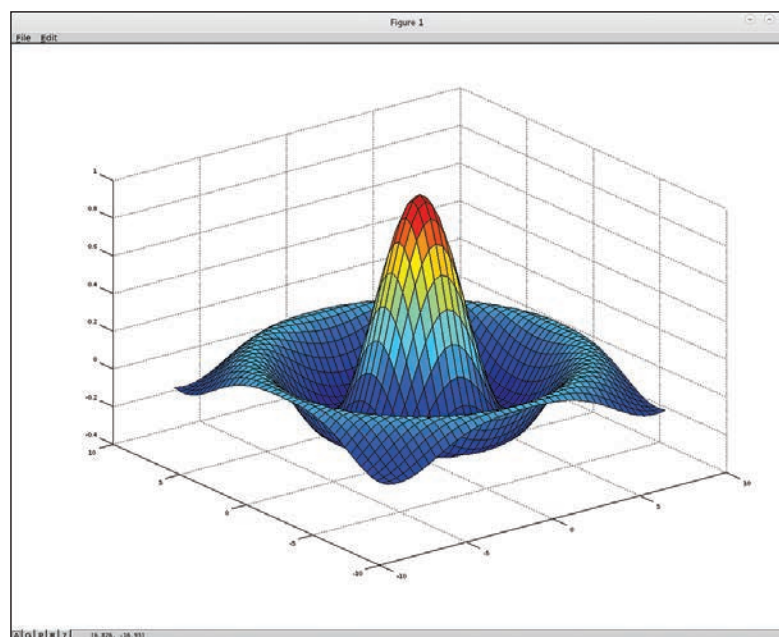
Перейдем же прямо к делу и войдем в его вкус. Простейший способ воспользоваться Octave — в качестве калькулятора, представив сложение, вычитание, умножение, деление и возведение в степень операторами +, -, *, /, ^ соответственно и вводя выражения с командной строки.

```
octave:5> (7 * 2^(-1)) + 1.4
ans = 4.9000
octave:6> x = (7 * 2^(-1)) + 1.4;
octave:7> y = 22.0;
octave:8> x+y
ans = 26.900
```

Octave присваивает результат переменной ans, но можно его присвоить в произвольной переменной. Обратите внимание на точку с запятой (;), завершающую первые две строки. Если завершить ею выражение, то результат не будет выводиться на экран, что очень полезно, если этим результатом может быть большой массив данных. Попробуйте ввести большие числа (100,100), и поймете, что я имею в виду.

В Octave есть все обыкновенные математические операторы (например, asin, cos, eps, format, inf, max, min, pi, sin, tan и т.д.). Чтобы узнать детали, воспользуйтесь функцией help, скажем, **help log**.

Чтобы воспользоваться Octave с максимальной полнотой, стоит освежить в голове свои знания по матричной алгебре, так как основным типом данных здесь является матрица. В Octave матрица представляется в виде массива чисел в квадратных скобках []; строки разделяются точкой с запятой (;), а столбцы — запятой (,) или пробелом. Вектор представляет собой особый вариант матрицы. Вектор-столбец — это матрица размерностью n×1, а вектор-строка — матрица размерностью 1×n. Для преобразования вектора-строки в вектор-столбец и наоборот можно воспользоваться знаком транспонирования ('). Символ % используется для отделения комментариев от кода. Итак, для начала создадим несколько матриц.



➤ Сомbrero Octave, оно же — «мексиканская шляпа». В командной строке наберите **sombbrero** — и вуаля! Нажмите на кнопку ?, чтобы узнать, как изменить графическое представление.

```
octave:11> x = [1,2;3,4] % создает матрицу 2 x 2
octave:12> y = [1 2 3 4] % создает вектор-строку 1 x 4
octave:13> y = [1 2 3 4]' % создает вектор-столбец 4 x 1
octave:14> y = [1;2;3;4] % создает вектор-столбец 4 x 1
Для создания вектора-строки с равномерно увеличивающимися
элементами можно воспользоваться конструкцией <пер-
вый элемент><итерация><последний элемент>, где значе-
ние параметра <итерация> по умолчанию равно 1, и его можно
опустить.
octave:15> y = 1:4
y = 1 2 3 4
octave:16> y = 1:0.5:2
y = 1.0000 1.5000 2.0000
```

Нам часто бывают нужны вектора и матрицы с равномерным набором чисел, и создавать их вручную, особенно если они большие, может быть очень утомительно. К счастью, в Octave есть несколько простых команд вам в помощь. Команда **linspace()** позволяет указать первое, последнее значение и количество элементов: так, для создания вектора из 20 элементов со значениями между 1 и 12 мы воспользуемся командой **x = linspace(1,12,20)**. В Octave также есть похожая функция **logspace()**.

Скрипты и функции

По умолчанию Octave выполняет операции над массивами поэлементно, поэтому при вызове **exp(v)** мы получим экспоненту от каждого элемента **v**. Для выбора *i*-го элемента массива он указывается после имени массива в круглых скобках. Для извлечения элементов массива можно опять же воспользоваться диапазоном, **v(4:length(v))** — то же самое, что и **v(4:end)**:

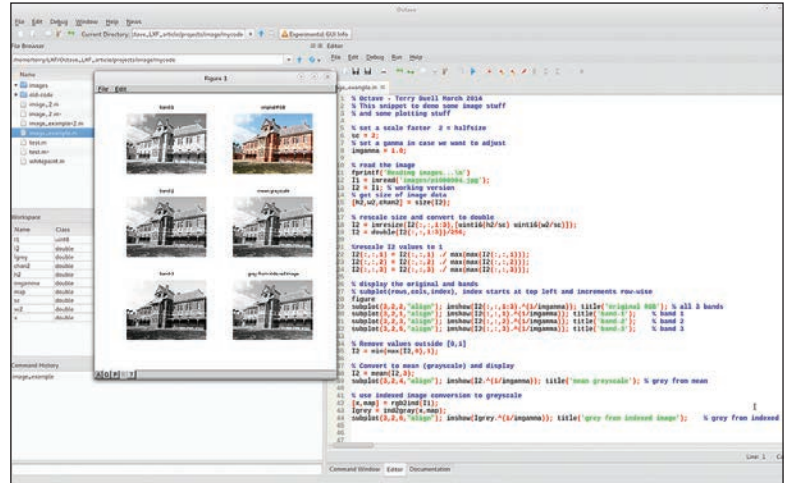
```
octave:17> v = [8,3,4,5,0,6,2]; % определение v
octave:18> v([1,3,7]) % извлечь 1-й, 2-й и 7-й элементы v
ans = 8 4 2
```

Для извлечения элементов из матрицы используется синтаксис (строка, столбец), а целая строка или столбец обозначаются двоеточием (:).

```
octave:20> b = [1,2;3,4] % определение матрицы b
b = 1 2
    3 4
octave:21> b(:,1) % извлечь 1-й столбец
ans = 1
    3
octave:23> b(1,:) % извлечь 1-ю строку
ans = 1 2
```

В Octave арифметические действия с матрицами обозначаются простыми операторами. Чтобы умножить матрицу на скаляр, мы набираем **2 * A**, для перемножения матриц — **A * B**, а для получения произведения матриц **A'A** (транспонированная версия **A * A**) — **A' * A**.

ОК, но как с помощью всего этого решать реальные задачи, спросите вы? Хороший пример — решение системы уравнений. Для решения набора линейных уравнений **AX = b** воспользуемся



➤ В версии 3.8 обновлен графический интерфейс. Код скрипта отображается в окне редактора, перекрывающем графическое окно с результатами. Как его открыть, вы узнаете на стр. 98.

оператором деления слева, ****, **A \ b**. Вот пример, показывающий, как это просто. Наша система состоит из трех уравнений: **3x-2y+z=10**; **x+3y+2z=19** и **2x-y-z=-1**:

```
octave:28> A = [3,-2,1;1,3,2;-1,-1] % матрица коэффициентов
A = 3 -2 1
    1 3 2
    -1 -1 -1
octave:29> b = [10,19,-1] % правые части как вектор-строка
b = 10 19 -1
octave:30> (A \ b)' % решаем, используя b как вектор-столбец,
результат преобразуем в вектор-строку
ans = 3.0000 2.0000 5.0000
```

Эти три значения и есть *x*, *y* и *z*, удовлетворяющие всем трем уравнениям, и все это было бы ничуть не сложнее, если бы уравнений было больше трех.

В Octave есть встроенные функции для математических операций и многие специализированные функции. Пример — одна из встроенных функций для решения нелинейных дифференциальных уравнений вида **dx/dt = f(x,t)**, **x(t=0) = x0**. В руководстве по Octave (см. врезку «Ресурсы» внизу) описан отличный пример, который мы воспроизводим здесь. Вам нужно определить свою функцию, что можно сделать прямо в терминале, и Octave откроет другую строку приглашения, которая будет отображаться до тех пор, пока вы не завершите ввод выражением **endfunction**.

```
octave:31> function xdot = f(x,t)
> r=0.25;
> k=1.4;
> a=1.5;
> b=0.16;
```

Скорая помощь



Чтобы добавить ярлык для графического интерфейса Octave, запустите команду **octave --force-gui** и поставьте галочку Launch in Terminal (Запустить в терминале). После этого откройте **home**, затем **Desktop**, бросьте свой файл Octave в редактор и измените строку **Icon=** на **Icon=/usr/share/octave/3.8.0/image-lib/octavelogo.png**. Примечание: этот путь указан для Fedora.

Ресурсы

Если вы хотите собрать программу из исходников, последнюю версию можно загрузить с <http://ftp.gnu.org/gnu/octave>. На данный момент это Octave 3.8.1, релиз с исправлениями ошибок. Онлайн-документация доступна на <http://bit.ly/GNUOctave>. Еще несколько полезных документов: лекции Андреаса Стэйхела [Andreas Stahel] (<http://bit.ly/OctaveatBFH>), руководство по Octave Джона Итона [John Eaton] (<http://bit.ly/OctaveMan>) и Revision Pack Фотиоса

Казолиса [Fotios Kasolis] (<http://bit.ly/OctaveRev-Pack>). И хотя эти руководства сходны по материалу, примеры в них разные, и они могут очень пригодиться новичкам. Также есть перечень команд Octave, который немного устарел, но тем не менее очень удобен (<http://bit.ly/OctaveQuickRef>). Официальная страница Octave — <http://www.gnu.org/software/octave/support.html>, и на ней можно найти всяческую

полезную информацию, в том числе — как подписаться на список рассылки. Еще один удобный источник кода, который я недавно нашел — Сообщество Джона Буркардта [John Burkardt] в Университете Флориды (<http://bit.ly/MATLABSource>). Большая часть кода MATLAB будет работать в Octave с очень незначительными изменениями, но инструментарий MATLAB очень богат, и многих его функций в Octave может не быть.

Скорая помощь



Хороший способ познакомиться с приемами написания кода на Octave — заглядывать в файлы `.m`. Они сохраняются в виде текстовых файлов, и, например, с помощью команды `help fred` можно узнать, где находится файл `fred.m`, и открыть его в текстовом редакторе.

```
> c=0.9;
> d=0.8;
> xdot(1)=r*x(1)*(1-x(1)/k)-a*x(1)*x(2)/(1+b*x(1));
> xdot(2)=c*a*x(1)*x(2)/(1+b*x(1))-d*x(2);
> endfunction
octave:32> x0=[1;2]; % укажем начальные условия;
octave:33> t=linspace(0,50,200); % укажем значения по оси
времени (t) как вектор-столбец;
octave:34> x=lsode('f',x0,t); % интегрируем систему диффе-
ренциальных уравнений.
octave:35> plot(t,x)
```

Итак, что же здесь происходит? Функция `lsode()` вызывается с нашей функцией `f()`, начальными условиями и массивом времен, для которых мы хотим найти решение. `lsode()` вызывает `f()` для получения решения для каждого значения `t`. Мы выводим решение командой `plot()`, и для этого Octave откроет окно отображения. При следующем вызове `plot()` содержимое окна отображения данных будет перерисовано. Чтобы каждый график выводился в новом окне, предварите вызов `plot()` цифрой.

Хорошие графические возможности — неотъемлемая часть пакета анализа данных, так как благодаря им мы можем увидеть все это в виде, более наглядном по сравнению с числовым представлением. В Octave 3.8 для реализации графики используется `FLTK`, хотя можно выбрать и `gnuplot`, вызовом `graphics_toolkit('gnuplot')`; и переключиться обратно по `graphics_toolkit('fltk')`; В случае с `FLTK` трехмерные данные можно масштабировать и поворачивать, что помогает с визуализацией.

Функцию `f()`, которую мы определили в терминале выше, можно было записать в файл, например, `f.m`, и вызвать из скрипта, скажем, `solvecf.m`, представляющего собой простой текстовый файл, который можно создать в любом текстовом редакторе. Лучше взять редактор, который понимает синтаксис Octave, например, `gedit`, установив режим подсветки синтаксиса (View-Highlight Mode) в "Octave". Наш скрипт `solvecf.m` может выглядеть примерно так:

```
x0 = [1;2];
t = linspace(0,50,200);
```

```
x = lsode("f",x0,t);
plot(t,x)
pause
```

и если сохранить `solvecf.m` в том же каталоге, что и `f.m`, и открыть терминал из этого каталога, то, набрав `octave solvecf.m`, можно запустить Octave и скрипт. Обратите внимание, что последним оператором в `solvecf.m` я проставил `pause`. Благодаря ему скрипт перед завершением приостанавливается, и окно с данными остается открытым. Для закрытия этого окна нажмите Enter. Без команды `pause` окно с данными закроется сразу же.

Альтернативный подход, который нравится мне чуть больше — избавиться от паузы, запустить Octave и набрать `solvecf` без расширения `.m` в строке приглашения для запуска скрипта.

Наш код из файла `f.m` представляет собой самую базовую форму функции. Обычно некоторые данные передаются функции в виде переменных, которые могут быть скалярами, векторами или матрицами. Кроме того, функция возвращает некоторые значения, поэтому более общий вид функции будет выглядеть таким образом:

```
function return-var = имя (список аргументов)
тело функции
endfunction
```

`return-var` — это переменная, которая будет содержать значение или значения, возвращаемые функцией. Эти значения нужно определить до завершения тела функции. В качестве возвращаемых значений может выступать вектор `[x,y,z...]`. Octave также поддерживает списки аргументов и списки возвращаемых значений переменной длины, см. `help varargin`. Чтобы писать более сложные скрипты и функции, понадобятся другие операторы, и в Octave есть операторы `if`, `else`, `elseif`, `switch`, `case`, `while`, `do-until`, `for`, `break`, `continue`, а также стандартные операторы сравнения и булевы операторы, `<`, `<=`, `==`, `>=`, `>`, `!=`, `&&`, `||`.

Типичное использование некоторых из этих операторов показано в следующих примерах кода.

```
for i = 1:10
% здесь выполняются действия
endfor
yesno = "yes"
switch yesno
case {"Yes" "yes" "YES" "y" "Y"}
value = 1;
case {"No" "no" "NO" "n" "N"}
value = 0;
otherwise
error («неправильная величина»);
endswitch
```

Векторизованный код Octave позволит существенно (от 10 до 100 раз) увеличить скорость выполнения кода в большинстве случаев. Векторизация — это технология в программировании, при которой вместо обычных элементов в циклах выполняются операции с целыми векторами. Концепция векторизации не уникальна для Octave, но она важна, так как Octave — матрично-ориентированный язык.

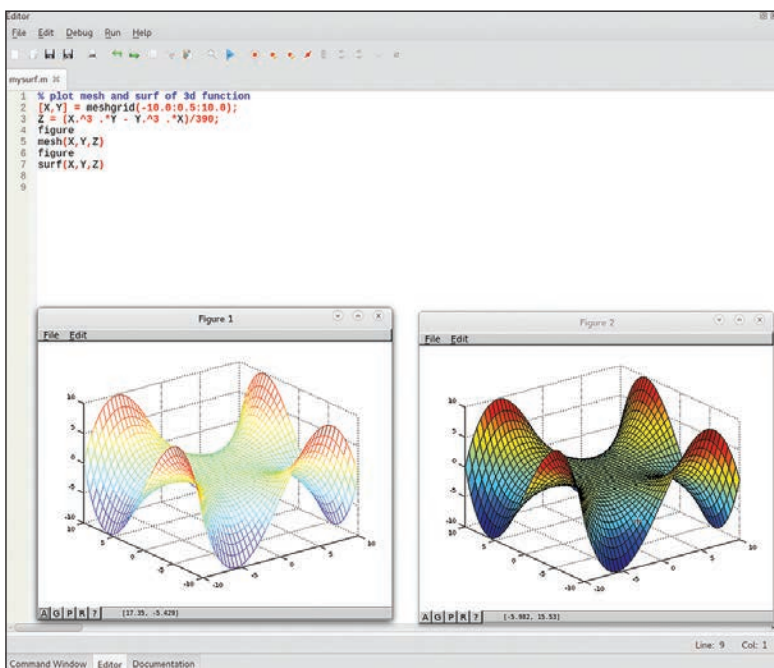
Цель векторизации в том, чтобы писать код, избегая циклов и используя операции целиком с массивами. Взгляните на этот простой пример из документации Octave:

```
for i = 1:n
for j = 1:m
c(i,j) = a(i,j) + b(i,j);
endfor
endfor
```

по сравнению с гораздо более простым

```
c = a + b;
```

в котором мы достигаем того же результата, что и в первом случае, но значительно быстрее. Сложение двух массивов 100×100



➤ Команда `meshgrid` используется для создания матрицы `[X,Y]` с использованием диапазонов, затем мы вычисляем `Z = f(X,Y)`. Для построения сетки используется команда `mesh`, а для сплошной поверхности — `surf`. С помощью этих двух команд можно представить любые трехмерные данные на регулярной сетке.

