

DVD ДАРОМ!



16 ВИДЕОКУРСОВ!

LINUX FORMAT

Главное в мире Linux

70 страниц учебников и статей!

- » WhatsApp помогает Pi
- » WordPress 4.3 изнутри
- » Хакинг сети из терминала

+ Лучший хромбук-2015: ищем, какой купить

Ноябрь 2015 № 11 (202)

**СПЕЦИАЛЬНЫЙ
ВЫПУСК!**

От Тукса:

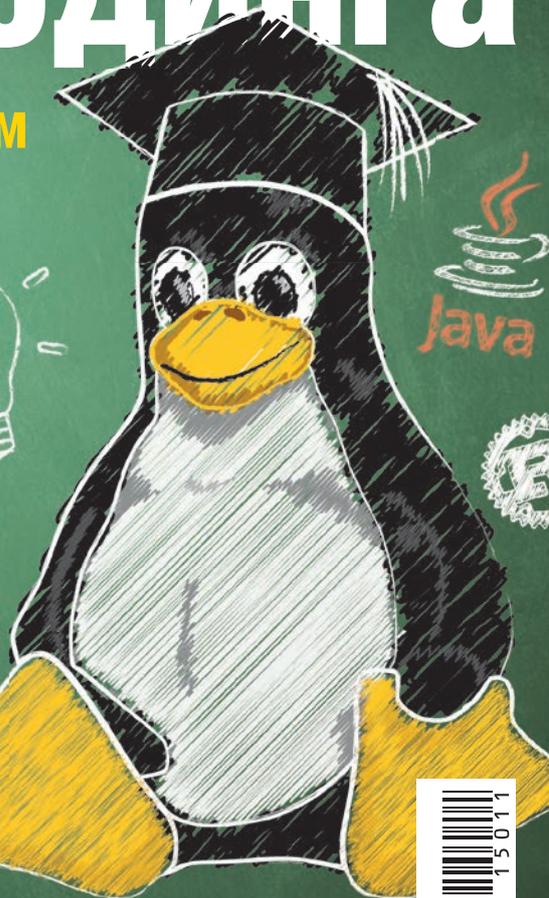
Академия кодирования

Все необходимое, чтобы стать классным разработчиком открытого кода

- ✓ Изучите Python сегодня!
- ✓ Создаем web-приложения
- ✓ Программируем на Raspberry Pi



Linux Rulez!



Java



```
for j in range(500) print('Linux Rulez!')
if sys.platform != 'linux': sad = True
```



Пропегандист-технар

« Существует масса возможностей улучшения производительности. »

Сара Новотны — о Nginx и OSCON с. 40

I ♥ Linux



Сделайте себе убунтофон

» Как залить Ubuntu Touch на свой Nexus



Копии файловой системы

Руководство по снимкам

» Файлы в надежных руках резервного копирования Linux



Приватность ZeroTier

Личная сеть VPN

» Храните анонимность онлайн по новейшей системе P2P



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

Linux center
www.linuxcenter.ru



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ

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 этот раз мы попросили наших экспертов поведать их любимые истории про кодирование, или как они начали программировать. И вот вам шесть рассказов о *Hello World*...



Джонни Бидвелл

Как у многих, моим первым опытом было написание программ на Basic для единственного на весь класс компьютера BBC B. Функциональные клавиши позволяли раскрашивать вывод, добавляя новую грань к операторам PRINT в цикле. Увы, аргументы этой команды *здесь не воспроизвести...*



Нейл Ботвик

Я начинал на университетской большой ЭВМ и Algol68, а потом «поднялся» до Pascal, но, в отличие от моих друзей, Fortran и Cobol сумел избежать. Это было немного давно [Ред.: — До н.э.?] — мало того, что мы употребляли перфокарты, так еще и первой нашей программой была *Salve Mundus*.



Алекс Кэмпбелл

Из области занимательных фактов: я начал программировать в институте, замахнувшись на компьютерный курс для продвинутых. Мы изучали C++, и я провалил тест, поскольку не мог уразуметь указатели... Ей-Богу!



Ник Пирс

Самая суть кодирования исключает меня из опроса. Могу лишь воскресить свои душераздирающие чувства, когда я в одном экземпляре сохранил на кассете игру с приключениями на Spectrum Basic и потерял ее... Мои родители всерьез опасались обезвоживания моего организма — так я рыдал.



Александр Толстой

Давным-давно (дамы еще не млели от моей бороды) я пробовал компилировать старую программу на новом GCC — понятно, безуспешно. Тогда-то я и выучился превращать в комментарии строки из Makefile и удалять спорные части исходника. Вот и все мои навыки кодирования по сию пору...



Валентин Синецын

А я уникален — моей первой программой был не *Hello World*. По-моему, она спрашивала у пользователя: «Ваше любимое блюдо?» и печатала: «Да, я тоже люблю \$dish». Впрочем, она была на Бейсике для БК-0010, и никакой интерполяции строк там еще не было...



Тридцатилетний спор

» На днях Windows исполнилось 30. Согласен, не тема для нашего журнала, но повод задуматься есть. Все эти годы вокруг этой системы не утихают споры, переходящие в «священные войны». Точнее сказать, трех этих систем: графической оболочки MS-DOS (первое десятилетие), 16-разрядной однопользовательской ОС и «взрослой» операционки в «клиентском» и серверном исполнении. По моему мнению, росту популярности ранней Windows более всего поспособствовали конкуренты. WordPerfect проспал «WYSIWYG-революцию», а Apple не разрешил клонировать свою архитектуру. С тех пор и поныне «Винда» — оптимальная платформа для выполнения *MS Office*, в чем и состоит главный аргумент ее сторонников. Серверные системы не имели столь явных преимуществ, но MMM (Маркетинговая Машина Microsoft) смогла это преодолеть. Я встречал win-админов, уверенных, что DNS, Kerberos и LDAP изобрели в Редмонде, штат Вашингтон. Основной аргумент сторонников — единообразный (и по логике работы, и по дизайну) интерфейс управления ОС, службами и приложениями. Согласен, хотя лично мне этот интерфейс не кажется ни логичным, ни удобным. Вопрос привычки... Три с половиной века назад, после 30 лет войны, родилась «архитектура» современного государства. В итоге «тридцатилетнего спора» кристаллизовалась архитектура современной ОС — открытая, модульная и с опорой на свободные стандарты.

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/linuxformat

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

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

Знаете всё о Linux?

ДА

НЕТ

Станьте автором
в журнале Linux Format!
linuxformat.ru/avtoram.phtml

Зарегистрируйтесь
на сайте
shop.linuxformat.ru



СОМНЕНИЯ

Скачайте бесплатно
архивные PDF-номера
журнала с сайта
linuxformat.ru/archive
и загляните на
wiki.linuxformat.ru



Версия для iPad
и iPhone доступна
в App Store

Выберите вид подписки

PDF-версия
на 6 месяцев

990 ₺

PDF-версия
на 12 месяцев

1800 ₺

Печатная версия
на 6 месяцев

1890 ₺

Печатная версия
на 12 месяцев

3480 ₺

Выберите вид доставки

Оплатите

Читайте Linux Format!

Станьте Linux-гуру

PDF-версия журнала Linux Format подойдет для тех, кто:

- Заботится о соблюдении прав деревьев
- Любит читать с экрана
- Мечтает получать каждый номер в день выхода журнала
- Хочет бесплатно скачивать содержимое DVD-приложения к каждому номеру

Печатная версия Linux Format понравится читателям, которые:

- Любят читать бумажные журналы
- Хотят получить в подарок подписку на PDF-версию Linux Format...
- ...а также диск с архивом журнала 2005–2014 гг.
- Порадуются новинкам открытого ПО на DVD-приложении к Linux Format в каждом номере

Способы доставки

- Курьером «ГНУ/Линуксцентра» по Москве и Петербургу
- Курьерской службой СПСР по России
- Почтой по России заказной или простой бандеролью
- Самовывоз из офиса «ГНУ/Линуксцентра» в Санкт-Петербурге
- Через пункты выдачи интернет-магазинов iml.ru в 11 городах России: Санкт-Петербург, Москва, Екатеринбург, Калуга, Нижний Новгород, Орел, Ростов-на-Дону, Тверь, Тюмень, Челябинск, Ярославль

Способы оплаты

- По квитанции в любом отделении Сбербанка
- Яндекс.Деньги, Webmoney
- Пластиковой картой Visa/MasterCard
- Наличными в офисе «ГНУ/Линуксцентра»
- Безналичный (для юридических лиц)



«ГНУ/Линуксцентр»
Санкт-Петербург,
пр. Медиков, 5, корп. 7
(метро «Петроградская»)
(812) 309-0686
www.linuxformat.ru

Содержание

«Научи человека ловить рыбу — и прокормишь его на всю жизнь». Маймонид

Обзоры

BunsenLabs Hydrogen 14

Из пепла CrunchBang величаво воспарил новый релиз дистрибутива-легковеса на базе Debian — понадеявшись раздуть пламя водородом. Обычно это срабатывает.



» CrunchBang возродился в BunsenLab.

Chromixium 15

Вариация Ubuntu, которая собезьянничала ChromeOS... безумство, доселе неслыханное! Предложила ли она нечто новенькое и крутое?

VirtualBox 5 16

При упоминании Oracle вас передергивает? Не будьте узколобыми: *VirtualBox* и всегда был удобным, а релиз 5.0 внес новую лепту в виртуальную потеху.

Dell Venue 10 7000 17

Быть Android-планшетом или Android-ноутбуком? Это устройство никак не придет к выводу. А мы пришли: оно нам нравится! За вычетом ряда недостатков...



» Android-устройство на все руки? Проверим.

Intel Core i7-5775C 18

Бежать ли со всех ног за этим новым процессором или обождать появления Skylake 6000? Расслабьтесь.

The Magic Circle 19

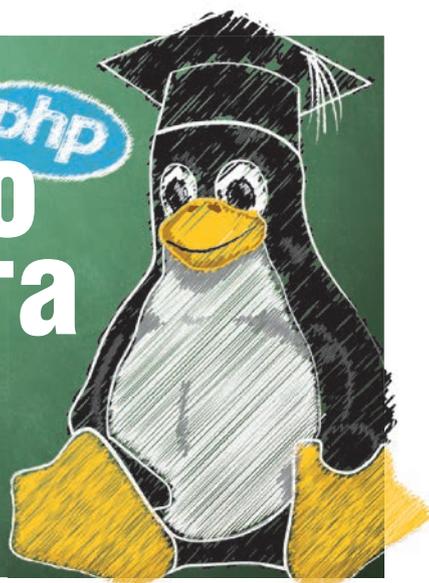
Играем в игру про то, как мы играем в игру, которая пытается чинить сама себя. Смех, слезы, взлом переменных. Реально смешно!

Back to Bed 20

В лунатическом состоянии прогуляемся по сюрреалистическим лестницам Мориса Эшера.

Тукс зовет в Академию КОДИНГА

Сделайте первые шаги к улучшению своих навыков программиста и станьте разработчиком СПО с. 30



Сравнение: Лучший хромбук-2015 с. 24



Интервью



Существует масса возможностей улучшения производительности в Сети.

Сара Новотны — о Nginx и OSCON с. 40

На вашем бесплатном DVD



UberStudent 4.1
 НОВЫЙ 64-БИТНЫЙ РЕЛИЗ!
 Ubuntu, пересобранный для студентов и преподавателей

wattOS R9
 Популярный дистрибутив с длительной поддержкой, который получил более 100 с 256 МБ ОЗУ

Эксклюзив!
 15 учебных видеокурсов от Ubuntu Project, Openstack и Linux с разбором

LINUX ЛУЧШИЕ ДИСТРИБУТИВЫ И БОЛЕЕ ТОГО! ВСЕ ДЛЯ БЫСТРОГО СТАРТА В LINUX

UberStudent 4.1 64-БИТНЫЙ, wattOS R9 32-БИТНЫЙ, OpenMediaVault 32-БИТНЫЙ

» Каждый месяц — только лучшие дистрибутивы

ПЛЮС: Эксклюзивные видеокурсы по Linux! с. 106

Побалуйте себя и любимых подпиской на LXF!



Доступно в AppStore!



www.linuxformat.ru/subscribe

Пользователям Raspberry Pi



Новости Pi 88

Что творится в Pi Foundation и какие проекты жаждут поддержки на Kickstarter.

Agobo 2 89

Мы разобрали и оценили новую робоплатформу.

Что такое Raspbian 90

Ускорьтесь с этим дистрибутивом на базе Debian: он ваш лучший друг на Pi и откроет вам мир СПО.

WhatsApp на Pi 92

Привяжите свой Pi к системе WhatsApp — и сможете не только писать сообщения, но и управлять Pi!



Ищите в номере

WordPress изнутри 46

На чем работает популярнейшая в мире система управления контентом? Естественно, на открытом ПО. Углубимся в нее и извлечем из нее максимум.



Приключения Tizen 50

OS Tizen — наследница нескольких проектов от нескольких компаний. В итоге ее заполучил Samsung, и теперь Tizen потихоньку распространяется на всевозможные устройства.

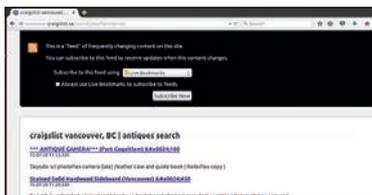
Академия кодига

Система: Сигналы 80

Учитесь подавать правильные сигналы и причаститесь к ядру, найдя с ним общий язык C. Сигналы бывают очень даже разные: некоторые и убить могут...

PHP: Агрегатор RSS 84

Ваши зачаточные знания о PHP расцветут на тренировке по комбинированию своих любимых новостных лент.



Постоянные рубрики

Новости 6

Вирусы вползли в роутеры, LibreOffice работает на оборонку, год настольного Linux настал в КНР, смартфоны служат дольше, военные билеты стали электронными, Линус неустрашим, СПО завоевывает Европу, а HP ставит Linux на коммутаторы.

Новости Android 22

Android-устройства запускают недетское ПО, BlackBerry приняла Android, Android махнули на MS Office, а Porsche обиделся на Google.

Сравнение 24

Рассмотрим поближе лучшие хромбуки 2015 года: Acer Chromebook 13, Acer Chromebook 15, Asus Chromebook Flip, Chromebook Pixel 2015, Dell Chromebook 11 2015.

Интервью LXF 40

Сара Новотны попала в OSCON благодаря случаю и сразу почувствовала себя там как рыба в воде.

Рубрика сисадмина 52

М-р Джолион Браун исследует свежие вести из мира контейнеров и завершает свою эпопею про Amazon Web Services криком о помощи. Не поможете?

Ответы 96

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
 Нейл Ботвик — про дистрибутивы, не загружающиеся с DVD, утерю адресной книги электронной почты, удаление Xfce из Ubuntu, преобразование файлов DJVU в PDF, восстановление неработающего загрузчика, торможение диска.

HotPicks 100

Отведайте горяченького! Лучшие в мире новинки свободного ПО: Calamares, Evilwm, Fish, gCAD3D, Geany, GNU Coreutils, Minetest, OpenClonk, OpenMediaVault, OwnCloud, Shotcut.

Диск Linux Format 106

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108

Не читали наш юбилейный выпуск? Ну, это позор, он же вообще бесплатный.

Через месяц 112

Битва года за звание лучшего дистрибутива: Mint против Ubuntu — кто кого соборет? И еще много чего интересного!

Учебники

Основы файловых систем

Снимки LVM — это просто 58

Нырнем в низкоуровневую систему резервного копирования Linux — моментальные снимки системы на лету.

Ключевые утилиты

GNU Coreutils 62

Освоим мощные инструменты командной строки Linux, начав с самых азов. Всё вам разжует и в рот положим.

Трюки Linux

Приватность ZeroTier 66

Нам всем есть что скрывать (не подумайте плохо...), вот и настроим анонимную сеть на PHP: это проще, чем вы думали.

Решение проблем

Сетевые инструменты 68

Ваш дистрибутив поставляется с кучей низкоуровневых инструментов, и мы вас научим с ними обращаться.

IP-телефония

Бэкапим перед стартом 72

Прежде чем названивать направо и налево, уберем свою систему. Резервная копия — это всегда умно.

Автоматизация

Макропроцессор M4 76

Даже сочинение оригинальных текстов можно автоматизировать — а иначе зачем человечеству и прогресс?



В ЭТОМ НОМЕРЕ: Вирусы в прошивках » СПО в Италии » Ноутбуки в Китае » Смартфоны служат дольше » ОС для силовиков » Речи Линуса » СПО шагает по Европе » Linux в коммутаторах

ВНИМАНИЕ: ВИРУС!

Прошивки под угрозой

Высокоустойчивое вредоносное ПО попало в маршрутизаторы.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

Замена прошивки маршрутизатора на модифицированную злоумышленниками версию больше не является чисто теоретическим риском. Специалисты Mandiant (дочерняя компания специализирующейся на кибербезопасности фирмы FireEye) выявили по крайней мере в 14 бизнес-маршрутизаторах Cisco в 4-х странах (Мексика, Украина, Индия, Филиппины), замену прошивки на бэкдор-версию. Бэкдор, названный SYNful Knock, предоставляет атакующим высокопривилегированный доступ к пораженным устройствам, сохраняющийся даже после перезагрузки.

SYNful Knock является модификацией ОС iOS, выполняемой на профессиональных маршрутизаторах и коммутаторах

от Cisco Systems. Специалисты Mandiant выявили его в моделях Cisco 1841, 8211 и 3825. Установка SYNful Knock стала возможной не из-за наличия уязвимости, но, вероятнее всего, по причине использования

Установка SYNful стала возможной не из-за уязвимости.

заданных по умолчанию паролей или похищенных учетных данных администратора. Cisco больше не продает вышеперечисленные модели, но нет гарантии, что уязвимость не затрагивает другие устройства.

Модифицированная прошивка реализует бэкдор-пароль для привилегированного доступа по протоколу Telnet и консоль, а также прослушивает команды, содержащиеся в специально созданных пакетах TCP SYN — отсюда и название SYNful Knock. Бэкдор способен использовать процедуру “knocking” для отдачи команд на ввод дополнительных вредоносных модулей в память маршрутизатора.

Позднее исследователи Shadowserver Foundation обнаружили еще как минимум 199 пораженных вредоносным ПО маршрутизаторов, 65 из которых находятся в США (всего же пострадали маршрутизаторы в 31 стране, в т.ч. 12 в Индии и 11 — в России).

ВНЕДРЕНИЕ СПО

Защищаться на свободе

Министерство обороны Италии переходит на LibreOffice и ODF.

Итальянские военные переходят на LibreOffice и Open Document Format (ODF). Как сообщает НПО Libreltalia Association, Министерство обороны Италии намерено в ближайшие полтора года установить LibreOffice на 150 тыс. своих рабочих станций, что будет вторым по величине развертыванием LibreOffice в Европе (крупнейшее — в МВД Франции, 240 тыс. ПК). Работы по миграции начнутся в октябре, а завершить их планируют в конце 2016 г. Libreltalia предоставит преподавателей для различных частей Вооруженных сил, а МО разработает серию онлайн-курсов в помощь переходу на LibreOffice (материалы будут опубликованы на условиях одной

из лицензий семейства Creative Commons). Решение о переходе — следствие принятого в июне 2012 г. закона, по которому свободное ПО с открытым исходным кодом для государственных учреждений страны должно быть приоритетным. МО является первой центральной правительственной организацией Италии, переходящей на открытый офисный пакет, но многие области и городские администрации, включая регионы Эмилия-Романья, Перуджа, Кремона, Мачерата, Больцано и Тренто, города Болонья, Пьяченца и Реджо-Эмилия уже используют LibreOffice.

Правительство Эстремадуры (Испания) подтвердило, что в его организациях

» Представитель МО Италии контр-адмирал Руджеро Ди Бьязе и президент Associazione Libreltalia Соня Монтедживе после подписания соглашения о сотрудничестве.



здравоохранения 10 тыс. ПК работает с открытыми офисными приложениями, и планирует сделать то же для 22 тыс. своих ПК. В Мюнхене (Германия) на LibreOffice переведены 17018 рабочих станций.

УЖЕ С LINUX

Где у нас год настольного Linux?

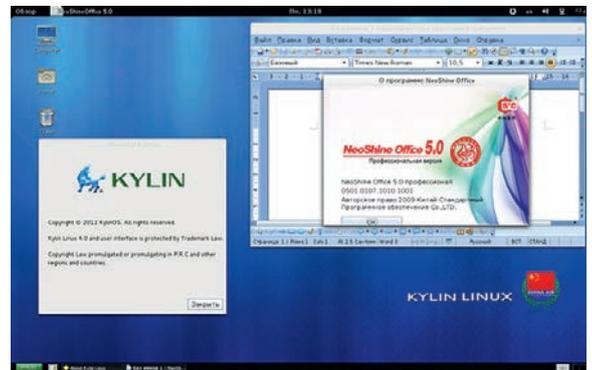
Dell и Hewlett-Packard продают в Китае ПК с предустановленной ОС GNU/Linux.

Издание *The Wall Street Journal* (WSJ) цитирует слова руководителя отделения Dell в Китае: «Этот год стал годом настольного Linux в Китае». Дистрибутив NeoKylin Linux поставляется на 42% продаваемых компанией в этой стране ПК, предназначенных, в первую очередь, для коммерческих и государственных структур (сферы, в которых специализируется Dell). Hewlett-Packard также поставляет в Китай компьютеры с предустановленным NeoKylin, однако конкретные цифры издания не приводит.

В 2010 г. компания China Standard Software (разработчик ПО для государственных нужд) и Оборонный научно-технический университет НОАК объединились с целью создания NeoKylin как безопасной альтернативы иностранному программному обеспечению (т.е. Windows). По мнению издания *TechinAsia*, NeoKylin основан на Ubuntu Kylin, разработанном Canonical для Китая.

Западные компании могут вести бизнес в Китае, но только являясь партнером или инвестором китайской фирмы. Как отмечает WSJ, Dell заявила о сотрудничестве с местной компанией Kingsoft по развитию «облачных» компьютерных решений, хотя пока не преодолены разногласия о владении пользовательскими данными, которые будут там храниться. Китайские компании все чаще используют иностранные инвестиции и разрабатывают продукты, успешно конкурирующие с западными аналогами; хороший пример тому — именно конкуренция Linux vs Windows. Да, глобально Linux занимает крохотную долю рынка — всего 1,6%, согласно NetMarketshare, оставаясь на этом уровне в течение многих лет. Однако он нашел свои специализированные ниши в определенных областях, таких как работа на суперкомпьютерах Китая Tianhe.

WSJ также отмечает значительное распространение пиратских копий Windows



➤ Исходный Kylin Linux от Canonical, на основе которого разработчиками КНР создан NeoKylin Linux.

в ряде стран и задается вопросом: возможно, Dell продает ПК с NeoKylin, который затем заменяют на пиратскую версию Windows? Однако, с учетом предназначения этих ПК для госструктур, такое предположение выглядит маловероятным.

ГЛЯДИШЬ, ЭКОНОМИЯ

Срок службы возрос

Потребители стали дольше использовать смартфоны, планшеты и ПК.

Потребители стали пользоваться своими планшетами, ноутбуками и смартфонами в течение более длительных периодов — к такому выводу пришла компания Gartner, и прогнозирует глобальное сокращение поставок всех типов устройств в этом году на 1%. Новый прогноз диаметрально противоположен ситуации трехмесячной давности, когда исследовательская фирма предсказывала общий рост поставок на 1,5% по сравнению с 2014 г. «Активность, направленная на замену всех типов устройств, снизилась, — пояснил аналитик компании Gartner Ранджит Атвал [Ranjit Atwal]. — Пользователи продлевают сроки службы своих устройств или не хотят менять их вовсе». Еще сильнее, на 13%, в 2015 г. упадут поставки планшетов: пользователи 7- и 8-дюймовых устройств такого типа все реже меняют их. Даже смартфоны используются потребителями дольше, несмотря на то, что их поставки, как ожидают, увеличатся в 2015 г. на 14% по сравнению с прошлым годом.

Для сравнения, поставки ноутбуков и настольных компьютеров снизятся в 2015 г. на 7,3% по отношению к 2014 г. Однако как Gartner, так и другие аналитики предсказывают резкое увеличение к 2017 г. продаж ПК с Windows 10, работающих на чипах Intel Skylake (уже в этом году Gartner предсказывает их рост на 4%).

Смартфон является более «персональным» устройством, чем ПК.

В числе 2395 млрд устройств, поставка которых ожидается в 2015 г., мобильные телефоны составят 1,9 млрд, а в 2017 г. — 2 млрд. При этом смартфоны в последние годы показали резкий всплеск среди мобильных телефонов: с 54% всех телефонов в 2013 г. до 75% в 2015 г. В 2017 г. смартфоны займут 89% рынка мобильных телефонов.

Такой успех объясняется ростом производительности и снижением стоимости (последнее прежде всего относится к моделям на Android). К тому же смартфон является более «персональным» устройством, чем ПК, которым нередко пользуются несколько человек.

Повышение вычислительной мощности смартфонов с большими экранами, так называемых «флаблетов [phablets]», также повлияло на снижение продаж планшетов. А определенная категория потребителей при выборе между ПК и смартфоном отдает предпочтение последнему из-за более низкой цены. Тем не менее, нынешние смартфоны не заменяются с такой же быстротой, как это происходило три–пять лет назад. Так, цикл замены iPhone возрос с 21,7 месяца два года назад до 27,4 месяца в этом году, что вынудило Apple реализовать программу Upgrade Program, позволяющую купить iPhone в рассрочку (нечто подобное, как ожидается, вскоре предложит и Samsung).

Электронные «военные билеты»

ОПК готова серийно поставлять силовикам защищенную ОС «Заря».

Объединенная приборостроительная корпорация (ОПК) готова начать поставки обновленной операционной системы «Заря», позволяющей безопасно работать с секретными сведениями и персональными данными. Система прошла все необходимые испытания и получила разрешение на серийное производство.

«В составе корпорации разработку „Зари“ по заказу Минобороны России ведет Центральный НИИ экономики, информатики и систем управления. В этом году ОС успешно завершила госиспытания и признана готовой к серийному производству, она соответствует 3-му классу защиты от несанкционированного доступа и 2-му уровню контроля отсутствия недеklarированных возможностей», сообщил директор департамента Объединенной приборостроительной корпорации Александр Калинин.

Такие ОС могут применяться как для полноценных рабочих мест, так и в качестве

основы для автоматизированных систем управления, для создания современных центров обработки данных (ЦОД) самого разного масштаба и уровня. Техника на базе этой ОС позволяет работать с информацией даже под грифом «совершенно секретно», что подтверждено сертификацией в системе Минобороны России.

Потенциальные потребители «Зари» — силовые ведомства.

«Подобное сочетание характеристик и возможностей на рынке сертифицированных защищенных операционных систем сегодня найти трудно, — говорит первый заместитель генерального директора ЦНИИ ЭИСУ Константин Солодудин. — „Заря“ учитывает опыт эксплуатации защищенных систем предыдущего поколения,

и архитектурно является развитием линейки военных операционных систем мобильной системы Вооруженных сил. Это существенно облегчает создание новых автоматизированных систем и портирование специального программного обеспечения из среды МСВС, позволяет не переучивать специально персонал для работы с новой системой».

Потенциальные потребители «Зари» — силовые ведомства, оборонный комплекс, коммерческие структуры, работающие с государственной тайной и персональными данными. На «Заре», в частности, будут работать ЦОДы, которые обслуживают создаваемую сейчас в России систему электронных «военных билетов» — персональных карт, содержащих всю учетную информацию о военнослужащем, включая состояние его здоровья.

Проект реализуется под руководством Генерального штаба Вооруженных сил Российской Федерации.

ГОВОРИТ ОСНОВОПОЛОЖНИК

А случись что...

Линус Торвалдс не боится попасть под автобус.

На проходившей в первой декаде октября в Дублине конференции Linuxcon EU Линус Торвалдс обсудил состояние Linux со своим давним другом, главным специалистом по Linux и Open Source компании Intel Дирком Хонделом [Dirk Hohndel].

По поводу безопасности Торвалдс отметил, что хотел бы увидеть еще более автоматизированную систему тестирования, поскольку в ядре существует слишком много мест для ошибок, чтобы людям было по силам проверить их. Однако в целом, по его мнению, ситуация с безопасностью ядра достаточно хороша, хотя Торвалдс остается реалистом. «Специалисты по безопасности всегда будут недовольны», считает он.

Когда речь зашла об оборудовании, Торвалдс заставил Хондела вздрогнуть, открыто расхваливая ARM. Отметив, что соперник архитектуры Intel делает огромные шаги, Торвалдс предсказал, что 2016 год

будет годом ноутбуков на ARM. «Не думаю, что мой работодатель одобрит такое предсказание», заметил на это Хондел.

Торвалдс не видит никаких проблем в поиске людей, готовых внести свой вклад, когда дело доходит до привлечения разработчиков ядра Linux. «Взяться за ядро и сделать небольшие патчи — довольно просто, хотя и немного страшновато. Меня больше беспокоит следующий шаг и поиск людей, которые уйдут от тривиальных патчей, став активными майнтейнерами». Торвалдс добавил, что увлеченный разработчик, который берется за ядро, а затем уходит, не очень полезен в долгосрочной перспективе. Действительно полезными Торвалдс назвал тех, кто готов еженедельно заниматься сопровождением все семь дней в неделю, хотя и признал, что это тяжело. Одним из способов облегчить деятельность майнтейнера является командный подход, который уже практикуется при работе над рядом подсистем ядра.



› Линус Торвалдс, отец-основатель и главный разработчик Linux.

Сам Торвалдс — майнтейнер ядра в целом, и возникает вопрос: что будет, если он, скажем, попадет под автобус? По словам Хондела, он уже просил Торвалдса быть осторожнее при пересечении улиц по пути к месту проведения мероприятия. «Там, где я живу, нет никаких автобусов, — ответил на это Линус. — А здесь они ходят по неправильной стороне дороги (в Ирландии левостороннее движение), и целят в тебя».

7 советов быстрого поиска работы от hh.ru

- 1. Определите цель.** Решите, кем вы хотите работать, как бы смешно это ни звучало. Точно сформулируйте вашу должность. Работодатель не найдет вас, если название резюме будет общим: «Менеджер» или «Начальник». Лучше уточните: «Менеджер по закупкам» или «Начальник строительной бригады».
- 2. Узнайте о своих способностях.** Вы все еще в поиске своего призвания? Пройдите онлайн-тест «Профориентация»* hh.ru/article/proforientation_promo и узнайте, какая работа вам больше всего подойдет.
- 3. Составьте резюме.** Сделать это на hh.ru легко. Главное — заполните все предлагаемые поля. Уделите особое внимание опыту работы и вашим достижениям — так вы покажете работодателю вашу компетентность.
- 4. Настройте процесс.** Подпишитесь на подходящие вакансии и получайте самые свежие на почту. А также скачайте мобильное приложение HeadHunter, чтобы искать работу в любое время, в любом месте.
- 5. Действуйте.** Откликайтесь на все интересующие вас вакансии. Пишите сопроводительные письма работодателю, поясняя, почему вас интересует эта вакансия.
- 6. Сделайте резюме заметным.** hh.ru/applicant/services Подключите «Яркое резюме»*, чтобы выделить резюме цветом, и «Автообновление»*, чтобы поднимать его в результатах поиска. Работодатели обратят на вас внимание.
- 7. Подготовьтесь к собеседованию.** Поздравляем, вас пригласили! Самое время подготовить ответ на вопрос: «Почему мы должны взять именно вас?». Узнайте максимум информации о компании и подготовьте небольшую речь о том, какой вы классный специалист.

И помните, что работа найдется для каждого!

ДАВНО ПОРА

Пустите в Европу

ЕК увеличивает долю Open Source в разработке ПО.

Для разработки публично распространяемого ПО Европейская комиссия (ЕК) стремится прежде всего использовать инструменты с открытым исходным кодом. С их помощью создана уже большая часть ПО ЕК, и за три года Генеральный директорат ЕК по информатике (DIGIT) намерен добиться «приоритета Open Source» во всех новых проектах ПО в Евросоюзе — так заявил DIGIT в начале октября на конференции Mindtrek OpenMind в Тампере (Финляндия). «Мы хотим реально содействовать большему распространению открытого кода, — заявил глава сектора DIGIT Пьер Дамá [Pierre Damas], представший на конференции стратегию OSS. — Это позволит повысить наши уровни адаптации и участия, и таким образом управлять движением вперед». В следующие три года DIGIT намерен изменить выбор инструментов, используемых ЕК для сотрудничества

онлайн, включая управление контентом, форумы и социальные сети. Так, уже запланировано перевести все сайты ЕК на открытую систему управления контентом *Drupal*.

ЕК выпускает ПО под свободной лицензией EURL (European Union Public License); примеры — CEF eID, eSignature и eDelivery. В том же ряду и принятие решений на базе Linux для Центров обработки данных ЕК, здесь стратегия сфокусирована на виртуализации. Пока на десктопах ЕК в основном проприетарные приложения (кроме браузеров и некоторых других инструментов). DIGIT собирается изменить ситуацию и здесь — например, путем трансформации и диверсификации своего ПО для автоматизации делопроизводства. Кроме того, структуры ЕК, отвечающие за выбор продуктов, могут сами определять свою политику, в зависимости от функциональности, общей стоимости владения и рисков.

LINUX ДЛЯ СЕТЕЙ

OpenSwitch от HP

Компания Hewlett-Packard анонсировала ОС на базе Linux.

С OpenSwitch в первую очередь предназначена для работы на сетевых коммутаторах и позволит сетевым инженерам «неограниченно внедрять инновации», а также упростит и ускорит создание спецсетей. Данная открытая платформа развивается при участии сообщества и поддержке компаний Acston, Arista, Broadcom, Intel и VMware, и ликвидирует привязку к вендору при развертывании специализированных сетей.

«Открытое ПО с его сообществом разработчиков ускоряет инновации и улучшает стабильность программной платформы, обеспечивая более мощную инфраструктуру для поддержки особых потребностей бизнеса и web-масштабных сетей», утверждает Марк Кэрролл [Mark Carroll], главный инженер HP Networking. Открытие инфраструктуры сетевой среды снимет проблемы кросс-совместимости проприетарных сетевых архитектур, скажем, при слиянии компаний или, как в случае HP, надвигающемся разделении. HP не понаслышке знакома с сообществом СПО,

и является одним из крупнейших участников создания ядра Linux и программной среды OpenStack. Первыми поддержат OpenSwitch открытые сетевые коммутаторы HP Altoline. Код новой ОС доступен в Git-репозитории: <http://git.openswitch.net/cgit>.

OpenSwitch поставляется с поддержкой протокола L2/L3, открытой облачной БД для персистентной и эфемерной конфигурации, инструментами настройки (типа CLI, REST, Puppet/Chef и *Ansible*) и системной БД для управления межмодульными коммуникациями — через web-интерфейс или CLI по наброскам проекта Quagga (также создавшим протоколы маршрутизации). **LXF**



➤ Новая ОС на базе Linux от HP упрощает создание специализированных сетей.

Новости короткой строкой

➤ На конференции DebConf решили прекратить формирование ISO-образов на CD для всех будущих обновлений Debian, начиная с Debian Testing и Debian 9.0 Stretch.
Источник: lists.debian.org

➤ ОАО «НПО РусБИТех» стало первой российской компанией, официально вошедшей в Консультационный совет The Document Foundation.
Источник: astra-linux.com

➤ Разработчики OpenMandriva создали свою страницу "OpenMandriva Games" с пошаговым руководством по запуску в Linux игр для Windows (типа *Dota 2*, *Left 4 Dead 2*, *Heroes of Newerth*) через Steam on Linux или PlayOnLinux.
Источник: games.openmandriva.org

➤ Creative Commons признала одностороннюю совместимость лицензии CC-BY-SA 4 с GPLv3.
Источник: creativecommons.org

➤ До конца 2016 г. Mozilla прекратит поддержку в Firefox плагинов с интерфейсом NPAPI (в т.ч. Silverlight и Java), оставив только поддержку Adobe Flash.
Источник: blog.mozilla.org

➤ Linux SPARC от Oracle строится на пакетах Oracle Linux 6 для x86_64, адаптированных под современное оборудование SPARC.
Источник: lists.debian.org

➤ После более двух лет разработки вышла NetBSD 7.0, со сборками для 58 архитектур систем и 16 семейств CPU.
Источник: www.netbsd.org

➤ Новая ОС Redox, на основе философии UNIX и принципе «все есть URL», создана на языке Rust и распространяется под лицензией MIT.
Источник: www.redox-os.org

➤ Mail.ru открыла исходники приложения для мобильной навигации Maps.me, со средствами просмотра карт OpenStreetMap оффлайн.
Источник: corp.mail.ru

➤ Марк Шаттлворт намерен включить в штатную поставку Ubuntu 16.04 LTS поддержку ZFS (модуль ZFSonLinux).
Источник: news.slashdot.org

➤ Сара Шарп [Sarah Sharp] и Мэтью Гарретт [Matthew Garrett] покинули ряды разработчиков ядра Linux из-за их и Линуса некорректного поведения.
Источник: sarah.thesharps.us

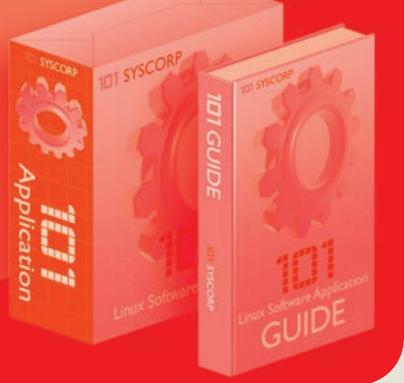
ВТР: Трансфузиология

«ВТР: Трансфузиология» — программный продукт, предназначенный для информатизации задач, связанных с забором, хранением и использованием донорской крови, ее компонентов и кровезаменителей, а также специальных лечебных процедур, выполняемых в отделении переливания крови.



Преимущества системы «ВТР: Трансфузиология»

- Разработка на базе свободного программного обеспечения (ПО с открытым кодом), что соответствует политике импортозамещения.
- Трехуровневая архитектура с тонким клиентом.
- Возможность интеграции с защищенным хранилищем персональных данных.
- Возможность проследить трансфузионную информацию от донора до реципиента и обратно.
- Предотвращение ошибок персонала на пути «от вены до вены» путем обеспечения информационного сопровождения и контроля.
- Эффективная организация планирования, распределения и подбора компонентов крови, анализа результатов и осложнений, проведения научных работ.
- Автоматическая загрузка результатов проведенных исследований из ЛИС (клинико-диагностической и бактериологической лабораторий и лаборатории цитогенетики).
- Автоматическое получение заявок на кровь и лечебные процедуры из МИС и передача обратно данных по выданным компонентам крови с прикрепленной этикеткой и результатов проведения лечебных процедур.
- Возможность проведения лечебных процедур, фиксации данных о полученных биоматериалах в рамках процедуры, печать процедурных листов.
- Автоматическая выгрузка данных по пациентам и отводам в ЕДЦ.
- Возможность оповещения доноров о приходах на сдачу крови и рассылки другой информации через СМС.
- При выездах сохранение связи с локальной БД путем использования «веб-клиента».