с использованием циклов занимает 0,123 с. Векторизованный код выполняется за 0,0000369 с.

Векторизация — предпочтительная технология для исключения циклов и ускорения кода, но заменить все циклы не всегда возможно. В этих случаях можно попробовать экспериментальный компилятор Octave — Just-In-Time (JIT) [англ. «точно вовремя»].

Octave — интерпретируемый язык, и каждый раз при появлении цикла Octave должен разбирать выражения в теле цикла перед их выполнением. С компилятором JIT это делается всего один раз, когда тело цикла транслируется в другой язык. Компилятор JIT — это новая возможность Octave, и на данный момент не все выражения в Octave можно ускорить с его помощью.

Для включения и выключения компилятора используется функция `jit_enable`. Функция `jit_startcnt` задает порог ускорения. Будут ускорены циклы с количеством повторений больше `jit_startcnt`. Также можно написать функции Octave на C, C++ и Fortran, скомпилировать их и вызывать из функций и скриптов Octave.

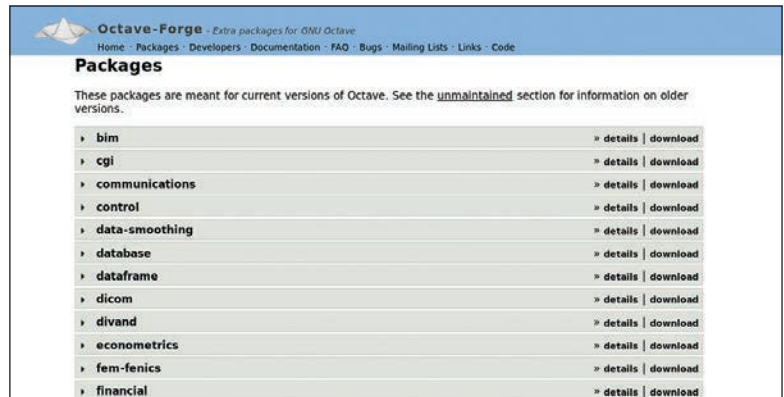
Команда `mkoctfile [-options] file ...` компилирует файл исходного кода в файл OCT. Код MATLAB, написанный для компиляции в MEX-файл MATLAB, часто можно скомпилировать в Octave, внеся лишь небольшие изменения в заголовочные файлы. Для компиляции файлов OCT нужно установить пакет `Octave-devel`.

Графический интерфейс Octave

GUI Octave, впервые появившийся в версии 3.8.0, все еще имеет экспериментальный статус, но работает очень хорошо. По умолчанию Octave по-прежнему запускается в терминале. Для запуска Octave в графическом режиме в терминале наберите `octave --force-gui`.

В нашем примере (см. рис. на стр. 89) показан новый GUI с запущенным редактором. Слева находится окно File Browser [файловый браузер], где можно найти сохраненные файлы скриптов и функций. Под ним находится окно Workspace [рабочая область] с информацией обо всех переменных рабочей области, а под ним — окно Command History [история команд]. Справа вы видите окно Editor [редактор] с файлом скрипта `image_example.m`. Там можно редактировать, сохранять и запускать скрипты, а также отлаживать их, устанавливая точки прерывания. Чтобы открыть файл в редакторе, дважды щелкните на нем в браузере файлов.


Эту правую часть окна можно занять окном команд (Command Window), и по существу получится то же самое, что и запустить Octave в традиционном режиме, или занять ее окном с пользовательской документацией (Documentation). Для переключения окон просто выбирайте соответствующие вкладки в нижней части окна. Поверх этого окна находится окно, созданное путем запуска скрипта `image_example.m`. Для очистки окон рабочей области, истории команд и окна команд можно выбрать соответствующие пункты меню Edit [Изменить] в верхнем меню.



Мы воспользуемся графическим интерфейсом для просмотра нескольких примеров. Код скрипта `image_example.m` (на DVD) позволяет оценить, как легко работать с изображениями.

В этом примере изображение считывается в массив (`imread`), масштабируется (`imresize`), преобразуется в оттенки серого двумя способами (`Mean` и `rgb2ind, ind2gray`), отображаются отдельные каналы изображения в форматах RGB и оттенках серого, после чего все эти результаты отображаются в одном окне с помощью функции `subplot`. Чтобы запустить этот скрипт для своего изображения, просто измените путь к изображению. Можете поиграть с разными командами и посмотреть, как это повлияет на результат. Для сохранения графика воспользуйтесь командой `print ("--deps", "foo. Eps")`, которая сохранит текущий график в файле формата EPS. См. `help print`.

Скрипт `run_hrv.m` (тоже помещен на DVD) вычисляет реакцию человека на вибрацию (Human Response to Vibration, HRV) по входному сигналу ускорения. Для новичка это может звучать сложновато, да так оно и есть, но нам интересны здесь лишь операции с данными. Данные HRV вычисляются во внешней функции, которая сюда не включена, и сохраняются в массиве ячеек, который затем записывается в CSV-файл с помощью метода `cell2csv` пакета `io`. Во время обработки вычисляется амплитудная спектральная плотность входного ускорения с помощью функции `pwelch` из пакета `signal`. С помощью функции `plot` результаты отображаются в отдельных окнах. В скрипте `run_hrv.m` обратите внимание на то, как первая запись массива ячеек `datavals` заполняется подходящими заголовками, затем вычисленные значения записываются в цикл, в котором поочередно обрабатываются все входные файлы. Сохраненный файл CSV можно загрузить в редактор электронных таблиц. В этом примере вы видите, как легко экспортировать результаты вычислений, и CSV — лишь один из форматов, которыми можно воспользоваться.

Чтобы узнать, чем еще Octave может вам помочь, взгляните на документацию во врезке «Ресурсы» (стр. 89) и на пакеты, доступные в репозитории Octave-Forge. 

➤ Пакеты *Octave-Forge*. Нажав на “details [подробности]”, вы получите описание всех функций пакета, а нажав на “download [загрузить]” — загрузите архив `gzip`.



Все специализированные функции удобно хранить в одном месте, например, `~/MyOctave`. Чтобы эти функции стали видимы в Octave, просто добавьте `addpath ("~/MyOctave")` в начале сеанса Octave. Этот путь можно сохранить в файл `~/octaverc`, если добавить к команде `addpath` еще `savepath`.

Инструментарии Octave

Дополнительные пакеты, или, на языке MATLAB, инструментари, для Octave доступны в *Octave-Forge* на <http://octave.sourceforge.net>. Во всем *Octave-Forge* есть 60 специализированных пакетов и три способа установки пакетов из них. Вы можете загрузить пакет с сайта и установить его командой `pkg install`, например, `pkg install pathio/image-2.2.1.tar.gz`. Может оказаться, что у этого пакета есть зависимости, которые нужно установить до него. Эти зависимости

должны быть перечислены на странице пакета *Octave-Forge*.

Второй способ — воспользоваться командой `pkg install --forge image`, которая установит пакеты из репозитория *Octave-Forge*. Для этой команды необходимы интернет-соединение и библиотека `cURL`.

Третий метод — установить пакеты из репозитория своего дистрибутива Linux, если они доступны.

Не все пакеты *Octave-Forge* загружаются автоматически, поэтому установка пакета не означает, что у Octave появится доступ к функциям пакета.

Команда `pkg list` в командной строке покажет все установленные пакеты. У загруженных пакетов в названии будет показана *. Чтобы принудительно загружать пакеты при загрузке системы, добавьте команды `pkg load` (например, `pkg load image`) в файл `~/octaverc`.

Erlang: Тестируем!

Андрей Ушаков возобновляет серию о проверке правильности программ, написанных на языке Erlang.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

В прошлый раз (LXF182) мы увидели, что процесс тестирования программных продуктов начинается еще с их разработки, когда программисты сразу же пишут тесты (модульные, функциональные, интеграционные и т. д.). Сегодня мы продолжим разговор и увидим, как выглядят тесты и как они пишутся.

Тестирование обычно проходит по следующему сценарию (независимо от того, ручное оно или автоматизированное): на «вход» подаются некоторые исходные данные, после чего на «выходе» получается некоторый результат. После этого полученные результирующие данные сравниваются с некоторыми заданными (эталонными) данными, хотя бывают такие тесты, в которых достаточно убедиться, что на «выходе» получены хоть какие-нибудь данные и не произошло прекращения работы из-за ошибки. Если полученный результат совпадает с эталоном, тест считается пройденным. Иначе считается, что мы тест не прошли; при автоматизированном тестировании об этом обычно уведомляют при помощи генерации некоторой специальной ошибки. Понимания этих принципов вполне достаточно, чтобы написать свою библиотеку для автоматизированного тестирования кода, но в большинстве случаев это будет изобретением велосипеда, т. к. для большинства платформ и языков предусмотрены общепринятые библиотеки (а то и несколько, как, например, библиотеки CppTest и GoogleTest для написания тестов на языке C++). Все эти библиотеки применяют следующий подход: в автоматизированных тестах расставлены специальные проверки. Если проверка выполнена успешно, то ничего не происходит. Если же проверка не выполнена, то генерируется специальная ошибка, после чего выполнение теста прекращается и тест считается не пройденным. Точно так же считается, что тест не пройден, если во время его выполнения сгенерируется какая-либо ошибка времени выполнения в исходном коде или в коде самого теста.

Давайте на простом примере рассмотрим, как выглядит подобный подход и подобные тесты. В качестве примера возьмем простой тест, написанный на языке C# с использованием библиотеки *NUnit* (операторы импорта соответствующих пространств имен мы здесь не приводим):

```
[TestFixture]
public class SimpleTestSet
{
[Test]
public void SimpleTest()
{
Assert.True(2 > 1);
}
```

```
Assert.False(2 < 1);
Assert.AreEqual(44, 40 + 4);
Assert.Throws<InvalidOperationException>(() => {throw new
InvalidOperationException();});
Double value = 0;
Assert.DoesNotThrow(() => value = Math.Cos(0));
Assert.AreEqual(1.0, value, Double.Epsilon);
}
}
```

В этом примере класс **SimpleTestSet** определяет набор тестов; для этого он помечен атрибутом **[TestFixture]**. Обычно к набору тестов относят тестовые сценарии для проверки одной и той же функциональности (класса, подсистемы и т. п.). В нашем примере набор тестов состоит ровно из одного теста — метода **SimpleTest**; этот метод помечен атрибутом **[Test]**. Тест состоит из 6 выражений проверки: все эти проверки определяются в классе **Assert** из библиотеки *NUnit* (в соответствии с идеологией *xUnit* эти проверки называются «ассертами»). Выражение **Assert.True** служит для проверки переданного выражения на истинность. Выражение **Assert.False** проверяет переданное выражение на ложь. Выражение **Assert.AreEqual** проверяет равенство некоторого выражения эталонному значению (которое, конечно, также может быть выражением). Выражение **Assert.Throws** служит для проверки того, что во время выполнения некоторого выражения сгенерирована ошибка («отброшено исключение», в терминологии языка C#) определенного типа. Выражение **Assert.DoesNotThrow** служит для проверки того, что во время выполнения некоторого выражения не сгенерировано никакой ошибки. И, наконец, последнее выражение **Assert.AreEqual** позволяет проверить на равенство два действительных числа с заданной точностью.

Может возникнуть вопрос: а для чего проверять возникновение ошибки при выполнении некоторого выражения? Ответ состоит в том, что генерация ошибки — это один из вариантов того, что может вернуться при вычислении выражения (помимо значения этого выражения, если оно есть). Иными словами, возникновение ошибок определенного типа при определенных условиях является контрактом данного выражения, точно таким же, как и возврат некоторого значения при заданных входных параметрах. И, очевидно, для более полного покрытия тестами нашего кода мы должны проверять и эти случаи.

А теперь перейдем непосредственно к языку Erlang и к написанию тестов на нем. Для этого воспользуемся библиотекой *EUnit*, доступной «из коробки» при установке среды выполнения Erlang. Первый шаг, который необходимо сделать при использовании библиотеки *EUnit* — это подключить ее. Делается это при помощи директивы **include_lib**; она должна располагаться после определения модуля (директивы **module**), но до первого определения какой-либо функции. Вот пример использования директивы **include_lib**:

```
-include_lib(<eunit/include/eunit.hrl>).
```

Подключение библиотеки *EUnit* (при помощи директивы **include_lib**) приводит к следующим эффектам:

» Автоматически создается экспортируемая функция **test/0** в текущем модуле (из модуля, в котором мы использовали директиву **include_lib**), и эту функцию можно использовать для запуска всех

Установка EUnit из репозитория

При работе со многими дистрибутивами операционной системы Linux появляется возможность устанавливать программное обеспечение из репозитория пакетов Linux. Язык и среду выполнения Erlang тоже можно установить из таких

репозитория. Однако при установке языка и среды выполнения Erlang следует обратить внимание на следующее: в некоторых версиях Linux (например, в Linux Mint 16) библиотека *EUnit* устанавливается как отдельный пакет.

Значения выражений в макросах

В языке Erlang все является выражением с некоторым значением. Это же справедливо и касательно макросов из библиотеки *EUnit*. Большинство макросов предназначены либо просто для некоторой проверки, либо для вывода информации на консоль. К таким макросам относятся следующие: **assert**, **assertNot**, **assertEqual**, **assertNotEqual**, **assertMatch**, **assertNotMatch**, **assertError**, **assertExit**, **assertThrow**,

assertException, **assertNotException**, **assertCmd**, **assertCmdStatus**, **assertCmdOutput**, **debugHere**, **debugMsg** и **debugFmt**. Возвращаемое значение таких макросов обычно не важно, но тем не менее полезно будет знать, что эти макросы возвращают атом **ok** (либо генерируют ошибку, когда в макросах проверки проверка не проходит). Макрос **cmd** выполняет команду, переданную ему в качестве

аргумента, и возвращает вывод этой команды в виде одной строки (при этом вывод нормализуется: символом разрыва строки становится одиночный символ LF). Макросы **debugVal** и **debugTime** вычисляют выражение (переданное в качестве аргумента), выводят некоторую информацию на консоль, связанную с этим выражением, и возвращают значение выражения.

тестов, определенных в текущем модуле. Функция **test/0** не будет автоматически создана в текущем модуле, если такая функция в нем уже есть или если запуск тестов отключен.

» Все функции вида **XXX_test/0** и **XXX_test_/0** (здесь **XXX** — любое разрешенное имя функции) автоматически экспортируются из текущего модуля (из модуля, в котором мы использовали директиву **include_lib**). Автоматического экспорта подобных функций не будет, если отключен запуск тестов и/или автоматический экспорт.

» Можно использовать ряд макросов для написания тестов (о них мы поговорим ниже).

Следующий шаг, который следует сделать при использовании библиотеки *EUnit* — это определить методы, содержащие интересующие нас тесты. Имена таких методов с тестами должны иметь следующий вид: **XXX_test/0**, где **XXX** — любое разрешенное имя функции (в данном случае выступающее как префикс). Если во время выполнения такого метода не генерируется ошибки, то считается, что тест выполнен успешно. В противном случае считается, что выполнение теста провалилось. Если тест не закончит свое выполнение в течение некоторого времени (например, заикнется), будет считаться, что выполнение теста также провалилось. Как уже говорилось выше, внутри теста мы делаем некоторые действия и обычно проверяем полученные результаты. Если результаты не совпадают с эталонными данными, то в таком случае можно сгенерировать ошибку. Можно также воспользоваться операцией соответствия шаблону; точнее, тем, что полученных данные не соответствуют шаблону — в этом случае ошибка генерируется автоматически. Но библиотека *EUnit* предлагает более понятную альтернативу подобным инструкциям: макросы **assertXXX** (эти макросы напоминают класс **Assert** и его методы из библиотеки *NUnit*, которые мы показали выше). Принцип работы этих макросов следующий: они проверяют некоторое условие (зависящее от вида макроса), и если результат проверки условия будет ложным, генерируется некое исключение.

Макросы **assertXXX** являются «сердцем» тестов, написанных при помощи библиотеки *EUnit*: практически в каждом таком тесте будет, по крайней мере, один из макросов **assertXXX**. Так что с этими макросами необходимо познакомиться поближе. Простейшим макросом является **assert(BoolExpr)**: он проверяет выражение **BoolExpr** на истинность. Макрос **assertEqual(Expect, Expr)** сравнивает значение выражения **Expr** с ожидаемым значением **Expect**. Макрос **assertMatch(GuardedPattern, Expr)** сравнивает значение выражения **Expr** с заданным шаблоном **GuardedPattern**, который может содержать выражение охраны (отделяемое от тела шаблона при помощи ключевого слова **when**) как с одной, так и с несколькими проверками. Когда проверок в выражении охраны несколько, только символ **;** будет допустимым разделителем между ними. Макросы **assertError(TermPattern,**

Expr), **assertExit(TermPattern, Expr)** и **assertThrow(TermPattern, Expr)** проверяют, что при вычислении выражения **Expr** сгенерирована ошибка, соответствующая шаблону **TermPattern**. Макрос **assertError** ожидает, что ошибка будет класса **error** (т.е. сгенерирована функциями **error/1,2**). Макрос **assertExit** ожидает, что ошибка будет класса **exit** (т.е. сгенерированная функциями **exit/1,2**). Наконец, макрос **assertThrow** ожидает, что ошибка будет класса **throw** (т.е. сгенерирована функцией **throw/1**). Вместо этих трех макросов можно использовать более общий макрос **assertException(ClassPattern, TermPattern, Expr)**. Он ожидает, что при вычислении выражения **Expr** сгенерируется ошибка класса **ClassPattern** (**error**, **exit** или **throw**), соответствующая шаблону **TermPattern**. Для макросов **assert**, **assertEqual**, **assertMatch** и **assertException** определены парные макросы **assertNot**, **assertNotEqual**, **assertNotMatch** и **assertNotException** соответственно — для них проверка условия будет успешна в том случае, когда результат проверки условия для исходного макроса будет ложным. Для макросов **assertError**, **assertExit** и **assertThrow** парных макросов нет. Это все макросы, которые нам доступны для проверки результата вычисления некоторого выражения (мы помним, что в языке Erlang все является выражением).

По поводу этих макросов отметим следующее. Во-первых, макросы **assertNotEqual**, **assertNotException** и **assertNotMatch** не документированы; об их наличии можно узнать, посмотрев исходный код библиотеки *EUnit* (из файла **eunit.hrl**). Так что по поводу этих макросов может возникнуть вполне логичный вопрос: стоит ли их использовать? Однозначный ответ, конечно, тут дать невозможно, но можно посоветовать следующее: наибольшая опасность, которая может возникнуть при использовании этих макросов — что авторы библиотеки *EUnit* как-то изменят их интерфейс или удалят. Но если посмотреть на эти макросы внимательно, станет понятно, что вероятность каких-либо изменений в них мала. Во-вторых, макрос **assertNotException** проверяет, что во время вычисления выражения не сгенерирована ошибка определенного класса, соответствующая некому шаблону. Проверить, что во время вычисления выражения не сгенерировано никакой ошибки, несколько нетривиально; следующий пример показывает, как можно осуществить подобную проверку:

```
?assertNotException(_, _, error(some_error_reason)).
```

К сожалению, при использовании данного подхода есть одна проблема: компилятор выдает предупреждение на данную проверку (и это предупреждение никак не обойти). Если использовать опцию трактовки предупреждений как ошибок при компиляции, то тест с данной проверкой нельзя будет скомпилировать. Чтобы обойти эту проблему, можно самому написать макрос, более тривиальный в использовании: он будет проверять, что вычисление выражения прошло без ошибок (без генерации каких-либо ошибок), и не будет вызывать предупреждений компилятора. Вот один из вариантов такого макроса:

»

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

Тесты без макросов `assertXXX`

Макросы `assertXXX` предназначены для проверки результата вычисления выражений по некоторым исходным данным (здесь под результатом вычисления выражения мы понимаем как значение этого выражения, так и генерацию ошибки при вычислении). Если результат вычисления выражения не пройдет проверку, сгенерируется специальная ошибка, сигнализирующая библиотеке `EUnit` о том, что данный тест провалился. Если в результате

вычисления какого-либо выражения будет сгенерирована некоторая произвольная ошибка (никак не связанная с проверкой), то для библиотеки `EUnit` это также будет означать, что выполнение теста провалилось. Поэтому, если цель теста — просто проверить, что при вычислении некоторых выражений не происходит генерации каких-либо ошибок, вполне вероятно, что такой тест не будет вообще содержать макросы вида `assertXXX`.

Бывает ситуации, когда мы хотим просто проверить, что при вычислении некоторого выражения не сгенерировано какой-либо ошибки. Можно, конечно, в тесте просто вычислить значение этого выражения, но при таком подходе явно не видно наше желание сделать подобную проверку. Другими словами, в некоторых ситуациях необходимо для наглядности явно указать, что при вычислении выражения мы не ждем никаких исключений.

```
-define(assertDoesNotException(Expr), ?assertEqual(true, try
  (=_Expr), true catch _:_ -> false end)).
```

И, наконец, в библиотеке `EUnit` отсутствуют макросы для сравнения действительных чисел с произвольной точностью (такое сравнение нужно из-за ошибок вычислительной математики).

Помимо приведенных выше макросов для проверки результатов вычисления выражений, в библиотеке `EUnit` есть еще ряд макросов для запуска на выполнения некоторой внешней команды и проверки результатов ее выполнения. Макрос `assertCmd(CommandString)` запускает на выполнение внешнюю команду, задаваемую строкой `CommandString`, ожидает окончания ее работы и проверяет значение кода возврата. Если значение кода возврата будет 0 (то есть команда успешно выполнялась), то проверка пройдет успешно; в противном случае проверка не пройдет. Макрос `assertCmdStatus(N, CommandString)` делает то же, что и макрос `assertCmd`, за одним исключением: код возврата выполнения команды сравнивается не с 0, а с некоторым заданным значением `N`. Макрос `assertCmdOutput(Text, CommandString)` запускает на выполнение внешнюю команду, задаваемую строкой `CommandString`, ожидает окончания ее работы и сравнивает вывод, сгенерированный командой (в виде единой строки), с ожидаемым выводом `Text`. Перед сравнением вывод от команды нормализуется: символом разрыва строки становится одиночный символ LF. И, наконец, макрос `cmd(CommandString)` запускает на выполнение внешнюю команду, задаваемую строкой `CommandString`, ожидает окончания ее работы и сравнивает значение кода возврата с 0. Главное отличие этого макроса от остальных (в особенности от `assertCmd`) заключается в том, что макрос `cmd` в случае успешной проверки кода возврата возвращает весь вывод, сгенерированный командой, в виде единой строки. При этом вывод, сгенерированный командой, нормализуется: символом разрыва строки становится одиночный символ LF. Очевидно, что значение, возвращаемое макросом `cmd`, можно использовать далее в тестах.

Есть еще ряд макросов, использование которых упрощает написание тестов — в особенности их отладку. Это макросы для вывода отладочной информации на консоль. Макрос `debugHere` выводит на консоль текущий файл и номер текущей строки. Макрос `debugMsg(Text)` выводит значение выражения `Text` на консоль. При этом `Text` может быть обычной строкой, атомом или `IO`-списком. Макрос `debugFmt(FmtString, Args)` форматирует строку при помощи функции `io_lib:format/2` и выводит результат на консоль. Макрос `debugVal(Expr)` выводит как исходный код выражения `Expr`, так и его значение, а также возвращает значение выражения. Так, например, выражение `?debugVal(f(X))` выведет строку `"f(X) = 666"` и вернет 666 (при условии, что значение выражения `f(X)` равно 666 при данном значении аргумента `X`). И, наконец, макрос `debugTime(Text, Expr)` выводит комбинацию текста `Text`

и времени, необходимого для вычисления значения выражения `Expr`; этот макрос возвращает значение выражения `Expr`. Так, например, выражение `?debugTime(«some long calculation», f(X))` выведет строку `"some long calculation: 1.002 s"` (если для вычисления выражения `f(X)` понадобится 1,002 секунды) и вернет значение выражения `f(X)`.

Но хватит слов: посмотрим, как мы будем писать тесты и использовать описанные выше макросы на практике.

```
-module(eunit_example_tests).
-include_lib("eunit/include/eunit.hrl").
example1_test() ->
  ?assert(4 == 2*2),
  ?assertNot(5 == 2*2),
  ?assertEqual(4, 2*2),
  ?assertNotEqual(5, 2*2),
  ?assertMatch([1 | _Rest], [1, 2, 3, 4]),
  ?assertMatch([1 | Rest] when length(Rest) == 2; length(Rest) == 3, [1, 2, 3, 4]),
  ?assertNotMatch([2, _Rest], [1, 2, 3, 4]),
  ?assertNotMatch([1 | Rest] when length(Rest) == 1; length(Rest) == 2, [1, 2, 3, 4]),
  ?assertError(some_error_reason, error(some_error_reason)),
  ?assertExit(some_exit_reason, exit(some_exit_reason)),
  ?assertThrow(some_throw_reason,
    throw(some_throw_reason)),
  ?assertException(error, some_error_reason,
    error(some_error_reason)),
  ?assertException(exit, some_exit_reason,
    exit(some_exit_reason)),
  ?assertException(throw, some_throw_reason,
    throw(some_throw_reason)),
  ?assertNotException(error, _, 2+3),
  ?assertNotException(exit, some_error_reason,
    error(some_error_reason)),
  ?assertNotException(error, some_exit_reason,
    error(some_error_reason)),
  ?assertCmd("echo hello world !"),
  ?assertCmdStatus(127, "some-unknown-command"),
  ?assertCmdOutput("hello world !\n", "echo hello world !"),
  ?assert(?cmd("echo hello world !") /= ""),
  ?debugHere,
  ?debugMsg("Some debug message"),
  ?debugFmt("Some format message: -p", [{format_data, 666}]),
  ?assertEqual(4, ?debugVal(2*2)),
  ?assertEqual(ok, ?debugTime("some long calculating",
    timer:sleep(1000))).
```

В данном примере мы объявляем модуль `eunit_example_tests`, подключаем библиотеку `EUnit` (директивой `include_lib`)

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

и объявляем функцию `example1_test/0`, которая и будет содержать наш тест. Функция `example1_test/0` имеет вид `XXX_test/0`, так что, как говорилось выше, функция `example1_test/0` экспортируется из модуля `eunit_example_tests` автоматически; поэтому модуль `eunit_example_tests` не содержит директивы `export`.

Рассмотрим функцию `eunit_example_tests:example1_test/0` более пристально: она содержит наш тест, в котором мы используем все вышеописанные макросы (а некоторые и не один раз). Использование макросов `assert`, `assertNot`, `assertEqual` и `assertNotEqual` тривиально и не должно вызывать вопросов (не забудем, что имя макроса всегда начинается с символа '?'). По поводу макросов `assertEqual` и `assertNotEqual` следует заметить следующее: ожидаемое значение (некоторое predefined значение, с которым сравнивается значение выражения) всегда должно быть левым аргументом, хотя с точки зрения компилятора это не важно (но важно с точки зрения семантики этих макросов). С макросами `assertMatch` и `assertNotMatch` все несколько интереснее: шаблон, с которым мы сопоставляем значение выражения, может содержать выражение охраны (идущее после ключевого слова `when`). При этом выражение охраны может состоять из нескольких проверок, но для их разделения разрешено использовать только символ ';' (это означает, что выражение охраны истинно, если хотя бы одна из проверок в нем истинна). Если мы используем выражения охраны, например, для определения разных вариантов функций, то для разделения проверок в таких выражениях охраны можно использовать также символ ';' (означающий, что выражение охраны истинно, если все проверки в нем истинны). Однако в макросах мы такой разделитель между проверками использовать не можем, поскольку этот разделитель будет неотличим от разделителя аргументов в самом макросе (которым также является символ ';'). В нашем примере мы используем и простые шаблоны без выражения охраны, и шаблоны с выражением охраны с несколькими проверками. Использование макросов `assertError`, `assertExit`, `assertThrow` и `assertException` также достаточно тривиально: мы ожидаем, что в результате вычисления интересующего нас выражения сгенерируется ошибка определенного класса, соответствующая шаблону. Класс ошибки определяется методом, при помощи которого ошибка была сгенерирована, а сама ошибка определяется аргументом, передаваемым в метод для генерации ошибки. В нашем примере мы просто генерируем определенную ошибку определенного класса (в качестве выражения, значение которого необходимо вычислить) для проверки работы макросов `assertError`, `assertExit`, `assertThrow` и `assertException`. А вот использование макроса `assertNotException` несколько нетривиально: в нем мы проверяем, что не сгенерирована ошибка определенного класса, соответствующая шаблону. Этой проверке удовлетворяют следующие три случая, когда во время вычисления выражения

- 1) никакой ошибки не сгенерировано;
- 2) сгенерирована ошибка другого класса;
- 3) сгенерирована ошибка ожидаемого класса, не соответствующая заданному шаблону.

Все эти три случая мы показали в нашем примере.

Макросы, предназначенные для взаимодействия со внешним миром — `assertCmd`, `assertCmdStatus`, `assertCmdOutput` и `cmd` — сами по себе также достаточно тривиальны; нетривиальными могут быть команды, используемые в данных макросах. Но в нашем примере мы показываем, как использовать именно макросы. Поэтому в этих макросах мы используем очень простые команды (а точнее, ограничиваемся командой `echo`), которые всегда будут выполняться успешно. Исключение составляет только пример с использованием макроса `assertCmdStatus`: здесь нам нужно получить код возврата выполнения команды, отличный от 0. Для этого мы пользуемся следующим фактом: код возврата выполнения неизвестной команды в операционных

системах семейства *nix всегда равен 127 (один из стандартных кодов возврата). В примере для макроса `assertCmdStatus` мы пытаемся выполнить несуществующую команду и сравниваем код возврата со значением 127.

И, наконец, в завершение нашего теста мы демонстрируем использование отладочных макросов `debugHere`, `debugMsg`, `debugFmt`, `debugVal` и `debugTime`. Причем макросы `debugHere`, `debugMsg` и `debugFmt` просто выводят информацию на экран, а макросы `debugVal` и `debugTime` еще и вычисляют значение некоторого выражения и возвращают вычисленное значение. Поэтому в примерах с макросами `debugVal` и `debugTime` мы проверяем значение, которое они возвращают. Макрос `debugTime` предназначен в первую очередь для измерения времени вычисления некоторого выражения; в нашем примере для простоты в качестве такого выражения мы используем паузу в 1000 миллисекунд. При этом измеренное время, необходимое для вычисления этого выражения, будет всегда чуть больше этой паузы (и, что естественно, каждый раз разное). В результате выполнения этого теста мы получим на консоли нечто подобное:

```
eunit_example_tests.erl:27:<0.39.0>: <-
eunit_example_tests.erl:28:<0.39.0>: Some debug message
eunit_example_tests.erl:29:<0.39.0>: Some format message:
{format_data,666}
eunit_example_tests.erl:30:<0.39.0>: 2 * 2 = 4
eunit_example_tests.erl:31:<0.39.0>: some long calculating: 1.002 s
Test passed.
```

Тесты мы писать научились (хотя пока и самые простые) — теперь на повестке дня следующий вопрос: как их запускать. Для запуска тестов на выполнение достаточно использовать одну из функций `eunit:test/1,2`. В простейшем случае, функция `eunit:test/1` принимает имя модуля, находит все тесты в этом модуле и выполняет их. Однако и тут есть масса нюансов. Пусть, например, имя модуля, которое мы передаем в функцию `eunit:test/1`, будет `m`. Тогда библиотека `EUnit` будет искать тесты как в модуле `m`, так и в модуле `m_tests`, если такой модуль существует. Если модуля `m` не существует, но существует модуль `m_tests` и мы передадим в функцию `eunit:test/1` в качестве имени модуля `m`, то получим ошибку времени выполнения. И, наконец, если существуют модули `m`, `m_tests` и `m_tests_tests`, и мы передадим в функцию `eunit:test/1` в качестве имени модуля `m_tests`, то библиотека `EUnit` будет искать тесты только в модуле `m_tests`. В более сложных случаях, в функцию `eunit:test/1` можно передавать и другие объекты, например, список имен модулей с тестами. Но об этом мы поговорим в другой раз. Функция `eunit:test/2` позволяет задать список опций в качестве второго аргумента. На данный момент в качестве такой опции можно использовать только атом `verbose`. Использование данной опции приводит к тому, что библиотека `EUnit` выводит некоторую дополнительную информацию на консоль.