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Antergos: Наследие дедов

Дистрибутив Arch Linux — один из самых богатых репозиториями со свежайшими версиями пакетов. Но использование этого богатства затруднено отсутствием простого способа развертывания системы: у Arch нет инсталлятора, установка его выполняется из командной строки. Эта досадная для многих применителей особенность породила ряд «юзерофильных» клонов, один из которых — Antergos. Это галисийское слово на русский язык можно перевести как «дедовский»: он сохраняет полную двоичную совместимость с ArchLinux'ом. Однако снабжен простым и гибким графическим инсталлятором, запускаемым из Live-среды Gnome 3 или непосредственно. Он дает на выбор любой десктоп (Cinnamon, Gnome, KDE, Mate, *OpenVox* или *Xfce*), вместе с набором штатных приложений, и дополняется *LibreOffice*, *Firefox*, *Steam*, шрифтами Google Fonts плюс поддержкой репозитория AUR, в котором всего есть даже больше, чем в Греции... то есть в Ubuntu'вских PPA.

Для управления пакетами Antergos наследует систему *pacman*, в сопровождении графической оболочки *Pamac*. Она позволяет легко и быстро находить, устанавливать и удалять пакеты, а также подключать и отключать AUR: нет нужды выискивать отдельные репозитории PPA, как в Ubuntu, или рыскать по сайтам разработчиков в поисках дополнительного софта. Что и делает Antergos подходящим выбором как для начинающих применителей, так и для тех, «которые с претензиями». alv@posix.ru

Сегодня мы рассматриваем:

BunsenLabs Hydrogen 14
Из очередного пепла восстал очередной дистрибутив. Гори, гори, ясно, желаем мы этой горелке Бунзена. Но вообще-то она и без нас не коптит.

Chromixium 15
Модный тренд — перебираться на онлайн-приложения. А если вам их мало? Тогда получайте дистрибутив, который сочетает Web с настольной работой.

VirtualBox 5 16
С бурным ростом контейнерных технологий *VirtualBox* оттеснили было на задний план. Но старая гвардия не сдастся! Новый релиз доказывает, что он еще поборется.

Dell Venue 10 7000 17
Легким движением руки вы можете превратить этот планшет с замечательно ярким экраном в рабочую машину. Побольше бы ему оперативной памяти...

Intel Core i7-5775C 18
Этот процессор как бы от ворон отстал, но к павам не пристал. Не спешите его приобретать — есть смысл дождаться чего-то получше, тем более что Skylake не за горами.

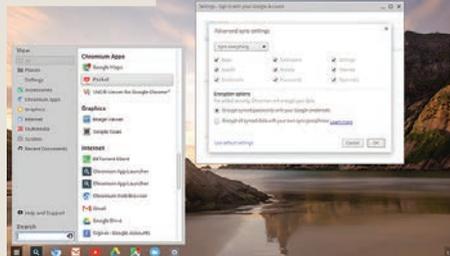
The Magic Circle 19
Мета-мышление или дурная бесконечность? Игра, которая воспроизводит игру, которая сама себя воспроизводит и починает на ходу, под перебранку разработчиков.



➤ Органический дисплей дает точнейшие оттенки черного.

Back to Bed 20
Игру придумали бравые студенты — таким не привыкать отыскивать путь в койку на автопилоте, в бессознательном состоянии. И интервью вполне себе сюрреалистический.

Chromixium



➤ Чего только ни придумают: перед вами — гибрид Ubuntu и ChromeOS!

The Magic Circle



➤ Рубануть мечом по ядовитому с виду грибу или, наоборот, с ним подружиться?

Сравнение: Хромбуки с. 24

Acer Chromebook 13



Acer Chromebook 15



Asus Chromebook Flip



Chromebook Pixel 2015



Dell Chromebook 11 2015



Звезда нетбуков вспыхнула — и закатилась: их, похоже, вытесняют хромбуки. Надо срочно разобраться, чем же, собственно, хороши эти придатки Всемирной паутины.

BunsenLabs 8.1

Юный дистрибутив на базе Debian подбирает оставшиеся кусочки CrunchBang, наследуя легкость для отважных поклонников *apt-get* вроде **Алекса Кэмпбелла**.

Вкратце

» Легковесный дистрибутив на базе Debian, преемник CrunchBang Linux. Близок к CrunchBang++.

Свободные и открытые проекты живут в дарвинистском электронном мире: одни успешны и живут долго; другие умирают еще до полноценной реализации; а третьи умирают после хорошей жизни. В феврале закончилась разработка CrunchBang Linux (часто сокращаемо как #!). Если сегодня заглянуть на <http://crunchbang.org>, то сайт отсылает на проект BunsenLabs, официально называя этот дистрибутив преемником #!.

Как и #!, Bunsen основан на Debian GNU/Linux. Версию 8.1, на Debian Jessie, уже окрестили Hydrogen, и, как и самый распространенный элемент во Вселенной, она минималистична.

Предлагаемое по умолчанию программное обеспечение — легковесное. Как и #!, Bunsen избегает окружения рабочего стола в пользу оконного менеджера *Openbox*. Панель задач *Tint2* появляется в верхней части стола, а *Conky*, в простой конфигурации, украшает правую сторону, показывая существенно важные параметры системы и удобный список горячих клавиш.

Пользователи, не привыкшие к *OpenVox*, найдут этот интерфейс довольно скудным и могут удариться в панику, глядя на меню. Простой щелчок правой кнопкой мыши открывает текстовое меню, где есть все основные приложения. Меню — это простые XML-файлы, которые при необходимости легко настроить. Пользователи могут редактировать свои меню в `~/config/openbox/rc.xml`, а общесистемная конфигурация находится в `/etc/xdg/openbox/rc.xml`.

Однако рабочий стол — не единственное, что ориентировано на облегченный



» Вид BunsenLabs довольно скуден (или дзен, в зависимости от точки зрения) по сравнению с другими, вследствие минималистского подхода *OpenVox*.

профиль. Основные утилиты рабочего стола (типа файлового менеджера *Thunar*) заимствованы из *Xfce*, а *Iceweasel* (производный от *Firefox*) и простой текстовый редактор *Leafpad* замыкают список имеющихся приложений. Все это означает, что Bunsen быстр даже на старом оборудовании. Интерфейс выглядит четким; хотя он наследует тусклую наружность CrunchBang и CrunchBang++, его внешний вид можно изменить.

Поджиг для горелки

Установщик выглядит как стандартный из Debian, без дополнительной символики. Менее 700 МБ ISO-образа легко помещается на установочный 1-ГБ USB-брелок. Компактный инсталлятор имеет свою цену: потребуется подключение к Интернету для всего, кроме самого необходимого. Если завершить установку в автономном режиме, нескольких приложений, показанных установленными в меню *OpenVox*, не будет, как не будет записей о вашем репозитории в `/etc/apt/sources.list`.

При первой загрузке в окне терминала откроется скрипт и спросит вас, хотите ли вы установить такие пакеты, как *LibreOffice*. В меню *OpenVox* также имеются ярлыки, запускающие скрипты для установки таких программ, как *GIMP*, *Inkscape* и т.д. Все это вам не удастся, если ваш файл `sources.list` заполнен неправильно.

Если вы окажетесь в таком положении, на <http://debgen.simplylinux.ch> есть генератор источников Debian. На <http://pkg.2ion.eu> лежит инструкция, как вручную добавить

репозиторий Bunsen. После добавления источников все должно прийти в согласие с миром. Следует отметить, что для Bunsen непригодны репозитории Ubuntu. Об этом говорится во время установки и во время первого запуска скрипта установщика. Скрипт даже заходит так далеко, что заставляет вас набрать «Я понимаю», заостряя на этом внимание.

BunsenLabs — дистрибутив хваткий, но в нем всё может пойти наперекосяк, если вы проявите небрежность. Отсутствие официальной документации может выставить его трудным для новичков, но комбинация #!-форумов и интернет-изобилие знаний о Debian должны обеспечить достаточное руководство для тех, кто ищет. Опытный пользователь Linux насладится легкостью и простотой Bunsen. **LXF**



Свойства навскидку



Openbox

Не надо паники! Простое текстовое главное меню OpenVox открывается обычным щелчком правой кнопки мыши.



Может бабахнуть!

Использование репозитория Ubuntu — лишь одна из нескольких вещей, способных угробить BunsenLabs.

LINUX FORMAT Вердикт

BunsenLabs 8.1 Hydrogen

Разработчик: BunsenLabs
Сайт: <http://bit.ly/BunsenLabs>
Лицензия: GPLv3

Функциональность	7/10
Производительность	9/10
Удобство в работе	8/10
Документация	7/10

» Молодой, скоростной, неотяченный дериват Debian — это если его не поломать.

Рейтинг **8/10**

Chromixium 1.5

Жаждете уйти на web-приложения, но жаль бросать любимые настольные программы? **Шашанк Шарма** выяснит, дает ли Chromixium лучшее из обоих миров.

Вкратце

» Дистрибутив использует внешний вид Google Chrome OS, но с Ubuntu в качестве базы. Легкий дистрибутив пытается объединить облачные и настольные приложения. В результате получился быстрый и гибкий дистрибутив, хорошо подходящий для старого оборудования. Доступный только для 32-битных машин, он быстрее даже с 1 ГБ ОЗУ, но лучше больше. См. также: Chromium OS и Peppermint OS.

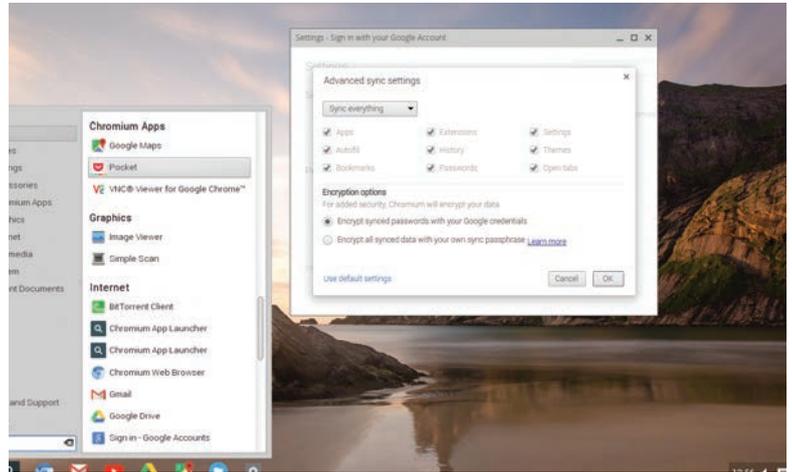
Chrome OS от Google хороша, если вам хватает онлайн-приложений. Chromixium предлагает расширить преимущества Chrome OS, грамотно добавляя возможности и функциональность обычной настольной ОС, что позволяет запускать и настольные программы.

Chromixium основан на релизе Ubuntu 14.04 LTS. Разработчик грамотно построил его, используя компоненты без холестерина, такие как оконный менеджер-полуплегковес *OpenVox*, и получив в результате дистрибутив, нетребовательный к ресурсам и пригодный для работы на старых ноутбуках и нетбуках. В соответствии с этим, он доступен только в виде 32-битного ISO с возможностью установки.

Но Chromixium не просто очередной дистрибутив на базе Ubuntu, использующий *OpenVox* ради экономии ресурсов. Разработчик отобрал каждый компонент, аккуратно увязал их вместе и отполировал дистрибутив до блеска. Chromixium включает куски из *Xfce*, *LXDE* и даже из *Gnome*. Дистрибутив использует композитный менеджер *Compton* и док-панель *Plank*, чтобы рабочий стол смотрелся опрятно и привлекательно.

Основным приложением дистрибутива является браузер *Chromium*, оснащенный плагином *Google PepperFlash*. Его можно использовать для доступа ко всем сервисам Google без каких-либо проблем. При желании можно войти в свою учетную запись Google, который потом синхронизирует все ваши настройки из различных интернет-служб Google.

По умолчанию пускатель [launcher] рабочего стола дистрибутива имеет иконки для доступа к Gmail, YouTube, Google-Диску



» Проект выпускает сервис-паки между релизами, чтобы установить важные обновления.

и Документам Google. Можно использовать панель поиска пускателя для поиска и добавления других онлайн-приложений и сервисов из интернет-магазина Chrome, а затем поместить их на док-панели для облегчения доступа.

В Web и за столом

Онлайн-приложения дополнены массой настольных и инструментами настройки; среди них — просмотрщик изображений *GPicView*, *Transmission*, *Brasero* и *Parole Media Player*. Можно расширить дистрибутив и установить больше программ через менеджер пакетов *Synaptic*. Менее опытные пользователи Linux могут использовать web-приложение для установки программ из каталога приложений Ubuntu.

Дистрибутив также включает целый ряд небольших изящных утилит для тонкой настройки системы; все они доступны в Панели управления Chromixium. Опять же, просматривая элементы Панели управления, видим намерение разработчика выбрать легкое программное обеспечение без ущерба для функциональности. Одним из примеров таких инструментов является *systemback* для создания и восстановления резервных копий и снимков системы.

Дистрибутив сфокусирован на web-приложениях, а для доступа к обычным настольным приложениям необходимо щелкнуть правой кнопкой мыши на рабочем столе. Впрочем, последний релиз включает меню *Cardapio*, которое можно перетащить из Панели управления в док-панель как стандартное меню.

Если вас не устраивает браузер *Chromium*, после установки можно переключиться на *Google Chrome*, следуя инструкции в вики. На машинах с менее чем 1 ГБ ОЗУ дистрибутив загружается дольше, прежде всего потому, что наряду с некоторыми другими инструментами и утилитами сразу запускается *Chromium*. Это также означает, что как только он начинает работать, web-приложения запускаются почти мгновенно.

Chromixium — отнюдь не первый дистрибутив, который попытался совместить обезжиренный характер облачной Chrome OS с удобством обычной настольной ОС. Но, благодаря вовлечению в процесс разработки активного сообщества пользователей, несмотря на то, что была только пара предыдущих релизов, он имеет признаки зрелого и стабильного дистрибутива. **LXF**



Свойства навскидку



Лучшее обоих миров

Здесь мирно сосуществуют онлайн- и традиционные настольные приложения.



Выбор компонентов

Один из козырей дистрибутива — разнообразие приложений и инструментов.

LINUX FORMAT Вердикт

Chromixium 1.5

Разработчик: Rich Jack
Сайт: www.chromixium.org
Лицензия: GPLv3 и др.

Функциональность	8/10
Производительность	7/10
Удобство в работе	8/10
Документация	6/10

» Сочетает простоту Chrome OS с мощью полноценного настольного Linux, но мало документации для новичков.

Рейтинг 7/10

VirtualBox 5.0

После почти пяти лет инкубации, последний релиз VirtualBox взволновал Шашанка Шарма множеством столь необходимых функций.

Вкратце

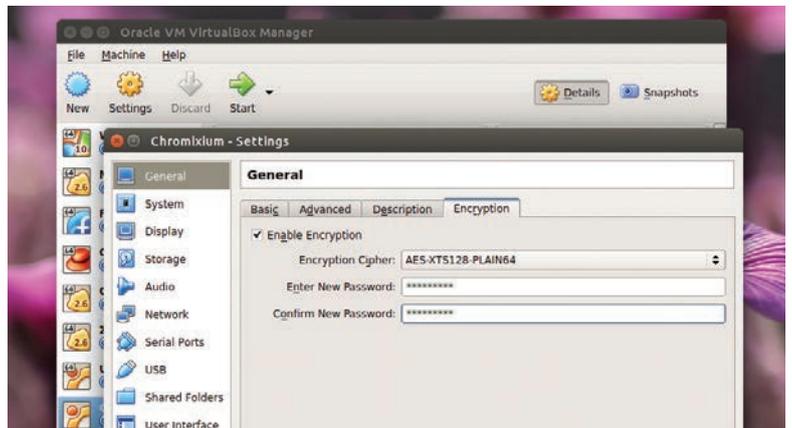
» Предназначенный для домашних и корпоративных пользователей, последний выпуск популярного настольного приложения виртуализации является переломным релизом *VirtualBox*. Потерпев серьезное поражение от таких альтернатив, как *Docker*, *VirtualBox 5.0* внес много улучшений и может похвастаться лучшей производительностью. Помимо полной поддержки USB 3.0, релиз предлагает поддержку паравиртуализации и шифрование образа диска для повышения безопасности. См. также: *QEMU* и *VMWare Workstation*.

Последний крупный релиз VirtualBox, v4.0, был вскоре после приобретения Sun Oracle'ом в 2010 г. С тех пор настоящее ПО виртуализации засиделось на минорных релизах и упустило концептуальные сдвиги в области виртуализации, в частности Docker'a и контейнеров в целом. Но *VirtualBox 5.0* доставляет веские аргументы в пользу решений виртуализации на основе гипервизора.

Новый релиз включает несколько новых функций, и не все из них принимают вид настраиваемых параметров в интерфейсе пользователя. Например, начиная с этой версии, *VirtualBox* предоставляет более широкий набор команд CPU для гостевой ОС. Приложения, работающие в виртуальных средах, смогут употреблять их для повышения производительности.

Изюминкой v5.0 является добавление методик паравиртуализации, а именно Hyper-V для Windows и KVM для Linux. Это должно значительно повысить производительность гостевой ОС, благодаря встроенной в нее поддержке виртуализации. Добавление паравиртуализации также должно привлечь больше пользователей, которые раньше использовали другие решения для создания виртуальных машин на базе KVM.

Другой важный момент — шифрование диска. С небольшим ущербом для производительности приверженцы конфиденциальности могут шифровать новые и имеющиеся виртуальные диски, используя алгоритм AES со 128- или 256-битными ключами шифрования. Но после включения этой функции вы теряете возможность перезагрузить свою машину без присмотра,



» Теперь *VirtualBox* позволяет запускать виртуальные машины в трех режимах: обычный оконный интерфейс, в отдельном окне или окне без панелей.

поскольку потребуется ввести пароль шифрования диска при ее загрузке. И учтите, что эта функция включается только после установки проприетарного расширения *VirtualBox Extension Pack*.

Прочие новые функции, зависящие от *Extension Pack*, включают поддержку устройств USB 3.0 и двунаправленный обмен. Гостевые ОС теперь могут напрямую распознать устройства USB 3.0 и работать на полной скорости 3.0. Интерфейс также позволяет настроить виртуальную машину для поддержки устройств USB 1.1, 2.0 или 3.0. Благодаря улучшенному двунаправленному обмену, можно использовать "drag-and-drop" между хостовой и гостевой системами на всех хост-платформах для гостевых Windows, Linux и Solaris.

Дополненный интерфейс

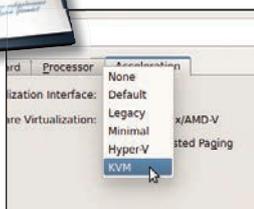
В целом интерфейс для создания и управления виртуальными машинами тот же; ряд разделов в установках виртуальной машины теперь расширили функциональность, с новыми настраиваемыми параметрами. Так, в разделе Система в вкладке Ускорение появилось выпадающее меню для выбора интерфейса паравиртуализации, а в разделе Хранение данных стало можно отмечать виртуальные диски с горячей заменой в дополнение к SSD-накопителям. Раздел Дисплей обзавелся новыми функциями на вкладке Захват видео: можно задать разрешение для записываемого видео, выбрать частоту от 1 до 30 кадров в секунду, а также битрейт от 102 до 1024 кбит/с. *VirtualBox* также оценивает размер записываемого видео по заданным параметрам.

В окне Настройка добавлен новый раздел «Интерфейс пользователя» для настройки такового для управления виртуальной машиной. Кроме включения или отключения полного меню, также можно при желании удалить конкретные подменю: например, сохранить меню Машина и удалить только функцию Сброс из интерфейса для конкретной виртуальной машины, чтобы заставить пользователей закрыть его должным образом.

В общем, в *VirtualBox 5.0* сделано достаточно, чтобы обосновать статус главного релиза. С паравиртуализацией и поддержкой USB 3.0 он получил новые возможности для начинающих и средних пользователей. Возможность запуска зашифрованных виртуальных образов является настоящим подарком для всех приверженцев безопасности данных. **LXF**



Свойства навскидку



Паравиртуализация

Гостевые ОС теперь могут использовать встроенную поддержку виртуализации вдобавок к интерфейсам KVM и Hyper-V.



Шифрование

Шифрование виртуальных дисков с промышленного стандарта шифрами AES в режиме XTS со 128- или 256-битными ключами.

LINUX FORMAT Вердикт

VirtualBox 5.0

Разработчик: Oracle Corporation
Сайт: www.virtualbox.org
Лицензия: GPL 2 и PUEL

Функциональность	8/10
Производительность	8/10
Удобство в работе	8/10
Документация	8/10

» Этот богатый функциями главный релиз много чего предлагает домашним и корпоративным пользователям, возвращая *VirtualBox* на пьедестал.

Рейтинг **8/10**

Dell Venue 10 7000

Кевин Ли пробует Android-планшеты, вообразившие, что они стали профессиональными благодаря добавлению модной клавиатуры.

Спецификация

- » **ОС:** Android 5.0
- » **СРУ:** Intel Atom Z3580 4-ядерный 2,3 ГГц
- » **ОЗУ:** 2 ГБ
- » **Дисплей:** 10,5 дюйма 1600×2560
- » **GPU:** PowerVR G6430
- » **Диск:** 32 ГБ
- » **Камеры:** 8 МП и 2 МП
- » **Связь:** Wi-Fi 802.11a/b/g/n/ac, Bluetooth 4.0, GPS
- » **Батарея:** до 7 ч, 7000 мА·ч
- » **Габариты:** 243,4×195,4×6,2 мм

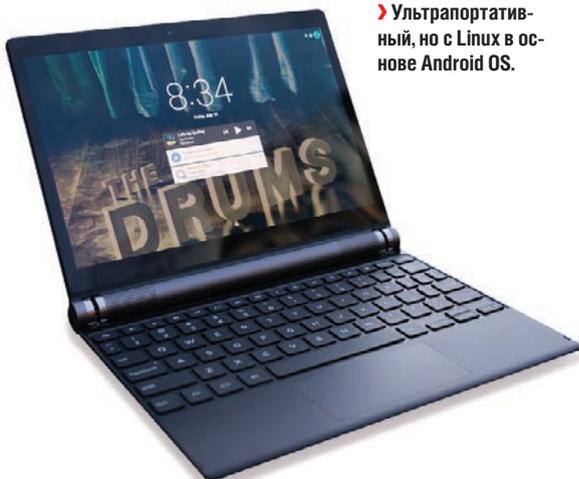
От Nexus 9 до Lenovo Yoga Tablet 2 было немало попыток создать Android-устройства для продуктивной работы. Нельзя винить производителей за попытки. Ведь обещание устройства, способного быть и личным мобильным кинотеатром, и офисом, заманчиво.

Venue 10 7000 является одним из таких устройств от Dell, и как часть линии Dell 7000 премиум-класса, этот 10,5-дюймовый планшет с 2560×1600 OLED дисплеем на голову выше типичного Android-планшета. Более того, четырехъядерный 2,3-ГГц процессор Intel Atom Z3580 дает этому планшету преимущества как рабочей машине. В то же время, он наделяет iPad Air 2 и Nexus 9 лучшим звуком динамиков и клавиатурой, на которой захочется работать.

Как и его меньший 8-дюймовый брат, Venue 10 7000 имеет тонкую алюминиевую рамку с цилиндрическим шарниром, выступающим с обеих сторон планшета. К сожалению, Dell с 10,5-дюймовым планшетом не поставляется с подставкой. Вместо этого для удержания планшета требуется дополнительно клавиатура Venue Bluetooth, чтобы использовать его в любом из дополнительных режимов. Клавиатура идет за £101, если не брать в комплекте.

Клавиатура закрепляется в канале ствола шарнира металлическими зажимами. Соединение настолько плотное, что планшет можно хорошенько встряхнуть, удерживая за клавиатуру, без боязни, что зажимы вылетят. Но не ждите, что Bluetooth-клавиатура будет работать, когда она не прикреплена к планшету, так как она не имеет собственного питания.

» **Ультрапортативный, но с Linux в основе Android OS.**



» **Конструкция Venue 10 делает клавиатуру необходимой.**

Качественный дисплей

Безусловно, самой привлекательной особенностью Venue 10 является его OLED-дисплей. Разрешение экрана 2560×1600 заставляет все выглядеть лучше: от цифровых комиксов и сайтов до фильмов и видео с YouTube. Важнее, что планшет просто выглядит лучше, потому что он дает потрясающе яркие цвета без какого-либо перенасыщения, что обычно досаждало у OLED-экранов. В то же время органический дисплей дает более точные оттенки черного, чем любой ЖК, что пригодится при охоте в темных сценах «Игры престолов».

К сожалению, недостаточные 2 ГБ личной памяти могут убить многозадачность на планшете. В любой момент времени, примерно половина ОЗУ планшета съедается службами Android, работающими в фоне, в том числе предустановленной копией антивируса McAfee. Запуск слишком многих приложений, таких, как Google Docs и Chrome, сразу ставит планшет на колени.

Dell оценивает длительность работы от батареи на Venue 10 7000 всего в семь часов, но в крайне редких случаях мы фактически смогли выжать еще более длительную работу без подзарядки — в 8 часов и 19 минут. Dell Venue 10 7000 опасно близок по цене к отличным ноутбукам, таким, как XPS 13 также от Dell. И это далеко

превосходит сумму, которую мы готовы выложить за комфорт Chromebook'a.

Безусловно, превосходящими конкурентами характеристиками Venue 10 7000 являются лучшая звуковая система и более длительное время автономной работы, плюс экран, дающий великолепное изображение — лучшее из всех. Тем не менее, этот планшет больше подходит для обычных задач вроде просмотра фильмов и web-страниц, чем для требовательной к нагрузке повседневной работы. А ведь гнались-то за производительностью, достойной рабочей машины? **LXF**

LINUX FORMAT Вердикт

Dell Venue 10 7000

Производитель: Dell
Сайт: www.dell.co.uk
Цена: £399 (с клавиатурой)

Функциональность	8/10
Производительность	7/10
Удобство в работе	8/10
Оправданность цены	7/10

» Отличный планшет, который тщится быть рабочим, но для многозадачности ему мало памяти.

Рейтинг 7/10

Intel Core i7-5775C

Первый процессор пятого поколения настольных процессоров Intel, наконец, появился... и огорчил **Дэйва Джеймса**.

Спецификация

- » **Название:** i7-5775C
- » **Сокет:** 1150
- » **Кэш:** 6 МБ
- » **Тип:** 64-битный
- » **SSE:** SSE 4.1/4.2, AVX 2.0
- » **Технология:** 14 нм
- » **Ядра:** 4
- » **Потоки:** 8
- » **Тактовая частота:** 3,3 ГГц
- » **Ускоренный режим:** 3,7 ГГц
- » **Требования по теплоотводу:** 65 Вт
- » **Макс. объем ОЗУ:** 32 ГБ DDR3
- » **Каналов:** 2
- » **Графический сопроцессор:** Iris Pro 6200
- » **Тактовая частота:** 300 МГц
- » **Макс. тактовая частота:** 1,15 ГГц
- » **Модулей:** 48
- » **OpenGL:** 4.3
- » **Мониторов:** 3
- » **Поддержка виртуализации:** VT-x, VT-d, VT-x EPT

Настольный процессор пятого поколения Intel, Core i7-5775C, наконец, оказался в нашем распоряжении. Привет, мистер Broadwell, почему вы так задержались? Если спросить у кого-нибудь из Intel, они скажут, что архитектура 14 нм была выпущена в 2014 г. И несмотря на это, первый полноценный процессор Broadwell мы получили только сейчас. Под словом «полноценный» мы имеем в виду оснащенный всеми возможностями четырехъядерный процессор с нормальной частотой и потенциалом для разгона.

С технической точки зрения Broadwell был выпущен в 2014 году. Точнее говоря, частично. Core M, двухъядерный многопоточный мобильный чип, был официально выпущен до Рождества, а несколько урезанных вариантов появились в ноутбуках весной. Означает ли это, что Intel решила уделять меньше внимания настольным процессорам? Не является ли это тревожным индикатором грядущего для семейства настольных процессоров Skylake? Если коротко, то нет.

В процессорах Broadwell всегда уделялось внимание энергосбережению, поэтому приоритет мобильных устройств стал неизбежен. И так как выпуск Broadwell еще задержался, влетев в выпуск следующего поколения Skylake, нет необходимости в выпуске полного диапазона процессоров для различных разъемов.

Тик-так

Итак, что же такое Broadwell? Это «так» для «тика» Haswell. Он позволяет ужать процесс производства до 14-нм чипа, по сути остающегося процессором, построенным по технологии Haswell. Благодаря этому чипы Broadwell получают внушительную энергоэффективность. И в первую очередь они интересны для разработчиков мобильных устройств.

Но они сохраняют некоторый интерес и для настольных пользователей, потому что с ними вы также получаете лучшую графику, реализованную в процессоре.

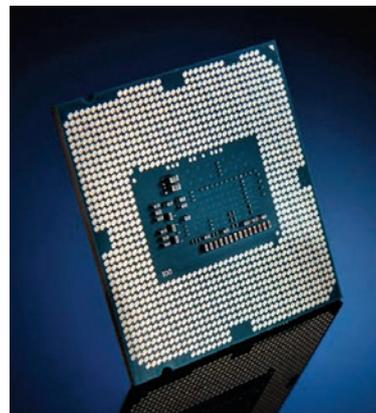
Это первое воплощение Iris Pro с разъемом, и в обоих процессорах i7-5775C и i5-5675C используется существенно улучшенное графическое ядро. По сравнению с компонентами HD Graphics 4600 в последних чипах Haswell «Devil's Canyon» количество модулей выполнения в Iris Pro 6200 увеличилось более чем в два раза.

В остальном вы видите классические показатели для Core i7 — четыре ядра с восемью потоками. Но поскольку это процессор класса C, вы получите тактовую частоту только в 3,3 ГГц с максимальным ускорением до 3,7 ГГц, хотя в обоих процессорах класса C оставлены незаблокированные умножители для разгона.

Причин обновляться с вашего текущего Haswell i7 и самых выдающихся процессоров i5, похоже, нет. Процессор i7-5775C значительно дороже i7-4790K, и с точки зрения номинальной тактовой частоты, а следовательно, и производительности в играх, это провал. Что касается тактовой частоты, тут тоже печально. Теоретически 5775C можно разогнать до 3,7 ГГц, но в наших тестах ни разу не удалось превысить 3,6 ГГц. Когда 4790K достигает 4,4 ГГц в номинальных тактовых частотах, это очень приличный отрыв в производительности. Он одинаков для всех наших тестов — тесты X264 отдали единодушную победу процессору последнего поколения.

Светлая сторона — показатели пропускной способности памяти. Процессоры Broadwell демонстрируют большие улучшения в этой области, обходя все, что мы видели за пределами древностей для сервера с процессорами Ivy Bridge-E.

Пиковая мощность с номинальными скоростями, которую мы наблюдали во время тестирования с большой нагрузкой — всего 104 Вт, а процессор нагрелся лишь чуть выше 50 °C. Учитывая низкую частоту, этого можно было ожидать. Но даже когда мы разогнали его до 4,2 ГГц, его температура все равно была не выше 60 °C, и он набрал не более 50 Вт дополнительной мощности. Эти улучшения действительно впечатляют.



» Новый i7 от Intel стоит около 300 ф.ст.

Этот восьмипотоковый процессор с трудом достигает мощности четырехъядерных процессоров последнего поколения. К сожалению, нам не удалось разогнать процессор дальше, просто потому что под рукой не было ничего теплоотводящего.

Этот процессор мог произвести впечатление в прошлом году и, возможно, даже сегодня, будь у него тактовая частота выше и цена ниже. Но этого нет, так что особых причин обращать внимание на данный процессор тоже нет. Да, у него впечатляющий графический сопроцессор. И да, возможно, версия Core i5 может служить обновлением с дешевого компьютера на Core i3 без отдельного графического сопроцессора. Но этот дорогой и малопроизводительный процессор найдет мало поклонников среди тех, кому нужно осязаемое увеличение производительности по сравнению со своим Haswell'ом. **LXF**

LINUX Вердикт
FORMAT

Intel Core i7-5775C

Разработчик: Intel
 Сайт: www.intel.com
 Цена: £ 299

Функциональность	9/10
Производительность	7/10
Удобство в работе	8/10
Оправданность цены	5/10

» Слишком мало, слишком поздно. Графика и энергоэффективность впечатляют, но производительность и цена — нет.

Рейтинг 5/10

Производительность ЦП	x264, кадров в секунду	Память, Гб/с	Игровой режим в среднем (мин.)	Мощность, Вт
Intel Core i7 5775C	45	20,14	44 (20)	104
Intel Core i7 4790K	53	17,73	52 (11)	195
Intel Core i5 4690K	40	17,72	52 (10)	122

The Magic Circle

Законченная игра о том, кто застрял в незаконченной игре, навела **Кристофера Ливингстона** на мета-мысль, что он не закончит этот обз...

Спецификация

- » **ОС:** Ubuntu 12.04 LTS, Steam OS и «большинство» дистрибутивов Linux
- » **СРU:** Intel Core 2 Duo, AMD Athlon X2 5200+
- » **ОЗУ:** 4 ГБ
- » **Диск:** 2 ГБ
- » **GPU:** Shader Model 4.0, 1 ГБ VRAM

The *Magic Circle* забрасывает вас в тестовую версию некой игры, которая уже годами находится в разработке, но все еще далека от завершения, и большинство геймеров, в наш век кампаний Kickstarter и Steam Early Access, это быстро просекут. То, что задумывалось как игра в стиле научной фантастики, превратилось в квест с мечами и магией, и продвигаясь по этому незаконченному миру, вы найдете несущие конструкции мозаичной космической станции, которые просматриваются сквозь завесу фантазии: роботы из былых времен соседствуют с волшебниками и гигантскими пауками.

Вскоре вы получите возможность внести изменения в игру посредством манипуляций с кодом. В частности, вы можете изменить поведение ее элементов. Например, захватив враждебное собаководное существо, вы сможете заставить ее стать вам не врагом, а другом. Назовите ей новых врагов, на которых она впоследствии будет нападать. Вы также можете полностью лишиться ее каких-либо качеств и оставить инертной, сохранив эти свойства в библиотеке, чтобы применить их к другим существам. Заметьте, ничто из этого не предполагает самостоятельного написания кода: все делается путем простого выбора слова из списка и вставки его в поле.

Сами напрямую вы эти свойства использовать не можете, и поэтому препятствия преодолеваются за счет изменения поведения вашего растущего зверинца. Скалы можно подвесить парить в воздухе, что обеспечит вам хороший плацдарм; зомби сделать огнестойкими, крыс наделять захватными лучами, а приемы рукопашной



» Сатира на игровую, где ваша задача — исправить недописанную игру.

вписать в код безобидным, самим по себе, грибам. Таким образом, у вас появляется несколько способов преодолеть препятствия, решить головоломки и победить врагов. Да и без головоломок выходит довольно весело: кто же устоит перед тем, чтобы отправить армию лояльных грибов, оснащенных лопастями вертолета и пульсорами, атаковать несколько огненных монстров?

Штampы и шутки

Вы — не единственный человек в игре: разработчики, в виде висящих в воздухе зрачков, работают над ней, пока вы играете, хотя в основном это сводится к тому, что они громко друг с другом препираются. Как и можно ожидать от игры о незаконченной игре, в *The Magic Circle* полно шуток о народном финансировании, разработке, игровых штампах и прессе про игры, и есть кое-какие довольно забавные наблюдения по поводу всего этого. Хотя тон игры изначально легкий и веселый, в конечном итоге, сюжет провисает под собственным весом. Сцены становятся все мелодраматичнее и надрывнее, появляются длинные отрезки, где разработчики спорят между собой, или ИИ обращается к вам, пока вы просто сидите в ожидании, когда они закончат (в Steam Early Access, пару таких сцен — хотя и не все — можно пропустить). Есть и другие проблемы: досадно, что в игре, где вы будете неоднократно проходить по одним и тем же местам, чтобы их исследовать, нельзя бегать (хотя между

отдельными узлами можно перемещаться быстрее). И хотя в предпосылке заложено, что игровой мир будет визуально непривлекательным, это дела не меняет. Устаревшей графике часто свойствен некий шарм, но это не тот случай.

Можете рассчитывать где-то на 4–6 часов игры, а по завершении истории становится доступна довольно приятная мини-игра. Вы также можете продолжить исследовать игровой мир, чтобы найти секретные области, аудиожурналы и все прочее из того хорошего, что вы, возможно, пропустили. Несмотря на имеющиеся проблемы, замысел *The Magic Circle* оригинален и хорошо воплощен, редактировать монстров для решения головоломок интересно на протяжении всей игры. **LXF**



» Возможность перепрограммировать поведение встречаемых объектов заставит вас пожалеть, что это невозможно в реальной жизни.

LINUX
FORMAT

Вердикт

The Magic Circle

Разработчик: Question
Сайт: www.magiccirclegame.com
Цена: £ 15

Сюжет	9/10
Графика	7/10
Увлекательность	5/10
Оправданность цены	6/10

» Неказистый с виду и малость зашкваренный на своем же сюжете, но в остальном умный и увлекательный творческий квест.

Рейтинг

8/10

Back To Bed

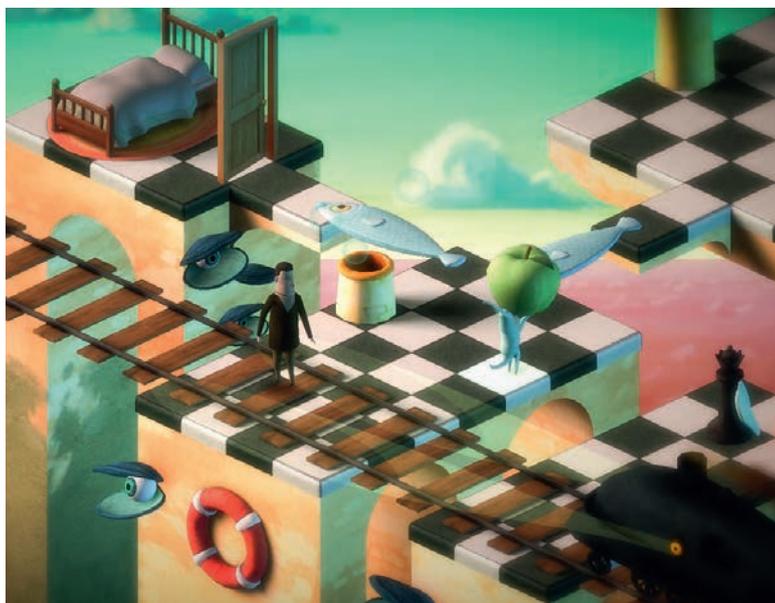
Самая настоящая «студенческая поделка»! Сюрреализм на экране, и это не просто слова — это то, с чем **Евгений Балдин** имел дело во время игры.