Сегодня мы только начали разбираться в том, какие средства для написания тестов есть в библиотеке `EUnit`. Также, мы написали специальный (синтетический) тест, демонстрирующий использование всех макросов. В следующий раз мы продолжим обсуждать возможности, доступные в библиотеке `EUnit`. [LXF](#)

Что такое IO-список

В документации по языку Erlang и его библиотекам можно найти следующее определение IO-списка: `iolist = maybe_improper_list(byte() | binary() | iolist(), binary() | []).` Из этого определения видно: IO-список — это список, элементами которого могут быть байты, байтовые строки и другие IO-списки. Вот такое рекурсивное определение...

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 I-Nex
- 2 Загрузке дистрибутивов с наших DVD
- 3 Параметры Grub
- 4 Запуск программ MS-DOS
- 5 Резервное копирование автоматом
- 6 Wicd и Mint

1 Установка I-Nex

В Мне понравился ваш рассказ об утилите настройки системы *I-Nex* (см. «По рецептам доктора Брауна», LXF182), и я захотел ею воспользоваться. Пытался установить ее несколькими способами, и ни один не увенчался успехом. Сначала я попробовал включить PPA I-Nex и установить утилиту командой `apt-get install`. Установка завершилась неудачно, со следующим сообщением:

```
У следующих пакетов есть неразрешенные зависимости:
```

```
i-nex : Зависит от python-configobj, но его невозможно установить
```

```
E: Невозможно исправить ошибки, у вас нерабочие пакеты.
```

Затем я зашел на сайт <http://i-nex.linux.pl> и загрузил 32-битную версию Deb-файла *I-Nex*. Она тоже не установилась из-за проблемы с *python-configobj*. Тогда я начал искать, как установить *python-configobj*. Я не пользуюсь Python, а на сайтах, которые мне попадались, были нужны слишком большие предварительные знания, поэтому я не смог ничего сделать.

Сейчас у меня полностью обновленная версия Mint 14, работающая на 32-битном процессоре — на старом ноутбуке Dell Optiplex 745. Для нормальной работы Windows его мощности недостаточно, но для Mint вполне хватает.

Билл Франклин [Bill Franklin]

Странное сообщение. Пакет *I-Nex* действительно зависит от *python-configobj*, пакета на Python с программами, которые облегчают чтение и запись конфигурационных файлов, но беспокоиться об этом не стоит. Мы попробовали установить *I-Nex* на виртуальную машину с Mint 14 и на Mint 16, воспользовавшись инструкциями с сайта *I-Nex*:

```
sudo add-apt-repository ppa:i-nexdevelopment-team/stable
sudo apt-get update
sudo apt-get install i-nex
```

Эти команды автоматически установили нам *python-configobj* вместе с другими зависимостями



Для установки информационной утилиты *I-Nex* в любой производный от Ubuntu дистрибутив воспользуйтесь PPA. Для других дистрибутивов доступны соответствующие установочные пакеты.

без каких-либо усилий с нашей стороны. Нам остается предположить, что у Вас не настроен необходимый репозиторий, хотя это и стандартный пакет Ubuntu. Проверьте список источников пакетов в Менеджере программ или прямо в `/etc/apt/sources.list`. В списке должна быть такая запись:

```
deb http://archive.ubuntu.com/ubuntu/ quantal main restricted universe multiverse
```

Если ее нет, добавьте этот репозиторий, выполните команду `sudo apt-get update` и попробуйте снова установить *I-Nex*. Загрузить Deb-файл можно с сайта сообщества Linux Mint 14 (<http://bit.ly/py-configobj>). Нажмите кнопку Install [Установить] на этой странице, чтобы загрузить и установить *python-configobj*, после этого Вы сможете продолжить установку *I-Nex*. При возможности исправьте ошибку с репозиторием, иначе Вы можете столкнуться с похожими проблемами при установке других пакетов.

Кстати, срок жизни Linux Mint 14 уже закончился, поэтому никаких обновлений, в том числе обновлений безопасности, больше не будет. Мы бы посоветовали обновиться до более новой версии — 16-я также скоро истечет, но к LXF185 уже прилагалась 17-я версия.

2 Дистрибутивы на DVD

Дистрибутивы с диска LXF183 принесли мне несколько проблем. В openSUSE 13.1 загрузка доходит до окна с ящерицей, и выводятся три иконки, но потом картинка исчезает, и на экране появляется много

диагональных линий. В Ubuntu 13.10 рабочий стол и иконки загружаются, зато не работают клавиатура и мышь. Mageia 4 нормально загружается, и все вроде бы работает, но я не могу воспроизвести видео из Интернета из-за неподходящей версии Flash. Я пользуюсь Linux, начиная с SUSE 7.3, и никогда не загружал последние версии Flash, полагаясь на то, что они включены в дистрибутивы.

Филипп Гай [Philipp Gaj]

Извините за проблему с Ubuntu — она вызвана обновлением, выпущенным прямо перед тем, как мы закончили работу над диском. Проблема проявляется только в некоторых системах, и ни в одной из тестированных нами систем ее не было. Мы подробнее рассказывали об этой проблеме в прошлом номере (см. «Ответы», стр. 96, LXF185).

Иконки на заставке KDE — не лучший способ определить момент «падения» рабочего стола, а Ваша проблема с openSUSE связана с драйвером видеокарты или повреждением оперативной памяти. Вы ничего не говорите о своей системе, но если в ней меньше оперативной памяти, чем бывает обычно, проблема может быть в этом. Live-дистрибутивам требуется гораздо больше оперативной памяти из-за объема данных, которые они вынуждены в ней кэшировать. При загрузке с ISO-образа openSUSE у Вас появится несколько дополнительных опций, в том числе безопасный режим. Чтобы сделать то же самое при загрузке с DVD, выберите пункт меню openSUSE KDE, нажмите E и добавьте следующие параметры в конец строки, которая обычно заканчивается на `silent quiet`:

```
apt=off acpi=off mce=off barrier=off ide=nodma
idewait=50 i8042.nomux psmouse.proto=bare irqpoll
pci=nommmconf vga=0x314
```

Вам почти наверняка не понадобятся все эти опции, но указав их разом, Вы загрузитесь быстрее, чем пробуя их по одному. Последний параметр устанавливает меньшее разрешение экрана, 800×600 (также можете попробовать `vga=0x317` для 1024×768). Меньшее разрешение снизит использование памяти и поможет решить проблемы видеодрайвера, связанные с автоматическим определением разрешения монитора.

В Mageia на Live CD проигрыватель Flash по умолчанию не установлен, но Вы сможете установить его, когда установите дистрибутив на свой жесткий диск. Зайдите в раздел Install & Remove Software (Установка и удаление программ) Центра управления Mageia [Mageia Control Centre], наберите `flash` в строке поиска, и Вы сможете

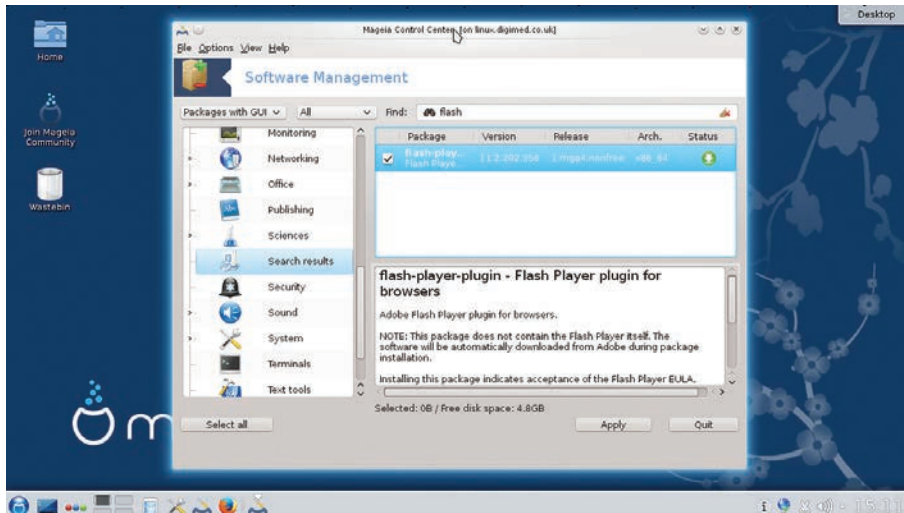
Терминалы и суперпользователи

Мы часто предлагаем в качестве решения проблемы ввести те или иные команды в терминале. Хотя обычно то же самое можно сделать с помощью графических утилит дистрибутива, такие решения будут слишком конкретными (будут зависеть от дистрибутива). Команды в терминале более гибкие и — самое главное — ими можно пользоваться во всех дистрибутивах. Команды настройки системы часто нужно выполнять от имени суперпользователя, называемого также **root**. Существует два основных способа это делать, в зависимости от используемого дистрибутива. Во многих дистрибутивах, особенно в Ubuntu и его производных, перед командой можно написать **sudo** — при этом будет запрошен пароль пользователя, и ему будут предоставлены привилегии **root** только на время выполнения команды. В других дистрибутивах применяется команда **su**, для использования которой требуется ввести пароль **root** и которая предоставляет полный доступ **root** до того момента, пока вы не наберете **logout**. Если в вашем дистрибутиве используется **su**, запустите ее один раз и выполняйте любые заданные команды без предшествующей **sudo**.

выбранный пункт. Удалите параметр **vga=** в строке **linux** и добавьте перед ней следующую строку:

```
set gfxpayload=1280x1024x8,1280x1024
```

Нажмите F10 для загрузки системы с этими изменениями. Если все получится, то имеет смысл сделать изменения постоянными, отредактировав файл **/etc/default/grub**. Найдите в нем закомментированный пример для **GRUB_GFXPAYLOAD_LINUX**, который Вы можете раскомментировать »



» По умолчанию Flash в Mageia не установлен, но это исправимо парой щелчков в Центре управления.

установить проигрыватель, после чего сайты с Flash должны начать работать. Из-за своей закрытости Flash всегда будет вызывать проблемы, так как разработчикам трудно интегрировать его с другими программами.

3 Графика Grub грубит

В Я пользуюсь openSUSE 13.1. После появления экрана 'Welcome to Grub' проходит еще два экрана, после чего я вижу следующее сообщение:

```
vga=0x307 is deprecated. Use set
gfxpayload=1280x1024x8, 1280x1024, before linux
command instead.
```

Я пользуюсь старой видеокартой Nvidia FX-5200 с памятью на 256 МБ, которая прекрасно справляется с рабочим столом KDE с кубиком. Я заглянул в **.bashrc** и **.profile**, но безрезультатно. Где мне найти команду, которая вызывает эту ошибку? Исправить ее нужно потому, что с ней система становится довольно нестабильной —

KDE зависит как минимум раз в день, и для нормальной работы приходится перезагружаться. stuarте9, с форумов

Эта ошибка не имеет отношения к рабочему столу, а команда, о которой идет речь — это команда **Grub**, находящаяся в меню загрузки. В ней указывается загружаемое ядро, и передаются параметры для этого ядра. Большинство параметров передаются напрямую, но **vga** — особый случай. Этот параметр задает разрешение начальной консоли, еще до запуска рабочего стола. Сообщение о том, что параметр устарел [deprecated], означает, что в будущем его удалят, но пока он работает, так что это чисто информационное сообщение.

Если Вы намерены избавиться от этого сообщения, вам нужно изменить конфигурацию **Grub**, и выполнить это можно двумя способами. Для временного изменения настроек нажмите E в меню загрузки, это позволит отредактировать



Коротко про...

Initramfs

Заглянув в каталог **/boot** или в файл меню загрузчика, можно увидеть ссылки на файлы **initrd**. Это образы виртуальных дисков, или **ramfs**. Виртуальный диск, как следует из названия, это хранилище данных, которое выглядит как диск и находится полностью в оперативной памяти, а файл **initrd** — виртуальный диск, сохраненный в файл. Итак, зачем это нужно?

Initrd (виртуальный диск инициализации — **initial ramdisk**) — это виртуальный диск, загружаемый ядром при запуске. Этот диск становится корневой файловой системой, и с него запускаются скрипты настройки системы, по-

сле чего управление передается настоящему корневому разделу на жестком диске. Основная функция виртуального диска — загрузка модулей ядра. Дистрибутивы предназначены для работы на самом разном оборудовании, поэтому модули драйверов в большинстве своем подгружаемые. Будь они вкомпилированы в образ ядра, у нас бы получилось огромное ядро, которое загружалось бы медленно, потребляло бы большой объем памяти и было бы на 90% избыточным. На то в Linux и нужны подгружаемые модули ядра. Но из корневой файловой системы нельзя загрузить модули ядра, необходимые для монтирования самой этой системы, и нам нужен способ предварительной загрузки драйверов контроллера жесткого

диска, файловой системы и, возможно, некоторых других, таких как LVM или **dm-crypt**. Именно это делает виртуальный диск: скрипт **linuxrc** на нем загружает соответствующие модули, запускает все необходимые программы установки (например, для LVM или для зашифрованной корневой файловой системы) и затем переключается на корневой раздел на жестком диске.

Образ **initrd** — это файловая система в файле **cpio**, упакованном **gzip**, поэтому ее можно распаковать, смонтировать, изменить и упаковать следующими командами:

```
cd /mnt/tmp
zcat /path/to/initrd | cpio -id
#modify files here
find . -depth | cpio -o | gzip >/path/to/newinitrd
```

и изменить, или добавьте новую строку с этим параметром. Так или иначе, требуемая Вам строка должна выглядеть так:

```
GRUB_GFXPAYLOAD_LINUX=1280x1024x8,1280x1024
```

Еще нужно удалить ссылку на `vga=0x307` из `GRUB_CMDLINE_LINUX_DEFAULT` и, возможно, из `GRUB_CMDLINE_LINUX`. Не удаляйте эти параметры полностью, так как остальная их часть имеет значение. Перед созданием нового файла `grub.cfg` сделайте резервную копию старого — можете переименовать его, добавив расширение `.old`. Это позволит Вам, если что-то пойдет не так, загрузиться с Live CD и восстановить старое меню. Теперь повторно создайте меню `Grub` одной из следующих команд:

```
grub2-mkconfig -o /boot/grub/grub.cfg
```

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Какой из них воспользоваться, зависит от того, куда дистрибутив установил файлы `Grub` — в openSUSE 13.1 это каталог `/boot/grub2`. Чтобы Ваши изменения вступили в силу, замените файл `grub.cfg` новым. При этом создастся новый файл меню с правильными строками `set gfxpayload`. Теперь перезагрузитесь, и надоедливое сообщение исчезнет с Вашего компьютера навсегда.

4 LOCO-МОТИВЫ

В Я пользовался *LocoScript 1, 2, 3 и Pro* на компьютере с MS-DOS, и у меня было столько стенок и слез, когда я решил отставить все это, распрощавшись с XP... Нет ли случайно какой-нибудь программы для запуска MS-DOS в Ubuntu, чтобы я мог пользоваться *LocoScript Pro*?

В. Е. Хайн [V E Hine]

Для Linux доступна парочка эмуляторов MS-DOS — *DOSEMU* (www.dosemu.org) и *DOSBox* (www.dosbox.com). Оба эмулятора предназначены для запуска программ DOS в Linux, хотя *DOSBox* скорее нацелен на старые игры DOS. У нас нет копии *LocoScript*, поэтому мы не можем проверить, подойдут ли они Вам, но попробовать стоит. Оба эмулятора бесплатны и имеют открытый исходный код, их можно загрузить с помощью центра программ Ubuntu. Для запуска программы в *DOSEMU* можно либо запустить команду `dosemu`, которая откроет командную строку DOS, и вызвать программу из нее, либо передать запускаемую команду `dosemu` таким образом:

```
dosemu someprogram
```

Преимущество последнего подхода в том, что Вы сможете создать ярлык для запуска программы на рабочем столе и запускать ее оттуда, минуя терминал. Учтите, что с технической точки зрения *DOSEMU* — не эмулятор: он запускает FreeDOS, бесплатный клон MS-DOS. Если у Вас есть лицензионная копия MS-DOS, можно приказать *DOSEMU* использовать ее вместо FreeDOS. Подробности описаны на map-странице.