Вкратце

» Головоломки в сюрреалистичных декорациях. Все очень просто, так как игра для казуалов. Но иногда нужно и подумать.

Согласно авторской легенде, игра создавалась по воскресеньям группой датских студентов из разных университетов в холодном пакгаузе на берегу гавани города Ольбурга. Смело игнорируя баги, группа молодых энтузиастов, немного подгоняемых перспективой проставить в своем резюме информацию, что они делали Игру, все-таки ее закончили; они выставились на локальном конкурсе и выиграли его, потом еще и еще. Звезд особо с неба не хватало, но в 2013 г. добрались до финала Independent Games Festival Student Showcase (Сан-Франциско, США). В том же году довели идею до Kickstarter, попросили скромных 12K \$ и получили доверие чуть более тысячи «меценатов из толпы», а также 13K \$. В августе 2014 г. приключился релиз на Steam под все обещанные платформы.

В каком-то смысле история о том, как делалась игра, даже интереснее той истории, которая лежит в ее основе, но для казуальных игр эта самая история совершенно не главное. Главный герой Боб — лунатик.



» На железную дорогу выходить не стоит, а то кашалот собьет. Хорошая задумка, но это скорее декорация, нежели элемент игровой механики.

Засыпая, Боб начинает ходить во сне, причем делает это в мире, где перспектива течет, а шахматные фигурки летают и каркают, на фоне добрых декораций, навеянных произведениями Дали и Эшера. Для сохранения рассудка Бобу нужно добраться до кровати. Во сне можно сорваться в пропасть, попасться на пути будильнику или собаке, а единственный, кто от этого способен защитить — сам же Боб, но почему-то он на четырех ногах, с хвостиком и остроконечными ушками. Вернуться в постель он сам себе и помогает.

Натыкаясь на препятствия, Боб поворачивается по часовой стрелке, то есть его путь предрешен окружающей обстановкой. Четвероногий Боб может создавать временные препятствия, перетаскивая большие яблоки, и прокладывать рыбные мостики. Вот, собственно говоря, и все. Механика игровых загадок проста и незатейлива. Основным привлекательным элементом является сюрреалистическое окружение с порталами, элементами загадочных лестниц Эшера и каркающими шахматными фигурками.

Все тридцать основных уровней проходятся, совершенно без напряжения, за час с хвостиком — ну, может быть, за два. Затем они немного усложняются тем, что к двери, за которой находится кровать, добавляется требование ключа — это еще

некоторое время, отданное головоломкам. Графика отнюдь не фотореалистична — особенно это заметно по трехмерным моделям персонажей и потокам воздуха; но декорации сделаны весьма приятно. Двигать яблоки приходится «на время», и хотя всегда существует приемлемое решение, разработчики рекомендуют для управления геймпад, а не мышью.

Игру правомочно сравнить с не слишком сложным кроссвордом, и у нее уже поднабралось немало реальных любителей — хотя и не из самых «хардкорных», но что взять с «казуальщины»? **LXF**

» Некоторые стены можно использовать как пол. К сожалению, Боб этого не умеет, и поэтому Бобу приходится за него отдуваться.



Свойства навскидку



Будильник — враг

С будильником лучше не сталкиваться. Благо их можно перенаправить в пропасть. На сей раз этого сделать не получилось. Можно только полюбоваться на жидкие часы от Дали.



Постель — друг

Боб направлен к двери, за которой кровать, можно и отдохнуть. Шахматные фигуры иногда присаживаются на пол, и их можно сгонять — есть соответствующее достижение.

LINUX FORMAT Вердикт

Back To Bed

Разработчик: Bedtime Digital Games
Сайт: bedtimedigitalgames.dk
Цена: 149 руб. в Steam

Сюжет	5/10
Графика	8/10
Увлекательность	7/10
Оправданность цены	6/10

» Приключения упорно спящего лунатика на пути к постели. Декорации не блестящие, но необычные, и порой довольно органично включены в загадки. Решение есть всегда.

Рейтинг 7/10

Доверяйте свой контент только профессионалам

Сеть доставки контента CDNvideo с 2010 года обеспечивает качественным, разнообразным и бесперебойным сервисом клиентов в России, СНГ, Европе и Юго-Восточной Азии.

Бесперебойные онлайн-трансляции



Высокая скорость и качество видео
Вещание на неограниченную аудиторию
Корректное отображение на любых устройствах
Множество дополнительных опций

Мультимедиа по запросу



Стабильно высокая скорость загрузки
Без буферизаций и зависаний
Не требует капитальных затрат
Снижение нагрузки на каналы связи

Ускорение интернет-сайтов



Улучшает позиции в поисковой выдаче
Положительно влияет на конверсию
Повышает лояльность пользователей
Улучшает пользовательский опыт в регионах

Подробности на сайте <http://cdnvideo.ru> и по телефону +7 495 212-02-46

CDNvideo – ведущий провайдер услуг сети доставки контента (CDN) в России и СНГ.
Ёмкость сети составляет более 500 Гбит/с, в том числе в Москве – более 300 Гбит/с.
Доступность сети в 2013-2015 гг. составила более 99,99%.



РАЗРАБОТКИ

CrossOver и Wine идут на Android

На Android-устройствах можно будет запускать ПО для Windows.

Благодаря приложению *CrossOver* для Android, которое CodeWeavers обещает представить еще до конца 2015 г., на Android-устройствах можно будет выполнять приложения, изначально предназначенные для Windows. Версия *Wine* для Android появилась несколько лет назад, и работа над ней продолжается. Сегодня приложения для Android могут выполняться в среде Windows, а уже очень скоро станет возможным и обратное.

Компания CodeWeavers реализует коммерческий уровень совместимости, позволяющий приложениям для Windows работать в среде Linux и Mac OS X. Ее решения также помогают разработчикам портов Windows-игр и других приложений для Mac и Linux при сборке пакетов своих программ обеспечить их работоспособность. Основа этих решений — open-source проект *Wine*, разработку которого финансирует CodeWeavers. Небезынтересно, что

CodeWeavers продолжает работать над поддержкой DirectX 11 для *Wine*.

Бесплатная tech-preview версия *CrossOver* для Android будет доступна до конца текущего года. Соответствующие патчи опубликуют и в открытом

Уже существуют и ноутбуки, и настольные ПК на Android.

проекте *Wine*, после чего станут доступными сборки *Wine* для Android. Тем не менее, есть и ложка дегтя: версии *CrossOver* и *Wine* для Android будут работать только на Android x86, т.е. в системах Android на процессорах Intel или AMD. Поддержка типичных Android-систем, на процессорах ARM, не заявлена.

Версия Windows RT никогда не предусматривала выполнения настольных приложений, поэтому desktop-программ, доступных для Windows на ARM, просто нет. Чтобы позволить приложениям Windows x86 выполняться на процессорах ARM, в составе *CrossOver* или *Wine* должна была бы содержаться виртуальная машина. По этой же причине Windows-приложения не будут работать при установке *Wine* в среде Linux на хромбуках с ARM-процессорами.

Есть и более очевидная проблема. Большинство Windows-приложений рассчитаны на работу с полноценной клавиатурой и мышью, а не на сенсорный экран и виртуальную клавиатуру. Однако уже существуют и ноутбуки, и даже настольные ПК на Android. Наконец, пользователь может выбрать Android-планшет на базе Intel.

СТРАТЕГИЯ

BlackBerry меняет ОС

BlackBerry официально подтвердила, что работает над созданием Android-смартфона.

Опубликованы финансовые итоги BlackBerry за II квартал 2015 г. Доходы этой канадской компании снизились до \$490 млн, по сравнению с \$916 млн годом ранее, а валовая прибыль упала, поскольку продажи смартфонов BlackBerry остаются на уровне \$0,8 млн в течение трехмесячного периода (годом ранее этот показатель составлял \$2,1 млн). Попыткой остаться на плаву выглядит официальное заявление, что компания работает над созданием устройства на базе Android, названным BlackBerry Priv (предположительно, сокращение от "privacy").

Генеральный директор BlackBerry Джон Чен [John Chen] высказался по данному поводу так: «Я подтверждаю наши планы по запуску Priv, Android-устройства, в названии которого прослеживается верность традициям BlackBerry и приоритеты защиты частной жизни наших клиентов. Priv сочетает лучшие традиции безопасности и производительности BlackBerry с расширенными

возможностями экосистемы мобильных приложений платформы Android».

В BlackBerry надеются, что Priv, появление которого в течение многих месяцев было предметом оживленных слухов онлайн, убедит определенную часть пользователей iOS и Android отказаться от своих платформ в его пользу, что увеличит нынешние 0,3% доли глобального рынка смартфонов, которые занимает канадская компания. Однако крайне маловероятно, чтобы Android и iOS, которым сейчас принадлежат 82,2% и 14,6% рынка соответственно, в ближайшее время уступили кому-то свои лидирующие позиции.

Предполагается, что BlackBerry Priv объединяет 5,4-дюймовый дисплей 1440×2560 QHD с выдвижной клавиатурой, что, по мнению компании, должно вызвать wow-эффект на корпоративном рынке. Недавние утечки информации также указывают на наличие 18-МП камеры, процессора Qualcomm Snapdragon 808 1,8 ГГц, 3 ГБ ОЗУ и «практически

стоковой» версии Android 5.0 Lollipop. В США новое устройство от BlackBerry поступит в продажу в ноябре, с поддержкой четырех основных мобильных операторов этой страны.



» Намеченный на ноябрь выпуск BlackBerry Priv на базе Android — попытка канадского производителя смартфонов остаться на плаву.

ЭТО БИЗНЕС

Согласились полюбовно

Asus будет предустанавливать продукты *MS Office* на свои Android-устройства.

Мicrosoft прекращает судебное преследование Asustek за использование касающихся Android патентов, в обмен на обещание тайваньской компании размещать в разрабатываемых ею мобильных устройствах на базе Android приложения *Microsoft Office*. 2 октября Asus и Microsoft сделали совместное заявление о расширении предыдущего соглашения по лицензированию между двумя компаниями, которое охватывает производимые Asus телефоны и планшеты на базе Android, и «программное обеспечение, устройства и сервисы Microsoft». Соглашение предусматривает «более тесную интеграцию» между двумя компаниями, включая «предустановку сервисов продуктивности Microsoft Office на выпускаемые компанией Asus смартфоны и планшеты на базе Android», а также сотрудничество, направленное на разработку новых продуктов.

«Это соглашение имеет большое значение для обеих компаний, — заявил Ник Психоджес [Nick

Psyhogeos], президент Microsoft Technology Licensing LLC. — Помимо обеспечения дальнейшего улучшения наших продуктов, оно открывает двери сотрудничеству между Microsoft и ASUS, которое возможно только на принципах взаимоуважения и координации интеллектуальной собственности».

Для Microsoft это не первое соглашение типа «патент-за-Office»: в феврале подобным образом был улажен патентный спор с Samsung. Несколько месяцев спустя корейская компания уже устанавливала пакеты некоторых приложений Microsoft на телефоны и планшеты Galaxy (OneNote и OneDrive, например, были предустановлены в unlocked-версии нового Galaxy Note 5, а также в телефонах для Sprint и T-Mobile; в версии для Verizon, однако, оба приложения отсутствовали).

Таким образом, имеет место классическое соглашение «услуга за услугу», благодаря которому Microsoft расширяет экосистему своего Office.

Неизвестно, продолжит ли Microsoft «расширять» существующие лицензионные соглашения



► Microsoft и ASUS уладили патентные споры: первая прекращает судебные преследования, а вторая предустановит Office на всех своих будущих Android-устройствах.

с другими лицензиатами, однако несомненно, что это — наиболее простой для Microsoft способ «убедить» производителей оборудования портить свои сервисы на как можно большее число платформ.

РАЗБОРКИ

Претензии Porsche к Google

Android Auto считывает данные двигателя?

Но к каким данным может получить доступ приложение? Google удалось убедить около 40 автопроизводителей использовать Android Auto, но Porsche выбрал Apple и iOS. Согласно информации издания *Motor Trend*, Porsche объясняет свое решение тем, что ПО от Google собирает данные о скорости автомобиля, положении дроссельной заслонки, температуре охлаждающей жидкости и масла, оборотах двигателя. Google отверг претензии немецкого автопроизводителя: «Мы очень серьезно относимся к тайне частной жизни, и не собираем такие данные. Пользователи сами выбирают информацию, которой им делиться с Android Auto, а система обеспечивает им hands-free во время движения и более точную навигацию через GPS автомобиля».

Android Auto действительно мониторит некоторые данные — например, движется ли автомобиль — получая доступ к GPS, средствам рулевого управления, аудиосистеме и данным о скорости вращения колес. Android Auto — это не встроенное микропрограммное обеспечение автомобиля, а просто приложение, работающее на смартфоне,

который соединяется через USB с дисплеем приборной панели. Телефон выполняет свою работу по визуализации карт, проигрыванию музыки и т.д., используя приборную панель в качестве второго сенсорного экрана, а аудиосистему в качестве внешних динамиков. Android Auto также реагирует на голосовые команды от водителя и пассажиров.

Аппаратные средства приборной панели, прямо или косвенно, соединяют телефон с CAN-шиной, которая является «центральной нервной сис-

Android Auto — это приложение, работающее на смартфоне.

темой» автомобиля. При разработке CAN-шины не предусматривались какие-либо средства безопасности, поэтому любой контроллер на двухпроводной шине может получить доступ к другой. Таким образом, возник вполне закономерный

вопрос: что мешает Android Auto получить доступ к любым типам данных от CAN-шины? Решение предлагает платформа OpenXC, своего рода брандмауэр между Android-устройством и шиной CAN, использующая переходник CAN-to-USB: устройство сможет запросить и получить только строго определенную информацию.

Android Auto управляет аудиосистемой автомобиля. Это означает, что у системы есть возможность записи в CAN-шину. Таким образом, скомпрометированное устройство — например, пораженное вредоносным ПО, которое использует уязвимость Stagefright — способно подать деструктивную команду в «ствол головного мозга» автомобиля? Возможно, в автомобилях, совместимых с Android Auto, шина CAN разделена таким

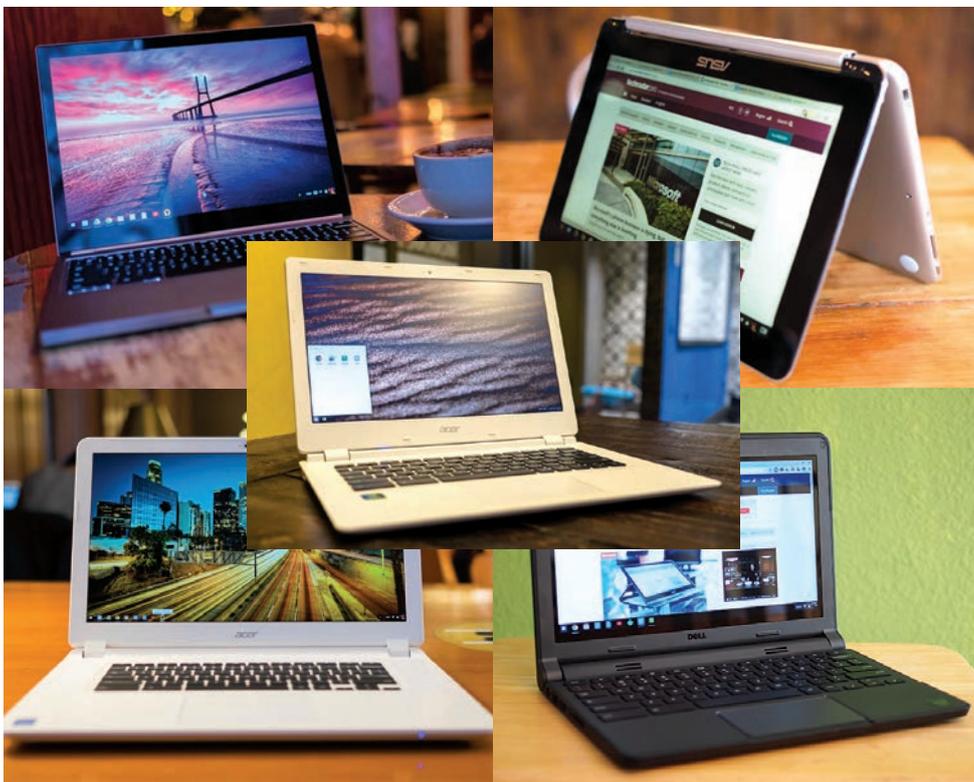
образом, что аудиосистема блокирована шлюзом от аппаратных средств управления двигателем? Возможно, управление аудиосистемами осуществляется без доступа к шине CAN? Ответа от Google на эти вопросы пока не последовало. **LXF**

Сравнение

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

Хромбуки

Раздумываете о том, какой гаджет от Google, работающий на Linux, покупать следующим? **Нейл Мор** берет несколько хромбуков на тест-драйв.



Про наш тест...

Важное свойство хромбуков — их сосредоточенность на облаке, с интерфейсами, основанными на браузере, и Chrome OS. Во многом это облегчает тестирование, потому что можно сосредоточиться на других областях: качество сборки, включая клавиатуру и тачпад; физические свойства; возможности подключения; производительность; длительность заряда и свойства для мультимедиа, например, качество экрана и аудио. В зависимости от ваших нужд, приоритеты различаются, но разбиение на разделы поможет вам оценить каждую модель независимо от нашего общего вердикта. И если длительность заряда вам важнее мощности, то ваше решение может быть кардинально отличным от нашего. По счастью, правда в том, что все протестированные модели хороши, и, по нашему убеждению, любая из них способна осчастливить своего владельца. Приятно также видеть, что система Linux занимает 7% рынка ноутбуков.

Наша подборка

- » Acer Chromebook 13
- » Acer Chromebook 15
- » Asus Chromebook Flip
- » Chromebook Pixel 2015
- » Dell Chromebook 11 2015

Мы постоянно рассказываем о хромбуках, и причина вовсе не в том, что мы поддерживаем платформу для нишевых ноутбуков, а в том, что они стали вполне жизнеспособным с коммерческой точки зрения продуктом, занимающим лидирующие позиции на рынке — и так уж получилось, что на них устанавливается Linux и многие дистрибутивы Linux для них родные. Мы не утверждаем, что они идеальны в смысле свободы ноутбуков, но вряд ли такое возможно в принципе, с прошивкой CPU как она есть, пока не появится полностью открытая аппаратная платформа.

Итак, в настоящем положении вещей, это лучшее из доступного пользователям открытого кода — но, подбросив на весы цену, выбор, производительность и качество оборудования, мы в итоге получим явную победу для пользователя Linux. Последний раз мы рассматривали хромбуки в Сравнении ровно год назад, и с тех пор наблюдались солидные подвижки. Google наконец-то выпустил новую версию своего высококачественного Pixel, и теперь многие хромбуки используют Intel Celeron 3205U, хотя имеется большой выбор опций.

Впечатляет то наблюдение, что многие производители принимают исследовать

разные форм-факторы, добавляя функции планшетов, стиль ультрабуков, стандартные размеры ноутбуков, повышая разрешение экрана и многое другое. И притом в целом производители умудряются по-прежнему предлагать свои продукты по соблазнительно низкой цене, характерной для хромбуков, и с поразительной длительностью заряда батарей. Да, некоторые модели начали перемещаться в ценовую категорию стандартных ноутбуков, но причина тут обычно в их спецификациях. Так что нужна ли вам школьная рабочая лошадка, офисный инструмент или нечто для работы в пути, хромбуки будут отличным выбором.

Качество сборки

Не развалится ли он на части, и можно ли его ронять?

У хромбуков исторически была репутация, скажем так, имеющих не лучшую сборку. Большинство при их описании употребили бы такие слова, как «пластиковый» и «дешевый». Однако это относится к хромбукам былых времен. Ныне же, хотя цена на большинство хромбуков удерживается низкой, их качество сборки повысилось, а более высокая цена также означает высококачественные материалы и качество сборки.

Asus Flip является отличным тому примером, с его металлическим корпусом, тонким профилем и алюминиевыми шасси в пику Apple. Все это также придает ему исключительно солидный вид, абсолютно не утяжеляя его, а благодаря закругленным краям его удобно и приятно держать. Его толщина всего 15,6 мм, и весит он 890 г — неплохо и для планшета.

Google Chromebook Pixel — еще одна модель с алюминиевыми шасси, однако Google предпочел исключительно утилитарную эстетику дизайну с острыми квадратными углами. Хотя это придает хромбуку очень современный вид, в руках его

держат не слишком удобно. Этот дизайн также делает его относительно объемным, и его вес составляет 1,5 кг при толщине всего 15 мм. Основная примета дизайна — довольно необычное в наши дни соотношение экрана 3:2.

Перейдем от очень тонкого и металлического корпуса к куда более массивному и сделанному из пластика корпусу Acer Chromebook 15. Это 15-дюймовый хромбук, так что стоит ожидать, что он будет тяжелее, и он весит 2,1 кг, хотя надо отметить, что это меньше, чем у большинства 15-дюймовых ноутбуков Windows. Его толщина тоже больше — 24 мм, и опять же, сравните с 30 мм ноутбука Acer Aspire E5 Windows. Он из пластика, однако Acer использовал тканевую текстуру, которая позволяет ощущать его в руках более надежно и придает ему более натуральный и солидный вид.

А теперь — к самой маленькой части спектра: если Dell Chromebook 11 и можно назвать крошечным, то он создан особо прочным. Прицеливаясь на школы, Dell мобилизовал весь свой опыт и создал устройство армейского уровня, которое способно



» Во время тестов Dell было трудно побить (в прямом смысле). Он хоть и маленький, но дюжий.

выдержать воздействие любого предмета, брошенного в него ребенком. При этом его толщина составляет всего 21 мм, а вес — 1,24 кг, и для мобильного устройства он выглядит очень неплохо. По размеру и весу он меньше, чем Acer Chromebook 13: это 13-дюймовое устройство имеет толщину 18 мм и весит 1,5 кг. Оно предназначено для дома и для студентов, но при сборке явно упустили из вида усиленные шасси, имеющиеся в Dell. Зато этот хромбук сумел найти разумную грань между портативностью и стилем.

Вердикт

CB Pixel 2015
★★★★★
Dell CB 11 2015
★★★★★
Acer CB 13
★★★★★
Acer CB 15
★★★★★
Asus Flip C100
★★★★★

» Dell самый прочный, зато Pixel всех красивее.

Дополнительные функции

Небольшие штрихи, которые их выделяют.

Обычно 15-дюймовый экран на ноутбуке не привлекает особого внимания, но для хромбуков это редкость, и Acer CB 15 считает своим козырем 15,6-дюймовый антибликовый экран. Что мы и отметили; во всем остальном это довольно незамысловатое предложение. Практически то же можно сказать и об Acer Chromebook 13, что не стоит воспринимать негативно: мы очень оценили его 13-дюймовые шасси. У Pixel тоже есть ряд интересных моментов: он легко узнаваем благодаря своим уникальным прямоугольным пропорциям, не говоря уже о многоцветной световой панели сзади экрана, которая отображает «основные» цвета логотипа Google. Весьма изобретательно, что эта панель по совместительству является также индикатором заряда батареи, и она включается каждый раз, когда вы дважды стукнете по тыльной части крышки.

Еще более удобная и самая яркая новая функция Flip — его способность трансформироваться в планшет и другие режимы.



Дисплей — с сенсорными способностями — соединен с основным ноутбуком поворотной осью, которая в режиме планшета отгибается на 360 градусов. Удобно также, что можно поставить Flip на бок, на клавиатуру, и экран развернется куда вам надо. Новая кнопка Virtual активирует «взрывной» вид всех ваших открытых окон и приложений, плюс виртуальную клавиатуру.

Dell предлагает ряд «жестких» функций, например, шарнир для поворота на 180°, чтобы расположить экран плоско, не отгибая его назад. Кроме того, он водонепроницаемый, предлагает сенсорный экран и имеет на крышке интересный индикатор внимания в классе. Явная звезда данной категории — Flip, но у каждого есть своя причина существования в экосистеме.

» Выдающаяся функция Asus Flip — его превращаемость в планшет.

Вердикт

Asus Flip C100
★★★★★
CB Pixel 2015
★★★★★
Dell CB 11 2015
★★★★★
Acer CB 13
★★★★★
Acer CB 15
★★★★★

» Asus Flip — самый гибкий по части дополнительных функций.

Производительность

А мой быстрее твоего!

Помимо срока заряда батарей, в этом разделе мы рассмотрим производительность в плане отзывчивости, чистой скорости и емкости памяти. Бессмысленно иметь очень крутой с виду или даже самый портативный хромбук на свете, если он не в состоянии справиться больше чем с двумя вкладками. Ну и, кроме того, желая

работать с полноценным дистрибутивом Linux поверх Chrome OS, вы должны быть уверены, что процессор и память вашего хромбука позволят вам это сделать.

В первую очередь мы здесь протестируем Chrome OS и проверим ее способность к работе в Сети с многочисленными вкладками браузера

без серьезного торможения. Вы сможете оценить разнообразные процессоры, от малобюджетного Celeron до куда более солидного Core i5, и важно также понимать отличие от самого последнего, пятого поколения процессоров Haswell, которое значительно увеличивает срок жизни батарей и улучшает возможности 3D.

Acer Chromebook 15 ★★★★★

Внутри CB15 жужжит солидный двоядерный процессор Intel Celeron 3205U Broadwell. Его можно считать обновленной версией Intel Celeron 2955U, стоявшего на многих хромбуках в 2014 г.

Более новые CPU могут похвастаться частотой 1,5 ГГц, а также некоторыми усовершенствованиями архитектуры, которые уменьшают энергопотребление и увеличивают скорость примерно на 10%. Несмотря на несколько вкладок, напигованных GIF'ками котят, Chromebook 15 отнюдь не уронил производительности и был практически идеальной многозадачной машиной. Если вам важны цифры — в тестах он показал 13328 в Octane, 2332 мс в Mozilla Kraken и 334 мс в SunSpider. Единственной заметной проблемой было то, что в спецификации для Великобритании он идет только с 2 ГБ памяти DDR3, тогда как весь мир получает 4 ГБ ОЗУ, и это выглядит известной дискриминацией.



Acer Chromebook 13 ★★☆☆☆☆

Интересным моментом здесь является использование процессора Nvidia Tegra K1; это движение в том же направлении, что и у Asus Flip, основанного на ARM. K1 — это четырехъядерная архитектура, работающая на 2,1 ГГц и использующая Nvidia Kepler GPU и 2 ГБ (имеется модель с 4 ГБ) памяти. Увы, в погоне за самой низкой ценой и самой долгой жизнью заряда батарей Acer явно пожертвовал производительностью из-за выбора памяти. Мы наблюдали задержки и зависания при открытии вкладок, когда хромбук выполнял другие задачи. Добавьте эту слабую общую производительность к сравнительным тестам, и мы увидим, что он уступает первенство процессорам Intel ARM, демонстрируя 7000 в Octane (чем выше, тем лучше), хилые 4800 мс в Kraken и 600 мс в SunSpider (для этих двух чем меньше, тем лучше). Это на одном уровне с Flip с его процессором ARM и более высокой производительностью GPU, но вдвое хуже результатов Intel.

Клавиатура и тачпад

Смогу ли я печатать, не выламывая пальцы?

Вы можете вообразить, что к текущему моменту производители ноутбуков уже освоили дизайн достойных клавиатур и тачпадов для мобильных устройств, однако осторожность никогда не вредит, особенно если речь идет о достоинствах среднестатистического хромбука.

Например, Chromebook 15 прогибается при нажатии на нижнюю полосу тачпада. Ничего особо страшного, но все же это раздражает, ведь в остальном это чудесный тачпад большого размера в стиле MacBook. Увеличению размера поспособствовала

приличная полноразмерная клавиатура, в которой нет только цифровой клетки.

На противоположном конце размерной шкалы Acer Flip все же умудряется втиснуть полноразмерную клавиатуру. Создается ощущение, что клавиши нажимаются достаточно сильно и уходят довольно глубоко, несмотря на тот факт, что клавиатура размещена в углублении, которое оставляет менее 2 мм между ней и корпусом хромбука. Было бы неплохо иметь трекпад побольше, но для этого просто физически нет места, а работает он превосходно.

Стандартная клавиатура Chromebook 13 отлично размещена и достаточно отзывчива, с тачпадом идеального размера. Конструкция клавиатуры не дает ей прогибаться, и печатать очень удобно. Клавиши достаточно громко щелкают, и вам не придется проверять, не пропустили ли вы букву.

О Pixel мало что можно добавить, поскольку клавиатура с подсветкой и точно настроенный трекпад превосходны. Водонепроницаемая и почти полноразмерная клавиатура Dell — на том же уровне: ею приятно пользоваться, а трекпад отзывчив.

Вердикт

- Acer CB 13 ★★★★★
- CB Pixel 2015 ★★★★★
- Asus Flip C100 ★★★★★
- Dell CB 11 2015 ★★★★★
- Acer CB 15 ★★★★★

» Превзойти Pixel по качеству ввода непросто, но Acer это почти удалось.

Asus Chromebook Flip ★★☆☆☆☆

Даже внутренне этот хромбук доказывает, что не лыком шит благодаря процессору 1,8 ГГц Rockchip 3288-С и ARM Mali-T624 GPU. Это четырехъядерный Cortex-A17 CPU, который, по заявлениям ARM, обеспечивает солидный прирост производительности — на 60% больше, чем Cortex-A9, при 20% снижении энергопотребления при той же нагрузке. У него нет никаких проблем с работой в браузере, даже при 15 открытых вкладках и потоковом воспроизведении музыки. Время от времени некоторые социальные сети, перегруженные мультимедиа-контентом, типа Facebook, приводили на Flip к зависанию экранов, но это несчастный случай. Поскольку он основан на ARM, вы вряд ли будете запускать на нем полноценный настольный дистрибутив, и эта версия идет всего с 2 ГБ памяти, хотя имеется и 4-ГБ модель. Наши сравнительные тесты показали 6795 для Octane (чем выше, тем лучше), 5447 мс в Kraken и 686 мс в SunSpider (чем меньше, тем лучше).

**Dell Chromebook 11 2015** ★★☆☆☆☆

Dell недалеко ушел от своих более мощных собратьев, используя чуть более старый Intel Celeron N2840, который является вариантом Bay Trail Atom. Это очень хитрый выбор Dell, поскольку он работает на 2,16 ГГц и способен на бросок в 2,6 ГГц, хотя при этом использует половину мощности Celeron 3205U — ну и всё же остается двуядерным. Dell имеет 4 ГБ DDR3 и 16 ГБ SSD.

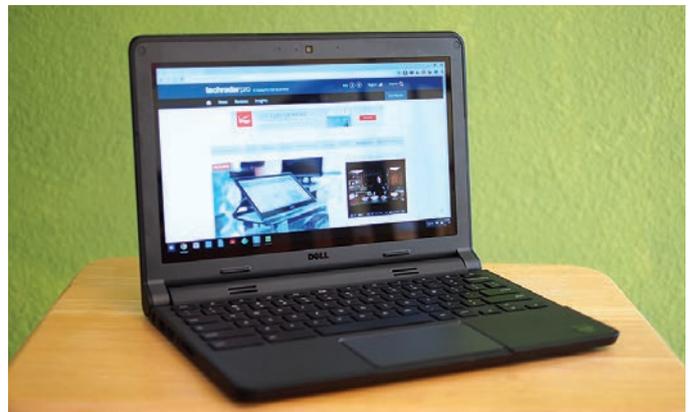
При обычном использовании с 15 вкладками, работающими в трех разных окнах браузера, никаких снижений скорости не наблюдалось; Chromebook 11 быстр, отзывчив и не показывает никаких зависаний. В тестах процессор Atom продемонстрировал свою слабую сторону, плетясь позади остальных процессоров Celeron с 8355 в Octane (чем выше, тем лучше), 3620 мс в Mozilla Kraken и 521 мс в SunSpider (чем меньше, тем лучше). И это отодвигает Chromebook 11 назад, на позицию между более медленными моделями ARM Flip и Intel Celeron 3205U.

Chromebook Pixel 2015 ★★★★★★

Он всегда был большим папочкой в мире хромбуков. Он собран на последнем, 5-м поколении Intel Core i5 5200U, что сильно помогает и позволяет ему солидно опередить остальные модели Сравнения.

Это двуядерный четырехпоточковый процессор 2,2 ГГц с турбо-режимом 2,7 ГГц. Google выпускает модель LS (Ludicrous Speed — До смешного быстрый) с Core i7, что на самом деле глупо. Помимо этого, имеется аж 8 ГБ DDR3 и 32 ГБ накопителя SSD.

Pixel — невероятно быстрая машина: web-страницы загружаются молниеносно, прокрутка плавная, и даже масштабирование щипком на тачскрине работает отлично. Однако по-настоящему машина отличается при воспроизведении потокового видео на 1080p и даже 4K-видео на YouTube. Наши тесты вернули потрясающие 24000 для Octane (где чем выше, тем лучше), 1428 мс в Mozilla Kraken и 298 мс в SunSpider (чем меньше, тем лучше).



Возможности соединения

Что я могу подключить и к чему подключиться?

Хромбуки по определению являются онлайн-машинами, предназначенными для подключения к облаку, так что эти особи должны предлагать сверхпрочное беспроводное соединение. Мы также рассчитывали на поддержку Bluetooth по умолчанию — а как иначе вы сможете насладиться *White Snake*, программируя на Python и не нарушая покоя и тишины своей местной библиотеки?

Начнем сверху: Chromebook Pixel предлагает двухканальный Wireless AC через карту Intel 7260 Wi-Fi и Bluetooth 4.0LE,

но еще больше впечатляет наличие двух портов USB-C и двух портов USB 2.0, что полностью покрывает все ваши нужды в ближайшем обозримом будущем.

Flip также радует в этой области, с беспроводным AC, Bluetooth 4.1 и видеовыходом micro-HDMI, но предлагает только два порта USB 2.0. Acer Chromebook 15 тоже демонстрирует гибкость, с беспроводным AC, Bluetooth 4.0 и выходом HDMI, а также одним портом USB 3.0 и еще одним USB 2.0. Acer 13 во многом, как и следовало ожидать, имеет такие же спецификации, однако

его два порта USB — USB 3.0. И, наконец, Dell, даже со своим собственным чипсетом Bay Trail, предлагает двухканальный беспроводной AC, Bluetooth 4.0, выход HDMI 1.4 и один порт USB 3.0 плюс один порт USB 2.0.

Все протестированные здесь устройства обладают встроенной web-камерой 720p, объединенным разъемом для наушников и микрофона и слотом SD-карты; у Dell предусмотрен второй слот SD с блокирующим устройством. Приятно видеть, что о самом главном для пользователей настолько хорошо позаботились.

Вердикт

CB Pixel 2015
★★★★★
Acer CB 13
★★★★★
Acer CB 15
★★★★★
Asus Flip C100
★★★★★
Dell CB 11 2015
★★★★★

» Pixel лидирует, со своим крутым USB-C, но они все хороши.

Живучесть батарей

Смогу ли я весь день упиваться «Игрой престолов»?

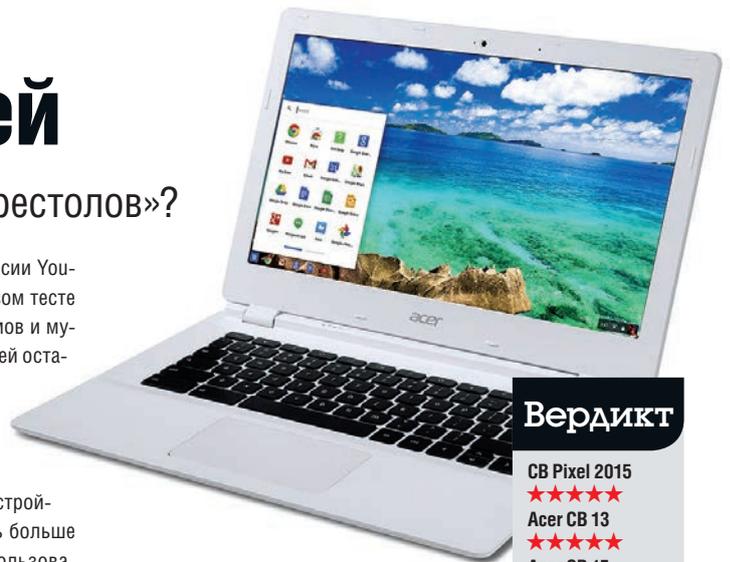
Чрезвычайных причин быть особенно упорными в работе сутками напролет у хромбуков нет [Ред.: — А Linux-to!], и тем не менее они хороши. Флагман, Pixel, заявляет о 12 часах работы батарей. В тестировании в реальных условиях работы устройства целый день на 50% яркости, включая двухчасовую сессию Netflix, мы получили восемь с половиной часов, и это весьма удивительно, при процессоре Core i5. Мы спокойно можем представить себе, что он проработает целый день. Похожая ситуация с Acer Chromebook 15, который работал чуть меньше восьми часов при тех же условиях.

Acer Flip был на другом уровне, показав восемь часов и десять минут работы в нашем реальном тесте. Он работал с десятью вкладками, при этом на нем писалась часть этого Сравнения, воспроизводилась Google

Music, еще был час Hangout и сессии YouTube. В более обычном семичасовом тесте при воспроизведении двух фильмов и музыки в течение трех часов, у батарей оставался заряд 50%.

Dell должен бы тоже быть долгоиграющим со своим Vay Trail CPU. Dell заявляет 10-часовой заряд батарей. Наши тесты при настройке яркости на 60% показали чуть больше восьми часов непрерывного использования, включая несколько открытых вкладок, ввод текста и видео с YouTube.

У нас остается Acer 13, и в реальной жизни с настройкой яркости экрана на 50% мы заполучили 8 часов 56 минут работы с воспроизведением видео. Это потрясающий результат; а второй запуск с музыкой, открытыми вкладками и редактированием текста позволил пересечь отметку в де-



Вердикт

- CB Pixel 2015 ★★★★★
- Acer CB 13 ★★★★★
- Acer CB 15 ★★★★★
- Asus Flip C100 ★★★★★
- Dell CB 11 2015 ★★★★★

» Как Pixel, так и Acer 13 демонстрируют отличную производительность.

» За живучесть батарей мы отдаем первенство Pixel, но результат Acer 13 тоже впечатляет.

вать часов. И так, нашим первым выбором становится Pixel, за которым следует Acer 13 с экстра-долгим результатом; однако же ни один из наших кандидатов нас здесь не разочаровал.

Дисплей и аудио

А мои глаза и уши выдержат эту сессию «Игры престолов»?

Начнем с Pixel, который щеголяет своими хипстерскими пропорциями 3:2 и чудесным 12,85-дюймовым 2560×1700 сенсорным дисплеем IPS. Это здорово для редактирования документов или фото, но не для просмотра фильмов. Технология IPS предлагает богатые цвета и яркость с незначительными тенями.

Нельзя сказать того же о Dell: его экрану не-IPS, 11,6-дюймовому HD на 1366×768 не хватает широких углов обзора, и он не такой яркий, как нам хотелось бы, особенно при солнечном свете. Однако удивительно,

что звук из стереонаушников был громким и чистым. Фактически, на самой большой громкости производимый Chromebook 11 звук заполнял небольшую комнату, однако мы услышали некие искажения звука на громкости более 80%, и при этом уровне звуку также не хватало насыщенности.

Acer Chromebook 13 предлагает 13,3-дюймовый не-сенсорный дисплей 1366×768, темный и тусклый. Он примерно такой же, как в Dell, но там, по крайней мере, есть опция сенсорной работы. Модели не для Великобритании, по крайней мере, предлагают

дисплеи Full HD 1080p. Но приличная работа аудио несколько компенсирует невысокое качество дисплея.

Вполне возможно, что Asus предусмотрел на Flip широкую окантовку, чтобы облегчить его применение в качестве планшета. Оно и понятно, но тем не менее отвлекает, когда пытаешься использовать его как обычный хромбук. В остальном 10,1-дюймовый дисплей WXGA (1280×800) на IPS с multi-touch весьма достойно справляется с работой — цвета не такие богатые, как у Pixel, но зато хорошие углы обзора, и он яркий. Мы даже рады сообщить, что Asus Flip предлагает неплохой набор колонок для прослушивания музыки и просмотра мультимедиа.

И напоследок у нас остается самый большой дисплей, Acer Chromebook 15. Это идеальная машина для воспроизведения потоковых мультимедиа. 15,6-дюймовый дисплей Full HD с разрешением 1920×1080 и подсветкой LED позволяет очень отчетливо различать все, от значков до текста, и облегчает просмотр на расстоянии. Вы будете благодарны Acer за выбор яркой панели IPS, которая предлагает самые широкие углы обзора из нами виденных. Микрофоны встроены в клавиатуру, и хотя звук получается не совсем полным, но вполне достойным.



» IPS-дисплей Chromebook 15 предлагает широкий угол обзора.

Вердикт

- Acer CB 15 ★★★★★
- CB Pixel 2015 ★★★★★
- Asus Flip C100 ★★★★★
- Acer CB 13 ★★★★★
- Dell CB 11 2015 ★★★★★

» Модели IPS, как обычно, уходят с призом.

Хромбуки

Вердикт

Нас искренне восхищает не только широчайший выбор хромбуков на рынке и их производительность, но и масса опций. Эта область явно наполняется жизнью и разнообразием, а производители не боятся пробовать инновационные функции на расширяющемся рынке. Это означает, что каковы бы ни были ваши нужды, под них найдется модель хромбука, способная их реализовать. Мы признаем, что, возможно, ваши потребности не вошли в критерии нашей оценки, но факт тот, что все модели в тесте этого года превосходны, и мы сомневаемся, что отыщется много разочарованных.

Начнем с мощнейшего Google Pixel 2015: несмотря на его впечатляющую производительность и прекрасный дизайн и сборку, мы опасаемся, что реально его почти никто не купит. Большинство покупателей хромбука будут отыскивать цену ниже £299, если не в районе £199, поэтому его близкая к Mac цена, вероятно, отпугнет львиную

долю покупателей. Если же ваши карманы достаточно набиты, вы не будете разочарованы, и это будет лестной темой для обсуждения, когда он вклинится в море MacBook Pro, наводняющих большинство конференций.

Почти на противоположной стороне спектра Dell Chromebook 11 2015 показывает возможности самой дешевой стороны рынка, и в целом предназначен не для обычного потребителя, а скорее для образовательных учреждений, которые, как правило, покупают эти устройства оптом и которым нужно нечто прочное и практичное. Он никого не разочарует своей выносливостью, однако его главным недостатком является экран. То же можно сказать об Acer 13: несмотря на превосходное качество сборки, клавиатуру и трекпад, его подводит малая мощность и дисплей. В противоположность им, Asus Flip доставляет вам множество



► Побеждает всемогущий Pixel, если исключить проблему цены.

приятных моментов — от изящного металлического корпуса до возможности конверсии и тачскрина. Если вы ищете новый планшет, это определенно хорошая альтернатива. И, наконец, Acer Chromebook 15 идеален для всех, кому нужен полноценный ноутбук, но с простотой хромбука — в частности, большой экран на 1080p делает его особенно привлекательным.

Каковы бы ни были ваши нужды, под них всегда найдется модель хромбука.

I

Chromebook Pixel 2015 ★★★★★

Сайт: <http://bit.ly/ChromebookPixel2015> Цена: £799

» Идеальная сборка для хромбука — просто зажмурьтесь на ценник.

IV

Dell Chromebook 11 2015 ★★★★★★

Сайт: www.dell.co.uk Цена: £199 (2 ГБ)

» Отличный выбор, если ваш хромбук будет постоянно получать пинки и падать.

II

Acer Chromebook 15 ★★★★★★

Сайт: www.acer.co.uk Цена: £229

» В качестве более крупного устройства — ноутбука Acer идеален по всем аспектам.

V

Acer Chromebook 13 ★★★★★★

Сайт: www.acer.co.uk Цена: £180

» Если отвлечься от экрана, то вы получите достойное время работы и лоск за хорошую цену.

III

Asus Flip C100 ★★★★★★

Сайт: <http://bit.ly/AsusCBFlip> Цена: £249 (4 ГБ)

» У него маловато мощности CPU, но в остальном этот хромбук просто блистательный.

Обратная связь

Мы не рассказали о вашем любимом хромбуке, или вы учудили с ним нечто уникальное? Напишите нам: lxf.letters@futurenet.com.

Рассмотрите также...

Хромбуки — не единственный фрукт на рынке портативных устройств, но если говорить о Linux, то возможности оказываются весьма ограниченными. Что касается других хромбуков, есть еще Toshiba Chromebook 2, и на горизонте маячит качественный Acer Chromebook C910, в котором будут те же спецификации, что и в Google Pixel 2015.

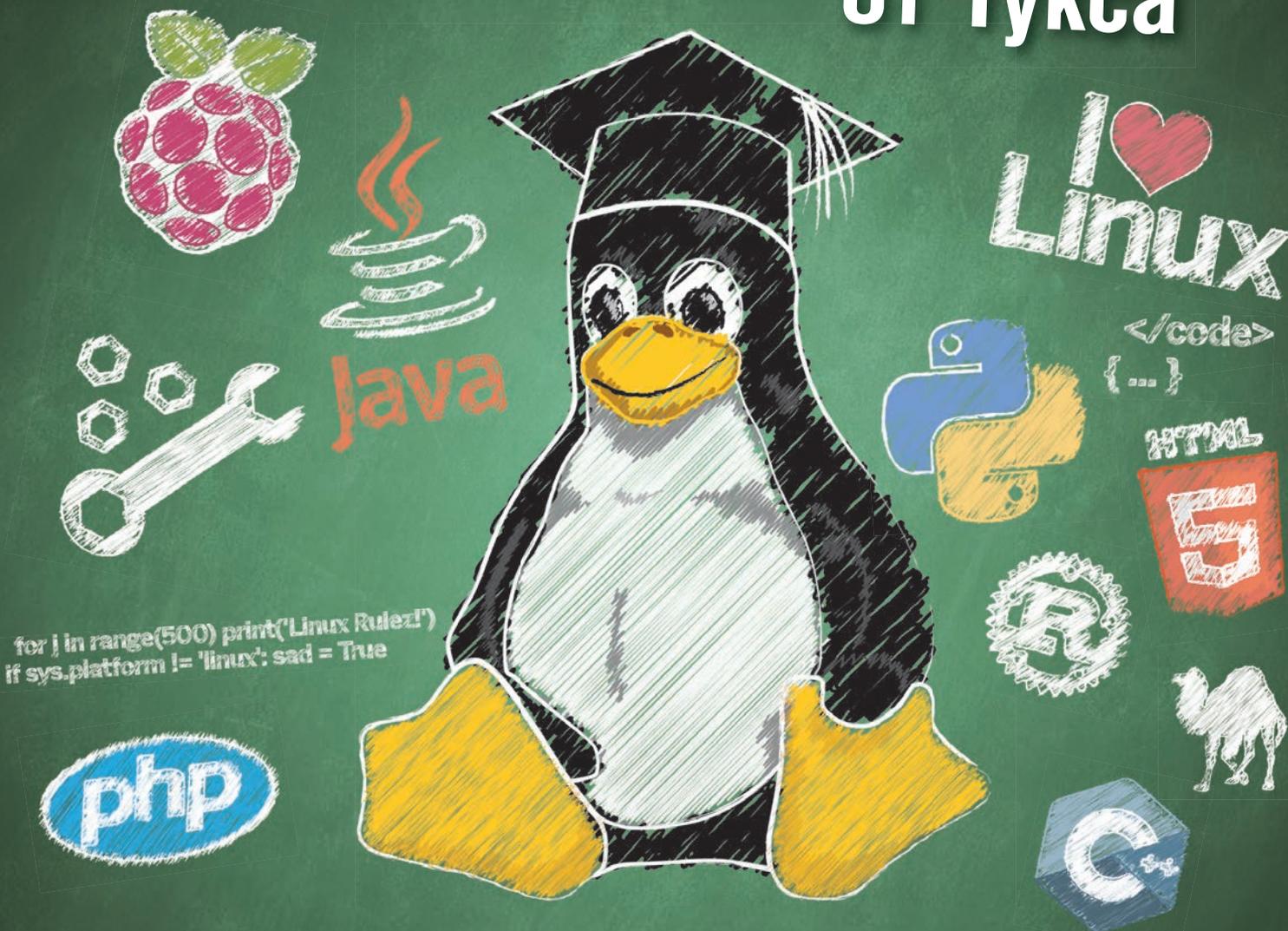
За пределами развивающегося мира хромбуков можно приискать более амбициозное устройство Android, например, Dell Venue 10 7000 (см. Обзоры, стр. 17), однако Android пока что не годится для настоящей многозадачности.

Нам всегда казалось, что работа на Android — своего рода компромисс, и сами устройства не всегда работают настолько хорошо, как должны бы.

Опции полноценного ноутбука с Linux на борту по-прежнему досадно ограничены; тем не менее мы наблюдаем, как отдельные магазины, например, Ebuyer, предлагают (странноватую) модель от Hewlett-Packard, и, разумеется, продолжается сотрудничество Lenovo и Ubuntu, с утвержденными сборками для многих моделей ноутбуков Lenovo. **LXF**



Идем в Академию кодинга от Тукса



Хотите научиться кодировать, но боитесь спрашивать? Ваш дружелюбный гид в мире кода — Джонни Бидвелл.

Программирование нынче в моде. Не умея программировать на Python, вы рискуете стать посмешищем в глазах своих же друзей... Это не пустая болтовня. Знание программирования является одним из основных для энтузиаста открытого кода. Будь то базовые скрипты *Bash* или умение читать PHP, знание программирования — больше, чем полезный навык: это основа. Сей-

час, когда программированию учат в школе, самое время отточить свои навыки, чтобы, если маленькая Джени попросит помочь запрограммировать на Python ее *Minecraft* из Pi, вы без страха приняли бы за дело.

Знакомство с программированием полезно и в работе с Linux: оно поможет вам эффективнее использовать терминал, откроет перед вами новые карьерные возможности или решит вам проблему

с некорректной работой web-страницы. Кроме некоторых затрат времени, вам это не будет стоить ни гроша, и для вас, как для пользователя FLOSS, доступ к любому языку лежит не далее [apt-get](#).

Итак, вашу руку! За несколько минут создадим забавную игру на Python, а потом выясним, какие языки подходят именно вам и вашим проектам и как вы справитесь с новой школьной программой и даже web-разработкой.

Знакомимся с программой

Э то будет очень плавное введение в программирование на Python: мы познакомимся с основами этого языка, а затем воспользуемся модулем *Pygame* для создания простой игры. То, что нам это удастся, является свидетельством мощи Python и *Pygame*: большая часть монотонной работы из процесса написания игры убрана, и после знакомства с элементарными конструкциями программирования мы, по большей части, будем работать с интуитивными командами.

Для начала следуйте инструкциям во врезке, чтобы выяснить, какая у вас версия Python, и установить *Pygame*. Используйте ли вы Raspberry Pi или большой компьютер, начните с командной строки. Откройте терминал (*LXTerminal* в Raspbian) и запустите Python 2.7 с помощью

```
$ python2
```

Альтернатива — запустить *IDLE*, а оттуда — командную оболочку, выбрав Python 2.7 Shell в меню Window. И то, и другое вызовет интерактивный интерпретатор Python, в котором у приглашения `>>>` можно вводить команды, да немедленно их и проверять. Здесь-то мы и создадим нашу первую программу Python, тщательнейшим образом введя следующее заклинание:

```
>>> print('Hello World')
```

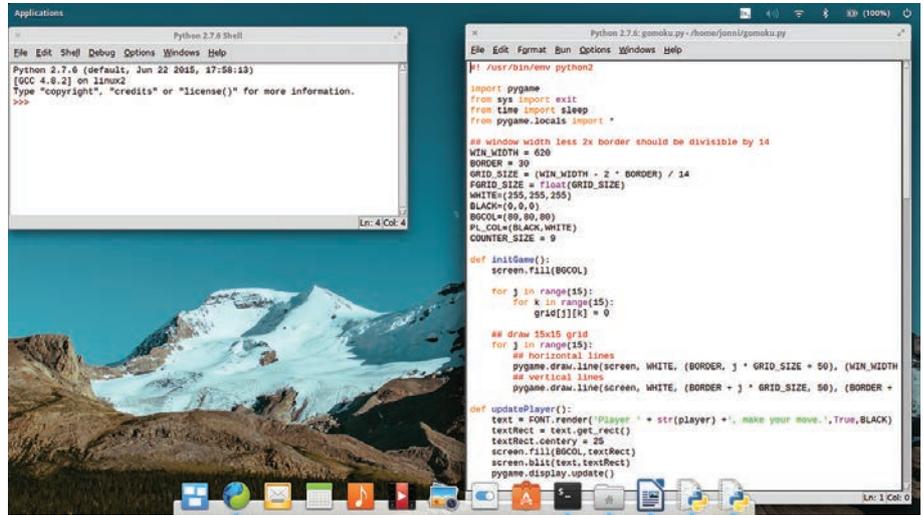
Как вы, вероятно, догадались, нажатие на Enter заставит Python отобразить общемировое приветствие. Из интерпретатора можно в любой момент выйти, нажав `Ctrl+D` или скомандовав `exit()`. Для создания более крупных программ есть смысл работать в текстовом редакторе или в *Integrated Development Environment* [интегрированной среде разработки] вроде *IDLE*, но интерпретатор — отличная возможность протестировать небольшие фрагменты кода. Поэтому сейчас для введения основных концепций и конструкций кода мы воспользуемся им. Для начала введем понятие переменной:

```
>>> name = 'Мафусаил'
```

```
>>> age = 930
```

```
>>> print(name, ' возраст ', age, ' лет.')
```

То есть мы можем присваивать переменным значения, изменять их по своему желанию и применять оператор печати, чтобы узреть их на экране.



➤ Среда разработки *IDLE* создана специально для Python. Вы можете установить ее на Ubuntu (или в данном случае на ElementaryOS) с помощью `apt-get install idle`.

Технически, скобки в строке с `print` (печать) нужны только в Python 3, но вреда от них нет, и это неплохая практика — по возможности писать универсальный код, способный работать в обеих версиях. Переменным в Python автоматически присваивается тип на основании их содержимого. Поэтому `name` [имя] получит тип `string` (строка символов), а `age` [возраст] — `integer` (целое число). Можете проверить это, введя `type(name)` и т.п.

Некоторые типы способны переводиться в другие — можно сделать `age` числом с плавающей запятой (с дробной частью) по команде

```
>>> age = float(age)
```

```
>>> age
```

Когда в интерпретатор вводится имя переменной, показывается ее значение, поэтому вы легко можете увидеть внесенные вами изменения. Целые числа и числа с плавающей запятой можно конвертировать в строковые переменные функцией `str()`. Python может пойти и в обратную сторону, например, конвертируя строковую переменную во `float`, но это работает, только если исходная строковая переменная выглядит приблизительно

так же, как и ее целевой тип: `float('10.0')` сработает, а `float('Rumplestiltskin')` — нет.

Как деление работает по-разному для `float` и `int`, так и сложение работает по-разному для строковых переменных. Здесь оператор `+` обозначает соединение: вторая строка лепится к концу первой. Таким образом,

```
>>> 'Hello ' + 'world'
```

```
'Hello world'
```

```
>>> str(123) + str(456)
```

```
'123456'
```

```
>>> 'Бетельгейзе' * 3
```

```
'БетельгейзеБетельгейзеБетельгейзе'
```

Последняя строка показывает, что строковые переменные можно умножать — а вот операции деления и вычитания для строковых переменных не определены.

Типы данных задают способ внутреннего представления данных, и результаты могут иметь весьма хитрые различия. Например, в Python 2 оператор деления `/` работает по-другому, если один из аргументов — `float`:

```
>>> 3/2
```

Устанавливаем Python и Pygame

Если вы используете Raspbian на Raspberry Pi или любую разновидность настольного Linux, то велики шансы, что у вас уже установлена как минимум одна версия Python (а то и две). Хотя последний релиз (1.9.2) теперь совместим с Python 3, ни один дистрибутив пока что не предлагает этой версии, поэтому в данном руководстве мы будем придерживаться Python 2 (2.7, если быть точным). Проверьте, какая версия Python стоит у вас по умолчанию, введя

```
$ python -V
```

Если вы получите результат, начинающийся с 2, то все отлично. Если, однако, вы видите нечто с 3, проверьте наличие у вас Python 2 командой

```
$ python2 -V
```

Некоторые дистрибутивы (из них выделяются Arch Linux и недавно вышедший Fedora 22) не предлагают серию 2.7 по умолчанию. Однако установка *Pygame* найдет ее в качестве зависимости, так что давайте сделаем это прямо сейчас. Пользователям дистрибутивов на базе Debian (включая Raspbian на Pi) надо устанавливать *Pygame* командой

```
$ sudo apt-get install python-pygame
```

Пользователи других дистрибутивов найдут пакет с похожим названием (в Arch это *python2-pygame*) и должны суметь установить его через соответствующий менеджер пакетов, то есть *pacman*, *yum*, *zypper* или еще какой-нибудь. Большинство дистрибутивов комбинируют со всеми установленными версиями Python среду *IDLE*; если вы не можете найти для нее значка, попробуйте запустить команду `idle` или `idle2`. Если ничего из этого не выйдет, поищите в репозитории своего дистрибутива.



```
>>> 1
```

```
>>> 3/2.
```

```
>>> 1.5
```

Разницу результатов обеспечила точка. Заметьте, что мы вводили просто 2., поленившись набирать 2.0. Главное в Python — краткость. (Да и зачем усложнять себе жизнь?) Если вам приходится делать много вычислений с плавающей запятой, вы рано или поздно столкнетесь с погрешностями округления. Посмотрите на такую красоту:

```
>>> 0.2 * 3
```

```
0.6000000000000001
```

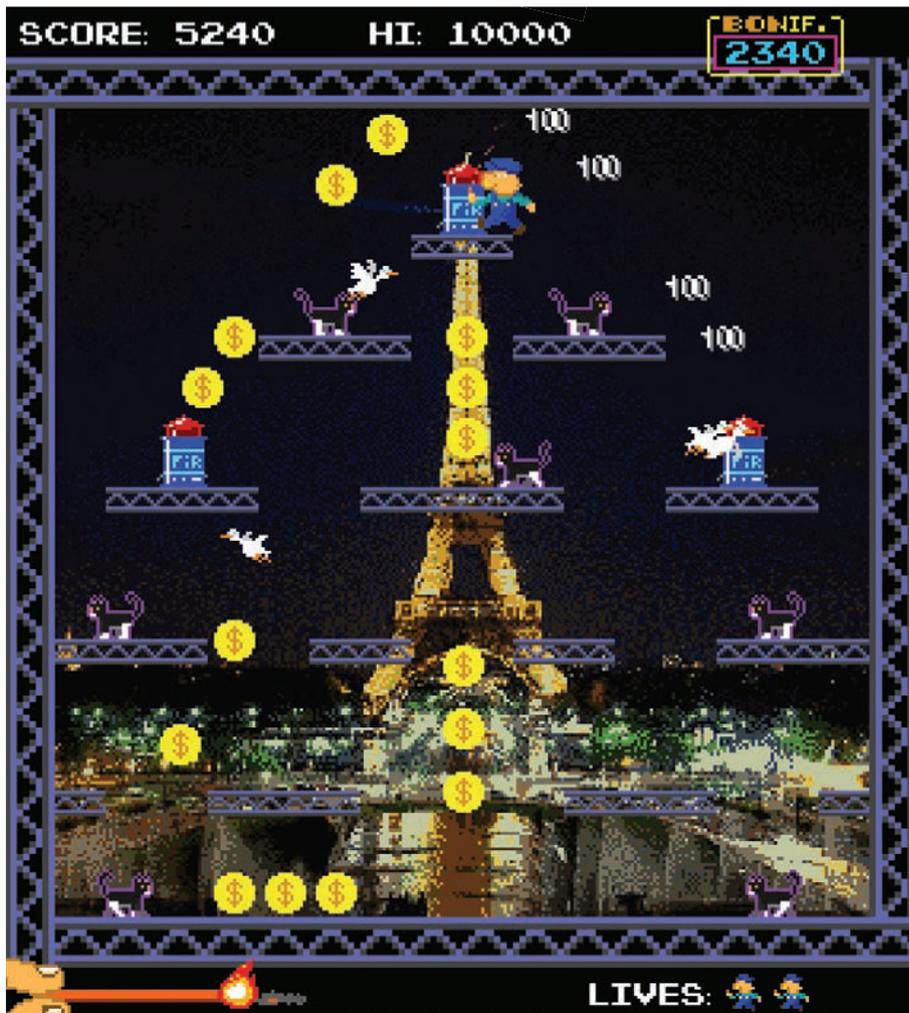
Подобные странности возникают, когда десятичные дроби переводятся в бесконечные двоичные. Иногда это проходит без последствий, но вам следует о них знать. Обойти их можно, преобразовав переменные с плавающей запятой в целые или используя функцию `round()`, что даст только требуемое вам число десятичных знаков. Мы увидим все это на практике, когда будем программировать нашу игру *Gomoku*.

Лезем в петлю

Очень часто программистам надо повторять почти одно и то же действие многократно. Например, вносить пункты в список, добавлять итоговое значение в каждый ряд таблицы или хотя бы вычитать энергию у всех врагов, только что сметенных лазерным ударом. Повтор одного и того же действия с каждым пунктом, строкой таблицы или врагом вручную привел бы к появлению затянутого и трудного для восприятия кода. На это у нас есть циклы [loop, *англ.* «петля» — например, скромный цикл `for` ниже. Если нажать Enter после первой строки, приглашение `>>>` превратится в ... Дело в том, что интерпретатор понимает: далее пойдет внутренний блок кода, и строка (строки), следующие за компонентом `for`, «принадлежат» этому `for`. Такие блоки кода должны быть смещены вправо; обычно для этого используется четыре пробела, или вы можете поставить сколько хотите, но уже не меняя выбранное количество. Если вы не отодвинете вправо вторую строку, Python вас обругает. Ввод пустой строки после оператора `print` завершит блок кода и запустит наш цикл.

```
>>> for count in range(5):
```

```
... print("iteration #", count)
```



► На *Pygame* созданы самые разные игры. Эта, *You Only Get One Match*, использует огромное количество фейерверков, но ограниченное число средств поджига. Вы найдете ее на <http://bit.ly/LXF202-onematch>.

И вот что здесь происходит. Мы ввели новую переменную — целочисленную, с именем `count`. Удобно присваивать переменным значения названия; в данном случае становится ясно (даже при отсутствии других навыков программирования), что должен произойти некий подсчет. Функция `range()`, используемая отдельно, возвращает список из ряда целых чисел. О списках мы поговорим буквально через минуту, а сейчас нам будет

достаточно знать, что `range(5)` выглядит так: `[0, 1, 2, 3, 4]`, можете проверить это в интерпретаторе. Итак, наша переменная `count` будет выполнять перебор всех этих пяти значений, и строка `print()` будет выводиться пять раз — по разу на каждое значение в ряду.

Другой тип цикла — `while`. Вместо перебора по списку наш симпатичный цикл `while` будет сновать по блоку кода до тех пор, пока не нарушится

Как играть в Gomoku

Gomoku — сокращенное *gomokunagabe*, что приблизительно переводится с японского как «пять предметов в ряд». На самом деле игра появилась в Китае около 4000 лет назад. Игроки по очереди помещали фишки на пересечении квадратов с целью формирования сплошных рядов (горизонтальных, вертикальных или диагональных) из 5. Традиционно поле было размером 19×19 квадратов, но мы решили выбрать меньшее по размеру поле, 15×15, используемое в некоторых разновидностях. Мы не включали в игру ИИ (это было бы слишком сложно в руководстве для начинающих), так что вам придется самим искать партнера для игры.

В порядке альтернативы можете обратиться к одной из множества онлайн-версий, а у пользователей KDE на рабочем столе есть похожая игра *Govu*.

Можно запросто выработать некие стратегии: например, всегда блокировать одну сторону «открытой линии из трех» вашего противника или создавать препятствие «ломаной четверке». Однако на достижение подлинного мастерства уйдут годы. Основной набор правил в том виде, в котором мы его применили, очень на руку тому игроку, у которого первый ход (по традиции он играет черными). В принципе, работа Л. Виктора Эллиса [L. Victor Allis] продемонстрировала, что хороший игрок

(точнее, игрок должен быть превосходным) всегда выигрывает, если начинает игру первым. Чтобы минимизировать действие этого принципа, на крупных турнирах используется стартовая стратегия под названием *swap2*. Первый игрок помещает на поле две черных фишки и одну белую, и затем второй игрок или выбирает цвет, или размещает еще одну черную и еще одну белую фишку на поле и дает первому игроку возможность выбрать цвет. Вы можете изменить код, чтобы принудительно использовать *swap2*, но вполне можно применять это правило и без изменения кода: просто не обращайтесь внимания на первые подсказки «Игрок x, ваш ход».

некое условие. В следующем примере таким условием является заявление пользователя, что он родился после 1900-го, но до 2016 года.

```
>>> year = 0
>>> while year < 1900 or year >= 2015:
... year = input("Введите свой год рождения: ")
... year = int(year)
```

И снова сам цикл смещен вправо, и снова вам требуется ввести пустую строку, чтобы он сработал. Для сравнения значений мы применили операторы «меньше (<)» и «больше или равно (>=)». Условия можно соединять логическими операторами `and` [И], `or` [ИЛИ] и `not` [НЕ]. Пока год имеет неподходящее значение, мы продолжаем задавать вопрос. Изначально он настроен на 0, что заведомо меньше 1900, поэтому цикл у нас запускается гарантированно. Мы использовали функцию `input()`, которая возвращает любую строку, введенную пользователем. Строку мы храним в переменной `year`, конвертировав ее в целочисленную переменную, чтобы не сбить сравнения в строке `while`. Во всем, что касается вводимой пользователем информации, стоит быть предельно осторожным: злонамеренный пользователь может ввести определенную информацию, которая приведет к сбою, что в нашем примере не так уж страшно, но все же чревато, если, скажем, это случилось с `web`-приложением, обращающимся с важными базами данных. Вы можете изменить дату 1900, если считаете, что ваша программа может пригодиться тем, кому уже больше 115 лет. Точно так же можно изменить и 2015, если вы хотите отогнать (правдивых) малолеток.

Чисто по списку

Мы уже упоминали списки, и в потрясающем проекте, который нас ждет впереди, будем активно их употреблять, поэтому необходимо рассказать, что же это такое. Списки в Python являются гибкими конструкциями, которые хранят серии индексированных пунктов. По указанным пунктам нет никаких ограничений: они могут быть строковыми переменными — `string`, `int` [целочисленными переменными], другими списками или сочетаниями вышеперечисленного. Списки определяются посредством заключения разделенного запятыми списка в квадратные скобки. Например:

```
>>> myList = ['Яблоко', 'Банан', 'Крыжовник']
```

Единственный подвох здесь в том, что списки имеют нулевой начальный индекс, поэтому мы получаем доступ к первому пункту нашего списка с помощью `myList[0]`. Если у вас чересчур человеческий образ мысли, то более логичными для вас будут списки с индексом 1. Python их ничуть не уважает, и если вы думаете по-человечески, готовьтесь к появлению классических ошибок смещения на единицу.

Вот так мы можем изменить последний пункт в списке:

```
>>> myList[2] = 'Ктулху'
```

Списки можно декларировать менее буквально — например, если мы хотим инициализировать список со 100 нулями, можно сделать так:

```
>>> zeroList = [0 for j in range(100)]
```

Это называется списочное выражение.

Другой пример —

```
>>> countList = [j for j in range(100)]
```

результатом чего будет список, содержащий целые числа от 0 до 99 включительно, чего в Python 2 с тем же успехом можно добиться посредством `range(100)`. Однако наша концепция более мощная: например, мы можем получить список квадратов с помощью оператора потенцирования (возведение в степень) `**` —

```
>>> squareList = [j ** 2 for j in range(100)]
```

После этого экспресс-курса мы готовы приступить к программированию нашей собственной игры. Вы найдете весь код на диске (в Учебниках, в файле `gomoku.py`) или на <http://pastebin.com/FRe7748B>, поэтому глупо было бы воспроизводить его здесь. Вместо этого мы сосредоточимся на интересных моментах, в некоторых случаях предлагая более удобоваримый фрагмент, с которым вы сможете поиграть и, надеемся, увидеть, как он превращается в версию программы.

Чтобы приступить к работе и посмотреть игру в действии, скопируйте `gomoku.py` в свою домашнюю директорию и запустите ее командой

```
$ python2 gomoku.py
```

С другой стороны, если вы хотите посмотреть на код, откройте тот файл в `IDLE` или в своем любимом текстовом редакторе. Разумно будет начать с первой строки... Выглядит она так:

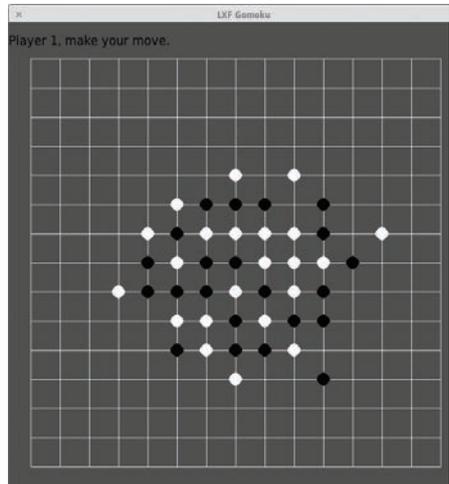
```
#!/usr/bin/env python2
```

Эту строку Python игнорирует (как и все строки, начинающиеся с `#`), однако она используется оболочкой, чтобы определить, как исполнять файл. Для поиска и запуска исполняемого файла в Python 2 мы здесь используем утилиту `env`, которая должна присутствовать на всех платформах. Чтобы этот хитрый трюк сработал, вам следует превратить файл `gomoku.py` в исполняемый — это делается из командной строки (если вы скопировали файл в свою домашнюю директорию или еще куда-нибудь, где вы имеете право выдавать разрешения) при посредстве

```
$ chmod +x gomoku.py
```

Вы увидите, что теперь вы можете запускать игру более лаконично:

```
$. /gomoku.py
```



» Одна из множества напряженных настольных баталий, выигранных Джонни. [Ред.: — Ты, небось, играл сам с собой.]

Далее у нас три конструкции, две из которых (`pygame` и `sys`) просты. Модуль `pygame` легко справляется со всем, что относится к игре — мы на самом деле только прошли по поверхности, коснувшись рендеринга графики и шрифтов. Из модуля `sys` нам нужна единственная функция, `exit()`, чтобы мы могли аккуратно закрыть игру по ее окончании. Вместо того, чтобы импортировать весь модуль `sys`, мы импортируем только эту функцию. Последняя строка импорта — просто для удобства: мы уже импортировали `pygame`, которая дала нам доступ к `pygame.locals`, набору констант и переменных. Мы использовали только те, которые относятся к мыши, клавиатуре и выходу. Присутствие здесь этой строки означает, что мы можем получить доступ, скажем, к любым событиям кнопки мыши с помощью `MOUSEBUTTONDOWN`, не предвзято ее `pygame.locals`.

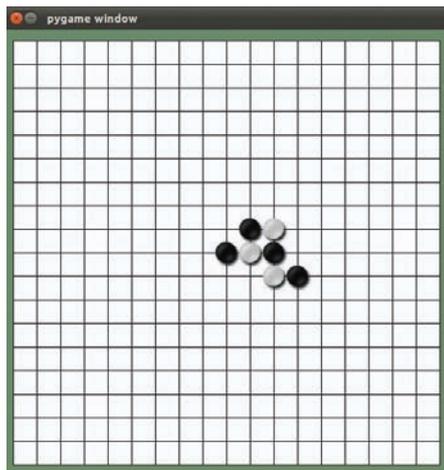
Все это в (Py)game

По всей программе вы будете замечать, что имена некоторых переменных написаны заглавными буквами, а некоторых — нет. Те, что написаны заглавными буквами — или из `pygame.locals`, или их надо считать константами, то есть такими штуками, которые не меняют своего значения на протяжении всей игры. Большинство из них объявлено после конструкции импорта и управляют такими параметрами, как размер игрового окна и фишки. Желая изменить цвет фишек, например, на красный и синий, вы можете заменить значения `WHITE` и `BLACK` на `(255,0,0)` и `(0,0,255)` соответственно. Эти переменные являются кортежем (структурой, похожей на списки, которая не может изменяться), задающим красные, зеленые и синие цветовые компоненты.

Далее вы увидите серию блоков, начинающихся с `def`: — это определения функций, и, как в случае с другими блоками кода в Python, они сопровождаются смещением вправо. Функция `initGame()` инициализирует игровую область. Вот простая версия, которая показывает, что делает эта функция:

```
WIN_WIDTH = 620
GRID_SIZE = (WIN_WIDTH) / 14
WHITE=(255,255,255)
BLACK=(0,0,0)
BGCOLOR=(80,80,80)
def initGame():
    screen.fill(BGCOLOR)
    for j in range(15):
        pygame.draw.line(screen, WHITE, (0, j * GRID_SIZE), (WIN_WIDTH, j * GRID_SIZE))
        pygame.draw.line(screen, WHITE, (j * GRID_SIZE, 0), (j * GRID_SIZE, WIN_WIDTH))
    pygame.init()
    pygame.display.set_caption("LXF Gomoku")
    screen = pygame.display
    set_mode((WIN_WIDTH,WIN_WIDTH))
    initGame()
    pygame.display.update()
```

Если добавить три строки импорта в самое начало этого кода, получится вполне рабочая программа Python. Функция `initGame()` не делает ничего, пока она не будет вызвана последней строкой; к тому времени мы уже инициализируем `Pygame`,



➤ Графическая *Go*токи Джозля Мюриэля [Joel Murielle] доступна на сайте *Pygame*. *Pygame* избавляет вас от работы со спрайтами, печально известными (наравне с зльфами) своей способностью создавать проблемы.

зададим имя нашего окна и установим размер окна в 620 пикселей. Все переменные задаются вне определений функций, следовательно, пять констант заглавными буквами в начале и `screen` доступны внутри определений функций; они именованы глобальными переменными. Переменные, определенные внутри определений функций, называются «локальными» — они прекращают свое существование после выхода из функции, даже если у них такое же имя, как и у глобальной переменной; это тоже стоит учитывать в вашей будущей деятельности как программиста. Переменная `screen` относится к «холсту», на котором будет изображена наша игра, и попозже она будет активно использоваться. Первое действие функции `initGame()` — раскрасить этот холст в приятный оттенок серого (если вы сменили цвет, то и на здоровье). Потом мы в цикле нарисуем горизонтальные и затем вертикальные линии, для нашей сетки 15 × 15. Ни одно из этих художеств не появится, пока мы не велит *Pygame* обновить изображение, отсюда и последняя строка.

Особо пронизательные читатели заметят, что сетка слегка выходит за края. Причина в том, что

изображение 15 равноудаленных параллельных линий делит поле на 14, однако 620 (размер нашего окна) на 14 не делится. Но когда мы добавим границы окна, поскольку мы хотим размещать фишки также и по краям, 620 окажется очень хорошим числом; к тому же нам было лень его менять. И хотя по краям это смотрится не совсем аккуратно, все же это свидетельство мощи *Pygame* и простоты Python, которые и позволяют нам такое сделать всего несколькими строками кода. Но не будем забегать вперед: наша игра пока что не делает ничего.

Тонкие моменты

Начиная отсюда и далее, мы будем говорить о настоящем коде, поэтому все фрагменты кода, которые мы будем упоминать, не будут работать в изоляции — они нужны для подчеркивания определенных тонкостей. Вы заметите, что переменная `FONT` не определена вместе с другими константами: причина в том, что нам нельзя использовать поддержку шрифтов *Pygame*, пока не вызван метод `init()` в *Pygame*.