В *LocoScript* используется проприетарный формат файлов, поэтому если Вы не сможете запустить *LocoScript*, то, возможно, и не сможете открыть свои файлы. Это следует учитывать для

любой программы, и вдвойне для той, что запускается в эмуляторе старой ОС. Поэтому следует архивировать свои документы в открытых форматах, которые будут читаться еще долгие годы. Сам я никогда не пользовался *LocoScript*, хотя знаю, что у него множество преданных пользователей. Но если Вы хотите гарантировать доступ к данным в будущем, стоит перейти на подходящий текстовый процессор. Существует много открытых текстовых редакторов, обладающих хорошими возможностями — например, *LibreOffice* и *AbiWord*. Переход на них может быть не простым, но сравните его с болью от утраты всех созданных с такой любовью документов!

5 udev переусердствует

В Я создал скрипт резервного копирования, который сбрасывает все мои файлы на второй диск с помощью *rsync*, затем архивирует их с *tar* и упаковывает с *gzip*. Получающийся файл имеет размер около 5 Гб. Этот файл я хочу скопировать на флэшку, чтобы хранить на ней копию, но хочу обойтись без этой суеты и мечтаю, чтобы я мог просто вставить флэшку USB, а она бы автоматически смонтировалась, затем на нее скопировался бы файл копии `tar.gz`, и потом мне отправилось бы письмо о том, что все готово. Для этого я использую следующее правило *udev*:

```
ACTION=="add&*", ATTRS{idVendor}=="8564",
ATTRS{idProduct}=="1000", RUN+="/root/scripts/
mount_USBBackup.sh"
```

Скрипт монтирует флэшку, копирует файл и посылает мне электронное письмо. Все работает, но сильно досаждают одна мелочь. После того, как я вставляю флэшку, письмо отправляется мне 11 раз!

Arthur_Dent, с форумов

Здесь у Вас несколько проблем, хотя пока Вы и столкнулись только с одной из них. Когда Вы вставляете флэшку, генерируется несколько событий *udev*, а Ваше правило соответствует слишком большому количеству событий, поэтому и срабатывает несколько раз. Чтобы увидеть события, которые создаются, когда Вы вставляете флэшку, запустите в терминале команду:

```
udevadm monitor
```

и затем вставьте флэшку. Уменьшить количество событий, которым соответствует правило, можно, проверяя подсистему. Для этого воспользуйтесь командой

```
udevadm monitor --subsystem-match=usb
```

или командой

```
udevadm monitor --subsystem-match=scsi
```

Затем добавьте в свое правило один из следующих параметров:

```
SUBSYSTEMS="scsi"
```

```
SUBSYSTEMS="usb"
```

Запустив скрипт всего один раз, Вы обнаружите, что его нельзя запускать в спешке. Параметр `RUN` в правилах *udev* желателен использовать только для тех программ, которые выполняются быстро, поскольку он блокирует

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, поскольку проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам необходимо знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства.

Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму.

Не уступающий в удобстве альтернативный вариант — *Ishw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени `root` и приложите файл `system.txt` к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
```

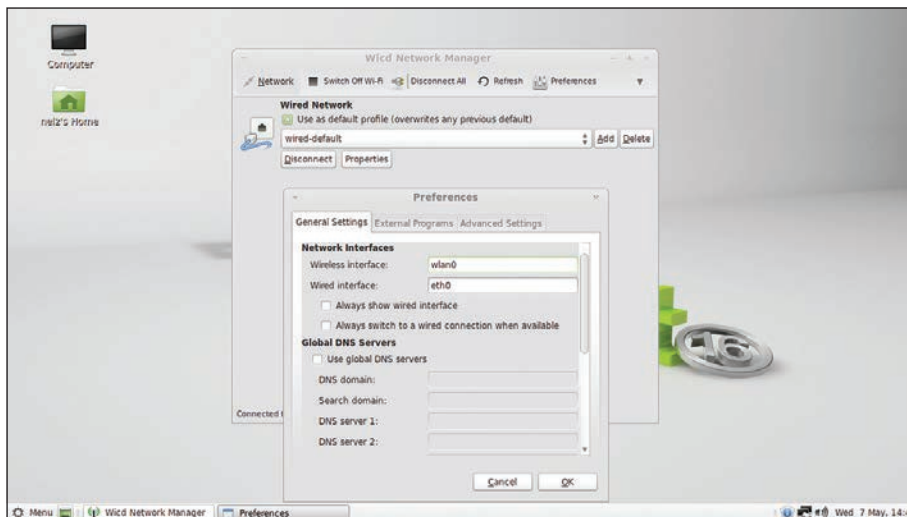
```
lspci >>system.txt
```

```
lspci -vv >>system.txt
```

дальнейшие действия *udev* до своего завершения. Для защиты от этой блокировки *udev* завершит все команды `RUN`, которые не вернут никакого значения по истечении заданного времени, которого может быть недостаточно для копирования большого файла на флэшку. Нам нужно, чтобы *udev* мог отправить сигнал, который подхватит кто-то другой, кто на самом деле выполняет процесс резервного копирования. Сделать это можно по-разному, но нам нравится *incrond*. Это демон, который следит за указанными файлами и каталогами с помощью функции ядра *inotify* и запускает команды в соответствии с набором правил *incron*. И если Ваше правило монтирует флэшку, например, в каталог `/media/backupstick`, можно заставить *incrond* наблюдать за этим каталогом. Установив *incrond* обычным образом, создайте файл `/etc/incron.d`, содержащий следующее:

```
/media IN_CREATE,IN_MODIFY /path/to/script $#
```

Каждая строка в *incrontab* состоит из трех полей. Первое поле — отслеживаемый файл или каталог, второе — разделенный запятыми список отслеживаемых событий, а третье — запускаемая команда. `$#` заменяется именем файла, сгенерировавшего событие, и перед записью на флэшку Ваш скрипт сможет проверить, ту ли флэшку Вы вставили. Монтирует флэшку правило *udev*,



➤ Переключиться с *NetworkManager* на *Wicd* обычно просто, но в версии для Mint 16 есть неработающая символическая ссылка, которую нужно заменить.

но размонтировать ее и удалить точку монтирования должен Ваш скрипт.

5 Менеджеры сети

В Недавно я переформатировал свой жесткий диск и установил на него Linux Mint 16 с вашего DVD, как делал раз или два прежде. На этот раз, когда я загружаю *Wicd* из менеджера пакетов, открывается окно с сообщением “Wicd needs to connect to your network card [Wicd должна подключиться к вашей сетевой карте]”. Я нажимаю «подключиться», после чего появляется сообщение “could not connect

to Wicd’s D-Bus check WICD log file for error messages [Невозможно подключиться к D-Bus Wicd, проверьте лог-файл на наличие ошибок]”. До последней установки я запускал эту программу безо всяких проблем.

Джон Джексон [John Jackson]

О *Wicd* состоит из двух компонентов. Демон выполняется в фоновом режиме с правами root и обычно запускается менеджером сервисов при загрузке компьютера. Клиент запускается на рабочем столе и предоставляет Вам интерфейс управления демоном *Wicd*.

Для связи друг с другом демон и клиент используют D-Bus. Ошибка, которую Вы видите, означает, что либо сервис D-Bus не запущен, что маловероятно (иначе Вы бы видели и другие ошибки), либо не запущен демон *Wicd*. Поэтому первым делом установите пакет *Wicd*, который установит демона и клиентов. Затем попробуйте запустить *Wicd* из терминала. При этом Вы скорее всего увидите сообщение об ошибке, заканчивающееся на

```
os.symlink(dest, backup_location)
```

```
OSError: [Errno 17] File exists
```

Если Вы это видите, то проблема в том, что демон *Wicd* не смог запуститься из-за неправильной символической ссылки. Запустите команду

```
ls -l /var/lib/wicd/resolv.conf.orig
```

и Вы увидите неправильную символическую ссылку, выделенную красным. Исправьте ссылку следующей командой:

```
ls -sf /run/resolvconf/resolv.conf /var/lib/wicd/resolv.conf.orig
```

Теперь *Wicd* должен запуститься с терминала, командой