Давайте рассмотрим основной цикл игры в конце кода. Вводный компонент `while True:` предполагает, что этот цикл будет совершаться вечно. По большей части это правильно — нам надо постоянно проверять наличие событий, а именно нажатий на кнопку мыши или нажатий пользователем на кнопку `exit`, пока игра не будет закончена. Вполне очевидно, что мы выходим из цикла, когда приложение закрывается — нажатие на кнопку запускает событие `QUIT`, на которое мы реагируем с помощью функции `exit()` из пакета `sys`. Внутри основного цикла первое, что нам надо сделать — вызвать функцию `updatePlayer()`, которую вы найдете в строке 32. Она обновляет текст вверху экрана, который сообщает о том, чей сейчас ход, рисуя («блитуя») сначала цельный прямоугольник, так, что любой предыдущий текст стирается. Далее мы проводим цикл для событий в очереди событий; когда игрок пытается сделать ход, вызывается функция `tryCounterPlace()`, которая передает координаты мыши.

Чтобы отслеживать игру, мы здесь используем двумерный квадратный массив (список списков)

в переменной `grid`. Он инициализируется сплошными нулями, а когда игрок делает ход, вводится соответственно 1 или 2. Первой задачей функции `tryCounterPlace()` является синхронизация координат мыши в том месте, где пользователь щелкнул, с парой координат, которыми индексирована переменная `grid`. Конечно, пользователь может щелкнуть на пересечении линий не точно, и нам придется заниматься округлением. Если пользователь щелкнул за пределами сетки (например, слишком высоко над сеткой, тогда координата `y` будет отрицательной), то функция возвращается к основному циклу. В ином случае мы проверяем, не занято ли это место в сетке, и если не занято, рисуем там кружок и обновляем наш массив состояний `grid`. При удачном ходе наша функция возвращает значение `True`, и взглянув на строку 111 кода, мы увидим, что оно передает ход другому игроку. Но перед тем, как это будет сделано посредством вызова `updatePlayer()` в начале цикла, мы вызываем функцию `checkLines()`, чтобы посмотреть, не создалась ли во время последнего хода сплошная линия фишек, означающая победу. Вы найдете подробности того, как это проверить, во врезке.

Когда с помощью нашего рабочего алгоритма определения обнаружена победная фишка, вызывается функция `winner()`. Она заменяет текст вверху экрана сообщением, в котором оглашается победитель, и запускается цикл `gameover`. Он ожидает, когда пользователь нажмет на `R` для перезапуска или выхода. Если дана команда перезапуска, то приказ пользователя сохраняется, и, поскольку он обновляется немедленно перед вызовом `checkLines()`, результатом будет то, что следующий раунд начинается проигравший.

Это небольшой проект (всего около 120 строк, даже не сравнить с 487 байтами *Bootchess*, о которой вы можете прочесть на www.bbc.co.uk/news/technology-31028787), но его можно расширить, причем множеством разных способов. Можно, например, добавить графику, а также игру в сети; и, что весьма амбициозно, вполне можно задействовать некий рудиментарный ИИ, создав одиночный режим. На самом деле, именно последнее и было сделано...

Читая между строк

Часть Ключевого этапа 2 [школьной программы] подразумевает, что вы должны научиться понимать и программировать простые алгоритмы. Мы уже говорили о нашем базовом алгоритме игры — дожидаться нажатия на кнопку мыши (или выхода пользователя из игры), проверить, был ли ход по правилам, проверить, нет ли линии из пяти фишек, и т. д. В сердце этой последней стадии лежит наивный, но тем не менее уместный алгоритм определения, является ли ход победным.

Рассмотрим более простой случай, когда нас интересуют только горизонтальные линии. Тогда мы должны прогнать цикл сначала по рядам, а затем по столбцам нашего массива сетки. Для каждого элемента мы должны проверить, что он

ненулевой (т. е. на нем стоит фишка), и имеют ли четыре элемента справа от него то же значение. В Python это выглядит так:

```
for j in range(15):
    for k in range(10):
        pl = grid[j][k]
        if pl > 0:
            idx = k
            while grid[j][idx] == pl and idx < 14:
                idx += 1
            if idx - k >= 5:
                # действия на случай победы
```

Обратите внимание, что переменная `k` в внутреннем цикле имеет максимальное значение 9. Нам не надо проверять положение в ряду дальше этого

значения вправо, поскольку наш алгоритм достигнет тех положений, куда доходили бы потенциальные линии. Наша переменная `idx` эффективно измеряет длину любой линии; она увеличивается с помощью оператора `+=`, сокращение для `idx = idx + 1`.

Алгоритм легко адаптируется для вертикальных и диагональных линий. Однако вместо того, чтобы пользоваться четырьмя разными функциями, мы поступили умнее и создали одну общую функцию `lineCheck()`, которую вызвали четыре раза с параметрами, требуемыми для каждого типа проверяемых линий. Указанные параметры просто изменяют границы циклов `for` и способ увеличения или уменьшения позиции в сетке для каждого направления линий.



Языки: Обзор

Одним из величайших технологических достижений был компилятор IBM Fortran, который появился в 1950-х. Он позволял писать программы для компьютеров, используя нечто не столь неудобное, как машинный код. Fortran и сегодня все еще достаточно широко используется, и, хотя некоторые презрительно фыркают на этого динозавра, он остается весьма актуальным, особенно для научных ИТ. Тем не менее, добровольно изучать его никто не станет, и появилось множество самых разных других языков.

Традиционно вам предлагается выбор между трудными и быстрыми языками — такими как Java, C или C++ — и простыми и медленными, типа Python или PHP. Быстрые языки склонны компилироваться: прежде чем код может быть запущен, его надо переделать в машинный код. Динамические языки конвертируются в машинный код прямо на ходу. Однако на определенном уровне все языки программирования одинаковы — имеются некие базовые структуры, например, циклы, условия и функции, и различие заключается лишь в том, как все это преподносится.

Для тех, кто только-только осваивает программирование, такое изобилие очень осложняет дело. Мнения о том, какой язык учить первым, диаметрально противоположны, однако правда в том, что одного языка тут не существует, хотя для мелкого народца мы от души рекомендуем Scratch. Любый язык, с которым вы познакомитесь, будет то восхищать, то бесить вас. Однако начинающим мы бы не рекомендовали C или Haskell.

Весьма популярное мнение склоняется в сторону Python; мы бы и рады его поддержать, однако многих отпугивает разницей между Python 2 и 3 (см. LXF195). За Python говорит многое: это, вероятно, один из самых человеко-читаемых языков. Например, для удобства чтения следует в своем коде использовать сдвиг вправо, а в Python он делается принудительно. Благодаря этому, Python может избавиться от фигурных скобок, которые во многих языках применяются для целей вложенности. Ну и нет нужды ставить точку с запятой в конце каждой строки.

В Python имеется огромное количество дополнительных модулей — мы уже видели *Pygame* и наш любимый, API для программирования *Minecraft* на Pi.

Дружелюбный к новичкам

Другие языки, подходящие начинающим — JavaScript и PHP. Их популярность в основном происходит из их употребительности в Сети. JavaScript работает на стороне клиента (вся работа делается web-браузером), а PHP относится к стороне сервера. И если вас интересует web-программирование, любой из них отлично вам подойдет. Вам также понадобится изучить основы HTML и, вероятно, CSS, чтобы ваша программа выглядела привлекательно, но это на удивление несложно. С косметической точки зрения PHP не так опрятен, как Python, но очень скоро (совсем как в «Матри-

це») вы начнете видеть сквозь скобки и долларové значки. В категории доступных языков стоит также упомянуть Ruby. Он появился благодаря желанию своего создателя, Юкиhiro Матцумото [Yukihiro Matsumoto], получить нечто «столь же мощное, как Perl, и более объектно-ориентированное, чем Python».

Не проходит и дня без возникновения в популярных программах таких проблем, как переполнение буфера или проблем с использованием освобожденной памяти. Просто загляните на <https://exploit-db.com>. Все они сводятся к ошибкам программирования, но некоторые такие ошибки отследить проще, чем другие. Одна из проблем с быстрыми языками в том, что они небезопасны для памяти. Программист должен распределять память так, как требуется, и освобождать ее, когда она уже не нужна. Если этого не сделать, программа примется вытворять то, чего ее никто не просил.

Но, к сожалению, не менее 40 лет широкого применения C показали нам, что люди, с присущей им безалаберностью, не слишком преуспевают в этой задаче, и никакого улучшения ждать не приходится. В ответ на наши неутешительные результаты в данной области появляется новое поколение языков. Мы видим языки, разработанные в Google (Go), Apple (Swift) и Mozilla (Rust). Их общая цель — быть сравнимыми по скорости с C, но при этом гарантировать безопасность памяти, столь необходимую в этом несовершенном мире, который кишит злоумышленниками.



Любой язык будет то восхищать, то бесить вас, но за Python говорит многое.

Руст недавно отметил свой релиз 1.0, и, возможно, в один прекрасный день Firefox переписут на нем, но в данный момент там многовато странностей и капканов, которые не приведут в восторг пользователей традиционных языков. Для примера скажем, что совершенно правильная программа запросто может отказаться компилироваться. Компилятор Rust нацелен скорее на корректность, чем на полноту — всё, что он компилирует, имеет высокую степень гарантии качества, но Rust ни за что не скомпилирует код, где увидит хотя бы тень сомнения, даже если эта тень существует лишь в воображении компьютера. И так, кодерам придется лезть из кожи вон, однако результат того стоит: помимо безопасности памяти и умозаключения о типах, Rust также превосходно справляется с параллелизмом (несколько потоков и процессов), гарантируя потокам безопасность и свободу от условий «состязания».

»

Программные парадигмы и жаргон

В императивном программировании порядок исполнения по большей части фиксирован: все происходит последовательно. Наш пример *Gomoku* в основном выполнен в императивном стиле, но применение функций делает его более процедурным — исполнение прыгает с функции на функцию; однако все равно сохраняется непрерывный процесс. Объектно-ориентированный (ОО) подход идет еще дальше. Программы ОО определяют классы, употребляемые во многих экземплярах; каждый класс является шаблоном для объекта,

способного иметь собственные переменные (атрибуты) и собственный код (методы). Это облегчает некоторые задачи, в частности, задачу распределенного доступа к данным без обращения к вызовам длинных функций или к запутанным глобальным переменным. Также это влияет на производительность, и с этим довольно трудно разобраться. Очень мало языков являются ОО в чистом виде; исключения — Ruby и Scala. C++ и Java поддерживают некоторые процедурные элементы, но они находятся в меньшинстве.

Функциональное программирование (ФП) уходит корнями в логику, и оно не для слабаков. Это очень абстрактный стиль, в котором программы скорее подчеркивают то, что хотят сделать, чем то, как они хотят это сделать. Главное в ФП — отсутствие побочных эффектов: функции возвращают только новые значения, нет никаких глобальных переменных. Сюда относятся более логичные языки, такие, как Lisp, Scheme, Haskell и Clojure. Долгое время ФП было вотчиной науки, однако сейчас оно популярно также и в индустрии.



Программирование в школе

В сентябре 2014 г. Великобритания предприняла новаторский шаг, включив программирование в государственную образовательную программу. Когда об этой инициативе объявили в 2013 г., министр образования Майкл Гоув [Michael Gove] признал, что текущая программа по ИТ устарела — она «не полезнее, чем обучение детей отправке телекса или путешествию на дирижабле». Куда важнее было передать юным падаванам знания о коде. Умение программировать сильно востребовано, о чем свидетельствуют сложности в подборе квалифицированного персонала на технические должности в Великобритании.

С большой помпой запустили «Год Кода», который, однако, слегка утратил популярность, когда стали известны определенные подробности: к уже существующему финансированию обучения ИТ предполагалось добавить сущую мелочь, и большая часть этой суммы должна быть потрачена на обучение 400 «Ведущих учителей», которые впоследствии должны передать свое мастерство младшим учителям по всей стране. Любители позлорадствовать получают большое удовольствие от интервью в новостной программе BBC с руководителем Года Кода того времени: та открыто признает, что сама программировать не умеет, несмотря на свои же утверждения о том, насколько это важно для понимания мироустройства.

Возможности обучения

Если оставить в стороне критику и насмешки, нас невероятно радует, что дети в возрасте 5 лет уже сейчас изучают темное искусство синтаксиса, семантики и символизма. Так что бойтесь, родители, скоро ваше чадо будет не просто вопить, вопрошая вас: «А множественное число от мангуста — это как?», а потребует разъяснений насчет рекурсии и абстракции. Хорошо это или плохо, но многие частные репетиторы выиграют от количества озабоченных родителей и запутавшихся детей, которое появится в результате реализации новой программы обучения ИТ. Однако есть более здравые направления, где можно найти помощь.

Во-первых, вы всегда можете найти отличные руководства здесь, в LXF — просто еще раз взгляните на предыдущие пять страниц; это исключительный источник знаний. В Сети также есть множество свободных ресурсов. Некоторые из них (скажем, официальная документация по Python) детям покажутся суховатыми, но мы настоятельно рекомендуем родителям приобретать эти навыки вместе со своими отпрысками. Если оживить старую машину, установив на нее чистую систему Linux, у вас появится отличная платформа для этого. Все необходимые вам программы — свободные, а такие дистрибутивы, как Mint и Ubuntu, просты в освоении.

Помимо отдельного ПК, другим отличным способом создания обучающей платформы станет Raspberry Pi. Если вы хотите использовать более



➤ Корпус FUZE дает вам все необходимое, чтобы приступить к работе с регистрами или учинить какую-нибудь пакость с помощью GPIO.

старую модель B+, вы сможете ее приобрести за £25, и она отлично подключится к вашему телевизору. Конечно, придется найти еще клавиатуру, мышь, SD-карту, кабель HDMI и, возможно, беспроводной адаптер, если нельзя протянуть кабель Ethernet к вашему роутеру. К счастью, имеется немало готовых наборов (например, Kano, который позиционируется как компьютер DIY [Do It Yourself — *англ.* «сделай сам»]), способных предоставить вам эти дополнительные периферийные устройства для Pi, которые защитят его от грязи и царапин. Нам особенно понравились такие варианты, как, например, FUZE, который превращает Pi в более традиционное, хотя и более массивное устройство всё-

узнав, что они не будут привязывать этот продукт к разрабатываемой ими платформе Windows 10, незавершенная бета-версия которой вышла в конце июля. В этих устройствах имеются датчики гравитации и движения, а также Bluetooth и несколько кнопок. Есть пять колец GPIO, к которым можно присоединять другие датчики или иные хитроумные приспособления. Их можно программировать на целом ряде языков, в том числе C++, Python, JavaScript и Blocks — визуальном языке программирования. Все предоставленные Micro-



Без программирования программа по ИТ была «не полезнее, чем обучение детей отправке телекса».

в-одном. Да, крошечный форм-фактор Pi весьма привлекателен, но учитывая количество подключаемых устройств USB, все это легко превратить в такой миниатюрный ад из проводов.

Кстати о крошечных устройствах: дабы соответствовать новым образовательным стандартам, BBC собирается распространить около миллиона компьютеров “Micro: bit” для учеников 7-го класса. Они еще меньше размером, чем Pi, но у них нет никаких средств вывода, кроме матрицы светодиодов 5×5. Следовательно, в отличие от Pi, они не смогут функционировать как отдельные компьютеры — вместо этого их придется программировать с более мощного устройства. Microsoft щедро предоставляет ПО и обучение для этого предприятия, и мы испытали большое облегчение,

и soft редакторы кода сетевые, и вводимый сюда код должен быть скомпилирован перед загрузкой в Micro:bit. Пока среди предшествующего запуску затишь информация очень краткая, но, похоже, вся эта компиляция проводится на серверах Microsoft. Будет очень обидно, если редакторы кода нельзя будет использовать оффлайн или не будет возможности компилировать локально. Более подробную информацию см. на www.bbc.co.uk/mediacentre/mediapacks/microbit.

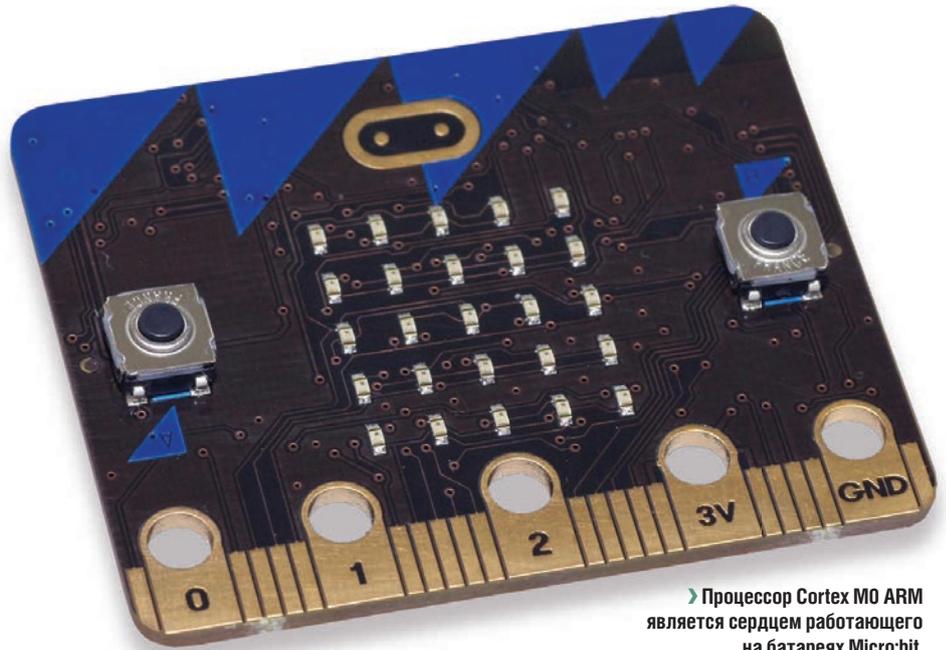
Производители Micro:bit не преминули подчеркнуть, что это устройство ни в коей мере не будет конкурировать с Raspberry Pi: оно, скорее, будет его дополнять. В Micro:bit имеется 20-контактный коннектор для соединения его с Pi или иным устройством, чтобы они могли работать в тандеме.

```
for j in range(500) print('Linux Rulez!')
if sys.platform != 'linux': sad = True
```

Подобное соединение будет совершенно необходимо Micro:bit для коммуникации с внешним миром. У устройства — на самом деле, у сочетания этого устройства с Pi — большой потенциал, однако присутствует и изрядная доля скепсиса по поводу того, возникнет ли такой же ажиотаж, как в 1982 г. при выходе BBC Micro. Тогда компьютеры были чем-то новым и волнующим, а в наши дни дети рассчитывают получить просто сплошной поток развлечений в виде шестисекундных видео с котятками или комментариев в соцсетях длиной 140 символов. И далеко не все дети с энтузиазмом воспримут идею о необходимости их программировать. Однако всё, что снижает, если не устраняет полностью, любое начальное препятствие к изучению программирования, нас радует. И мы также с нетерпением ждем, когда отстающие ученики начнут хакать друг у друга Micro:bit'ы или использовать их в коллективных DDoS-атаках на свою школьную инфраструктуру.

Рассмотрим ИТ-программу

Программа включает три Ключевых этапа. Первый, для детей 5-6 лет, вводит понятие алгоритма в самом общем смысле. Алгоритмы описываются в виде рецептов и расписаний, чтобы познакомить учащихся с идеей формализации инструкций. Второй этап (7–11 лет) вводит такие основные понятия, как циклы и переменные. Вместе с этим школьники будут учиться использовать web-сервисы и возможности сбора данных. Последний этап, для средней школы (11–14 лет), требует изучения не менее двух языков программирования и понимания двоичной арифметики, Булевой алгебры, функций и типов



➤ Процессор Cortex M0 ARM является сердцем работающего на батареях Micro:bit.

данных. Учащиеся также коснутся теории информации — по крайней мере, того, как разные типы информации могут быть представлены в виде потока битов. Вдобавок они получат представление о взаимодействии аппаратной и программной составляющих.

В процессе обучения они также осваивают жизненно важные навыки обеспечения конфиденциальности онлайн и информационной безопасности — это навыки, недостаток которых уже привел

к множеству корпоративных и правительственных оплошностей. В общем и целом, это весьма амбициозный проект; но, возможно, подобный радикальный шаг необходим, чтобы ликвидировать нехватку знаний и умений в данной области. При известном везении эта схема приведет к столь необходимой диверсификации в среде кодеров. Если это сработает и школьники будут учить Python наряду с китайским, а попутно к Канту изучать Кохонена и Кнута, это будет потрясающе.

Клубы программирования



В Великобритании насчитывается более 2000 добровольных клубов программирования. Code Club, вооруженный £100 000, любезно предоставленными Google, предоставляет материал, а школы (или иные организации) предоставляют место и ресурсы для этой благородной внеклассной деятельности.

Программа Code Club нацелена на детей 9–11 лет и состоит из двух семестров программирования на Scratch, затем одного семестра web-дизайна (HTML и CSS), и завершается это все финальным семестром взрослого программирования на Python. Проекты очень тщательно разработаны, чтобы поддерживать в детях интерес: они будут ловить привидений и участвовать в гонках на лодках в Scratch, и развлекаться в Turtle, и следить за самыми последними Покемонами в Python — и это далеко не все.

Если у вас есть время и хоть какие-то знания, вам стоит попробовать себя в качестве инструктора. Все руководства уже готовы, и если вам удастся убедить местную начальную школу приютить вас, то к вам приставят учителя для поддержания порядка и обороны от хулиганов. Однако у вас потребуют проверки знаний. Code Club также предоставляет три специальных модуля для учителей, в чьи обязанности входит работать по новой образовательной программе по компьютерным технологиям.

Такой образ мысли сообщества совершенно в духе открытого кода; он открывает дверь в мир программирования для всех, без какого бы то ни было коммерческого интереса или корпоративного контроля. Феномен Code Club распространяется

по всему миру. Теперь их можно найти даже в Бахрейне и Исландии. Материалы по проекту уже переведены на 14 языков, и ожидаются дальнейшие переводы. Более подробную информацию см. на www.codeclub.org.uk.





PHP: Код и вычисления

РНР некогда означало Personal Home Page, но сегодня это рекурсивный акроним PHP: Hypertext Processor. Многие из ведущих мировых web-приложений, таких как *WordPress* и *OwnCloud*, написаны на PHP. Он широко используется рядом крупнейших сайтов, и некоторые из них настолько велики (Facebook), что им пришлось писать для него собственный движок (HHVM). Вместо использования традиционной настройки PHP (запуск в качестве модуля через web-сервер), давайте установим PHP-CLI, редакцию командной строки. Это избавит нас от необходимости настраивать *Apache* или нечто подобное (хотя ныне это довольно просто, так что вы, возможно, решите именно таким образом следовать нашему руководству, и это будет замечательно).

В дистрибутивах на базе Debian и их производных надо запустить

```
$ sudo apt-get install php5-cli
```

Другие дистрибутивы называют свои пакеты иначе — в Arch это просто *php*. В конечном итоге, если вы увидите, что ваш менеджер пакетов пытается установить web-сервер и всякую лишнюю ерунду, значит, вы выбрали неправильный пакет.

Мы будем работать в текстовом редакторе по вашему выбору — вы можете использовать *nano* и делать все из терминала, или выбрать нечто графическое, как, например, *Idle* или *Geany*. Что бы вы ни выбрали, ваша первая программа будет короткой — вот такой:

```
<?php
echo 'Hello World!';
?>
```

Сохраните ее как **hello.php** и затем запустите командой

```
$ php hello.php
```

Вероятно, вы уже знаете, что произойдет. Подобно тому, как JavaScript включен в страницы HTML через тэги `<script>`, код PHP должен быть окружен тэгами в первой и последней строке фрагмента кода. Блоки кода — которые включают, но не ограничиваются присваиванием переменных, циклами и компонентами `if...else...endif` — должны заканчиваться точкой с запятой (самым коварным знаком препинания). Если предполагалось, что вы включите вышеприведенный код в документ HTML на web-сервере, работающем на PHP, то на этой web-странице он будет приветствовать мир. Просмотр исходника страницы в браузере не отобразит кода PHP, только его результат... Мрак.

Давайте попробуем более сложный пример. На сей раз мы определим функцию с именем `fib`, которая принимает целочисленную переменную `$n` в качестве аргумента и возвращает *n*-е число Фибоначчи. Переменные в PHP предваряются доллар-выми значками, функции заключаются в фигурные скобки, а условия `if` — в круглые скобки:

```
<?php
function fib($n) {
    if ($n == 0) {
```

```
        return 0;
    }
    if ($n == 1) {
        return 1;
    }
    return fib($n - 1) + fib($n - 2);
}
echo fib(12);
?>
```

Даже не имея опыта программирования и знаний о последовательности Фибоначчи, вы, вероятно, сумеете догадаться, что «нулевое» число Фибоначчи — это 0, а следующее за ним — 1. Последующие числа Фибоначчи определяются как сумма двух предыдущих чисел, следовательно, остальная часть последовательности будет такой: 1, 1, 2, 3, 5, 8. Удивительно, но написание этого кода практически не требует усилий — вполне допустимо вызвать функцию из самой себя, и она не выдает отказа. Эта техника известна как рекурсия, и ее отлично иллюстрирует пачка кукурузных хлопьев с картинкой, на которой изображена пачка кукурузных хлопьев с изображенной на ней... Мо-

Многие из ведущих мировых web-приложений, таких как WordPress и OwnCloud, написаны на PHP.

жет, не совсем так — благодаря нашим двум базовым случаям глубина рекурсии ограничена, так что она не будет бесконечной. Так или иначе, это не самый эффективный способ расчета чисел Фибоначчи. Посмотрите на рисунок, чтобы увидеть структуру вызовов для `fib(5)` — более половины вызовов являются лишними. По мере роста *n* растет также и это удваивание усилий: посмотрите, что происходит, когда вы меняете вызов функции в предпоследней для вычисления 40-го числа Фибоначчи (это около 100 миллионов, и на вычисление у нас ушло около 45 с). Сверьтесь со врезкой на следующей странице, чтобы поправить дело.

Обратите внимание, что если пользователь не предоставляет аргумент `fib2.php` (из той самой врезки), PHP начнет ругаться, поскольку массив `$argv` не имеет записи с индексом 1. Значением `$argv[0]` всегда является имя самой программы, в данном случае `fib2.php`. Если аргумент предоставлен, мы конвертируем его из строковой переменной в целое (`int`). Это конвертирует неупотребимый результат в `int 0`, что и запускает наше малополезное сообщение об ошибке. К сожалению, это значит, что наша программа не работает с нулевым числом Фибоначчи; однако его легко запомнить. Со всем, что больше нуля, наш новый алгоритм будет на удивление быстро использоваться. Ну, не то, чтобы прямо со всем — 93-е число Фибоначчи больше, чем 2^{64} , и его нельзя сохранить как 64-битную целочисленную переменную. PHP

округляет его до числа с плавающей запятой (отображаемого в виде мантииссы и порядка) с ограниченной точностью, но в конечном итоге числа становятся слишком большими (у нас — от 1477) даже для этого.

Вероятно, стоит отметить наличие унаследованных издержек на каждом уровне рекурсии и в вызове функций, которые вы добавляете в свою программу. Поэтому если вам и правда нужна истинно немецкая эффективность, лучшим решением будет многократное повторение. В данном случае не будет ощутимого прироста скорости (`fib2 1476` отображает ответ почти мгновенно), поэтому мы оставим это в качестве упражнения. Вы должны бы знать о циклах `for` в PHP, что очень хорошо, потому что мы расскажем о них в следующем разделе.

Шифр Цезаря

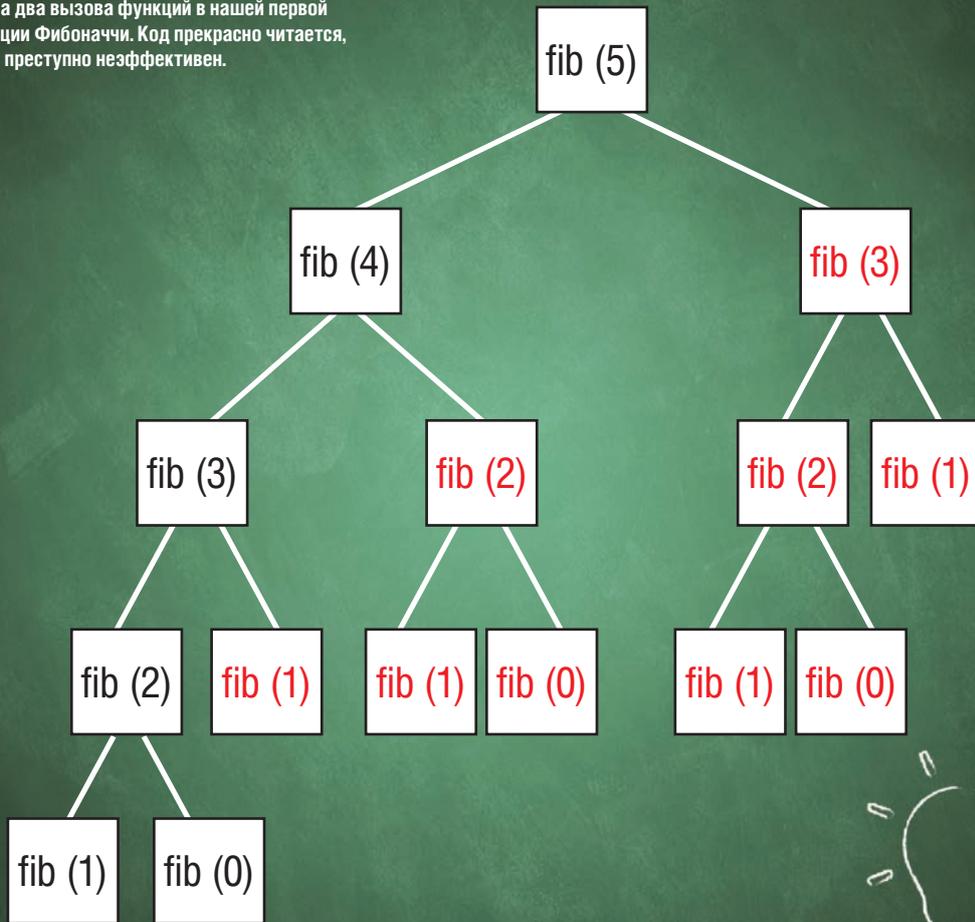
В завершение сегодняшнего урока обратимся к классической криптографии. Обязательно загляните еще раз в наш замечательный трактат по данной теме в **LXF189**, однако достаточно будет знать, что Юлий Цезарь шифровал сообщения, сдвигая

каждую букву на три позиции — меняя X, Y и Z на A, B и C соответственно. На компьютере добиться этого трудновато. Один из способов, достойных внимания — значения ASCII каждого символа простого текста: буквы верхнего регистра A–Z имеют значения 65–90, а нижний регистр a–z занимают ряд значений 97–122. Для простоты мы не будем касаться остальных символов (например, пробела и знаков препинания) в шифротексте. Итак, мы прогоним цикл по `$plaintext` символ за символом, добавим сдвиг значений в значения ASCII каждого буквенного символа, выполним все необходимые циклические переходы, получим представление для каждого символа его нового значения, и, наконец, приложим это все к нашему шифротексту.

Как только вы это сделаете, поиск значения ASCII будет делаться с помощью `$ord`, а символа — с помощью `$chr`:

```
<?php
function ccipher($plaintext, $shift) {
    $sciphertext = "";
    for ($j = 0; $j < strlen($plaintext); $j++) {
        $asciicode = ord($plaintext[$j]);
        if ($asciicode >= 65 && $asciicode <= 90) {
            $sciphertext .= chr(($asciicode + $shift + 13) % 26 + 65);
        }
        else if ($asciicode >= 97 && $asciicode <= 122) {
            $sciphertext .= chr(($asciicode + $shift + 7) % 26 + 97);
        }
    }
}
```

Красный текст демонстрирует зрящие усилия на два вызова функций в нашей первой функции Фибоначчи. Код прекрасно читается, но он преступно неэффективен.



Скорая помощь

Измените значение \$shift, чтобы создавать другие, столь же неэффективные шифры — значение 13 даст вам код ROT13, который популярен благодаря своему использованию для сокрытия ключевых фраз и спойлеров на многих форумах.



```

}
else {
    $ciphertext .= chr($asciicode);
}
}
return $ciphertext;
}
$plaintext = 'Veni vidi vinci';
$shift = 3;
echo ccipher($plaintext, $shift);

```

Циклы For иницированы в стиле C, предлагая список условий в скобках. Наша переменная-итератор \$j запускается с 0 и проходит по всей протяженности строковой переменной. Оператор and сокращен до &&, и мы используем оператор .= для соединения строковых переменных. Все достаточно просто, за исключением циклического перехода. Здесь мы используем оператор деления по модулю — %, который дает

остаток от деления, в нашем случае на 26. В первом случае нам надо, чтобы это был 0, если значение нашего сдвига равно 65, поэтому мы прибавляем 13, поскольку $65 + 13 = 78$, что делится на 26. Заодно мы добавляем 7 во втором блоке. Места для написания соответствующей функции cdecipher() не остается, но это лишь вопрос замены двух знаков плюс на знаки минус. Посмотрите, сможете ли вы восстановить таким же образом исходный текст. LXF

Более быстрая функция Фибоначчи

Программисты крайне озабочены написанием эффективного кода. И причина не в их любви к элегантности (хотя, как правило, она присутствует); как мы уже говорили, с неэффективного кода, возможно, очень хорошо начинать, но он может потом дико тормозить, как только ему будет предложено поработать с большей нагрузкой. Здесь мы можем использовать более умную функцию Фибоначчи, которая проходит два предыдущих числа в последовательности, так что не затрачивается впусую никаких усилий. Однако мы все равно хотим быть в состоянии вызывать нашу функцию с одним аргументом, поэтому мы сделаем дополнительные

```

?>
function fib2($n, $oldfib = 1, $olderfib = 0) {
    if ($n > 1) {
        return fib2($n - 1, $oldfib + $olderfib, $oldfib);
    }
    else {
        return $oldfib;
    }
}
if (!$argv[1]) {
    echo 'no';
}

```

```

}
else{
    echo fib2((int) $argv[1]);
}
?>
Помимо определения нашей новой улучшенной функции fib2(), мы также используем специальную переменную $argv, чтобы разрешить пользователю ввод данных в командной строке. Если сохранить этот файл как fib2.php, то 54-е число Фибоначчи будет найдено таким образом:
$ php fib2.php 54
86267571272

```

Техно- вестник

Мэтт Хэнсон берет интервью у влиятельного приверженца FOSS и менеджера сообщества *Nginx*.



Сара Новотны [Sarah Novotny] — пропагандист технологий и менеджер сообщества *Nginx*. В 2001 г. она основала компанию Blue

Gecko, занимавшуюся адми-

нистрированием удаленных баз данных и проданную в 2012 г. *DatAvail*. Она часто делает доклады об инфраструктуре и о стиле жизни компьютерщиков, а также отвечает за формирование программ *OSCON*. *OSCON 2015*

станет для нее последним в данном качестве, и мы встретились с ней, чтобы узнать, что привело ее в одну из крупнейших в мире конференций по открытому ПО и что ее так привлекает в *OSCON* и в среде компьютерщиков.

Linux Format: Вас называют «техническим евангелистом» — какую технологию вы проповедуете?

Сара Новотны: Я — большой сторонник технологии вообще. Хотя изначально занималась инфраструктурой и открытым ПО. Сейчас, как глава

сообщества и продвижения *Nginx*, я регулярно, письменно и устно, рассказываю о разных способах создания и работы с приложениями в современном Интернете, становящемся все сложнее и динамичнее, как и *Nginx*, и открытом ПО в сердце сетевой инфраструктуры.

LXF: Похоже, технологическая инфраструктура — ваша большая страсть. Как вы попали в эту сферу?

О ПОЛУЧЕНИИ RHEL 5

Мне обещали дать больше работы, когда я смогу подключить свой ПК к Интернету.

СН: В 1997 г. я бросила магистратуру, желая присоединиться к буму Интернет-компаний, и устроилась на работу в *Amazon*, тогда еще книжный магазин. Программировала я только на *Fortran*, так что в мои обязанности входило отвечать на вопросы: «Где моя книга?» в службе поддержки клиентов. Вскоре я начала писать инструменты на *Perl*, чтобы

помочь себе в работе, и одной из первых выступила за создание специальной команды разработчиков в этом отделе. Когда таковая была создана, я осознала, что до сих пор есть проблемы с серверами, позволяющими сотрудникам службы поддержки выполнять свою работу. Я стала убеждать в этом — то есть активно доставать — Сетевой операционный центр [NOC] и системных администраторов, которые их обслуживали. Каждое утро я звонила им и передавала от команды СП информацию о метрике и болевых точках. За что меня в конце концов наградили новеньким пейджером и статусом младшего системного администратора.

LXF: Это и было ваше знакомство с открытым ПО?

СН: Да, помимо этого позорного статуса и пейджера, мне предложили копию *Red Hat Linux 5* (кажется, на 3,5-дюймовых флоппи-дисках). И обещали дать больше работы, когда я научусь подключать свой настольный ПК к Интернету. Это было мое первое знакомство с открытым ПО и первое скомпилированное ядро. Спустя несколько дней (и множество вопросов со стороны



моих коллег), мне разрешили доступ к другим системам, помимо моей.

В Amazon использовалось много программ с открытым кодом, и я то и дело с ними сталкивалась, но удивительная познавательная сфера открытого ПО меня тогда не затянула, не было ни времени, ни места подумать о его более широком влиянии. Тогда это было просто ПО, которое я могла скачать.

Уйдя из Amazon, я, вместе с несколькими талантливыми коллегами, основала компанию, предоставляющую услуги администрирования удаленных баз данных. Изначально Blue Gecko поддерживала продукты Oracle, а я была системным инженером, давая рекомендации по архитектуре серверов. Один клиент попросил нас о поддержке MySQL, и поскольку знаний у меня на тот момент было в самый раз, чтобы испортить отношения с Oracle — я была там в лучшем случае младшим администратором БД — я решила освоить эту простенькую СУБД.

Она тогда была в версии 3.23, и, по сравнению с Oracle, имела более ограниченную функциональность и масштабируемость. В конце концов, из этой заявки на поддержку выросла полноценная открытая база данных. Я начала активно участвовать в сообществе MySQL — выступая с докладами, проводя встречи и став со-основателем подкаста.

LXF: Кроме того, вы являетесь организатором программ OSCON; как вы попали в конвенцию Open Source?

СН: На одной из последних конференций MySQL, которую проводили O'Reilly Media, я познакомилась с Джинной Блейбер [Gina Blaber], их вице-президентом по конференциям. Я поблагодарила ее за вклад в сообщество MySQL и свой собственный рост, и спросила, не могу ли помочь в развитии программы. В том же году я пересеклась с Джинной

на OSCON, и, по счастью, случилось так, что Элисон Рэндал [Alison Randall] тогда как раз решила уйти с поста со-председателя. Джина представила меня Эду Дамбиллу [Edd Dumbill], в то время другому со-председателю OSCON, и мы нашли общий язык. Как он выразился (в своем британском стиле): «Через 10 минут мы уже друг перед другом выкладывались». И за последующие 3 года я старалась почерпнуть от Эда как можно больше о главной конференции O'Reilly OSCON.

LXF: В чем состоят ваши обязанности как со-председателя?

СН: Со-председатели, включая меня, Мэтью МакКаллоу [Matthew McCullough] из GitHub и Рэйчел Румелитус из O'Reilly Media, отвечают за редакторское видение программы секций. То есть предугадать, что может быть интересно участникам, и что они должны знать за 9–12 месяцев до мероприятия. Мы намечаем, о чем следует рассказать, и следим, чтобы в программе соблюдался

О ВЛИЯНИИ OSCON

Когда сообщества пересекаются, можно узнать массу нового.

баланс между тем, что свежо и ново, и тем, что имеет непреходящую важность и может иной раз произвести революцию; между различными сообществами, разными уровнями навыков и темами, объединяя все это в цельное и сжатое мероприятие. При этом мы сильно полагаемся на комитет составителей программы, ежегодно прочитывающих более тысячи заявок на OSCON, чтобы консультировать нас по специфике любой технологии.

LXF: В чем, по-вашему, состоит своеобразие OSCON как мероприятия?

СН: Для меня, OSCON — это встреча племен и обществ, при пересечении которых можно узнать массу нового. OSCON — это событие, где находят друг друга разработчики открытого ПО и его пользователи. Обсуждения, которые я слышу и в которых принимаю участие, всегда заставляют меня активнее и иначе думать о проблемах, с которыми я сталкиваюсь в технических сообществах, включая мои собственные. Я получаю огромный интеллектуальный (и особый «технарский») заряд от этого смешения. Это самая ценная конференция, куда я каждый год приезжаю исключительно ради связей, которые завязываются с новыми сообществами.

LXF: К слову о сообществах, вы являетесь менеджером сообщества Nginx — сервера и обратного прокси, на котором работает ряд очень популярных сайтов, включая Netflix, Pinterest и Wordpress.

Не могли бы вы рассказать, в чем его особенность?

СН: У Nginx потрясающе много особенностей. Есть технические причины: легкость и невероятная эффективность архитектуры, управляемой событиями; практичность его стремления внедрять ограниченное количество вещей, зато очень и очень хорошо; перспективность, вследствие раннего внедрения SPDY. Есть и социальные факторы: Nginx стартовал как проект одиночки без какой-либо документации на английском (общепринятом техноязыке); техническое превосходство продукта способствовало его популярности в сообществе и переводу документации — в среде web-разработчиков он стал чем-то вроде тайного знака причастности. Когда мне впервые посоветовали его использовать, я не могла найти его в поиске — не знала, как это пишется.

Для евангелиста — это просто мечта способствовать развитию компании с такой потрясающей репутацией в техническом отношении и столь сильными позициями на рынке. Хотя наш сервер чаще всего используется среди 10000 лучших сайтов, по трафику мы только-только доросли до второй позиции. Существует масса возможностей улучшения производительности в Сети, что для меня означает — больше довольных клиентов. Мне хотелось бы это-му способствовать.

LXF: Параллельно с нашим разговором проходит OSCON 2015. Есть ли в этом году выступления, которые вам бы особенно хотелось бы посетить?

СН: Это мне всегда дается на OSCON труднее всего. Обычно я посещаю по два-три выступления. Но всегда с огромной завистью смотрю на расписание. В этот раз я попытаюсь послушать доклады «Как мой POODLE утратил свой Xen, увидев Ghost, погнавшись за Berserk и получив ShellShock вместе с Heartbleed» Констанцы Хит [Constanza Heath] из Intel, «99 способов загубить открытый проект» Брэндона Киперса [Brandon Keepers] из GitHub, «Говори „нет“, как босс!» Деб Николсон [Deb Nicholson] из Open Invention Network, «Ты кодишь



как сисадмин — Синдром самозванца у современного разработчика» Уэйда Минтера [Wade Minter] из AdWergx, «Выбирай скучную технологию» Дэна МакКинли [Dan McKinley] и «Подключение и использование микросервисов при масштабировании» Ника Шадрин [Nick Shadrin]... И да, это такая бессовестная реклама.

LXF: В октябре OSCON приходит в Европу, чему мы очень рады. Вы в этом участвуете?

СН: К сожалению, в OSCON Europe я не участвую. OSCON 2015 станет для меня последним в качестве со-председателя. Эта прекрасная возможность слагать историю открытого ПО была дана мне на пять лет. И как и раньше, такие перестановки несут свежие перспективы и изменения, по мере того, как новые со-председатели обретают собственные голоса. Мы, вместе с Натом, Элисон, Эддом, Мэтью, Саймоном и сообществом, заложили прочную основу для постоянного развития и укрепления позиций открытого ПО. Я надеюсь оставаться активным участником сообщества и способствовать тому росту аудитории и влияния, которого мы добились за время моего пребывания в этой должности.

LXF: И все же, без сомнения, OSCON вас будет не хватать! Забегая вперед, в этом году вы будете

выступать на LISA 15 в Вашингтоне. Какие темы вы затронете?

СН: Мой доклад на LISA 15 [Вашингтон, 8–13 ноября, <http://bit.ly/LISA2015>] будет введением в *Nginx*. Буду рассказывать о пути, который мы прошли от web-сервера, созданного в 2002 Игорем [Сысоевым; Igor Sysoev] — основателем *Nginx* для решения проблемы 10 000 соединений [C10K problem] [оптимизации сетевых сокетов для одновременной обработки большого числа клиентов], до платформы приложений. Теперь, обладая возможностями распределения нагрузки, кэширования, обратного прокси-сервера и потоковой передачи медиа, *Nginx* может использоваться для улучшения стабильности, масштабируемости и безопасности трафика HTTP или TCP. Выступление на LISA будет своего рода введением в технологию, с примерами построения конфигурации сначала статического web-сервера, затем добавления компонента обратного прокси в простое приложение, и до момента кэширования контента из этого приложения с последующим добавлением элементов API управления доступом типа шлюза.

LXF: Возвращаясь к *Nginx*, отчасти его появление было связано с растущей популярностью сайтов — с необходимостью обработки все большего числа запросов одновременно. С учетом

того, что Интернет продолжает расти, какие, по-вашему, новые проблемы с трафиком это повлечет для сетевых инфраструктур?

СН: Увеличение числа довольных потребителей ведет к увеличению трафика, и чем больше трафика идет на сайты и web-серверы, тем больше бит передается по планете через устаревающие сетевые структуры и код, рассчитанный на процессоры и ограничения памяти из прошлого века. Рисковые ситуации и затруднения будут возникать постоянно. В этом плата за рост и радости технологии. Отчасти это будут проблемы технические — как маршрутизация BGP в прошлом году и то, что преступники могут, не сходя с места, грабить все больше людей; отчасти — социальные: что значит неприкосновенность личной жизни в мире, где я и сам — продукт? Но инфраструктура будет развиваться, цифровые дороги будут чинить, и будут строить новые магистрали.

LXF: Насколько *Nginx* готов к такому будущему?

СН: Я думаю, *Nginx* вполне готов помочь нашему сообществу грамотно применять свою инфраструктуру для удобства пользователей. Помочь эксплуатировать существующее оборудование наилучшим образом, разумно и прозрачно распределять трафик для новых технологий (например, облачных) там, где это имеет смысл. Мы помогаем им плавно интегрировать и ускорять работу ряда сложных сред приложений, чтобы для конечного пользователя интерфейс информационной или развлекательной программы был простым и быстрым, а не раздражающе тормозным.

LXF: И как же *Nginx* готовится к появлению HTTP/2? Какие возникали проблемы?

СН: Ну, на данный момент HTTP/2 является спецификацией. Много чего еще надо проверить путем реализаций серверов, адаптации и, конечно, должного и недолжного использования. Я как-то читала в *New Yorker* статью о том, как в FDA [Управление по санитарному надзору за качеством пищевых продуктов и медикаментов США, — прим. пер.] одобрили новое снотворное: «после утверждения FDA проведут тест в условиях реальности — на двойную дозу, три бокала вина и использование ноутбука». Мне кажется, с техникой дело обстоит схожим образом. Спецификации пишутся и тестируются в условиях, заданных инженерами. Но HTTP/1.X научил нас, в первую очередь, тому, что технологи могут принять и применить любой инструмент так, что хоть восторгайся, хоть проклинай, в зависимости от того, как посмотреть.

Мы работаем над первой реализацией HTTP/2, и летом появятся экспериментальные релизы для любителей быть на передовой. Как водится в технологии, мы ожидаем, что в некоторых организациях новые протоколы повысят производительность, другим же придется пойти на более масштабные изменения, чтобы увидеть какую-либо пользу. Это одна из причин, почему *Nginx* не будет сразу переходить на HTTP/2 по умолчанию. Чтобы узнать больше о парадигматических изменениях, связанных с HTTP/2, стоит посмотреть презентацию Ильи Григорика [Ilya Grigorik] »



из Velocity — «Лучшие практики вчерашнего дня становятся для HTTP/2 противодействием», <https://youtube.com/URLTwZ3ehk>.

LXF: Вы часто упоминаете о сообществе — что и неудивительно, ведь вы — менеджер сообщества! Но какое значение сообщество имеет для проекта *Nginx*?

СН: Самое первоочередное. Сообщество приняло и взрастило *Nginx* с самого начала. Я люблю оценивать компании по тому, как они тратят деньги, и *Nginx* явно ценит свое сообщество. Я, как его представитель и евангелист, стала третьей, кого московская команда инженеров наняла со стороны. Первым стал генеральный директор, Гас Робертсон [Gus Robertson], затем наш финансовый директор, и потом я. По-моему, это говорит: во-первых, создай хороший продукт; во-вторых, отнесись к финансам ответственно; в третьих, поддержи сообщество.

LXF: Ну и открытое ПО, я полагаю, также важно для *Nginx*, но думаете, Heartbleed [критическая уязвимость в ряде версий OpenSSL, делающая возможной кражу данных, — прим. пер.] ударил по общественному мнению относительно открытого ПО и его безопасности? Как этому противостоять?

СН: Heartbleed стал для всех моментом прозрения. Безусловно, это напоминание о том, что открытое ПО — не панацея от всех бед. Открытость не означает, что проблемы с безопасностью решаются сами собой, архитектуры улучшаются, а продажи взлетают. Открытое ПО — это инструмент, а любой инструмент имеет свои сильные и слабые стороны. К первым можно причислить то, что самые великие умы отовсюду могут поучаствовать в решении проблемы в сетевом сообществе,

все участники которого преследуют общие цели. Я полагаю, открытость и смирение, свойственные открытому ПО при обнаружении уязвимостей, могут более рассудительным взять верх, когда страсти поутихнут.

LXF: Вы также часто говорите о том, что значит быть компьютерщиком — какие качества, по-вашему, для этого необходимы?

СН: Недавно мы с командой *Nginx* со всего мира решили опытным путем выяснить, что входит в эту культуру сейчас, дабы сохранить ее по ходу быстрого развития. Опираясь на те главные черты, что

Как бы ни расписывали профессию менеджера сообщества или евангелиста во всем блеске и славе публичных выступлений, куда важнее быть впереди в своей отрасли и находить себе в своем сообществе сторонников, столь же одержимых вашим проектом. Именно они формируют те отношения, которыми исподволь управляет лидер сообщества, усиливая, наставляя, ободряя и делясь информацией.

Наконец, найдите тех, кто работает на сообщество, и проводите с ними время. Когда мы стали не жалеть дней на совместные путешествия, выяснилось, что мы — очень сплоченный коллектив

О ПРИОРИТЕТАХ NGINX

- № 1: создать хороший продукт;
- № 2: быть финансово ответственным;
- № 3: поддерживать сообщество.

мы выявили в *Nginx*, можно сформулировать качества, определяющие стиль жизни компьютерщика: любознательность, открытость, прогрессивность и стремление к совершенству.

Любознательность и открытость поддерживают культуру обучения, а прогрессивность и стремление к совершенству определяют то, как эта культура реализуется в творчестве, усердии и высоких стандартах. Добавьте к этим ценностям *Nginx* двустороннюю обратную связь со вдумчивыми сообществами-партнерами, чтобы делиться информацией и тем, что создано, — и получите описание этого волшебного стиля жизни компьютерщиков, а также сообщества открытого ПО и цикла разработки.

LXF: Наконец, для тех из наших читателей, кто хотел бы стать лидером сообщества, с чего лучше начать?

СН: Начните с того, что вам по-настоящему нравится. Еще одно качество настоящего компьютерщика — легкая одержимость. Не в плане неадекватности, а в плане энтузиазма, вовлеченности, чтобы стоя на краю сказать: «Разве не круто?!» Когда вы получаете удовольствие от того, что узнаете и делаете, это заразительно, и вы естественно притягиваете других людей, кого привлекает то же самое. В основе сообщества должны быть увлеченность и взаимодействие. Когда проект, или инструментарий, или новый взгляд на мир вас увлекают, гораздо легче понять друг друга и увлечь тех, кто пока этого не разделяет. И затем внести свой вклад. Помочь новичку разобраться с IRC, или обнаружить и восполнить пробел в документации, или отловить найденные ошибки, сделать прополку, до которой ни у кого не доходят руки — прекрасный способ влиться в проект.

Далее, важно прислушиваться и делиться с другими. Многие в сообществе завязаны на то, чтобы находить и отмечать работу других, продираться через чужие жалобы в поиске и формулировании ответа, тем самым улучшая свое сообщество и продукт или проект для пользователей.

любящих свое дело, любящих учиться и делиться мастерством, что мы готовы обсуждать климат в сообществе и способы улучшить взаимодействие, пока не попадаем со стульев. Мы — странно-человеколюбивая группа технарей, которым одинаково приятно увидеть и новичка, вылезшего из своей раковины и принятого в сообществе, и новую технологию.

К слову, среди евангелистов, лидеров сообщества и тех, кто взаимодействует с разработчиками, которых я знаю, больше людей интровертных, чем экстравертивных. Так что, если вы интроверт, не думайте, что эта работа не для вас. Некоторые из самых активных и успешных людей в сфере управления сообществом — классические интроверты. **LXF**



Широкоформатая печать



Широкоформатая печать

- ◆ Наружная и интерьерная
- ◆ Стойкая к воде и УФ
- ◆ Плоттерная резка, печать на любых материалах

Широкоформатая экопечать

- ◆ Печать для рекламы на транспорте
- ◆ Печать для метро (стикеры, щиты, световые короба)
- ◆ Мобильные рекламные стенды, витрины
- ◆ Ширина 1,8 м, разрешение 1440 DPI

Наши преимущества

- ◆ Собственное производство
- ◆ Высокое качество
- ◆ Доставка
- ◆ Минимальные сроки и экономия ваших сил
- ◆ РИА и РА — скидки на всё!

Графический дизайн

- ◆ Полиграфической продукции
- ◆ Наружной рекламы
- ◆ Фирменный стиль
- ◆ Цифровая фотосъемка

Digital PRINT

(812) 591-79-80

(812) 591-79-81

diprint.spb.ru

Санкт-Петербург,
ул. Смольячкова, 19,
офис 405А
(м. «Выборгская»)

Вслед за WordPress 4.3

Эрик Манн описывает самые интересные новые функции очередного этапного релиза платформы с открытым кодом.

Еще в 2007 г. я начал вести блог в *WordPress*. Я профессионально работаю с платформой и вношу вклад в развитие ее кода с 2010 г. За это время я застал 17 основных релизов. И тем не менее, ничто еще не удивляло меня как писателя и разработчика так, как новый набор функций *WordPress 4.3*.

Костяк команды анонсировал начало разработки *WordPress 4.0* на конец апреля 2014 г. 23-й релиз в его истории, *WordPress 4.0* был объявлен этапным «нулевым» релизом. Проект возглавлял талантливый инженер, сосредоточенный на удобстве пользователя [UX], при поддержке сотен подкованных разработчиков со всего мира. Изначально у проекта хватало функций, чтобы восхитить любого фаната *WordPress*, и было невероятно любопытно наблюдать за тем, как обновление сузилось до набора отлично написанных, протестированных и проверенных функций.

Как и у каждого нового релиза, есть функции, которые нравятся всем пользователям, и есть функции менее интересные; однако, впервые на моей памяти, о каждой функции версии 4.0 есть что сказать. Впрочем, я сократил мой список пунктов «прямо не верится, что у нас это есть» до самых, по моему мнению, важных новинок.

Одной из главных целей платформы *WordPress* является «демократизация публикаций», и особенность, поставляемая в каждую версию — упростить писателю получение шанса быть услышанным онлайн на его собственных условиях. Поскольку большая часть мира говорит не на английском, это значит, что администратору *WordPress*

нужно выпускать контент в иноязычном формате. Обширный контингент неумолимых волонтеров переводит текст в *WordPress* на разные языки. Множество тем и плагинов также привязаны к переводу *WordPress*, давая возможность не-англоговорящим пользователям использовать всю платформу, включая опциональные расширения.

Вплоть до 4.0, однако, процесс установки и настройки был только на английском языке. Новичкам для установки приходилось либо пользоваться английским, либо искать кого-нибудь себе в помощь. Как легко представить, это было основной преградой к присоединению для людей, не владеющих английским.

WordPress 4.0 изменил это. Теперь платформа позволяет в процессе установки и настройки указать свой родной язык. Выбор языка, отличного от английского, запускает скачивание языкового пакета, автоматически

Ничто не удивляло меня так, как набор новых функций WordPress 4.3.



► Начальный экран, представленный установщиком WordPress 4.0, позволяет выбрать язык, автоматически переводя установщик.

перевода инструкции на всех экранах и самого администратора WordPress. Основная часть сообщества WordPress признает ценность нашей растущей международной аудитории. Впервые люди, не владеющие английским, могут устанавливать и настраивать WordPress, не нанимая дорогого подрядчика, чтобы он настроил программу.

Эти изменения сделали WordPress доступным для большего числа пользователей, чем за всю его историю. В мае этого года количество загрузок от человек, не владеющих английским, впервые превысило количество загрузок англоязычных, так что остается только ожидать дальнейшего всемирного распространения WordPress.

WordPress 3.5 совершенно переосмыслил медиа-галерею, значительно облегчив импорт и работу с изображениями внутри постов и страницы администратора. В версии 4.0 WordPress продолжает оптимизировать работу с мультимедиа, введя новую сетку для галереи.

В прошлом у редакторов был только один вариант просмотра записей в Media Library — в виде списка [List]. Это предоставляло постраничный список записей, каждая из которых снабжалась миниатюрой. Управление медиа на большом сайте подразумевало навигацию по довольно скучному списку изображений, охватывающему несколько страниц. Хотя бы немного попользовавшись

WordPress, вы поймете, что отображение в виде списка в релизах до 4.0 было далеко от идеала пользовательских ожиданий.

Управление медиа

Обновленная Media Library открывает новый режим просмотра Grid по умолчанию. Вам предоставляется богатая сетка увеличенных миниатюр, втягивающих контент, используя тот же принцип, что мы применяем при бесконечной промотке. Нажатие на какую-либо конкретную миниатюру загружает метаданные в модальном окне. Это также ускоряет процесс редактирования, не требуя переключаться между различными страницами в WordPress.

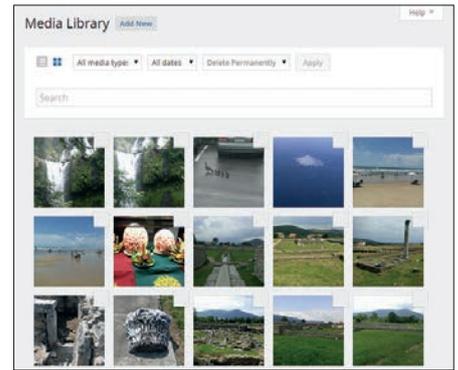
Без этой информационной мета-модели управления массой индивидуальных активов [assets, все составляющие сайта] было бы утомительным и затратным по времени процессом. Даже просмотр надписей и описаний приводил к вынужденной загрузке отдельной страницы. Уточненный опыт работы представляет все подробности вложений в легкодоступном месте, что экономит время и силы.

Каждая новая функция обращается к мощи и изобретательности сообщества WordPress.

Даже вложения, не являющиеся изображениями, вроде документов и ZIP-архивов, оказались в плюсе от новой трактовки сетки. Правда, ей недостает индивидуальных миниатюр, но представленные иконки MIME-типа в том же формате, что и увеличенное изображение миниатюр, безусловно, делает Media Library более унифицированной.

Обновления редактора

Наиболее употребительная функция WordPress — это редактор контента WYSIWYG [What You See Is What You Get — что видишь, то и получаешь]. Используйте ли вы WordPress только изредка для



► Сетка отображения Media Library включает увеличенные миниатюры, упрощая управление активами.

блоггинга, либо в качестве профессионального медиа сервиса новостей, или просто чтобы курировать статическое тело контента, вам в какой-то момент понадобится редактор.

А раз это самый заметный и востребованный элемент платформы, можно ожидать, что редактор получит достаточную дозу полировки в лю-

бом важном этапе; и WordPress 4.0 внес ряд обновлений в опыт редактирования и в сам редактор. Два волнующих изменения, достойных внимания: WordPress долго время был чемпионом по использованию oEmbed, облегчая способы делиться контентом одного сайта или платформы с другим. В релизе 4.0 поддержка oEmbed шагнула дальше, добавив новых провайдеров oEmbed для таких сайтов, как www.ted.com и www.collegehumor.com. Версию 4.0 также резко развили таким образом, что мы получаем oEmbed внутри редактора контента.

Раньше редакторы могли лишь копировать и вставлять URL ресурса oEmbed на чтение (таких как видео YouTube или сообщение Twitter) прямо ►

Взгляд изнутри: Хелен Хоу-Санди



После второго бета-релиза мне выпал шанс посидеть с Хелен Хоу-Санди [Helen Hou-Sandi], лидером релиза WordPress 4.0, чтобы узнать немного больше подробностей о том, каким она видела развитие проекта за прошедшие несколько месяцев.

Предыдущие версии платформы часто следовали заданному направлению общего развития проекта. Однако версия 4.0 была менее централизованна: «Мы предпочли улучшить то, что у нас уже было», объяснила Хоу-Санди. Как результат, большинство функций, планировавшихся к выпуску, основывались на предыдущих

успехах или фокусировались на обновлении элементов.

Примером последнего является интерфейс плагина WordPress. По словам Хоу-Санди, плагины являются «сердцевинной» того, что может сделать WordPress, но интерфейс оставался относительно нетронутым с 2009 г. Она объяснила это тем, что одной из главнейших задач релиза 4.0 было облегчить работу новым пользователям, и потому реконструкция способа, которым плагины одновременно бы обнаруживались и управлялись в WordPress, находилась в начале списка команды разработки.

По сути, Хоу-Санди говорит, что 4.0 должен был «по мелочам неуклонно продвигаться по всем фронтам», а не концентрироваться на какой-то одной области. Она выразила надежду, что конечный продукт сможет удовлетворить несколько групп — сделав WordPress проще для начинающих пользователей и в то же время расширив возможности новых API для разработчиков.

WordPress — очень живой проект, который постоянно прогрессирует с каждым новым релизом. Цель 4.0 такая же, как и у каждого релиза: сделать WordPress лучше.

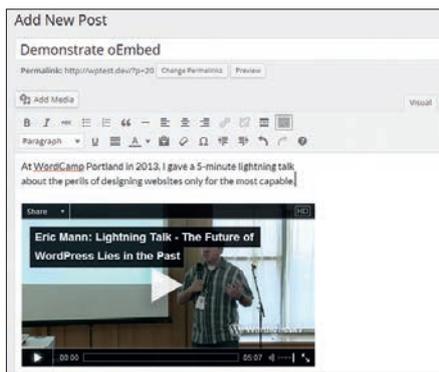
в редактор поста. При просмотре из внешнего источника эти «сырые» URL динамически превращались в более приглядные представления связанного материала. Хотя эта функция добавляла немало ценности для редакторов, ссылающихся на внешние источники, но также делала процесс оптимизации тела вокруг встроенного ресурса немного сложнее.

WordPress 4.0 изменил эту функцию, включив предпросмотр встроенного контента в режиме live. Когда ссылка на контент вставляется в редактор, WordPress автоматически сканирует ссылку, выбирая похожие ресурсы через oEmbed, создает представляющий предпросмотр [preview] и, наконец, отображает предпросмотр в редакторе.

Это подлинная реализация WYSIWYG для редактора WordPress, устраняющие мучения, вызываемые применением oEmbed для расширенного контента поста. Исходный URL для встроенного актива также будет всегда доступен в просмотре Text; но однажды увидев подключенное превью в действии, вы уже не ощутите необходимости покинуть редактор Visual и переключиться на другой вид для предпросмотра.

«Липкая» панель инструментов

Моя любимая новая функция — не столько добавление, сколько уточнение существующей модели поведения. В WordPress 4.0 панели инструментов для визуального редактора — и сверху и внизу редактора контента — прикреплены к окну браузера,



Когда oEmbed-восприимчивые ссылки появляются в контенте поста, WordPress создает предпросмотр соответствующего им ресурса.

когда вы прокручиваете страницу. По мере того, как вы пишете и ваша статья удлиняется, область контента автоматически расширяется с каждым словом. Панель инструментов вверху — с богатыми возможностями форматирования текста и встроенными функциями медиа — липнет к верху страницы, пока пишется ваш контент. Больше никаких

Верхняя панель инструментов липнет вверху страницы, даже по мере написания контента.

прокручиваний в окне, чтобы изменить параграф, форматирование или вставить ссылки.

При обратной прокрутке к началу статьи ниже меню редактора контента липнет к низу окна браузера. Редактируете ли вы контент в начале поста, его конце или посередине, у вас сохраняется доступ и к вашим инструментам форматирования, и к метаданным о посте, такой как количество слов.

Это довольно мелкое изменение в работе по редактированию дает значительное преимущество, особенно для писателей, создающих объемный контент. Попробовав новый «липкий» редактор, вы зададитесь вопросом, почему этой функции пришлось так долго ждать?

Я начал свою карьеру в WordPress в качестве разработчика плагинов, так что работа конечного пользователя с плагинами всегда был моим ключевым интересом.

Улучшение работы с плагинами

Плагины — основа создания функции WordPress как настраиваемой платформы, и в качестве подрядчика я провел множество часов, объясняя клиентам, как с плагинами работать. К сожалению, раньше экран плагинов в WordPress не был самым удобным. Хуже того, последние пять лет в нем не менялось ничего, кроме раскладки.

WordPress 4.0 ввел ряд изменений в способы работы новых пользователей с плагинами, как в области изучения, так и управления. Установщик плагина пытался помочь обнаруживать функции посредством базового поля поиска и достаточно

бессмысленного облака тэгов. Если вы заранее не знали, что искать, шансы найти что-либо новое для вашего сайта были достаточно ограничены. Новый установщик выделяет «избранные [featured]», «популярные [popular]» и «новые [new]» плагины, причем избранные разбиты на категории, такие как «функциональные» и «социальные».

На страницах установки плагина теперь, как и прежде, отображаются в виде детализированных плиток, с названием, кратким описанием и автором плагина. На плитках также отмечено число скачиваний и средний рейтинг плагина. Умная ссылка More Details [Подробнее] запускает перекрывающее отображение полной информации о плагине, взятой с WordPress.org — включая графический баннер.

Этот экран Details также показывает, когда доступны обновления плагина, выдавая журнал изменений проекта, оценки и информации о совместимости в доступ администраторам.

Улучшения настройщика

The Theme Customizer [Настройщик тем] также подвергся значительным изменениям с версией 4.0 и был переименован просто в Customizer. В соот-

ветствии с информацией WordPress 4.0 (<http://bit.ly/CustomizerImprovements>): «Настройка может относиться к чему угодно. В этом все дело... Customizer можно использовать для чего угодно, и мы бы хотели поощрить экспериментирование с различными способами его применения».

Панель виджетов

Поддержка виджетов добавилась в Customizer в предыдущей версии платформы WordPress, что позволило владельцам сайтов видеть в режиме live предпросмотр изменений боковой панели виджетов по мере их появления. Расширение области применения Customizer означает, что в грядущих релизах мы добавим в инструменты больше, так что виджеты удостоились собственной панели в интерфейсе пользователя Customizer.

Панели — это способ сгруппировать секции Customizer, позволяющий как по смыслу, так и визуально различать элементы, управляемые раздельно. Администраторы сайта могут на одной панели заняться их оформлением и настроить только цвета, отображение баннера и раскладку контента. Затем они могут копнуть глубже и через другую панель управлять контентом и организацией виджетов. Виджеты — всего лишь первая серия обещанных изменений системы Customizer.

API разработчика

Добавление панели API, хотя и выглядит лишь мелким улучшением самого инструмента Customizer, открывает разработчикам двери к большей гибкости будущих тем оформления. Каждый владелец сайта стремится к уникальному дизайну,

Новые функции...

WordPress 4.0 “Benny” вышел 4 сентября 2014 г., и даже с тех пор неуклонно выпускаются точечные обновления, последнее из которых — 4.3 — вышло в августе 2015. Вот список крупнейших новых функций, добавленных в течение последнего года.

WordPress 4.1 “Dinah” — Декабрь 2014

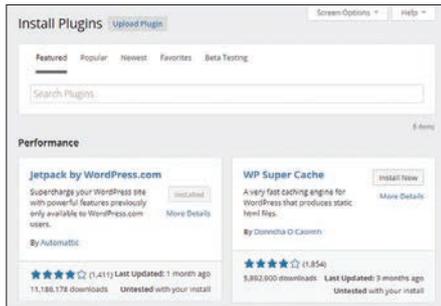
- Новое оформление ядра — 2015
- Целеустремленный режим редактора
- Панель редактирования встроенных изображений
- Выход отовсюду
- Добавленная поддержка Vine

WordPress 4.2 “Powell” — Апрель 2015

- Press This — новая система клавиатурных комбинаций при публикации
- Расширенная поддержка символов
- Мгновенная смена оформления
- Встроенные Tumblr и Kickstarter
- Оптимизированные обновления плагинов

WordPress 4.3 “TBC” — август 2015

- Новый редактор меню навигации
- Создайте свой Favicon [Favicon — значок web-сайта или web-страницы, — прим. пер.]
- Создание сильного пароля
- Новая система клавиатурных комбинаций при форматировании текста
- Полный редактор Press This



➤ Новая страница установки плагинов оптимизировала инструменты изучения, улучшив классификацию доступных плагинов.

но не у каждого есть время на создание собственного оформления — или средства, чтобы нанять разработчика.

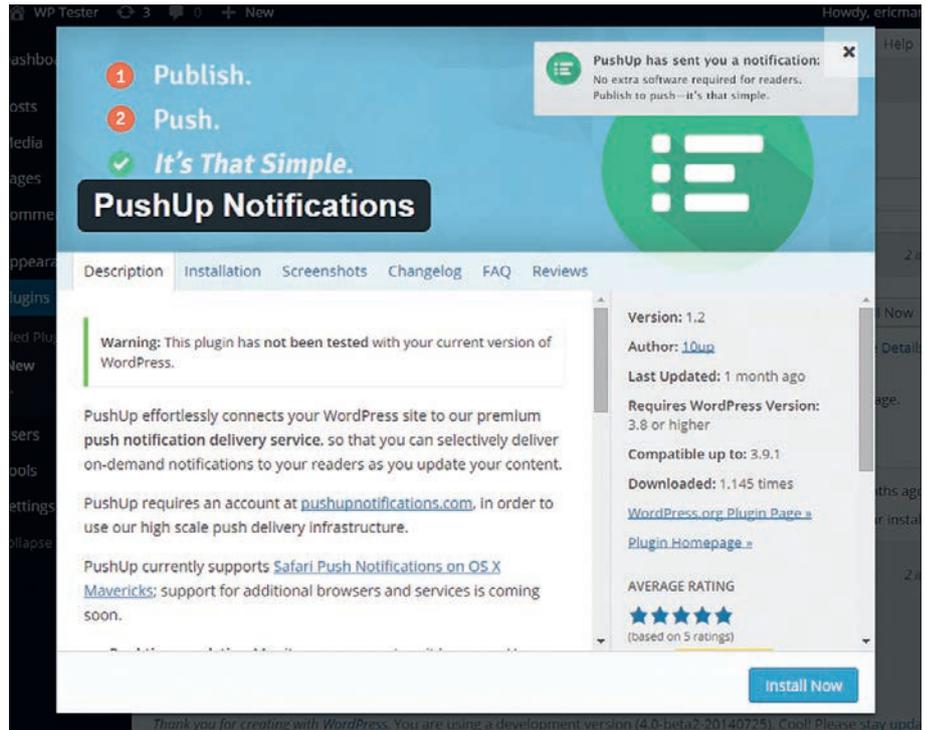
Гибкие темы оформления, поддерживаемые Customizer — простейший для неискушенных пользователей способ добраться до их сайта WordPress и сделать его действительно своим собственным. Редакторы — и писатели — могут изменять через Customizer дизайн домашней страницы, страницы About, содержимого архивов и даже больше. Customizer API добавляет внешнего лоска существующим опциям оформления API, вдохновляя разработчиков на то, чтобы позволить клиентам поднять капот их сайта.

Возможность помещать определенные функции под одну или более панелей упрощает процесс разделения одного типа настройки от других: например, функции общей темы оформления могут находиться на одном экране, а контекстные функции, представляющие данные только для вошедших пользователей — на другом.

Изменения, введенные WordPress 4.0, означают, что возможности Customizer ограничены лишь вашим воображением. К счастью, дизайнеры оформлений, как правило, склонны к творчеству.

Подведем итоги

Большинство сообществ разработчиков много внимания уделяют «нулевым» релизам. WordPress 2.0 получил полностью переделанный движок, который ввел Ajax для более хваткого административного интерфейса и редактор постов WYSIWYG, которым мы пользуемся по сей день. Далее, WordPress 3.0 претерпел объединение кода WordPress и WordPress MU (т.е. multi-user, многопользовательский),



➤ Детали модальности для плагинов по сравнению с предыдущими версиями обогатились информацией.

ради улучшения способностей платформы к хостингу нескольких сайтов в одной установке.

Некоторые пользователи могут сказать, что WordPress 4.0 не является столь же значительным для сообщества изменением, каким были его предшественники с нулевым номером. Честно говоря, основа сообщества WordPress никогда и не ставила своей первоочередной задачей превращать «нулевые» релизы в некое потрясающее событие. Взамен данный этап WordPress фокусируется на небольших итерациях большого набора функций, где новая версия будет выпускаться каждые четыре месяца для поддержания дальнейшего импульса развития.

Этот релиз совпал по номеру с 4 — хотя данный номер версии не предъясвляет никаких особых требований к набору функций. Однако рассмотренные в данной статье функции и бесчисленные небольшие исправления и оптимизации за кулисами WordPress 4.0 оказались ничуть не меньшим этапом, чем любой из «нулевых релизов», вышедших ранее.

Каждая функция в этом релизе обращается к силе и изобретательности сообщества, поддерживающего WordPress. Эти функции также показывают, какой импульс будет поддерживать WordPress в последующих версиях. LXF



➤ Когда плагины готовы к обновлению, ссылка More Details показывает журнал изменений, оценки и детали совместимости.

Как стать участником

У WordPress изнуряющий цикл релизов, нацеленный на новую версию каждые четыре месяца. Создание новых ярких функций в основе WordPress за такой сжатый срок отнимает много времени и требует большого участия сообщества.

Хотите стать участником команды разработчиков кода для WordPress? Лучший способ сделать это и вывести его на требуемый уровень — следить за основными блогами разработки: WordPress Core (<http://make.wordpress.org/core/>), WordPress UI

Group (<http://make.wordpress.org/ui/>) и WordPress Accessibility Group (<http://make.wordpress.org/accessibility/>).

Важно помнить, что даже люди, не имеющие отношения к разработке, могут сделать вклад в непрерывную разработку. Функции надо тестировать, об ошибках — докладывать, а системы — документировать. Вы можете помочь создать следующую версию WordPress, даже если в жизни не написали ни строчки кода. Не хотите ли

вы помочь WordPress двигаться вперед? Начните со вклада в следующие места:

- WordPress Support Forum (<http://wordpress.org/support/>).
- WordPress Code Reference (<http://developer.wordpress.org/reference/>).
- Станьте участником с помощью вашей локальной группы Meetup (<http://wordpress.meetup.com/>).
- Посетите WordCamp рядом с вами (<http://central.wordcamp.org/>).

Что такое TIZEN



Николай Абрамов обратил внимание на интересную операционную систему — родственницу Linux.

Tizen — операционная система с открытым исходным кодом. В названии “Tizen” сочетаются «связность» (от Tie — соединение), «активность» (от Rise — подъем) и «медитативные качества» (от Zen — Дзен). Tizen (произносится «тайзен») базируется на ядре Linux и предназначена для широкого круга устройств, включая смартфоны, интернет-планшеты, компьютеры, автомобильные информационно-развлекательные системы, «умные» телевизоры и цифровые камеры.

Предыстория создания

Прародителем Tizen является Moblin — специализированный дистрибутив Linux, предназначенный для запуска на ультра-портативных устройствах, оснащенных процессором Intel Atom. Проект Moblin был инициирован Intel в 2007 г. и позднее передан в ведение The Linux Foundation, преобразовавшись в проект MeeGo, в котором инженеры Intel принимали активное участие.

MeeGo — возможно, последняя попытка сделать традиционную мобильную ОС. MeeGo обязана своим существованием инженерам Intel и Nokia, которые работали над созданием независимых мобильных ОС (Moblin и Maemo), основанных на технологиях Linux, а в начале 2010 г. решили совместно создать единую ОС. Результатом стала выпущенная

всего через несколько месяцев операционная система MeeGo 1.0 Arlington, поддерживающая около двух десятков уже присутствующих на рынке нетбуков и частично коммуникатор Nokia N900, набравший к тому времени достаточно большую популярность.

MeeGo — это не полностью самостоятельная ОС, а скорее дистрибутив Linux, адаптированный для работы на разного рода портативных устройствах и встраиваемой технике, начиная от нетбуков и мультимедиа-систем автомобилей и заканчивая смартфонами и умными телевизорами. Практически все компоненты этой операционной системы были позаимствованы из настольных ОС, основанных на ядре Linux

Разработчики изначально позиционировали операционную систему как некий набор базовых компонентов, поверх которых можно создавать совершенно разные интерфейсы для различных устройств и рынков, сохраняя при этом их полную совместимость между собой. Операционная система имеет стандартизированный API, основанный на фреймворке Qt, который гарантирует совместимость вариантов ОС между собой, но дает полную свободу изменения графического интерфейса. Предлагалось использовать единый дистрибутив и несколько пресетов пользовательского интерфейса — User Experience (UX): Netbook.