```
sudo wicd start
```

На экране ничего не появится, и это хорошо — значит, сервис запустился в фоновом режиме. По умолчанию в Mint используется *NetworkManager*, поэтому его придется отключить в разделе автозапуска [Startup Applications] Центра управления Mint — там он называется просто Network [Сеть]. Сделав это и позаботившись настроить запуск *Wicd* в разделе Services [Службы] Центра управления, перезагрузите систему. Теперь клиент *Wicd* должен запуститься без ошибок. Его иконка появится в системном лотке. **15:35**



Часто задаваемые вопросы

DNCP

➤ Что такое DNCP?

Это сокращение от Dynamic Host Configuration Protocol [Протокол динамической настройки узла]. Подключаемому к сети компьютеру нужна определенная информация: его IP-адрес, адреса серверов DNS, а также куда перенаправлять запросы для компьютеров, ему не известных. Вместо сообщения всего этого вручную (т.н. статическая адресация), можно воспользоваться DNCP и получить эту информацию с сервера. Это упрощает работу с большими сетями и перемещение компьютера из одной сети в другую, как это часто бывает с ноутбуками.

➤ DNS? Еще одна аббревиатура...

Domain Name Service — сервис доменных имен, он преобразует

доменное имя, например, www.linuxformat.co.uk, в сетевой адрес, такой как 212.113.202.71.

➤ Как DNCP работает?

Клиент (ваш компьютер) отправляет в сеть широкоэвещательный «запрос обнаружения». Он принимается всеми компьютерами, но отвечают на него только серверы DNCP. Каждый сервер отправляет обратно «предложение [offer]». Если предложений несколько, клиент выбирает одно из них и запрашивает необходимые данные: IP-адрес компьютера, один или несколько серверов DNS и адрес шлюза.

➤ Шлюз? А это не из жизни водников?

Да, но здесь имеется в виду другой шлюз. Шлюз по умолчанию — это адрес, куда отправляется весь трафик для компьютеров вне вашей сети, например, для интернет-

серверов. Затем шлюз перенаправляет вам трафик и ответы на запросы.

➤ Где взять сервер DNCP?

Обычно отдельный пакет для него не требуется. В большинстве маршрутизаторов, в том числе широкополосных и/или беспроводных модемах/роутерах, есть сервер DNCP, который выделяет адреса для локальной сети. Кабельные модемы от интернет-провайдеров также используют DNCP для предоставления вам адреса при включении компьютера и модема. Если Вы захотите запустить собственный сервер DNCP для своей сети, то в дистрибутиве почти наверняка есть и сервер, и клиент (последний часто установлен по умолчанию). Поищите по строке “DNCP” в менеджере пакетов. Или же загрузите *dnsmasq* с сайта www.thekelleys.org.uk/dnsmasq.

➤ Я что-то слышал про срок аренды в DNCP. Что это?

При отключении соединения серверу DNCP направляется запрос «освобождения», благодаря которому сервер узнает, что выданный вам адрес больше не используется и может быть употреблен повторно. Если компьютер был отключен нештатным способом — например, произошел сбой электрической сети или вы вынесли ноутбук из зоны действия беспроводной сети — то сервер не сможет повторно воспользоваться вашим адресом. Для этого и необходим срок аренды — срок действия адреса. Клиент DNCP знает, что должен отправить запрос на обновление адреса в течение этого периода. Если клиент не обновил адрес, то он может быть освобожден автоматически, чтобы использоваться для кого-то другого, если возникнет такая необходимость.



LXF HotPicks



Ричард Смедли

Активист FOSS, Ричард колесит по вязким полям Interweb на своем верном SMED5000 в поисках новых залежей сокровищ открытого кода для вас.

WordPress 3.9 » view3dscene » GCC 4.9 » QEMU 2.0.0 » Nixos » youtube-dl » http » Crossfire » Micro Tetris » Infolist » MongoDB 2.6

Weblog и расширяемая CMS

WordPress

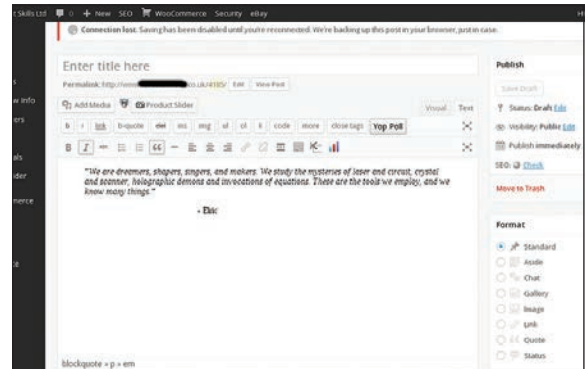
Версия: 3.9 Сайт: <https://wordpress.org>

С момента своего основания тесакцем Мэттом Малленвегом [Matt Mullenweg] и уроженцем Манчестера Майком Литтлом [Mike Little] — как ответвление более не поддерживаемой программы блогов 2/Cafelog в 2003 г., WordPress покорила невероятное количество сайтов — 75 000 000, и поддерживает индустрию ценой в миллиард долларов.

Ее новый релиз, Smith, назван в честь джазового органиста Джимми Смита [Jimmy Smith] и предлагает загрузку изображений методом drag-and-drop, ускоренный доступ к инструментам обрезки и поворота изображений и значительно улучшенный предпросмотр Gallery.

Пожалуй, самое полезное в нем (особенно если вы занимаетесь сайтом, где размещают контент пользователи без технической подготовки) — улучшенный видеоредактор.

Скорость, доступность и мобильная поддержка — дело хорошее; но реально облегчит жизнь администратора сайта возможность принять вставленный код из текстовых редакторов, например, MS



WordPress всегда был простейшим способом завести блог для пользователей — не-технарей; а теперь они могут вставлять текст прямо из документа Word.

Word, не связываясь с этими жуткими фоновыми стилями, который тащит за собой настольная Windows при вставке.

«WordPress работает на 75 миллионах сайтов в Сети.»

Подключение

Программа настройки тем стала чуть более WYSIWYG — перетаскивайте виджеты туда, где вы хотите их видеть, потом сохраняйте — и готово. Имеются и мульти-медиа-преlestи. Недавние релизы WordPress стабильно сочетали обновления интерфейса пользователя и функций; но в удобную CMS, сайт электронной коммерции и блог для миллионов пользователей его превратила удивительная гибкость и, как следствие, огромное количество модулей и тем — около 30 000 плагинов; самые популярные из них достигли показателя в 10 000 000 скачиваний. Частично этот успех обязан применению PHP, и люди без опыта программирования очень быстро осваиваются с модулем. Бывают осложнения с безопасностью и поддержкой, но более популярные плагины обычно написаны лучшими программистами.

Установка не сложнее, чем для сетевых программ на базе данных: рекомендуемые web-серверы — Apache или Nginx, PHP версии 5.2.4 и MySQL версии 5.0 или новее... вот и все основные зависимости. Создайте БД и пользователя MySQL со всеми привилегиями для доступа и внесения изменений. Отредактируйте файл wp-config.php по информации БД. Доделка в браузере — и сайт на WordPress готов.

Исследуем интерфейс WordPress

Меню приоритетов

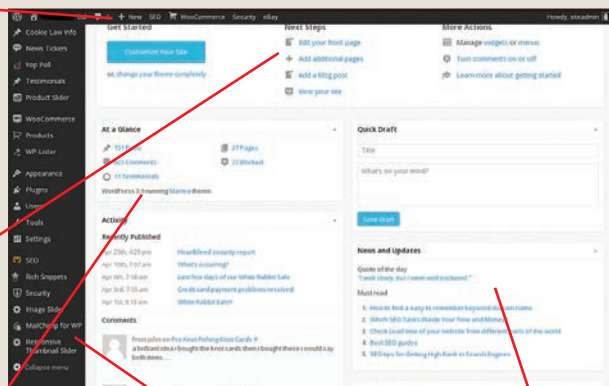
Хотя их и можно настроить индивидуально, но настройки администрирования WP по умолчанию — отличная отправная точка для большинства сайтов; там есть опции обновления сайта, электронной коммерции и безопасности.

Приступим

Быстрые ссылки на то, что вам в первую очередь нужно будет сделать на сайте. Закончив, можете закрыть это окно.

Обновление одним щелчком

Гарантирует, что существующие сайты очень быстро будут работать на самой последней версии — удобная функция.



Простые опции

Здесь есть из чего выбирать — темы, модули, настройка — и WordPress прилагает максимум усилий по содержанию их в порядке.

Настраиваемая панель информации

Вы можете удалить эти настройки по умолчанию с информационной панели и перестроить ее по своему усмотрению, чтобы проводить работу на сайте.

Браузер для форматов данных VRML и X3D

view3dscene

Версия: 3.14.0 Сайт: <http://castle-engine.sf.net>

Castle Game Engine только что дошла до версии 5.0.0. Игровой движок 3D и 2D под лицензией LGPL, написанный на FreePascal, с отличной поддержкой формата данных VRML/X3D 3D, сам по себе заслуживает внимания — и вы можете увидеть на наших страницах игры, использующие этот движок — но в этом месяце также появился новый релиз Castle Game Engine с браузером 3D-изображений, *view3dscene*.

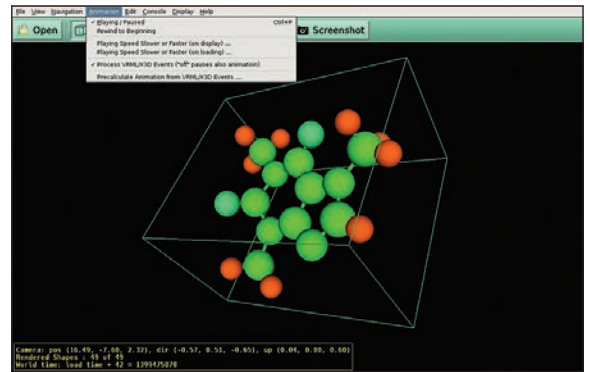
Для рендеринга он использует *OpenGL*. Вам также понадобятся *Libpng*, *Zlib*, *GTK+* и *GtkGLExt*. Вероятно, все они у вас уже есть, кроме последнего, который у вас появится, если вы предпочтете рабочий стол на *OpenGL*; или, если это не ваш случай, в вашем дистрибутиве будет пакет для него. Если вам также нужен звук, необходимы *OpenAL* и *OggVorbis*. При желании записывать ролики со своими 3D-изображениями понадобятся *ffmpeg* и программы конвертирования *ImageMagick* для работы с некоторыми форматами файлов.

Помимо этих зависимостей — неизбежных в контексте большинства настольных дистрибутивов — установка проста. Выберите 32- или 64-битную загрузку, распакуйте и просто запустите бинарник. У вас также будет папка с документацией в 3D-формате, дающая полную информацию о том, что делает *view3dscene* и как именно он работает.

Создаем анимацию

Простейший вариант — вызвать ее из командной строки с именем VRML или X3D в качестве аргумента; а если вы добавите ключ `--screenshot-range`, то сможете записать все свои действия. Следующая команда запишет вам две секунды видео при 25 кадрах в секунду:

«Режим Walk — с определением столкновений и гравитацией».



В проекте имеется множество 3D-моделей — возможно, вам уже знакома сия простая зависимость для многих гиков: это молекула кофеина.

`view3dscene caffeine.wrl --screenshotrange 0 0.04 50 caffeine-walkthrough.Avi`

Три цифры — это показатели для TIME-BEGIN [Время начала], TIME-STEP [Интервал отсчета] и FRAMES-COUNT [Число кадров].

Есть разные режимы навигации, от режима по умолчанию, *Examine*, где вы поворачиваете и перемещаете всю модель, до *Walk*, с определением столкновений и гравитацией, и *Fly*, который смахивает на *Walk*, но без гравитации. Есть текстуры, шейдеры GLSL, рельефное текстурирование и наложение теней, плюс множество функций анимации.

Полный пакет компиляции

GCC

Версия: 4.9 Сайт: <http://gcc.gnu.org>

Это ПО находится в самом сердце GNU/Linux. Есть и другие компиляторы, используемые для C и C++, но этот применяется почти всеми дистрибутивами и проектами для создания программ на вашем ПК — в том числе даже в языках программирования Perl и Python, которые и сами написаны на C. Версию 1.0, выпущенную в мае 1987 г., написал Ричард М. Столлмен [Richard M Stallman] как одну из первых частей проекта GNU, и сегодня GCC компилируется практически на всех платформах. Это определенно самый популярный кросс-компилятор.

Улучшения для всего

Данный релиз, хотя он включает улучшения для разных встраиваемых архитектур, не обходит вниманием и мейнфреймы, предлагая многочисленные реформы для платформы S/390. Естественно, недавно появившиеся устройства Intel, AMD и ARM усовершенствованы во многом.

Хотя более всего GCC популярен для C и C++ (для которых включает расширение OpenMP 4.0), он также предусматривает интерфейсы для Objective-C, Fortran, Java, Ada и Go, и опционально поддерживает другие языки.

Fortran продолжает получать усовершенствования, отражающие стандарты 2003 и 2008, а поддержка (грядущего) стандарта C++ 2014 продолжает появляться в GCC (с тэгом «экспериментальной»), включая общие (полиморфные) лямбды плюс разделители цифр в виде одиночной кавычки для удобства чтения. GNAT (компилятор GCC Ada) в этом релизе перешел по умолчанию на Ada 2012 вместо Ada 2005.

«GCC был написан как одна из первых частей проекта GNU.»



Документация GCC очень полная, но так и следует быть — ведь GCC весьма обширный и мощный.

Внутри не очень много каких-то особых изменений, хотя улучшения в самых труднительных местах впечатляют: «Потребление памяти при компиляции Firefox со включенной отладкой сократилось с 15 Гб до 3,5 Гб; время редактирования — с 1700 секунд до 350 секунд». А что же GCC 4.10? Самая впечатляющая новость в том, что UK Technology Strategy Board (агентство по инновациям Великобритании) финансирует работы Embecosm по улучшению GCC, чтобы оптимизировать компиляцию для обеспечения самого низкого энергопотребления. См. демонстрационный ролик на GNU Tools Cauldron в июле.

Эмулятор и виртуализатор машин

QEMU

Версия: 2.0.0 Сайт: <http://wiki.qemu.org>

Виртуализация изменила серверный ландшафт: теперь тысячи серверов умещаются в стойке средних размеров. Настольные системы она тоже преобразила, позволив проведение от простых разнородных экспериментов с дистрибутивами Linux и разными менее известными ОС до безопасной работы с полноценными средами разработки, отгороженными от остальной части вашего ноутбука, чтобы не повредить вашу рабочую среду, если вы совершите фундаментальную ошибку кодирования.

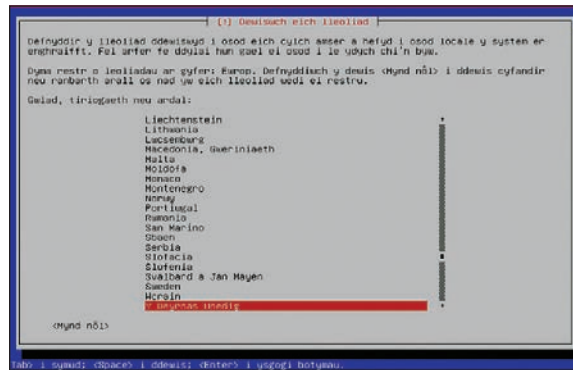
Именно в последнем случае QEMU проявляется во всем блеске, в качестве виртуализатора или эмулятора всей платформы оборудования на вашей локальной машине, что позволяет экспериментировать с версиями Grub2, заново открыть BeOS или запустить старое ПО SPARC на своем скромном ноутбуке от Intel.

QEMU может эмулировать всю систему, включая процессор и разные периферийные устройства — вероятно, куда

дружелюбнее по части драйверов, чем любое реальное оборудование; или может запускать процессы Linux, скомпилированные для одного CPU, на другом CPU, через User Mode Emulation. Его быстрые динамические трансляции делают его довольно скоростным эмулятором ARM, CRIS, i386, M68k (ColdFire), MicroBlaze, MIPS, PowerPC, SH4, SPARC или x86-64.

Платформы

Установите QEMU через свой менеджер пакетов (в репозиториях многих дистрибутивов уже есть бинарники), и вы получите множество пакетов *qemu-system-ARCH*, где ARCH [архитектура] — mips, x86 и т.д. Каждый обеспечивает эмуляцию целой системы. На x86, пакет x86 запускает



► Помимо того, что она эмулирует другие архитектуры, QEMU может предложить вам на пробу несколько отдельных установок вашего любимого дистрибутива.

KVM — виртуальную машину ядра (если у вас есть ее поддержка).

При наличии всевозможных настроек «оборудования» процессора, ОЗУ, дисков, разрешения экрана и т.д., неудивительно, что QEMU может похвастаться впечатляющим числом параметров команды — но вы можете вызвать их только при образе диска в качестве аргумента, и обычно это работает вполне нормально и переводит вас прямо на экран загрузки.

QEMU являет собой удивительный образец сотрудничества, со свыше 2500 обновлений от 179 авторов в этом последнем релизе. Если вы ищете повод для использования QEMU, просто попробуйте ее, и очень скоро вы оцените, как она облегчает жизнь многим разработчикам.

«Попробуйте Grub2, заново откройте BeOS или запустите SPARC.»

ОС с функциональным менеджером пакетов Nix

Nixos

Версия: 14.04 Сайт: <http://nixos.org>

В мире дистрибутивов на базе Debian и Ubuntu, с мощным вторым эшелонном альтернатив на базе Arch Linux и RPM, редко попадает независимый разработанный дистрибутив, созданный совершенно иным способом: на менеджере пакетов Nix.

Nix — чисто функциональный менеджер пакетов, который хранит каждый пакет в изоляции — что имеет результатом не просто отсутствие зависимостей, но и отсутствие контента в /bin, /sbin, /lib или /usr. Все сосредоточено в /nix/store; имена директорий заканчиваются именем пакета с префиксом-хэшем, который указывает на точную сборку и функциональные зависимости пакета. Это позволяет менеджеру пакетов иметь разные версии программ, просто различая их по хэшированному префиксу.

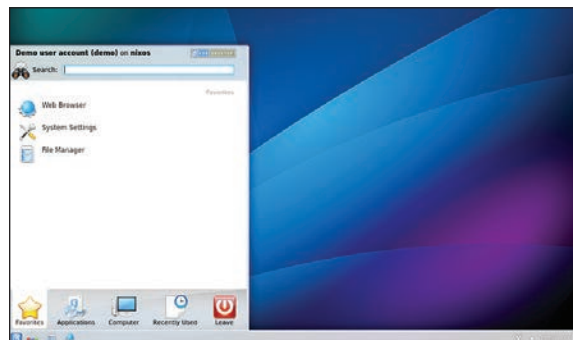
Вы, скажем, можете обновить библиотеку, не обновляя зависимую от нее программу, потому что обе версии библиотеки

могут спокойно сосуществовать. Возможны непривилегированные установки, но двум непривилегированным пользователям нельзя делить тот же самый пакет — он будет установлен дважды.

Живи — и пробуй

Самое потрясающее следствие способа работы Nix — это переключение версий: оно не просто делается из командной строки, но можно даже загрузиться в предыдущую настройку из Grub, если из-за обновления возник сбой загрузки. База данных транзакций Nix также весьма полезна, если ваша файловая система ext4 заменяет несинхронизированные файлы в хранилище Nix нулевыми байтами.

«Менеджер пакетов Nix держит каждый пакет в изоляции.»



► Минималистский рабочий стол KDE с live CD или с установки — однако менеджер пакетов Nix быстро изменит это.

Установка вполне стандартна: Nixos предоставляет live CD, с которого можно запустить команды для постоянной установки на жесткий диск. Этот live CD также открывает виртуальную консоль с *w3m*, показывающим полное руководство. Далее консоль открывается в игре *Rogue*.

Вы, конечно, можете изучить и увидеть зависимости файловой системы в действии и отсюда, но чтобы реально испытать функции менеджера пакетов, придется сделать установку на диск. В качестве альтернативы можно просто скачать менеджер пакетов Nix — или его ответвление семейства GNU, Guix, с интерфейсом Scheme вместо внешнего скриптового языка Nix; он используется в качестве официального менеджера пакетов GNU OS.

Командная утилита для скачивания видео

youtube-dl

Версия: 2014.05.05 Сайт: rg3.github.io/youtube-dl

Youtube-dl от Рикардо Гарсия Гонзалеса [Ricardo Garcia Gonzalez] — это скрипт для скачивания видео с YouTube и тому подобных сайтов, например, Vimeo. Это очень удобно для подготовки офлайн-презентаций и для того, чтобы обойти буферизацию в тех местах, где нет хорошего интернет-соединения — иными словами, для соединений в сельской местности повсюду, кроме Финляндии.

У этого пакета нет определенной лицензии СПО, но он четко объявлен Public Domain. Кстати, вам ведь не надо напоминать о необходимости уважать авторские права при скачивании видео? Это я так, на всякий случай.

Пакет *youtube-dl* в большинстве дистрибутивов не является приоритетным для обновления. К счастью, это скрипт Python; в нем нет увесистых зависимостей, так что установка проста. А раз это пакет Python, простейшим способом установки, видимо, будет

```
sudo pip install --upgrade youtube_dl
```

Но если вы не хотите идти путем установщика python, или *pip* не очень ладит с вашей системой, просто скачайте пакет, сделайте исполняемым и разместите где-нибудь в своем \$PATH.

```
sudo wget https://yt-dl.org/downloads/2014.05.05/youtube-dl -O /usr/local/bin/youtube-dl
```

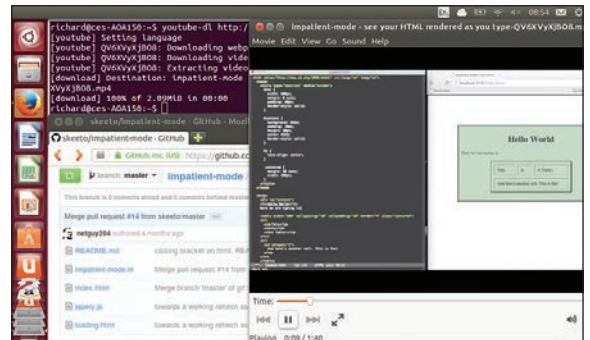
```
sudo chmod a+x /usr/local/bin/youtube-dl
```

отлично справится с работой, а *Curl* будет неплохой альтернативой *wget*.

Всепроникающий

Скрипт скачивает видео не только с YouTube: он справится еще с сотней сайтов. На некоторые из них вы, вероятно, в жизни своей не зашли бы, однако все основные

«Удобно для офлайн-презентаций и чтобы обойти буферизацию.»



Долой буферизацию — скачивайте видео с YouTube, чтобы смотреть его офлайн позднее. Отлично подходит для презентаций и дает плавное воспроизведение.

сайты здесь налицо, в том числе Vimeo, Facebook и Flickr.

Когда дойдет до работы, просто вызовите его из терминала, приложив URL YouTube (или другого сайта) в качестве аргумента, таким образом:

```
youtube-dl http://youtu.be/QV6XVyXjB08
```

Youtube-dl проанализирует сайт, найдет встроенное видео, скачает и сохранит его для вас.

Более ранние версии давали скачанному видео буквенно-цифровое обозначение, если вы не использовали переключатель **-o**. И эта опция позволит дать файлам имя по вашему усмотрению:

```
youtube-dl http://youtube.com/QV6XVyXjB08 -o emacs-web-demo.Flv
```

Интерактивный просмотрщик процессов Linux

htop

Версия: 1.0.3 Сайт: <http://hisham.hm/htop>

Всем знаком *top*, с его динамичным отображением процессов и системной информации в реальном времени. Он неплохо настраивается и имеет ряд интерактивных функций, включая поиск, и вдохновил немало своих подобию для применения к сети (*ntop* и *iftop*) и I/O (*iostat*). Есть и улучшенные программы просмотра работающих процессов — хотя они и не эмулируют некоторые странные трюки *top* Unix — типа «Отскакивающих окон [Bouncing Windows]» и «Окна Большой Птицы [The Big Bird Window]» — но зато восполняют недостаток функций.

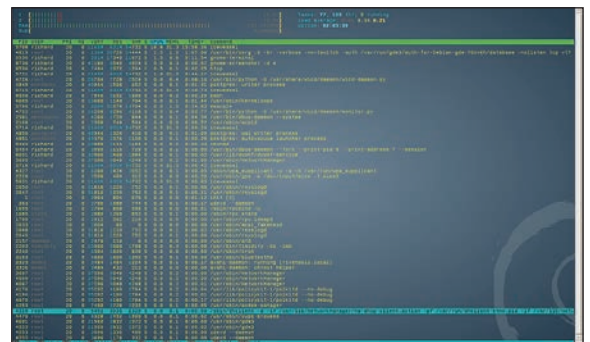
Программа *top* встроена в Unix стандартно. Она переключилась с *curses* на *termcap* сразу в 1986, ради эффективности потребления ресурсов, и с тех пор это не менялось. Как следствие, имена длинных процессов урезаны до 80-го столбца. *htop* использует *ncurses*, и поэтому его можно прокручивать как вбок (для путей длинных процессов), так

и вертикально. Кроме того, *htop* быстрее, чем *top*: он запускается быстрее, и не теряется, проверяя правописание, при каждом нажатии не назначенной клавиши.

Клавиши функций

htop также успешно распишет общее потребление CPU по ядрам (см. рис. справа). Можно осуществлять навигацию по указанному процессам, мышью или клавишами со стрелками, и осуществлять взаимодействие через клавиши F: F5 отобразит древовидную иерархию процессов; F7 и F8 позволяют менять приоритеты процессов, а F9 — всегда полезный kill для процессов, переставших отвечать. У нас у всех есть свои любимые мерзавцы!

«F9 — полезный kill для процессов, переставших отвечать.»



htop делает просмотр процессов быстрым и простым, с древовидным отображением, операциями одним нажатием и боковой прокруткой.

Можно осуществлять поиск по результатам процессов, фильтровать их и настраивать порядок столбцов и то, что отображается, а также метрики вверх; доступ к функциям дают оставшиеся клавиши F. Есть клавиши вызова полезных утилит по выделенному процессу, например, **I** для *Isof*, перечисления дескрипторов файлов, открытых процессом; **s** предоставляет «живое» обновление вызовов системы, заданных процессом с помощью *strace*; а **L** отображает вызовы библиотек с помощью *ltrace*. Поток пользователя и поток ядра можно скрыть (а потоки отдельного пользователя отображаются по ключу командной строки) — и выбрать разный порядок сортировки.

Без *htop* вы скоро будете как без рук.

HotGames Развлекательные приложения

Многопользовательская графическая ролевка с приключениями

Crossfire

Версия: 1.71.0 Сайт: <http://crossfire.real-time.com>

Crossfire появилась в 1992 году как клон *Gauntlet*, одной из первых многопользовательских аркад с ползанием по подземелью. Из одной внутренней карты и пяти заклинаний, *Crossfire* вырастила более 3000 карт с гораздо лучшей магией, 13 рас, 15 классов персонажей и 150 монстров.

Данный релиз добавляет привязки к клавишам для каждого персонажа, новые карты и кучу отладок. До загрузки клиента *GTKv2* понадобятся библиотеки *GTK-dev* и *xpmlib*. В Debian или Ubuntu вы отобьетесь командой