Конкуренты ОС Tizen.

Handset, Smartphone, Tablet, In-Vehicle infotainment и SmartTV.

Однако после заключения соглашения с Microsoft Nokia потеряла интерес к работе над этой операционной системой, и разработка MeeGo фактически завершилась. Судьба MeeGo очень показательна: операционная система с хорошей технической базой и поддержкой крупных компаний оказалась бессильна перед натиском iOS и Android из-за отсутствия понятной экосистемы для разработчиков и пользователей.

После отказа Nokia от работы над MeeGo Intel объединилась с компанией Samsung, организацией Linux Foundation и проектом LiMo для создания совершенно новой ОС Tizen. Последняя унаследовала большинство наработок MeeGo, но по сути стала совершенно новой ОС, а место Nokia в ее разработке заняла преимущественно Samsung, вместе с несколькими другими компаниями, включая NEC и Panasonic. ОС Tizen спроектирована на базовых компонентах MeeGo и предлагает новый подход к разработке приложений, основанный на технологиях HTML5 и JavaScript.

Идея Tizen заключается в том, чтобы использовать стек технологий Linux в паре с графическим интерфейсом, полностью основанным на web-технологиях, таких как HTML, JavaScript и CSS. Главным аргументом в пользу такой архитектуры здесь служит простота переноса и создания приложений

Помимо основного web-стека для создания приложений, в Tizen предусмотрен также Native Development Kit, позволяющий писать части приложений на низкоуровневых языках типа C и C++,

Достоинства и недостатки

Недостатками ОС Tizen являются:

- » Отсутствие развитой экосистемы приложений.
- » Недостаточная проработка некоторых модулей, например, системы многозадачности и синхронизации.

ОС Tizen обладает следующими преимуществами:

- » Меньшие требования к ресурсам платформ.
- » Максимальная открытость исходного кода.
- » Возможность разработки интерфейса приложений на HTML5, что должно обеспечивать
- быстроту портирования под эту ОС любых web-приложений.
- » Способность работать практически с любыми устройствами: ПК, ноутбуками, телевизорами, ресиверами, мультимедиа системами и даже с бытовой техникой.



► Индия получила Samsung Z1.

что необходимо для создания игр и других высокопроизводительных приложений. При этом для 99% обычных приложений производительности JavaScript будет достаточно, так как наиболее трудоемкие операции (проигрывание видео, музыки, шифрование и так далее) будут выполняться библиотеками, входящими в базовый комплект ОС и написанными на тех же C/C++.

Первый публичный релиз Tizen состоялся в мае 2012 г., когда разработчики выложили в Сеть исходные тексты первой версии операционной системы, SDK на базе Eclipse, эмулятор, основанный на QEMU, а также специальный инструмент для быстрого тестирования приложений, работающий прямо в браузере и эмулирующий API Tizen.

В сентябре 2014 г. стала доступна версия 2.0 операционной системы, которая, несмотря на значительный прыжок в номере версии, не принесла кардинальных изменений, за исключением расширенного API, более полной поддержки стандартов HTML5/W3C.

На текущий момент Tizen развивается в трех направлениях: Mobile, Wearable иIVI.

Выпуск устройств с ОС Tizen

В 2014 г. Samsung анонсировал телефон Samsung Z, продажи которого должны были начаться в России. Однако вскоре фирма отказалась от выпуска телефона, объяснив это стремлением занять нишу в бюджетном сегменте телефонов. В начале 2015 г. в Индии поступил в продажу Samsung Z1 на операционной системе Tizen.

В 2015 г. на платформе Tizen вышли первые модели телевизоров. Интерфейс телевизоров не претерпел серьезных изменений, но количество доступных приложений для этой платформы, по сравнению с моделями 2014 г., существенно сократилось.

Также выпущено уже несколько наручных часов с ОС Tizen: Gear, Gear 2 и Gear S2. К первым моделям



► Линейка часов SmartWatch.



► Так выглядит Tizen IVI 3.0.

часов Gear было немало претензий: нестабильно работал Bluetooth, плохо синхронизировались контакты, скорость работы оставляла желать лучшего. С выпуском модели Gear 2 ситуация исправилась: работа часов стала стабильной и были добавлены новые функции. К недостаткам первых двух моделей можно отнести поддержку синхронизации только с телефонами Samsung. Модель 2015 г. Gear S2 располагает удобной системой навигации и поддерживает большинство смартфонов.

Для желающих поучаствовать непосредственно в развитии самой операционной системы есть возможность добавления своего кода в репозиторий <http://review.tizen.org/git>, в котором находятся исходные коды всех компонентов. Чтобы воспользоваться этой возможностью, необходимо зарегистрировать учетную запись разработчика на странице <http://tizen.org/user/register>.

Также доступны для скачивания образы операционной системы для таких устройств, как Rasp-

Для желающих поучаствовать в развитии ОС есть возможность добавления своего кода в репозиторий.

В конце мая этого года на конференции разработчиков Tizen стало известно, что версия Tizen 3.0 выйдет с поддержкой персональных компьютеров и ноутбуков. Еще один вариант Tizen, который был представлен на конференции — это Tizen IVI (in-vehicle infotainment) для информационно-развлекательных систем автомобилей. Совместимое с ней оборудование демонстрировалось на машинах Land Rover.

berry Pi (<http://blogs.s-osg.org/bringing-tizen-to-a-raspberry-pi-2-near-you/>) и olimex (<https://github.com/leon-anavi/tizen-sunxi>).

Перспектива развития

Перспективы продвижения Tizen различаются для различных типов устройств. Основных факторов два: удобство интерфейса/отсутствие ошибок в самой операционной системе и ассортимент приложений для устройств.

Первое справедливо для всего диапазона устройств с данной ОС (ТВ, часы, камеры, телефоны) и со временем несомненно будет развиваться и дорабатываться. А вот быстрое наращивание ассортимента приложений в магазине для смартфонов является непростой задачей: для этого требуются крупные финансовые вложения и достаточно много времени. Возможно, частично данную проблему удастся решить реализацией встроенной виртуальной машины для запуска Android-программ, однако вопрос с их быстрым действием останется открытым. LXF

Ресурсы для разработчиков

Большая часть информации для разработчиков собрана на сайте <http://developer.tizen.org>. Для разработки приложений под Tizen необходимо установить SDK — комплексный набор инструментов для разработки стандартных приложений и приложений Tizen Web. Он состоит из IDE, эмулятора, полного набора программных средств, примеров программ и документации. Эти инструменты доступны для трех платформ: Ubuntu, Windows и OSX. На текущий момент выложен SDK для смартфонов и носимых устройств [wearable].



По советам м-ра Брауна

Джоллион Браун

В свободное от консультаций по Linux/DevOps время Джоллион обуздывает стартап. Его самая большая амбиция — найти причину пользоваться Emacs.

Проблемы сисадминов стран первого мира

Не знаю, как вы, а я считаю, что сейчас — лучшее время для работы системным администратором. Возникают новые сложные задачи — взять хотя бы скорость технического прогресса. За свою карьеру я прошел путь от поиска каждого отдельного компонента сервера и его подключения до ситуации, когда я фактически не знаю (и знать не хочу), где стоят мои системы. При этом на рынке все еще есть силы, которые стремятся снизить ценность навыков сисадмина. Стереотип много работающего и низкооплачиваемого сисадмина все еще существует. (Мой совет: бросайте такую работу и ищите новую!) Но в целом мне гораздо больше нравится работать сегодня, а не в «старые добрые времена». Популярность получила удаленная работа. Рост DevOps как культуры труда и описания работы (некоторые скажут, что это неправильно), как мне кажется, меняют представление о навыках сисадмина как о чем-то малоценном (и дешевом). Но самое большое изменение из всех — взрывной рост Интернета и количество бесплатных программ и информации.

И здесь лежит моя самая большая проблема: всего этого слишком много. Слишком много проектов, которые нужно проверять, и вещей, за которыми нужно следить. Жаловаться на это нелепо, но «вокруг явный перебор классных вещей, с которыми я хочу познакомиться». Возьмем хотя бы количество способов управления виртуальными машинами и контейнерами, которые сейчас есть. Их почти так же много, как фреймворков JavaScript! Существует соблазн просто прыгать от одной новинки к другой, но этак у меня будет много обрывочных знаний о большом количестве тем. Это хорошо, пока подкрепляется более глубоким пониманием нескольких тем поуже. Найдите нишу, которая вам интересна, и ныряйте в нее. Хороший универсал с рядом специальных навыков — всегда очень ценный член любой команды. jolyon.brown@gmail.com

Эзотерическое системное администрирование из таинственных закоулков серверной.



Kubernetes достиг 1.0

Средство разработки/управления контейнерами с открытым исходным кодом готово к полноценному применению, говорит Google, передавая проект новому фонду.

На июльской конференции OSCON (конференция по ПО с открытым исходным кодом O'Reilly) Google анонсировала, что *Kubernetes*, популярный открытый проект компании для управления контейнерами и развертывания контейнеров, который появился в 2014 г., достиг важного этапа — версии 1.0. Затем Google дополнила эту новость тем, что передает проект еще одной новой организации под покровительством Linux Foundation.

В Cloud Native Computing Foundation входят (среди прочих) Docker, Red Hat, IBM, Intel и Cisco. Если при чтении этой статьи у вас возникло дежавю, не волнуйтесь — это поразительно напоминает новости прошлого месяца о проекте Open Container, в котором было много тех же имен (кроме Microsoft

и Amazon, в данном случае). У этой новой организации есть миссия — создание и управление новой вычислительной парадигмой, построенной вокруг контейнеров, микросервисов и динамически планируемых приложений и операций. *Kubernetes* — первый проект, принятый под крылышко CNCF, хотя на сайте (<https://cncf.io>) говорится и о других кандидатах, которые затем будут «объединены» в некий единый продукт.

В июле Google также стал корпоративным спонсором OpenStack, чтобы целиком вобрать *Kubernetes* в облачную ОС с открытым исходным кодом (так себя описывает OpenStack). Хотя инженеры Google уже предоставили код, поддержка поискового гиганта также имеет символическое значение для OpenStack (и вполне может быть звонком от AWS со стороны Google). Интеграция *Kubernetes* существенно упростит использование контейнеров на ее платформе и станет еще одним барьером к ее принятию.

Google сообщает, что для достижения программной версии 1.0 около 400 разработчиков выполнили более 14000 добавлений кода в репозиторий GitHub программы. Это позволило получить систему, которая доказала свое успешное масштабирование на сотнях узлов с тысячами контейнеров, а также обладает возможностью управлять журналированием контейнеров, балансировкой нагрузки и поддерживает большое количество вариантов хранилищ. Высока вероятность того, что мы расскажем о *Kubernetes* в будущих номерах — следите за этой рубрикой!



» Вам нужен *Kubernetes*, потому что программ для управления контейнерами не может быть слишком много...

Часть 3: Amazon и Ansible

В последней части своего пути к отказу от использования браузера **Джолион Браун** показывает, как перевести большую часть AWS под контроль *Ansible*.

В двух предыдущих номерах **LXF** я рассмотрел облако Amazon Web Services и поместил элементы примера инфраструктуры облака под контроль *Ansible*. В этой последней части я расскажу о последней части сервисов и о том, как *Ansible* может взаимодействовать с ними. В конце прошлого месяца я начал экспериментировать с RDS — сервисом реляционных баз данных. Я хотел создать в нем новую базу данных, заменив базу данных WordPress по умолчанию более облачной версией. Но сначала мне нужно получить представление о структуре базы данных.

Сейчас я могу создать резервную копию существующей базы данных с помощью стандартной утилиты вроде *mysqldump*. Подключившись к своему экземпляру EC2 по SSH (к тому, на котором запущена установка Bitnami WordPress), я могу подключиться к MySQL от имени root обычным способом:

```
$ mysql -u root -p
```

Пароль root для MySQL создается случайно при создании образа (на сей раз я использовал образ Bitnami). Он записывается в файл `/var/log/boot.log` (получите его командой `$ sudo grep -i password /var/log/boot.log`), и я только хочу убедиться, что он правильный, выполнив подключение. Сделав это, я могу вывести список баз данных командой `show databases;` — меня интересует база данных `'bitnami_wordpress'`. Затем я могу выйти и сделать резервную копию этой базы данных для использования в RDS:

```
$ mysqldump -u root -p bitnami_wordpress > /tmp/wordpress.sql
```

Теперь я могу скопировать ее в свой локальный репозиторий *Ansible* командой `scp`. На своем локальном компьютере я хочу поместить ее в каталог `files` для роли `rds`. Из корневого каталога репозитория *Ansible* я запускаю следующую команду:

```
$ scp -i keys/lxfkeys bitnami@<IP-адрес экземпляра EC2>:/tmp/wordpress.sql roles/rds/files/
```

(Помните, что если вы не уверены, какой IP-адрес подставить, его можно получить командой `/etc/ansible/ec2.py --list`.)

Файл совсем небольшой — в моем случае около 40 КБ, поэтому он загрузится за пару секунд.

Теперь с помощью этого файла я могу создать новую копию базы данных в RDS. Мне нужно изменить файл `main.yml` в каталоге `roles/rds/tasks`, который сейчас выглядит так:

```
---
- name: Create RDS instance
  local_action:
```



```
module: rds
command: create
instance_name: lxf-database
db_engine: MySQL
db_name: bitnami_wordpress
size: 10
instance_type: db.m1.small
username: mysql_admin
password: 1nsecure
region: «{{ region }}»
wait: yes
register: rds
- name: debug rds
  debug: var=rds
```

Так в RDS будут созданы экземпляр и база данных `bitnami_wordpress` внутри него. Однако перед этим мне также нужно изменить настройки групп безопасности, которые мы задали в прошлом месяце, чтобы разрешить подключения MySQL — в файле `roles/security_group/tasks/main.yml`:

```
---
- name: Create security groups
  local_action:
    module: ec2_group
    name: «lxf-sec»
    description: "Группа безопасности для lxf"
```

» Как я ни старался, мне не удалось настроить RDS для виртуального частного облака, для которого я хотел... Пришлите мне ответ открытой, пожалуйста!

»

Еще одна, более облачная база данных...

В прошлом я рассказывал о MySQL-сервисе Amazon RDS, но в этом месяце одна из альтернатив прошла стадию бета-версии — это *Aurora*. Анонсированный в 2014 г., это еще один вариант реляционной базы данных для пользователей AWS, пусть и с небольшими отличиями. Она позволяет достичь пятикратной производительности MySQL на том же «железе» и совместима с версией 5.6 популярной СУБД. Впрочем, судя по слухам, это внутренняя реляционная СУБД (подробности о каком-то общем коде с MySQL проследить сложно). Amazon зазывает клиентов, желающих попробовать, миграцией на новую СУБД

без простоя. Бета-тестирование, как утверждает Amazon, показало, что каждый экземпляр *Aurora* может обеспечить до 100 000 операций записи и 500 000 операций чтения в секунду, похоже, благодаря тесной интеграции СУБД с виртуализованным слоем хранения данных с SSD, созданным специально для рабочих нагрузок баз данных.

Наряду с обычными облачными преимуществами (возникающие проблемы обрабатываются в фоновом режиме, причем работа базы данных не прерывается; развертывание, применение заплат и резервное копирование выполняются автоматически),

судя по информации на странице продукта, эта СУБД может автоматически определять сбои базы данных и перезапускать ее без необходимости восстановления от сбоев или повторной сборки кэша базы данных. При полном сбое экземпляра *Aurora* автоматически выполнит восстановление с помощью реплики чтения (общее количество реплик чтения — до 15). Все это звучит очень впечатляюще для тех, кто в теме. Автоматическое масштабирование и невероятно быстрая репликация (10–20 мс) всяко привлекли мое внимание. Эх, было бы у меня нечто достаточно важное для хранения в *Aurora*...

```
region: "{{region}}"
vpc_id: "{{ vpc_id }}"
rules:
- proto: tcp
  from_port: 22
  to_port: 22
  cidr_ip: 0.0.0.0/0
- proto: tcp
  from_port: 80
  to_port: 80
  cidr_ip: 0.0.0.0/0
- proto: tcp
  from_port: 3306
  to_port: 3306
  cidr_ip: 0.0.0.0/0
```

Конечно, разрешать подключения к порту MySQL отовсюду, как сделали мы (указав 0.0.0.0/0) — это УЖАСНАЯ идея; не делайте так в своих системах. Но в примерах я могу позволить себе быть чуть беспечнее.

Теперь о проблеме с Ansible, которую мне, к сожалению, не удалось решить. Это создание экземпляра базы данных в соответствующей группе безопасности. С тяжелым сердцем мне пришлось вернуться обратно в консоль AWS, затем в окно RDS и изменить экземпляр, выбрав группу безопасности lxf-sec вместо группы по умолчанию (см. скриншот на стр. 53). Уверен, что причина в моей собственной инертности, а не в какой-либо проблеме со стороны Ansible. (Если у вас это получится, пожалуйста, сообщите мне!)

Теперь я хочу восстановить файл mysqldump в созданный экземпляр. Проще всего это сделать, как я обнаружил, создав новый файл, который я назвал create_db.yml:

```
---
- hosts: localhost
  sudo: no
  connection: local
  tasks:
  - name: create the bitnami database
    mysql_db:
      login_host: <name of amazon rds instance>
```

```
login_password: «!nsecure»
login_user: «mysql_admin»
login_port: «3306»
name: «bitnami_wordpress»
state: import
target: roles/rds/files/wordpress.sql
```

Этот файл использует модуль mysql_db Ansible, чтобы восстановить файл wordpress.sql, который я загрузил какое-то время назад. Чтобы воспользоваться этим модулем, мне понадобилось установить пакет python-mysqldb в Ubuntu командой apt-get install. Мне также пришлось выяснить имя экземпляра (нечто вроде 'lxf-database.c8qoqjukkdv.eu-west-1.rds.amazonaws.com'), снова запуская команду \$/etc/ansible/ec2.py --list, и добавить его в файл. Теперь я могу выполнить эту однократную задачу следующей командой:

```
$ ansible-playbook create_db.yml --private-key=keys/lxfkeys -i inventory
```

Она (надеюсь) сработает и создаст для меня базу данных. По ее завершении я смогу подключиться к экземпляру MySQL напрямую: \$ mysql -u mysql_admin -p --host=lxf-database.c8qoqjukkdv.eu-west-1.rds.amazonaws.com

Когда RDS снова покажет командную строку >mysql, я смогу выполнить команду show databases; и быть вознагражден тем, что база данных bitnami_wordpress теперь запущена и работает в новом, более облачном, чем когда-либо прежде, расположении. Для таких задач удобно создавать готовые файлы YAML, хотя я здесь предаю идемпотентный идеал. Иногда прагматизм берет верх...

Но как заставить наш WordPress использовать эту базу данных? Просто измените строку define('DB_HOST', 'localhost:3306') в файле /opt/bitnami/apps/wordpress/htdocs/config.php. Здесь может помочь и Ansible — модуль lineinfile идеально подходит для таких вещей.

Переходим к ec2_facts

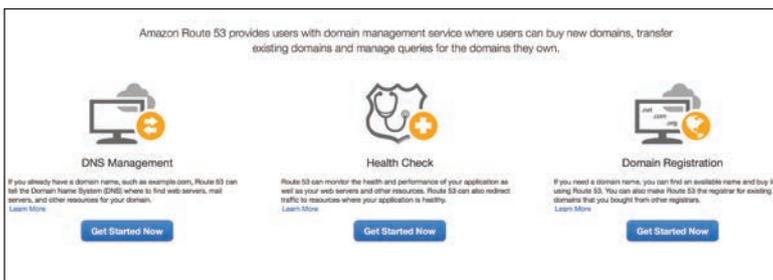
Вероятно, вы увидели, как возможность ссылаться на динамические переменные между запусками могла бы существенно помочь в этом случае. В нашем коде есть примеры жестко закодированных значений переменных. Это отлично подходит для примеров, но я также хотел бы использовать это внутри сценариев воспроизведения. В этом может помочь модуль ec2_facts. Чтобы показать его возможности, я могу добавить новую роль listfacts в файл site.yml (сразу после роли wpcli). Я могу создать файл roles/listfacts/tasks/main.yml, который выглядит так:

```
---
- name: get EC2 facts
  action: ec2_facts
```

Это дает подборку сведений, которые я могу использовать, поскольку они заданы как факты Ansible. После этого я могу использовать их в других местах.

```
ok: [52.18.164.120] => {"ansible_facts": {"ansible_ec2_ami_id": «ami-51345f26», «ansible_ec2_ami_launch_index»: «0»,
```

➤ **Route 53. Наш славный редактор почему-то не в восторге от идеи передачи мне управления linuxformat.com для целей этой статьи...**



Совместимость с PCI и AWS

Каждый, кому приходилось иметь дело с системой, обрабатывающей информацию о кредитных картах, знаком с PCI DSS — промышленным стандартом безопасности данных кредитных карт (Payment Card Industry Data Security Standard). Это набор рекомендуемых практик и мер безопасности, предназначенных для предотвращения мошенничества с кредитными картами. Он вводит контроль над данными и снижает их уязвимость. Однако все эти практики применимы к любым системам, где хранятся данные клиентов, и каждый, кто разрабатывает такую

систему, должен как следует изучить стандарт (каким бы скучным это ни казалось). Многие вещи для системных администраторов очевидны (и, я думаю, в некоторых случаях намеренно описаны довольно туманно) — например, построение безопасной сети или необходимость постоянного тестирования и мониторинга.

Компании, расположенные в верхней части спектра PCI — обычно это крупные предприятия, вложившие немалые деньги в локальную инфраструктуру — по моему опыту, более сдержанны

насчет публичных облаков вроде AWS. Поэтому Amazon (и другие облачные провайдеры) потратили массу времени и усилий на то, чтобы сделать свою инфраструктуру и сервисы безупречными и более привлекательными для таких клиентов. Большая часть сервисов, о которых я упомянул на этих страницах, были проверены на соответствие стандартам. Поэтому гораздо удобнее воспользоваться обзором PCI от тех, кто уже проходил процедуру аудита для PCI. Подробности — на aws.amazon.com/compliance/pci-dss-level-1-faqs.

```
«ansible_ec2_ami_manifest_path»: «(unknown)», «ansible_
ec2_block_device_mapping_ami»: «/dev/sda1», «ansible_
ec2_block_device_mapping_ephemeral0»: «sdb», «ansible_
ec2_block_device_mapping_ephemeral1»: «sdc», «ansible_
ec2_block_device_mapping_ephemeral2»: «sdd», «ansible_
ec2_block_device_mapping_ephemeral3»: «sde», «ansible_
ec2_block_device_mapping_root»: «/dev/sda1»
```

Получи пинок на шоссе... 53?

Amazon предлагает удобный сервис для управления DNS. Сервис Route 53 (он назван по номеру порта, используемого для трафика DNS) предлагает всё полагающееся сервису. Для взаимодействия с сервисом есть API или web-интерфейс, и вариантов взаимодействия очень много. Хорошее и глубокое представление о сервисе вам даст ролик “Amazon Route 53 Deep Dive” на YouTube (<http://bit.ly/LXF202-route53>). В *Ansible* есть модуль, способный взаимодействовать с этим сервисом, и мне довольно легко назначить приватный IP-адрес экземпляру EC2 с записью DNS под своим контролем.

Пусть у меня под управлением есть домен (скажем, сильные мира сего в редакции **LXF** сошли с ума и передали мне управление над linuxformat.com). Я могу создать публичную зону (Public Hosted Zone) в AWS, тогда я получу имена серверов имен. Эти имена надо затем сообщить провайдеру, чтобы запросы DNS направлялись на сервера имен Route 53 Amazon. Сделав это, я смогу

добавить в свою конфигурацию *Ansible* роль, среди прочего — с такими строками:

```
- name: Provision an instance
  local_action:
    module: route53
    command: create
    zone: linuxformat.com
    record: «blog.linuxformat.com»
    type: A
    ttl: 3600
    value: “{{ public_ip }}”
```

У этого модуля есть удобный параметр `overwrite`, который обновит запись или создаст новую, если таковой не существует. У сервиса Route 53 есть несколько параметров высокого уровня. В крупных конфигурациях можно направлять трафик на сайты с наименьшей задержкой, выполнять маршрутизацию по географическим данным (по аналогии с сервисом, предоставляемым CloudFront) и выполнять отработку отката на второй экземпляр, чтобы избежать простоев во время обновления. Внутренним ресурсам также можно назначать записи DNS, не открывая их для внешнего Интернета, что при определенных обстоятельствах бывает очень удобно.

Как всегда, место на страницах ограничено, но надеюсь, что этот обзор AWS и *Ansible* хоть чем-то вам полезен. Если у вас есть вопросы, пишите мне на jolyon.brown@gmail.com, буду рад ответить.

Краткое введение в... Redis

Ищете молниеносно быстрое хранилище данных для сайта? Но без накладных расходов, как с обычной базой данных? Возможный ответ — *Redis*.

В грядущих статьях я иногда буду давать краткое описание программ или пакетов, которыми я пользовался и которые, на мой взгляд, могут вам пригодиться. Первым будет *Redis*, сильно нашумевшее несколько лет назад хранилище пар «ключ/значение» из семейства программ “NoSQL”. На самом деле, называть *Redis* таким хранилищем не совсем честно — разработчики описывают его как «сервер структур данных, поддерживающих различные типы значений». Наряду с традиционными строковыми парами «ключ/значение» *Redis* также поддерживает в качестве значения более сложные структуры данных (такие как списки и свертки). Важнейшие преимущества *Redis* — скорость и количество доступных клиентов. Я видел, как *Redis* использовали в качестве очень эффективного кэша сеанса, где он оказался очень надежным и простым в настройке. У него есть преимущество по сравнению (например) с *Memcached* в долгосрочности хранения данных — способность сделать данные постоянными. Он также отлично подкреплён массой превосходной документации и очень активным сообществом разработчиков.

Текущая работа на данный момент сосредоточена в основном на кластеризации — сейчас версия *Redis* 3.0.2, а в версии 3.0 появился *Redis Cluster*, распределённая реализация *Redis* с автоматическим сегментированием данных и устойчивостью к ошибкам. *Redis* — однопоточная программа, и чтобы насладиться преимуществами лишнего процессора в многоядерной системе, проще всего запустить ещё один сервер (т.е. ещё один процесс *redis*).

Но в целом *Redis* очень быстр — согласно сайту проекта, в средней системе Linux можно достичь показателя в 500 000 запросов в секунду, и ограничивающим фактором является не процессор, а память. Память он использует довольно разумно — с миллионом ключей используется всего несколько сотен МБ (в зависимости от количества полей и их структуры). В то же время, в системе

► *Redis* — пример программы из семейства NoSQL. Сверх меры разрекламированная несколько лет назад, программа с тех пор постоянно улучшается.

хорошо иметь небольшой объем подкачки (и, пожалуй, стоит установить флаг ядра `vm.overcommit_memory`). Заставить ядро из-за нехватки памяти уничтожить память в кэше сеанса, которая не была записана на диск — плохая ситуация.

Другие варианты применения *Redis* включают очереди сообщений, обработку множеств, операции с типами и более общий web-кэш. В состав *Redis* входит удобная утилита командной строки *redis-cli*, с ее помощью я без особых усилий выполнял базовый мониторинг. Также было приятно получить этим методом кое-какую статистику.

Итак, *Redis* — хороший вариант для кэша сеанса. Сравнительно новые функции кластера также стоят внимания. Иногда исчезновение временных данных может не играть большой роли, но для сервисов, у которых утрата сеанса означает «выбрасывание» пользователя из приложения, сохранность этих данных может быть очень важной. Если хотите узнать больше, обратитесь к документации на <http://redis.io> и короткому, но удобному интерактивному руководству на <http://try.redis.io>. **LXF**

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



А если у вас Android, подпишитесь на электронную версию журнала Linux Format на сайте shop.linuxformat.ru!

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Замена FDL и не только

Пора кончать с этим Кунг-Фу.
Мак-Клейн. Крепкий орешек 4.0

Было долгое и нудное юридическое исследование, и наконец-то Creative Commons завершили процесса признания совместимости лицензий CC-BY-SA 4.0 и GPLv3. Теперь в проект с общей лицензией GPL третьей версии можно включать электронные объекты под лицензией CC-BY-SA четвертой.

Лицензия GPL создавалась для программных исходников, и ее анонсирование затачивалась под него, но мир-то состоит не только из кода. Документация должна была по задумке охватываться лицензией GNU FDL, но вы почитайте, что нужно сделать, чтобы ее применить. Оно, конечно, не смертельно в обязательном порядке процитировать несколько страниц лицензии, но довольно неудобно, особенно если тексты небольшие. При создании же Creative Commons авторы отдельно озаботились удобством автора, желающего сообщить о своем выборе; даже хорошо узнаваемые значки придумали.

FSF уже давно признала удобство CC-BY-SA по сравнению с GFDL, разработав спецмодификацию FDL под номером 1.3. Благодаря этому огромный и важнейший для сообщества проект Википедия автоматически перешел на CC-BY-SA тогда еще версии 3.0. Теперь же возможность использования документации для GNU под лицензией Creative Commons подтверждена официально. Естественно, документацией можно не ограничиваться, но тексту CC-BY-SA особенно к месту.
E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Делать снимки 58
Сейчас вылетит птичка! — говорит **Ник Пирс** и создает моментальный снимок системы, чтобы откатиться к нему на случай возникновения проблем.



Пользоваться утилитами 62
Дмитрий Пантелеичев рассматривает базовый набор утилит, без знания которых нельзя считаться линуксоидом.



Настраивать сеть 66
Чувствуете потребность в виртуальной приватной сети? Тогда вам к **Нейлу Ботвику**: он умеет разворачивать такие сети за считанные минуты.



Починять сеть 68
Под чутким руководством **Михалиса Цукалоса** разберемся с неполадками в сети, освоив полезные утилиты. Главное — понять, в чем проблема.



Припасать копии 72
Справившись с соблазном тут же воспользоваться свежесобранной мини-АТС, **Максим Черепанов** начал с мер предосторожности.



Осваивать макропроцессор 76
Хотя автоматизация написания романов **Дмитрию Кузнецову** не видится возможной, он зашел по этому пути довольно далеко.

АКАДЕМИЯ КОДИНГА



И новичкам, и гуру!
Всегда полезно будет познать нечто доселе неизвестное

Подаем сигналы 80
Д-р **Крис Браун** занялся сигналами всерьез — вплоть до таких, которые способны на убийство... процессов в системе, а не другого чего.

Сливаем новости 84
При содействии **Кента Ельчука** произведем анализ RSS-лент и отберем самые актуальные новости — например, по трудоустройству.

СНИМКИ LVM: Копируем диск

Ник Пирс быстро и легко откатывает все нежелательные изменения в системе, применяя моментальный снимок Logical Volume Management.



Наш эксперт

Ник Пирс уже 21 год пишет статьи и долгие годы возится с Ubuntu. Его самое выдающееся достижение на данный момент — мульти-медиа-сервер Mini-PC для Raspberry Pi, запускающий Plex.



Одной из самых притягательных причин для реализации Logical Volume Management (LVM) на ваших жестких дисках являются моментальные снимки [snapshot]. Если вы когда-либо устанавливали программы с ошибками и затем тратили годы, пытаясь устранить причиненный ущерб, или просто вынуждены были ждать, пока не получите полную резервную копию, чтобы взяться за тестирование, то моментальные снимки значительно облегчат вашу жизнь.

Моментальные снимки — это тома, специальным образом созданные на вашем диске и отражающие состояние диска на момент создания. Начиная с этого момента и далее, блоки вашего диска проверяются на предмет изменений, и эти изменения записываются в том моментального снимка. То есть у вас есть безотказная резервная копия, которую можно употребить в случае, если произойдет худшее и ваша система станет нестабильна. Моментальные снимки можно использовать и не в столь крайней ситуации: например, если вы поймете, что вам не нравятся внесенные вами изменения и вы хотите мигом вернуть вашу систему к моменту изготовления снимка, вместо того, чтобы вручную отменять все эти изменения.

Изначально моментальные снимки были доступны только для чтения, так что изменения в блоках вашего жесткого диска записывались только в ваш корневой раздел после того, как копия исходного блока была сделана в томе моментального снимка с точными изменениями, записанными в таблицу исключений. Это значит, что любые файлы, которые вы добавляете, изменяете или удаляете — например, через установку, обновление или удаление прило-

жений — эффективно защищаются посредством моментального снимка. Если они вызывают проблемы на вашем компьютере, возвращение к моментальному снимку отменит их и вернет компьютер к тому состоянию, когда снимок был сделан.

С появлением LVM2 моментальные снимки стали read/write по умолчанию и более гибкими. Как мы выясним далее, теперь можно использовать том моментального снимка в качестве песочницы, позволяя легко и просто превратить ваш компьютер в тестовый полигон без необходимости возиться с полными резервными копиями или виртуальными средами. Он работает, записывая изменения в моментальный снимок вместо основного раздела, что дополнительно облегчает восстановление ваших исходных настроек в случае проблем. Если впоследствии вы решите сохранить свои изменения, то вы просто объединяете содержимое вашего тома моментального снимка с основным разделом.

На нашем уроке мы рассмотрим, как использовать моментальные снимки в Ubuntu — как в качестве простого инструмента для резервных копий/восстановления, так и в качестве виртуального тестового полигона. Мы используем версию Ubuntu LTS 14.04.2, и если у вас другая версия, действия могут отличаться.

Готовимся к снимку

Моментальные снимки работают только на жестких дисках, настроенных с использованием Logical Volume Management. Вам предлагается эта опция при первой установке Ubuntu — если вы тогда отметили Use LVM with the new Ubuntu installation, то все отлично (можете перейти к следующей главе). А если LVM на вашей системе не был настроен, у вас есть два варианта: первый — форматировать или конвертировать ваш диск в LVM, а второй включает совершенно иной подход.

Если вы хотите использовать моментальные снимки в том виде, в котором они предоставляются LVM, ваш первый вариант — форматирование диска — означает, что понадобится переустановить Ubuntu с нуля, на сей раз отметив во время установки важную опцию LVM. Во многих случаях данный подход оказывается непрактичным, и неплохо вам знать, что существует обходной путь, который сохранит существующие настройки. Местами он довольно неуклюж, и если что-то пойдет не так, вы в итоге все равно потеряете установленную систему, поэтому сделайте полную резервную копию своей системы, прежде чем начинать — попробуйте опцию live CloneZilla (<http://clonezilla.org>), если ищете нужный инструмент. Не пропускайте эту стадию: если конвертирование не удастся, восстановите свою резервную копию и попробуйте снова.

Быстрейший и простейший способ конвертировать в LVM — взвалить всю черную работу на бесплатную утилиту *blocks*. Это относительно безболезненный процесс, однако есть небольшое

Скорая помощь

Если вы включили LVM при установке Ubuntu, вы увидите, что Grub и ваши загрузочные файлы находятся на собственном разделе за пределами LVM. Если предполагается, что ваши изменения включают модификацию ядра или какие-то другие реформы в разделе /boot, сделайте их резервную копию отдельно с помощью такого инструмента, как CloneZilla или dd.

Объем тома моментальных снимков

Одна из самых хитроумных проблем по части томов моментальных снимков — это их объем. Если у вас огромный жесткий диск, довольно просто отвести под них больше, чем, по вашему мнению, им когда-либо понадобится; ну, а если вы ограничены в свободном месте? Беда в том, что если пожадничать, и ваш том моментальных снимков впоследствии заполнится изменениями, снимок будет считаться недействительным, и вы не сможете его применить — даже для восстановления вашей системы с этой точки. Если у вас достаточно свободного неиспользуемого места, вы можете изменить раз-

мер тома до того, как он переполнится, обеспечив ему дальнейшую нормальную работу. Это можно сделать вручную с помощью Logical Volume Manager путем выбора вашего тома моментальных снимков под Logical View и щелчку по Edit Properties, чтобы расширить его. Также технически возможно заставить LVM следить за томом, и когда он достигает определенного порога, расширять его автоматически. Для этого отредактируйте файл `lvm.conf`:

```
sudo gedit /etc/lvm/lvm.conf
```

Задайте в файле поиск `snapshot_autoextend_threshold` и измените значение на 80, чтобы автома-

тически расширять его, как только том заполняется на 80 % (вы можете уменьшить это значение до 50). Желая изменить величину, на которую расширяется том, измените значение `snapshot_autoextend_percent` (по умолчанию 20 %). Теперь сохраните файл, закройте `gedit` и перезапустите, чтобы изменения вступили в силу.

К сожалению, на момент написания данной статьи настройка, о которой мы рассказали, как выяснилось, не работает в Ubuntu, но об этой проблеме сообщено на Launchpad (см. <http://bit.ly/LVMAutoExtendBug>).

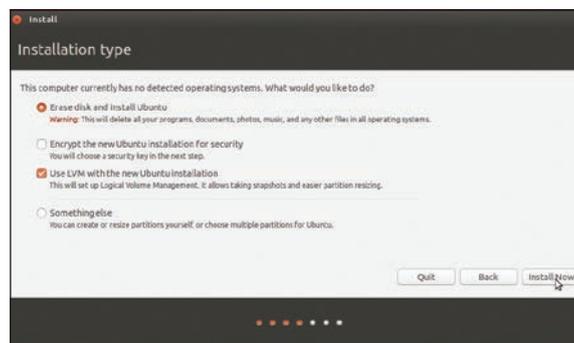
осложнение: `blocks` требует более ранней версии Python (3.3). Не волнуйтесь, окно Convert your drive to LVM в точности показывает весь предстоящий вам процесс. Когда диск будет конвертирован, не спешите перезагружаться, поскольку ваша инсталляция будет незагружаемой. Без паники — сначала надо отремонтировать `Grub`, и это будет сделано без шума и пыли инструментом `Boot Repair`. Перейдите на <https://help.ubuntu.com/community/Boot-Repair>, там есть полное руководство по его установке и использованию. Просто следуйте рекомендованным настройкам восстановления, и вы увидите, что Ubuntu снова загружается, на сей раз с работающим LVM. Даже проделав это, не удивляйтесь, если увидите, что `Grub` жалуется на отсутствие поддержки записи `diskfilter`. Это не критичная ошибка — секунд через пять все выяснится, и ваш компьютер нормально загрузится; это подтвержденный недочет `Grub` (см. <http://bit.ly/GrubDiskfilterBug>), и отладка появится скорее рано, чем поздно.