```
apt-get install libgd-tools libgd2-xpm-
devlibsdl1.2-dev libsdllimage1.2-dev
libgtk2.0-0 libgtk2.0-dev libglade2-0
libglade2-dev
```

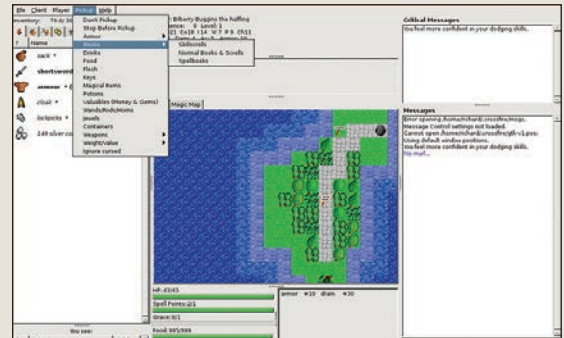
После обычной установки через `./configure, make, sudo make` запускайте довольно старомодный интерфейс.

При первом запуске вам предстоит создать учетную запись *Crossfire*; в меню видно, сколько человек играет на каждом сервере, и можно выбрать тот коллектив, где вам будет наиболее комфортно.

Создание персонажа

Вам также понадобится создать персонажа, если у вас нет готового. Выберите имя, расу и род занятий, а затем карту, где вы хотите появиться. Вы устанавливаете начальный уровень навыков вашего персонажа, распределяя баллы между Харизмой, Телосложением, Силой, Мудростью, и т.д. Навыки улучшаются по мере

«Вырастила 3000 карт, 13 рас, 15 классов и 150 монстров.»



► Наш полурослик, Бильберри Баггинс [аллюзия на Бильбо Баггинса, — прим. пер.], обещает многое, но сначала он должен пройти через Остров Руководства...

их упражнения в течение игры. Система вполне сбалансирована, но сперва стоит прочесть руководство.

ПО сервера тоже выходит под свободной лицензией, и его запуск дома дает возможность проверить ваши собственные карты. Системные требования минимальны даже на публичных серверах — а на домашнем сервере она должна запуститься даже вместе с вашим мультимедиа-сервером, а то и с Raspberry Pi.

Классика аркад

Micro Tetris

Версия: 1.2.1 Сайт: <http://troglorbit.com/tetris.shtml>

Cогласно Freecode, *Micro Tetris* только что обновился. Мы считаем, что данное обновление могло бы выйти и пораньше, но всё равно это хороший повод снова взглянуть на *Micro Tetris* со товарищи на troglorbit.com.

Проистекшая из работы Джона Тромпа [John Tromp] для конкурса International Obfuscated C Code Contest в 1989-м, эта крошечная реализация использует только управляющие последовательности ANSI и команду Unix `stty` для настройки и восстановления экрана — здесь нет каких-либо внешних зависимостей от библиотек, помимо стандартной библиотеки C типа *uClibc*, что делает её очень подходящей для самых малых встраиваемых сред — хотя бы таких, как те самые часы IBM на Linux, 13-летней давности, которые пылятся где-то на дне ящика стола. Эта игра поместится в любом вашем

встраиваемом устройстве, если только у него есть хоть какой-нибудь консольный интерфейс — и, возможно, даже будет спрятана, как пасхальное яйцо.

Хоть изначально код был умышленно запутан [obfuscated], его переписали в более читаемом стиле C, и лет пять назад вышел релиз 1.0. С тех пор он был отлажен и приобрёл новые свойства, например, постоянную таблицу рекордов.

Странное обаяние

Перейдите на сайт игры, и вы также увидите *Micro Snake*, *Micro FTPD* и прочие интересные примеры кода. Мы, как всегда, наслаждались возможностью заново

«Эта игра поместится в любом встраиваемом устройстве.»



► Да, это обычный тетрис; но чтобы в него сыграть, достаточно библиотеки C, и он поместится где угодно!

открыть для себя радости тетриса — эта простая разновидность динамической головоломки странно-привлекательна для человеческого мозга, хотя бросить играть в неё гораздо проще, чем в более сложных головоломках. Однако даже отвлекаясь от интересного игрового процесса — если вы намерены отточить свои навыки работы в C, возможно, с целью перехода на встраиваемые системы, вам, возможно, захочется взглянуть на минималистский код Фрека Видейка [Freek Wiedijk] и Йоахима Нильссона [Joachim Nilsson] и увидеть, на что способен C даже в малом объеме.

Редактор блоков списка INFO для RIFF (AVI, WAV и т.д.)

Infolist

Версия: 20140504 Сайт: infolist.sourceforge.net

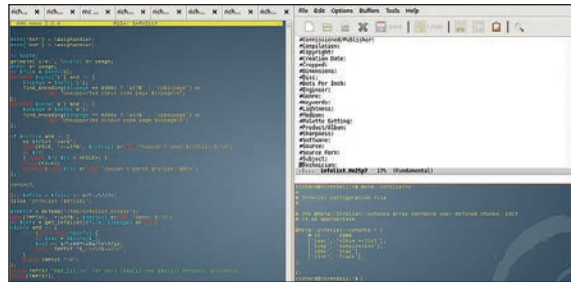
Э тот аккуратный скриптик Perl от талантливого кодера-шведа Матса Петерсона [Mats Peterson] позволяет редактировать блоки INFO list в любой файл RIFF (Resource Interchange File Format). Файлы RIFF включают повсеместно распространенные форматы WAV и AVI, а также недавно возникший формат WebP, и все они хранят данные маркированными блоками, с опциональным блоком INFO, обычно находящимся недалеко от начала файла. Информация в блоке INFO обычно размещена в виде ряда предустановленных категорий, таких, как копирайт (ICOP), комментарии (ICMT) и артист (IART); но авторы контента (да и любой) свободно могут определить и добавить категорию, но с оговоркой: это способно вызвать неверную интерпретацию подблоков ID некоторыми программами.

Простая установка

Чтобы запустить *infolist*, просто распакуйте его и поместите бинарник в свой

путь (например, `~/bin`) — возможно, изменив резервный редактор в скрипте, если вы не ладите с *Vi* — и включенные библиотеки Perl туда, где Perl их увидит (например, в `/usr/lib/perl/5.18`). Можно взять файл шаблона `.infolist`, но тогда смените массив `@Meta::InfoList::xchunks` на нечто соответствующее тому, что вы редактируете. Существующий массив `Album Artist; Compilation; Disc; Track` отлично подойдет для вашей коллекции One Direction CD, но не вполне — для вашей коллекции записей птичьего щибета.

➤ Настройте свои `@Meta::InfoList::xchunks` в *Infolist* так, чтобы не добавлять `Album Artist` к информации по звуковым эффектам для игры.



Хранилище документов без схем

MongoDB

Версия: 2.6 Сайт: <http://mongodb.org>

MongoDB — высокопроизводительное хранилище данных с открытым кодом, ориентированное на документы и хранящее данные во вложенном состоянии — двоичная форма JSON позволяет осуществлять спецзапросы вложенных данных, не действуя логических схем; это идеально для сайтов, хранящих много данных при скромном бюджете, и позволяет компаниям и проектам быстро расти. Но *MongoDB* применяют и огромные, успешные проекты, от *bit.ly* и *Foursquare* до Большого адронного коллайдера CERN. Релиз 2.6 БД NoSQL заявлен как большой шаг вперед, со «всесторонними ключевыми расширениями сервера, потрясающим новым инструментом автоматизации и важнейшими для предприятия функциями».

Движок исполнения запросов переписан для лучшей масштабируемости. Появилась возможность возвращать наборы результатов любого объема; агрегация

поддерживает переменные, добавилось операций по обработке наборов данных и их компоновке. Новый протокол записи улучшает поддержку операций с массивами. Сняты ограничения для *maxIncomingConnections*. Безопасность и улучшение аналитики включают шифрование TLS, определяемые пользователем роли, ревизию системы и редактирование на уровне полей. Установка бинарников с сайта проста, и незачем ждать, пока новый релиз появится в пакетах вашего дистрибутива. **LXF**

➤ Установка базы данных ради того, чтобы с ней познакомиться, требует времени — сперва попробуйте *MongoDB* с JS на try.mongodb.org.

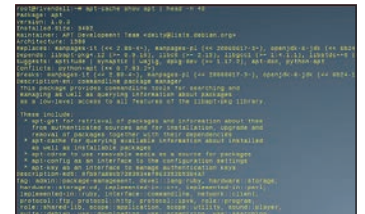


Также вышли

Новые и обновленные программы, тоже достойные внимания...

➤ APT 1.0.3

Advanced Packaging Tool для Debian, Ubuntu и относящихся к ним дистрибутивов. wiki.debian.org/Apt



➤ APT добавляет поддержку файлов настройки по производителем.

➤ Yocto Project 1.6

Шаблоны, инструменты и методы для создания индивидуального встраиваемого Linux. www.yoctoproject.org

➤ xterm Patch #304

Стандартный эмулятор терминала добавил экспериментальную поддержку ReGIS. <http://invisible-island.net/xterm>

➤ neslisp 0.0.17

Пакет компилятора lisp для архитектуры 6502. <http://sf.net/projects/neslisp>

➤ libnode 0.4.2

Реализация Node.js на C++ со сниженными требованиями к ресурсам. <https://github.com/plenuno/libnode>

➤ pydiction 1.2.3

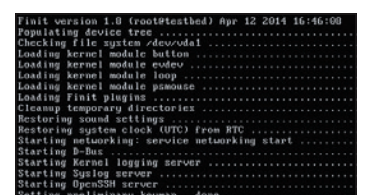
Vim с автодополнением по Tab для Python, Numpy, Django, Twisted, Pygame, OpenGL и т.д. http://www.vim.org/scripts/script.php?script_id=850

➤ Mentoliga 2.6.0.0

Менеджер Football League под GPL для Windows; должен работать на WINE (?). <http://mentoliga.wix.com/mentoliga>

➤ Finit 1.9

Маленькая и быстрая замена SysV, запускает сервисы параллельно. <http://troglobit.com/finit.html>



➤ Обратный инжиниринг *syscalls*, демона быстрой инициализации для EeePC.

На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Улучшим

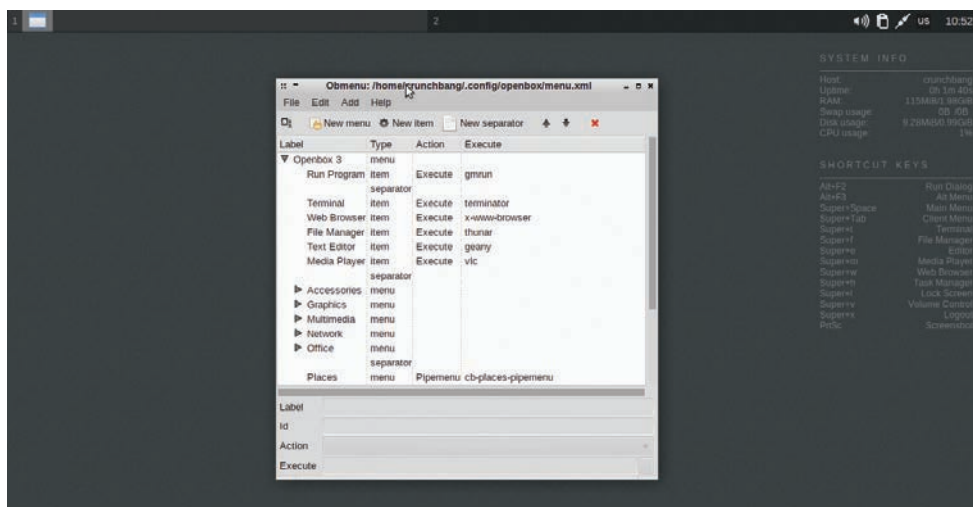
В Linux масса приятных моментов, но то, что мы ныне принимаем как должное — это live CD/DVD. На заре дисков *Linux Format* (в первых 19 выпусках были только CD) такого явления, как live CD, вообще не было. Дистрибутивы ограничивались чистой установкой: вы загрузились с CD, а тот разбивал ваш жесткий диск на разделы и ставил дистрибутив. Версий «до покупки попробуйте» не было. Переломил ситуацию Knoppix: это не первая попытка live CD, но он первый реально справлялся с самым разным оборудованием. Затем другие дистрибутивы начали выпускать диски live — но только в этом режиме, без возможности установки, пока PCLinuxOS (по-моему) не объединил обе функции. За ним последовал Ubuntu... и это сделалось настолько привычным, что когда один дистрибутив из Сравнения этого месяца вдруг загрузил установщик, я был поражен.

Теперь мы полагаем, что можно попробовать любой дистрибутив, не трогая имеющихся установок, и в подавляющем большинстве случаев правы. А значит, нам выдан доступ ко всем инструментам Linux на любом ПК, который мы можем перегрузить, не говоря уж о превосходных инструментах восстановления. Еще одно очко в пользу Linux.

Neil

Дистрибутив Linux

CrunchBang



О некоторых дистрибутивах шумят заголовки прессы (не всегда заслуженно), а есть и такие, что спокойно делают свою работу без особой гласности. CrunchBang явно относится к последней категории. Даже его рабочий стол отличается сдержанностью — в простых серых тонах.

Но не давайте первому впечатлению себя обмануть: это отличный дистрибутив с сильным сообществом. CrunchBang основан на Debian, а поверх него пристроен рабочий стол *OpenBox*; результатом является компактный, быстрый и стабильный дистрибутив с широким выбором программ.

Дистрибутив Linux

Elive

Elive — это live CD на основе E (хитро зашифровано, а?), менеджера окон *Enlightenment*. *Enlightenment* — одна из тех программ, над которыми стоит потрудиться: настройка по умолчанию вполне разумна, но приложив усилия, вы создадите прекрасные рабочие столы, все так же быстрые и легковесные.

В Elive такой рабочий стол есть. Если вам нравится анимация рабочего стола, вам понравится то, как изменяется масштаб значков на док-панели. Это бета-версия нового релиза, и в ней пока нет программы установки на жесткий диск, хотя когда вы будете читать этот материал, она уже вполне может появиться.

» Важно
ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.



Manjaro

Это еще один дистрибутив-скромник, прекрасно справляющийся со своими задачами. Он выделяется среди всех этих производных Ubuntu и Debian, поскольку основан на Arch Linux — дистрибутиве, который обретает популярность в качестве основы для других. Arch — мощная система, но не самая простая для новичка, и Manjaro в этой версии придает ей больше дружелюбия с помощью быстрого рабочего стола *OpenBox*.

ДИЕТИЧЕСКИЕ ДИСТРИБУТИВЫ



Лучшие дистрибутивы-легковесы Linux — установите сейчас, ускорения для!

А ТАКЖЕ: 40 супер-эффективных программ, 10 книг о Linux и многое другое

Новичок в Linux? Начните отсюда!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент MS Office?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

Из журнала

ПО с малым потреблением ресурсов

40 программ, выбранных за скорость и легковесность.

Коды руководств

Программы и примеры для учебников этого месяца.

Ответы

Все ваши проблемы с открытым кодом, решенные в этом выпуске.

LXFHotPicks

Crossfire

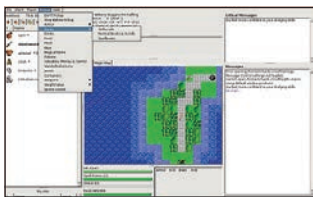
Совместная многопользовательская графическая ролевка с приключениями.

GCC

Compiler Collection 4.9 с gcc.gnu.org. Полный пакет для компиляции.

htop

Интерактивный просмотрщик процессов Linux.



Infolist

Редактор блоков списка INFO для видеоформатов RIFF (AVI, WAV и т.д.).

Micro Tetris

Представитель классических аркадных игр.

MongoDB

Хранилище документов данных без схем.

Nixos

ОС с функциональным менеджером пакетов Nix.

QEMU

Эмулятор и виртуализатор произвольных машин.

view3dscene

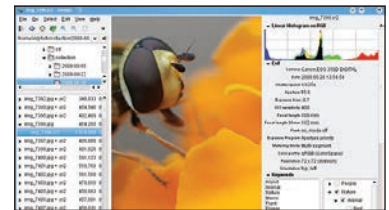
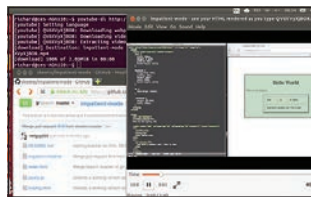
Браузер для форматов данных VRML и X3D.

WordPress

Приложение для блогов и расширяемая CMS.

Youtube-dl

Программа командной строки для скачивания видео сами понимаете откуда.



И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязанные быть в каждой операционной системе.

Hardinfo Инструмент для тестирования ОС.

Ядро Исходник последней версии ядра.

Memtest86+ Проверьте свое ОЗУ на сбои и ошибки.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создание загрузочных дисков в Windows.

SBM Независимый от ОС менеджер загрузки с простым интерфейсом.

WvDial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

The Cathedral and the Bazaar (Собор и Базар) Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Intro to Linux Руководство для новичков.

Linux Kernel in a Nutshell (Ядро Linux в двух словах) Введение в ядро от магистра хакеров Грегга Кроа-Хартмана [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для системных администраторов.

Словарь Linux Linux от A до Z.

Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Осваивайте скрипты оболочки.

Расширенное руководство по созданию скриптов Bash Больше информации для опытных пользователей.

Руководство сисадмина Основы работы с системой.

Обзор инструментов Инструменты GNU.

Подкасты



Сезон 6 Эпизод 3

О да, это эпизод 3, а не переодетый эпизод 2. Как нам кажется. Ну что мы можем знать, запертые в студии звукозаписи, где кондиционер ревет, как Боинг-747 на взлете?

Новости Новый член команды появляется как раз к сокращению штатов, дроны Linux взмывают в небо, а FCC становится отчасти бесполезным.

Открытия недели LxQt — милый новый легковесный рабочий стол, XBMC 13 Gotham наконец-то вышел, и он весьма хорош, вся Дания попала в *Minecraft*, GoboLinux — интересный ремикс, плюс обсуждаем ноутбук Open Source Novena и менеджер файлов *Jurassic Park 3D SGI*.




Пропустили номер?



Закажите его через сайт www.linuxformat.ru в «ГНУ/Линуксцентре»! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти всего пара минут!

Прямо сейчас для заказа доступны следующие номера:



LXF183
Май 2014

250 руб.

- » Битва дистрибутивов Который лучше?
- » Среда разработчика Не просто редактор
- » GnuPG Модная тема криптографии
- » Android+Linux = Беседа телефона с компьютером

LXFDVD: Mageia 4, openSUSE 13.1, Ubuntu 13.10, LibreOffice 4.2, VMS Empire 1.11, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_183/
PDF-версия: shop.linuxformat.ru/elxf_183



LXF184
Июнь 2014

250 руб.

- » Стол поудобнее Обустроим рабочее место
- » Все под контролем Мониторинг системы
- » OpenELEC Чисто конкретно для медиа
- » Какой портрет, какой пейзаж! Облагородим фотки

LXFDVD: Mint 16 Mega Mix, Ubuntu 14.04, Krita 2.8, The Legend of Edgar, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_184/
PDF-версия: shop.linuxformat.ru/elxf_184/



LXF185
Июль 2014

250 руб.

- » Смастерим компьютер Это проще, чем вы думали
- » Сам себе Википедия Выбираем движок wiki
- » UEFI Отправим BIOS на покой
- » Minecraft Pi Игры в блочной вселенной

LXFDVD: Ubuntu 14.04 LTS, MX-14, Trisquel 60.1, Linux Mint 17, Grml 2014.03, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_185/
PDF-версия: shop.linuxformat.ru/elxf_185/

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через www.linuxformat.ru/subscribe/ или www.linuxcenter.ru, получают электронную версию в подарок! На сайте shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF.

Подписывайтесь на сайте www.linuxformat.ru/subscribe/

Телефоны отдела подписки:

- » Санкт-Петербург (812) 309-0686
- » Москва (499) 271-4954



Linux Format VKontakte

Вступайте в нашу
группу vk.com/linuxform

На странице LXF ВКонтакте вы найдете:

- » Новости о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше, этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: CentOS 3671.0, Debian 2014, SaliX 14.1

Calculate Linux 13.19

64-разрядная сборка с рабочим столом Xfce

Live



Август 2014
LXF DVD 186

LINUX
FORMAT

Август 2014
LXF DVD 186

LINUX
FORMAT

ДИСТРИБУТИВЫ ДИСТРИБУТИВЫ



Лучшие дистрибутивы-легковесы Linux —
установите сейчас, ускорения для!

А ТАКЖЕ: 40 супер-эфективных программ, 10 книг о Linux и многое другое

Содержание

LINUX FORMAT

Сторона 1

ДИСТРИБУТИВЫ С МАЛЫМ ПОТРЕБЛЕНИЕМ РЕСУРСОВ

Cunchbang 11
Elive 2.2.2 beta
Manjaro Openbox 0.8.9
Puppy Slacko 5.7

ДОКУМЕНТАЦИЯ — 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Bash Scripting Подробное руководство по программированию на Bash
bourne Shell Scripting Начальное руководство по программированию на Bash
Cathedral Bazaar Классический текст Эрика Раймонда [Eric S Raymond] «Сбор и базар»

The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian

Dive Into Python Учебник по программированию на Python

Intro to Linux Начальное руководство по Linux на Python

Linux Dictionary Словарь Linux, объясняющий специфическую терминологию

Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков —

Греггом Кроа-Хартманом [Greg Kroah-Hartman]

System Administrators Guide Руководство по базовому администрированию Linux

GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

Сторона 2

ДИСТРИБУТИВЫ

Calculate Linux 13.19 с рабочим столом Xfce, 64-разрядная сборка Live (ISO-образ)
Deephin 2014, 64-разрядная сборка Live (ISO-образ)
CoreOS 3671.0, 64-разрядная сборка Live (ISO-образ)
SaltX 14.1, 64-разрядная сборка (Установочный ISO-образ)

НОТРИСКС

Crossfire Многопользовательская графическая ролевка с приключениями
GCC Полный пакет для компиляции
htop Интерактивный просмотрщик процессов Linux
Infocast Редактор блоков списка INFO для RIFF (AVI, WAV, и т.д.)
Micro Tetris Классика аркад
MongoDB Хранилище документов, без схем
Nixos OS с функциональным менеджером пакетов Nix
CEMU Эмулятор и виртуализатор машин
view3dsScene Браузер для форматов данных VRML и X3D
WordPress ПО для блогов и расширяемая CMS
Youtube-dl Командная утилита для скачивания видео

Пожалуйста, перед использованием Асдного Диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

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 feedback@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех этапах производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аналитический сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных в вашей жесткой копии диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с нами, чтобы узнать больше о нас и о нашей политике в отношении данных.
Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-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 (скорее всего, в вашей системе присутствует только 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
```

Для записи ISO-образа вам осталось набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени 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?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам 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.



UNIXFORUM.org



Есть вопросы?
Задавайте!

<http://unixforum.org/>

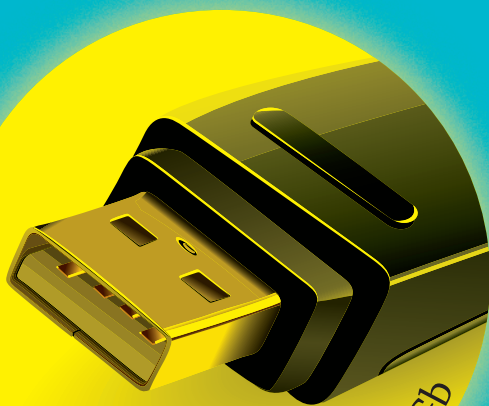
Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств
в разных городах России
позволит вам оптимизировать
процессы логистики и доставки товара

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



ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 14.04

А ТАКЖЕ
версии для юриди-
ческих лиц —
с лицензионным
договором
присоединения

USB Flash 8 Gb

495 рублей

www.linuxcenter.ru/shop/distros/usb/



UnixEducationCenter

ОБУЧЕНИЕ И СЕРТИФИКАЦИЯ ОС LINUX



ПО ВСЕМ ВОЗМОЖНЫМ УРОВНЯМ:

RHCSA – RED HAT CERTIFIED SYSTEM ADMINISTRATOR

RHCE – RED HAT CERTIFIED ENGINEER

RHCA – RED HAT CERTIFIED ARCHITECT

www.unixedu.ru

+7 (812) 611-15-75 mail@unixedu.ru

Санкт-Петербург, Черноморский пер., д.4

ГНУ/Линуксцентр
приглашает на работу!



ВАКАНСИЯ: Разработчик систем электронного документооборота

ТРЕБОВАНИЯ:

- » Опыт внедрения и сопровождения систем электронного документооборота на базе Alfresco
- » Умение описать бизнес-процесс
- » Знание SQL, понимание принципов построения и функционирования баз данных
- » Знание СЭД, отличных от Alfresco, приветствуется

ОБЯЗАННОСТИ:

- » Разработка систем с нуля, сопровождение
- » Настройка системы в соответствии с требованиями бизнеса
- » Написание инструкций для конечных пользователей

ПОДРОБНЕЕ: www.linuxcenter.ru/vacancy/

Футболки GNU/Linux

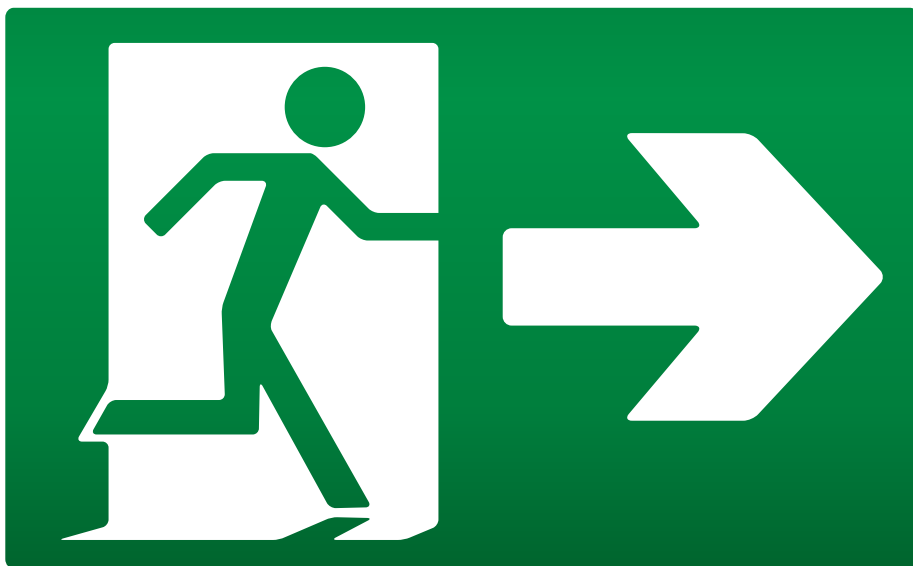
БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/



Windows



В сентябрьском номере

Как сбежать с Windows

Руководство по отмене Windows и переходу на Linux. Расскажите друзьям! Расскажите коллегам! Время идти.

Отведаем Wine

Ну, мы перешли... и все же скучаем по *Baldur's Gate*... Запускаем свежий релиз *Wine* для потехи в Windows.

Безопасный VPS

Мечтаете о дешевом, но безопасном частном сервере — VPS? Разумеется, да; и мы проведем вас по всем стадиям процесса!

Автомобили... на Linux

Машины без водителей революционизируют мир, и в сердце у них будет Linux... точнее, в мозгах.

Содержание будущих выпусков может меняться — вдруг наш автомобиль от нас сбежит...



Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф-077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова, Ирина Шулакова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ЗАО «Мезон.Ру»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. 6/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 11084

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Заместитель редактора Мэтью Хэнсон [Matthew Hanson]

matthew.hanson@futurenet.com

Научный редактор Джонни Бидвелл [Jonni Bidwell] jonni.bidwell@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Джолон Браун [Jolyn Brown], Терри Дьюэлл [Terry Duell], Марко Фиоретти [Marco Fioretti], Мэтью Хэнсон [Matthew Hanson], Кевин Ли [Kevin Lee], Нейл Мор [Neil Mohr], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Ричард Смедли [Richard Smedley], Евгений Балдин, Петр Семилетов, Александр Толстой, Андрей Ушаков, Алексей Федорчук

Ассистенты по выпуску Алекс Саммерсби [Alex Summersby], Эстер Ньюмен [Esther Newman]

Иллюстрации Элли Уолтон [Eily Walton], Шейн Коллиндж [Shane Collinge], Саймон Миддлвик [Simon Middleweek], iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 1225 442244, email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция): пр. Медиков, 5, корп. 7. Тел. +7 (812) 309-0686

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел./факс +7 (499) 271-4954

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 диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds]. «GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

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

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

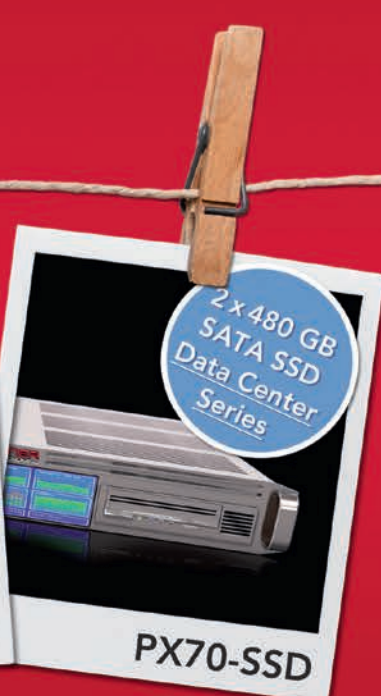
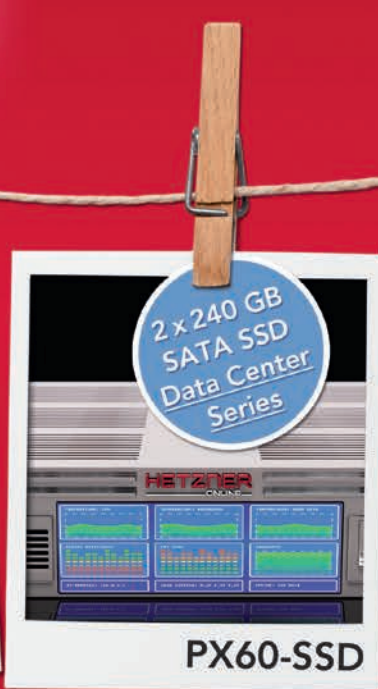
309-06-86

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

Linux  center

СВЕЖИЕ РАЗРАБОТКИ!

Новинка!



ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 2 ТБ 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 ТБ Трафик*

2600

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX60-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 240 ГБ 6 Гбит/с SATA SSD Data Center Series
- 20 ТБ Трафик*

3000

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX70

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 4 ТБ 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 ТБ Трафик*

3000

рублей в месяц + Установка 3700 рублей

ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 480 ГБ 6 Гбит/с SATA SSD Data Center Series
- 30 ТБ Трафик*

3700

рублей в месяц + Установка 3700 рублей

ВКЛЮЧЕНО:

- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта



Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!

RU.HETZNER.COM



* Нет платы за превышение. При превышении 20 ТБ/месяц (PX60, PX60-SSD, PX70), 30 ТБ/месяц (PX70-SSD) скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 80 рублей за каждый дополнительный ТБ.