Если в вашей системе не включен LVM и у вас нет никакого желания его реализовывать, вы можете получить часть функций моментального снимка, используя комбинацию `rsync`, `diff` и `Cron` с помощью интерфейса KDE `Back In Time`. Это позволит вам выбрать папку, резервную копию которой вы хотите сделать, затем сохранить ее на диск или раздел резервного копирования. Место на диске минимизировано путем записи только изменений, и `Back In Time` позволяет автоматически обновлять моментальные снимки согласно установленному графику, чего нельзя делать с моментальными снимками LVM. Поищите в своем менеджере пакетов (мы используем Ubuntu и Software Center) 'snapshot' и установите версию `Back In Time` (root), чтобы разрешить доступ ко всем вашим файлам.

Настройте том резервных копий

Моментальные снимки LVM работают благодаря созданию тома на неиспользованной части вашего жесткого диска. А значит, понадобится припасти место — одним из двух способов: либо изменить размер вашего первичного диска, высвободив достаточно места для моментальных снимков, либо использовать другой жесткий диск или раздел. Где бы вы ни решили отвести место, 10 ГБ будет более чем достаточно, но если вы ограничены в объеме, то хватит и 2 ГБ (а то и меньше), если планируемые вами изменения незначительны. Если же вы обнаружили, что том ваших моментальных снимков (или томá, если вы храните многочисленные изменения) выходит за отведенные рамки, вы всегда успеете добавить ему места, по обстоятельствам.

Мы постараемся отвертеться от работы в терминале с помощью бесплатного инструмента под названием Logical Volume Man-



agement, который можно найти в *Software Center*, задав в поиске LVM. Предупреждение: он плохо ведет себя с дисками RAID (если это ваш случай, найдите во врезке справа на полях *Скорая помощь* необходимые вам команды терминала). Вы также можете использовать инструмент для изменения объема своего первичного раздела или отведения свободного места, откуда бы вы его ни взяли. Если вы пойдете по этому пути, вам понадобится запустить Logical Volume Management с вашего live CD, что означает установку его оттуда через *Software Center*.

После установки запустите Logical Volume Management. Если вы хотите изменить размер корневого раздела, выделите его под Volume Groups > Volume name > Logical View и выберите Edit Properties. Уменьшите объем вашего тома так, как вам надо. Щелкните по OK и подождите, пока объем изменится. Затем выберите Logical View, где Unused Space [свободное место] станет видимым как часть группы томов. Не создавайте здесь ничего — здесь будет храниться ваш будущий том (или тома) моментальных снимков.

Если вы используете вторичный диск, вам незачем целиком отводить его под моментальные снимки — если на диске уже есть данные, сперва используйте `GParted`, чтобы изменить объем раздела для высвобождения требуемого места. После этого создайте пустой неотформатированный раздел на образовавшемся свободном месте. Затем вы можете переключиться на Logical Volume Management, где найдете новый раздел под Initialised Entities. Выберите в списке раздел, который вы хотите использовать (используйте Properties, чтобы определить нужный по его объему), и щелкните по Initialise Entity, а затем по Yes. Отныне диск больше не будет показываться под Unallocated Volumes — щелкните по Add to existing Volume Group, выберите существующую группу томов и щелкните по Add. Теперь освободившееся место доступно для использования.

► Простейший способ добавить Logical Volume Management в вашу систему — отметить нужную опцию при первой установке Ubuntu.

Скорая помощь

Logical Volume Management (LVM) не ладит с системами RAID, и если вы относитесь к этой категории, понадобится исследовать LVM через терминал. См. <https://wiki.ubuntu.com/Lvm>, чтобы начать.

»

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

Если вам не хватает места для моментальных снимков, для увеличения количества доступного места можно применить сочетание обоих методов — уменьшения объема первичного раздела и распределения свободного места из других разделов.

Теперь ваша система готова делать моментальные снимки; простейший способ сделать это — через сам Logical Volume Management. Откройте его с помощью программы запуска приложений и при требовании введите пароль. Когда появится основное окно, раскройте Logical View под Volume Groups, выберите свой корневой раздел и щелкните по кнопке Create Snapshot. Если у вас появится сообщение об ошибке модуля dm-snapshot (что при первой попытке весьма вероятно), откройте отдельное окно терминала и наберите `sudo modprobe dm-snapshot`.

Щелкните по Create Snapshot снова. Дайте вашему тóму моментальных снимков удобное и описательное имя — `testbed`, `rootsnap` или нечто подобное. Вы можете отвести ему все оставшееся место (щелкнув по Use remaining) или указать точный объем с помощью окна размеров LV или бегунка. Оставьте оба окна Mount неотмеченными и щелкните по ОК. После короткой паузы, пока LVM перезагружается, вы увидите обзор новой структуры. Поздравляем — вы только что сделали свой первый моментальный снимок системы!

В этом пункте надо решить, как вы собираетесь использовать свои моментальные снимки. Простейший метод — использовать их только для отката, и тогда дальше вы работаете, как обычно.

Изменения записываются в основной раздел, и если случится нечто нежелательное, вы сможете восстановить снимок через терминал по команде

```
sudo lvconvert --merge /dev/<volgroup>/<snapshotname>
```

где `<volgroup>` — название группы тома вашего LVM, а `<snapshotname>` надо заменить на имя, которое вы дали тóму моментальным снимкам при создании; если вы нетвердо его помните, откройте Logical Volume Management и выберите свой моментальный снимок под Logical View. В списке вы увидите и имя моментального снимка (Logical Volume Name), и имя группы (Volume Group Name).

Когда вы нажмете Enter, вам скажут, что снимок нельзя объединить с 'orep origin volume', что обычно означает необходимость перезагрузки системы — исходные файлы моментального снимка заменят ваши изменения, и ваш корневой раздел успешно вернется ко времени, когда был сделан моментальный снимок. Если ваши изменения не дают вам загрузиться, вы можете запустить команду с вашего live CD.

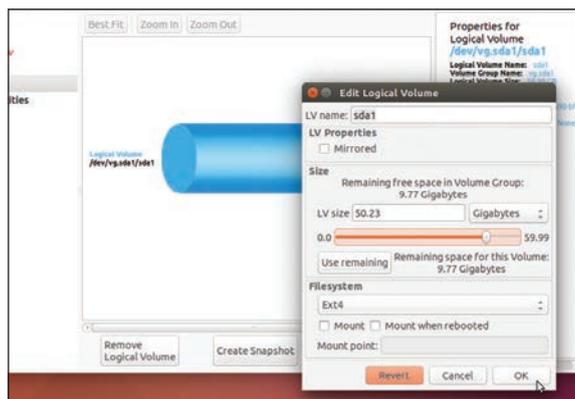
Если вы все еще не можете загрузиться после запуска команды, зажмите Shift при загрузке, затем выберите Advanced Options и выберите более старую версию ядра. После того, как она успешно загрузится, используйте обновление ПО, чтобы ваша установка снова стала современной.

Виртуальный тестовый полигон

Вы также можете использовать моментальные снимки несколько иным способом, посредством временного монтирования тома снимков при загрузке. Изменения вы будете записывать туда, а не в основной раздел. Если что-то пойдет не так, можете просто перезагрузиться в основной раздел, и ваши изменения магическим образом испарятся; затем сотрите моментальный снимок и начните заново. А если все нормально, можете объединить содержимое вашего тома моментальных снимков с основным разделом, чтобы увековечить изменения. Пошаговое руководство (см. внизу стр. 61) покажет, как это делается.

Учтите, что если вы хотите продолжить тестирование после перезагрузки, вам придется каждый раз следовать шагу 2 руководства, чтобы вручную заставить *Grub* использовать ваш раздел с моментальными снимками. Если это вам надоест, создайте инструментом вроде *Grub Customizer* пункт загрузки для своего тома моментальных снимков и при каждом запуске компьютера

➤ Простейший метод обеспечить место для ваших моментальных снимков — взять его из вашего корневого раздела.



Конвертируйте свой диск в LVM

Вначале загрузитесь в Ubuntu и установите *lvm2* через *Software Center*. Теперь откройте окно терминала и наберите `sudo nano /etc/fstab`, чтобы удостовериться, что диски перечисляются по их UUID (если нет, понадобится отредактировать ссылки — определите их по `sudo blkid`). Затем наберите `sudo updateinitramfs -u`, чтобы перекомпилировать образ *initramfs*.

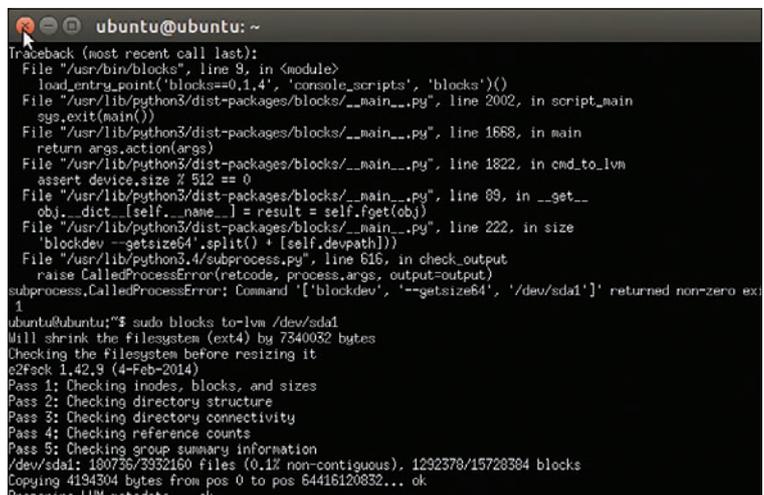
Перезагрузитесь в свой Ubuntu live CD, откройте окно терминала и наберите следующие команды:

```
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:g2p/storage
$ sudo add-apt-repository ppa:frull/deadsnakes
$ sudo apt-get update
$ sudo apt-get install python3.3
$ sudo apt-get install python3-blocks bcache-tools
```

Теперь наберите такую команду в терминале, чтобы выполнить конвертирование:

```
blocks to-lvm <device>
```

Замените `<device>` на информацию об устройстве — обычно `dev/sda1`, и теперь раздел должен быть конвертирован. Запомните присвоенные имя группы раздела (`vg.sda1`) и имя логического раздела (`sda1`).



➤ Если LVM в вашей системе не настроено, незачем стирать свой диск и начинать заново: его можно конвертировать, установив *lvm2*.

выбирайте этот пункт. Чтобы сделать это, откройте окно терминала и наберите следующее:

```
$ sudo add-apt-repository ppa:danielrichter2007/grubcustomizer
$ sudo apt-get update
$ apt-get install grub-customizer
```

После установки запустите *Grub Customizer*, выберите свой пункт по умолчанию и щелкните по Edit. Нажмите Ctrl+a, а затем Ctrl+c во вкладке Source, чтобы скопировать существующий текст. Щелкните по OK, затем создайте новый пункт. Нажмите Ctrl+v во вкладке Source, чтобы вставить существующий текст, затем настройте строку, начинающуюся с `linux`, чтобы она указывала на ваш раздел моментальных снимков, как указано в шаге 2 руководства. Щелкните по OK, а затем щелкните по new имя, чтобы именовать его.

Начиная с этого момента и далее, *Grub* должен автоматически появляться при запуске, но если это не так, переключитесь во вкладку General Settings в *Grub Customizer* и отметьте show menu под Visibility; это позволит выбрать ваш раздел моментальных снимков в качестве используемого. Наконец, когда вы закончите с моментальными снимками, у вас будет выбор — удалить или

объединить его изменения с вашим основным разделом. Однако не забудьте удалить пункт в *Grub Customizer*.

Тома моментальных снимков обычно существуют столько, сколько вам надо — с одним исключением (см. ниже). Покончив с разделом моментальных снимков, можете оставить его в покое, но когда необходимость в нем отпадет, его следует удалить. Простейший способ сделать это — через Logical Volume Management: просто перейдите в раздел моментальных снимков под Logical View и щелкните по кнопке Remove Logical Volume, которую вы найдете; вы также можете удалить раздел, выбрав сам Logical View — щелкните на томе моментальных снимков, чтобы выбрать его, а затем по Remove Selected Logical Volume(s).

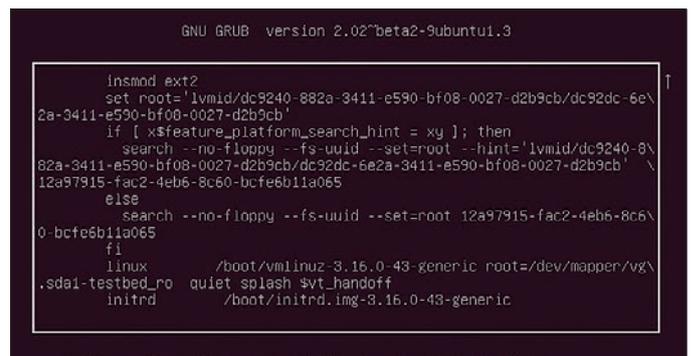
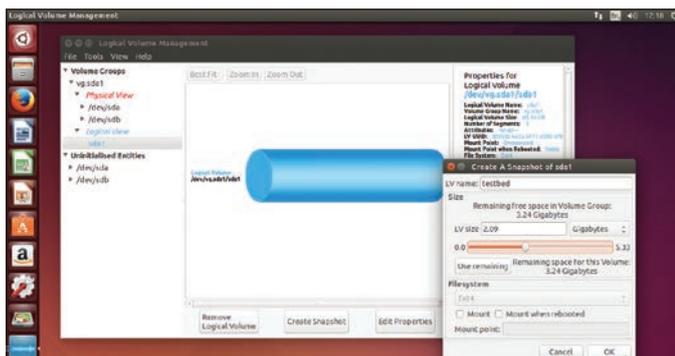
Прочитайте предупреждение и щелкните по Yes. По удалении моментального снимка то место, которое он занимал, перейдет в раздел неиспользуемого, позволяя отвести его под новые моментальные снимки.

Разделы моментальных снимков удаляются автоматически только когда они объединяются с основным разделом по команде `lvconvert --merge`. После объединения моментальный снимок удаляется, а место автоматически отмечается как неиспользуемое. **LXF**

Скорая помощь

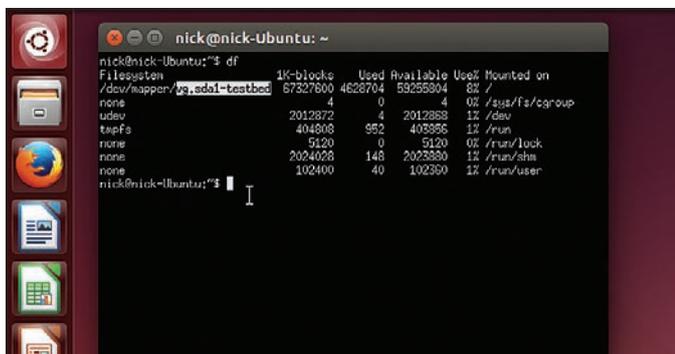
Отслеживать текущее состояние моментальных снимков в LVM легко. Запустите его и выберите снимок из окна Logical View. В Properties, на правой стороне, есть полезная информация, в т.ч. о работе со Snapshot, что поможет вам следить за наличием мест. Чтобы обновить вид, выберите View > Reload.

Моментальные снимки как тестовый полигон



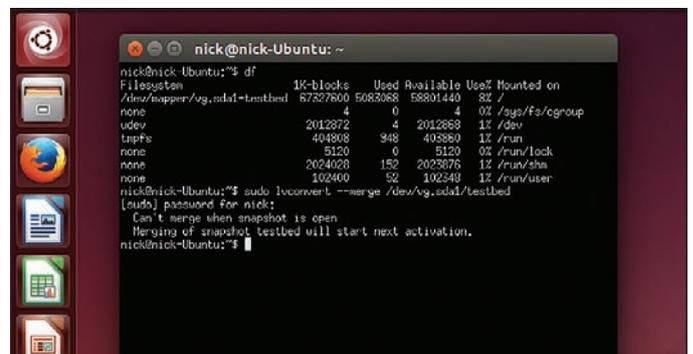
1 Сделайте моментальный снимок

Создайте моментальный снимок, как сказано в основном тексте. Убедитесь, что вы отвели достаточно свободного места вашему разделу моментальных снимков, чтобы вместить все изменения, которые вы планируете протестировать. Если вы не уверены, щелкните по Use remaining, чтобы отвести все доступное место моментальным снимкам. После этого закройте LVM. Откройте окно терминала, наберите `df`, нажмите Enter и запомните ссылку `/dev/mapper/`.



2 Временная загрузка

Перезагрузите свой компьютер, удерживая Shift при загрузке. Когда появится меню *Grub*, нажмите e, чтобы получить доступ к тексту скрипта *Grub*. Используйте стрелки, чтобы определить строку, начинающуюся с `linux`, которая относится к `/dev/mapper/<volgroup>-<snapshot>`, где `<volgroup>-<volname>` соответствует тому, что вы записали с `df` в предыдущем шаге. Измените `<snapshot>`, чтобы оно соответствовало имени, которое вы дали своему моментальному снимку.



3 Используйте тестовый полигон

После входа снова откройте терминал и используйте `df`, чтобы проверить, что вы используете том моментальных снимков, а не основной раздел. Удовольствие в этом, можете поэкспериментировать и убедиться, что все изменения записываются в ваш раздел моментальных снимков, а не в основной. Вы можете отменить эти изменения в любое время, просто перезагрузившись в нормальном режиме, что вернет вас к вашему основному разделу.

4 Объединение изменений

Закончив тестирование, вы можете сохранить произведенные изменения, вновь объединив ваш моментальный снимок с основным разделом. Откройте окно терминала и наберите следующее:

```
$ sudo lvconvert --merge /dev/<volgroup>-<snapshot>
```

Вам скажут, что объединение произойдет во время следующей перезагрузки, так что понадобится перезагрузиться, чтобы завершить процесс.

Пакет GNU Core Utilities: часть 1

Дмитрий Пантелеичев представляет незаменимый пакет из программного окружения GNU, затеяв захватывающее путешествие по пакету.



Наш эксперт

Дмитрий Пантелеичев попробовал в жизни многое, от 1С до Silverlight. И пришел к выводу, что надо стать свободным: как от наркотиков, так и от владельцев прав на коммерческое программное обеспечение.

Что такое проект GNU? Редко кто из нас, произнося слово “Linux”, вспоминает, что название это не совсем справедливое. Linux — это только ядро нашей любимой операционной системы. Основное же программное окружение, которым мы пользуемся, появилось гораздо раньше ядра. И называется это программное окружение GNU. Следовательно, всю ОС следует называть GNU/Linux. Проект GNU появился в 1983 г., а основал его Ричард Столлмен. Специально ради развития своего проекта он в 1984 г. уволился из Массачусетского технологического института (MIT), чтобы всё свое время посвятить написанию свободного ПО. Идеология, продвигаемая Ричардом Столлменом, стала ответом на коммерциализацию системы UNIX, которая начала тормозить развитие технологий, так как разработчики и пользователи оказались сильно зависимыми от правообладателей. Программное окружение, разрабатываемое в рамках проекта GNU, и по сей день остается основным средством администрирования системы.

Главный ресурс проекта GNU в Интернете — сайт <http://gnu.org>. Там любой желающий может ознакомиться с историей и идеологией проекта, узнать, как подключиться к разработке, оказать другую помощь, а также скачать самые свежие версии программ. Количество программ, разрабатываемых в рамках проекта GNU, ныне довольно велико. Многие из них имеют графический интерфейс (например, *GnuCache*, программа финансового учета). Официального минимального списка «базовых» компонентов нет, но в качестве «базового» списка можно назвать следующие компоненты:

- » **GNU Core Utilities** Набор базовых утилит, изучение которых будет начато на этом уроке.
- » **GNU Toolchain** Набор инструментов для разработчика, включающий компилятор, ассемблер, компоновщик, отладчик, средства для распространения программы для конечных пользователей.
- » **GNU C Library** Версия стандартной библиотеки C для ОС GNU.
- » **Bash** Командная оболочка.
- » **Gnome** Графическая оболочка.

Пакет GNU Core Utilities

На нашем уроке мы начнем рассматривать набор утилит GNU Core Utilities (или *coreutils*). Этот набор считается абсолютно базовым и необходимым, и предполагается, что он установлен по умолчанию в любом дистрибутиве Linux. Тем не менее, прежде чем приступать к прохождению этого учебника, убедимся, что пакет у нас установлен. Для этого введем команду любой утилиты из этого пакета с флагом `--version`. Например, команду `mkdir` (создание каталога), `chmod` (изменение прав доступа к файлу) или любую другую.

```
mkdir --version
chmod --version
```

В результате на экране должен появиться текст, похожий на показанный на рис. 1. Прочитать подробно про этот пакет и все его компоненты можно, набрав в консоли команду

```
info coreutils
```

Можно также открыть справку в формате `info` сразу на странице, где содержится информация по нужной программе. Для этого надо к `info` приписать название нужной программы:

```
info mkdir
```

По любой утилите также можно вызвать справку, набрав ее команду с ключом `--help`:

```
mkdir --help
```

Можно также вызвать справочные страницы, где содержится информация по данной программе. Справочные страницы выводит команда `man` с названием нужной утилиты в параметре, например,

```
man mkdir
```

Здесь надо добавить, что разные командные оболочки могут содержать дополнительные встроенные функции. Такие функции исполняются внутри самой командной оболочки, а не внешними программами. Например, оболочка *Bash*, наиболее популярная в среде пользователей Linux, содержит команду `cd` (изменение текущего каталога). Однако в пакете GNU Core Utilities вы не найдете утилиты `cd`. Это встроенная команда *Bash*. Встроенные команды будут изучаться в учебнике, посвященном *Bash*. Здесь же они просто будут коротко объясняться по ходу примеров.

Существуют также системные утилиты для выполнения административных функций, запускаемые суперпользователем. Здесь они также будут коротко объясняться по ходу дела. Примеры таких утилит — *useradd*, *usermod*, *userdel*, они предназначены для администрирования учетных записей пользователей.

Просмотр данных о пользователе

Рассмотрим утилиты из пакета GNU Core Utilities, позволяющие просмотреть информацию о пользователе: *id*, *logname*, *whoami*, *groups*, *users*, *who*. Одна из самых простых из них — *whoami*. Она всего-навсего выводит имя текущего пользователя системы, то есть отвечает на вопрос «кто я». Набираем в консоли команду

```
whoami
```

и вот что видим (в моем случае):

```
dima
```

Перед нами появляется имя пользователя системы. Здесь нас может заинтересовать, нет ли возможности добавить какой-нибудь ключ, чтобы получить информацию поподробнее. За этими сведениями обратимся к справке по утилите. Наберем имя утилиты с ключом `--help`:

```
whoami --help
```

Почитав вызванную справочную систему, мы видим, что, кроме двух знакомых нам ключей `--help` и `--version`, никакие другие ключи здесь не предусмотрены.

Перейдем к следующей утилите пакета — *logname*.

```
logname
```

```
dima
```

Вывод такой же, как у *whoami*. Возникает мысль, что эти утилиты тождественны друг другу. Так ли это?



Скорая помощь

Меню сайта <http://gnu.org> предусматривает перевод содержимого сайта на множество языков, в том числе и русский.

Чтобы проверить это, давайте создадим нового пользователя. Создавать нового пользователя имеет право только суперпользователь-root. Зайдем в систему как root. Для этого введем команду `su`

Появится приглашение ввести пароль суперпользователя. После успешного ввода пароля мы попадаем в систему с правами root. Создадим пользователя... скажем, с именем thomas. Для этого воспользуемся командой `useradd`. Она вызовет системную утилиту из числа тех, которые необходимы для администрирования системы. В пакет GNU Core Utilities она не входит.

```
useradd thomas
```

Далее, нам надо задать этому пользователю пароль. Это делается с помощью системной утилиты `passwd`:

```
passwd thomas
```

В консоли появится приглашение ввести пароль. После ввода пароля появится приглашение ввести его повторно. Когда мы успешно введем пароль два раза, появится сообщение: «все данные аутентификации успешно обновлены». Выйдем из учетной записи суперпользователя с помощью команды `exit`, ратем войдем в систему как наш новый пользователь:

```
su thomas
```

Снова появится приглашение ввести пароль, на сей раз — пароль нашего нового пользователя. После ввода его пароля мы войдем в систему как этот пользователь. Набрав команду `whoami` снова, мы увидим такой результат.

```
whoami
```

```
thomas
```

Теперь вызываем команду `logname`:

```
logname
```

```
dima
```

Результат уже другой! Итак, наш эксперимент показал: команда `whoami` выводит имя текущего активного пользователя, а команда `logname` — имя пользователя, от имени которого мы вошли в систему. Не забудем выйти из-под нового пользователя:

```
exit
```

Переходим к новому компоненту нашего пакета: посмотрим, что собой представляет утилита `users`. Она выводит имена всех пользователей, вошедших в систему. В данный момент никто, кроме нашего текущего пользователя, в систему не входил. Но почему же его имя отображается два раза?

```
users
```

```
dima dima
```

Для ответа на этот вопрос вызовем утилиту `who`. В результате выведутся имена пользователей системы с указанием, в какое время и через какие устройства те входили в систему. Видно, что пользователь входил в систему два раза через два разных устройства.

```
who
```

```
dima tty8 2015-08-26 22:09 (:0)
```

```
dima pts/1 2015-08-26 22:09 (:0)
```

Откроем на экране еще одну консоль (чтобы были две консоли одновременно) и введем еще раз `users`. Вот что получилось.

```
users
```

```
dima dima dima
```

Теперь введем `who`:

```
who
```

```
dima tty8 2015-08-26 22:09 (:0)
```

```
dima pts/1 2015-08-26 22:09 (:0)
```

```
dima pts/2 2015-08-26 22:09 (:0)
```

Что это означает? Это означает, что теперь пользователь зашел в систему три раза через три разных устройства (см. также рис. 2).

Первое из них — это эмулятор терминала (TTY-устройство). Как известно, каждое физическое устройство в UNIX-подобных системах представлено в виде специального файла, то есть интерфейса, который выглядит, как файл, а на самом деле является программным входом в физическое устройство. Однако, в отличие

```
dima@dima-System-Product-Name ~ $ mkdir --version
mkdir (GNU coreutils) 8.21
Copyright (C) 2013 Free Software Foundation, Inc.
Лицензия GPLv3+: GNU GPL версии 3 или новее <http://gnu.org/licenses/gpl.html>
Это свободное ПО: вы можете продавать и распространять его.
Нет НИКАКИХ ГАРАНТИЙ до степени, разрешенной законом.

Автор программы -- David MacKenzie.
dima@dima-System-Product-Name ~ $
```

► Рис. 1. Текст, отобразившийся при вводе `mkdir --version`.

от других, TTY-интерфейс не связан с каким-либо аппаратным устройством: это программный эмулятор поведения настоящего терминала. Поскольку в списке, выведенном командой `who`, этот программный терминал обозначается как `tty8`, выходит, что в нашей системе несколько таких терминалов? Да! И, как видно по списку пользователей, мы вошли в систему через терминал №8.

А можно ли зайти в систему через другой виртуальный терминал? Оказывается, можно. Для переключения виртуальных терминалов служит сочетание клавиш `Ctrl+Alt+` (функциональная клавиша с номером, соответствующим номеру терминала). Так, чтобы попасть на терминал №8, надо нажать сочетание `Ctrl+Alt+F8`. Но давайте попробуем попасть в терминал №1 — `Ctrl+Alt+F1`.

Перед нами предстал черный экран с приглашением `login:`. Да, здесь графическая оболочка не запущена, и поработать придется в режиме командной строки. Вводим имя пользователя. Появится приглашение ввести пароль — введем его. И если пароль правильный, то, наконец, появляется приглашение командной оболочки.

Вводим команду `who` и видим такую картину:

```
dima tty8 2015-08-26 22:23 (:0)
```

```
dima tty1 2015-08-26 22:23
```

```
dima pts/1 2015-08-26 22:23 (:0)
```

```
dima pts/2 2015-08-26 22:23 (:0)
```

Видно, что наш пользователь теперь зашел еще раз через устройство `tty1`. Вернемся в терминал №8: `Ctrl+Alt+F8`. О радость! Снова появилась наша графическая оболочка с открытыми в ней двумя программными консолями. Закроем обе консоли (для этого в каждой из них можно ввести команду `exit`) и снова перейдем в терминал №1 — `Ctrl+Alt+F1`. Мы увидим, что в системе остались только два пользователя, зашедшие через `tty8` и `tty1`, а пользователи, зашедшие через `pts/1` и `pts/2`, исчезли.

```
dima tty8 2015-08-26 22:35 (:0)
```

```
dima tty1 2015-08-26 22:35
```

Исходя из этого, мы можем сделать вывод, что устройства `pts/1` и `pts/2` соответствуют запущенным экземплярам программной консоли. На самом деле это так называемые псевдотерминалы. Псевдотерминал — это виртуальное устройство, относящееся к программному терминалу, как «ведомый» по отношению к «ведущему» (`slave` по отношению к `master`). Псевдотерминалам назначаются окна консоли в графической среде, а также сеансы удаленной связи с системой по протоколу SSH или Telnet.

Итак, мы незаметно подобрались к утилите `who`. Давайте, для начала, посмотрим, какие есть варианты вызова этой утилиты. Вызовем ее с ключом `--help`, чтобы посмотреть справочную систему:

```
who --help
```

Вот здесь вариантов действительно много. Сразу обращает на себя внимание, что большинство ключей имеет два варианта написания: короткий и длинный. Например, можно ввести команду `who` с коротким ключом:

```
who -h
```

А можно и с длинным:

```
who --heading
```

В обоих случаях результат будет одинаковым — дополнительно будут выведены заголовки столбцов.

```
ИМЯ ЛИНИЯ ВРЕМЯ КОММЕНТАРИЙ
```

```
dima tty8 2015-08-26 22:09 (:0)
```

```
dima pts/0 2015-08-26 22:09 (:0)
```

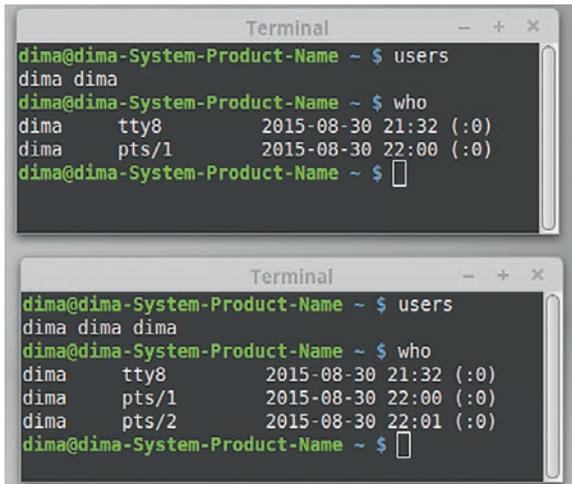
```
dima pts/1 2015-08-26 22:09 (:0)
```



По любой утилите пакета GNU Core Utilities можно вызвать справку, набрав ее команду с ключом `--help` (пример: `mkdir --help`), а также открыть раздел справочной системы с описанием этой программы (пример: `man mkdir`). Можно также почитать справку в формате `info`, как по всему пакету (`info coreutils`), так и по каждой отдельной программе (пример: `info mkdir`).

»

Рис. 2. При открытии второй консоли в системе появляется еще один пользователь.



Внимательно почитайте справочную систему и попробуйте запустить команду `who` с каждым ключом. Например, вызов с ключом `-b` (или `--boot`) не покажет пользователей, но зато выведет время последней загрузки системы: `who -b`. А вызов с ключом `-a` (`--all`) выведет вообще всю информацию насчет этой утилиты: `who -a`.

Перейдем к утилите `groups`. Как можно выяснить, вызвав справочную систему (вы не забыли, что для этого надо набрать `groups --help?`), эта утилита отображает данные о членстве пользователей в группах. Если пользователь не указан, отобразится группа, к которой принадлежит текущий процесс. В нашей системе в данный момент есть пользователь `dima`, входящий в группу `dima`, и пользователь `thomas`, входящий в группу `thomas`. По вызову команды `groups` отобразится название группы текущего процесса.

```
groups
dima
```

Если мы вызовем эту команду с указанием имени пользователя, то будет выведено имя пользователя и его группа.

```
groups dima
dima : dima
groups thomas
thomas : thomas
```

Давайте немножко изменим эту ситуацию. Переименуем группу `dima` в группу `students`. Для этого нужно использовать системную утилиту `groupmod` и зайти в систему как суперпользователь.

```
su
groupmod -n students dima
exit
```

Посмотрим, что у нас получилось.

```
groups dima
dima : students
```

Теперь поместим в эту же группу и пользователя `thomas`.

```
su
usermod -G students -a thomas
exit
```

Смотрим, что получилось.

```
groups thomas
thomas : thomas students
```

Результат: пользователь `thomas` теперь состоит в двух группах.

Переходим к последней утилите этого раздела — `id`. Она выводит имена пользователей и их группы вместе с идентификаторами пользователей и групп, а также контекст безопасности процесса (если доступна система SELinux, т.е. Linux с улучшенной безопасностью). Если вызвать эту утилиту, указав имя пользователя в параметре, будут выведены только имя, идентификатор и групповой идентификатор пользователя, и имена и идентификаторы групп.

```
id thomas
uid=1001(thomas) gid=1001(thomas)
```

`группы=1001(thomas),1000(students)`
Тот же результат получится, если в параметре указать пользователя не по имени, а по идентификатору:

```
id 1001
Если вызвать ее с ключом -Z, будет выведен только контекст безопасности процесса. Команда с этим ключом будет работать только в том случае, если система поддерживает SELinux.
```

```
id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Контекст безопасности процесса — это свойство процесса, применяемое в системе SELinux и включающее его пользователя-владельца, роль, тип и уровень.
```

Просмотр информации о системе

Переходим к следующему классу программ из пакета GNU Core Utilities — средствам отображения информации о системе: `date`, `arch`, `nproc`, `uname`, `hostname`, `hostid`, `uptime`. Начнем с утилиты `arch`. Она просто выводит название архитектуры компьютера:

```
arch
x86_64
```

Никаких ключей (кроме обычных `--help` и `--version`) `arch` не имеет.

Утилита `nproc` выдает сообщение о количестве процессоров.

Еще одна утилита, `uptime`, выдает информацию о текущем времени, времени непрерывной работы системы, количестве пользователей и о средней загрузке системы. Время непрерывной работы системы вычисляется с момента загрузки до текущего момента. Средняя загрузка отображается в виде трех значений, которые представляют собой среднее количество работающих процессов плюс количество неактивных процессов, ожидающих ввода информации [uninterruptable sleep state] за последние 1, 5 и 15 минут.

```
uptime
17:24:29 up 8 min, 2 users, load average: 0,46, 0,40, 0,18
```

Заметим, что утилита `uptime` присутствует не во всех системах, а только в тех, где установлены средства для определения времени загрузки. Поэтому, разрабатывая переносимые скрипты, предполагаемые к запуску на множестве разных систем, не следует ожидать, что команда `uptime` будет работать везде. Кроме того, эту же команду могут предоставлять другие программы, установленные в системе. Так, если мы введем эту команду с ключом `--version`, мы можем увидеть совсем необычный результат.

```
uptime from procpss-ng 3.3.10
```

Это означает, что команда `uptime` была вызвана не из пакета GNU Core Utilities, а из программы `procpss-ng`, средства для получения информации о процессах.

Следующая утилита, `hostname`, отображает имя хоста, то есть данного компьютера, как устройства, имеющего уникальную идентификацию в сети. Эта команда, как и команда `hostid`, которая отображает не имя хоста, а его числовой идентификатор, тоже может отсутствовать в системе, а предоставляться другими программами. Учтите это при написании переносимых скриптов.

Далее рассмотрим утилиту `uname`. Она печатает сведения о системе. Так, с ключом `-s` (или без ключа) она показывает информацию об имени ядра операционной системы.

```
uname -s
Linux
```

С ключом `-r` — версию ядра операционной системы:

```
uname -r
3.16.0-38-generic
```

С ключом `-o` — имя операционной системы:

```
uname -o
GNU/Linux
```

А с ключом `-a` — всю информацию по данной утилите.

Мы подошли к последней утилите этого раздела — `date`. На ней остановимся подробнее, так как работа с ней будет несколько сложнее. Первая ее функция — это вывод текущих даты и времени:

```
date
Сб авг 29 19:13:09 MSK 2015
```

Ключ `-u` позволит вывести время не по местному часовому поясу, а по всемирному координированному времени.

```
date -u
Сб авг 29 16:13:09 UTC 2015
```

Дату и время можно отображать в нескольких форматах. Например, можно отобразить его так.

```
date +%d-%m-%Y
Результатом будет
29-09-2015
```

Или так:

```
date +"%d %B %Y"
29 Август 2015
```

Или даже так:

```
date +"%H ч %M мин %S с"
19 ч 39 мин 57 с
```

Как вы уже, наверное, догадались, формат должен начинаться со знака "+", и выражение после знака "+" может заключаться в кавычки. Кавычки необходимы в тех случаях, когда внутри формата присутствуют пробелы, потому что иначе пробелы будут восприниматься, как разделы между параметрами команды. Спецификаторы `%H`, `%M`, `%m`, `%u` используются для обозначения элементов даты и времени, например, текущего года, месяца, часа, дня в месяце, дня в году, и т.д. Все это описано в справочной системе. Некоторые спецификаторы могут выводить сразу несколько элементов в определенном формате.

```
date +%R
19:46
```

Формат можно сочетать с любым ключом. Например, время, показанное выше, можно вывести по всемирному координированному времени.

```
date -u +%R
16:46
```

Вторая функция утилиты `date` — это вывод не только текущего, а вообще всякого произвольного времени, заданного в параметре. Дата и время задаются строкой после ключа `-d`:

```
date +"%d %B %Y" -d "2015-02-01"
```

Или, как вариант,

```
date +"%d %B %Y" --date="2005-02-01"
```

Вы не забыли, что ключи могут иметь короткий и длинный варианты? Результат:

```
01 Февраль 2015
```

Если параметр представляет собой пустую строку, то выводит время, соответствующее полночи текущей даты.

```
date -d ""
Сб авг 29 00:00:00 MSK 2015
```

Общие правила формата ввода даты можно продемонстрировать следующим списком. Все они правильно вводят одну и ту же дату: 24 сентября 1972 года.

- » 1972-09-24 » 24 Sept 72
- » 72-9-24 » 24 Sep 72
- » 72-09-24 » Sep 24, 1972
- » 9/24/72 » 24-sep-72
- » 24 September 1972 » 24sep72

Строковое обозначение месяца предусмотрено только на английском языке: January, February, March, April, May, June, July, August, September, October, November, December. Возможно также трехбуквенное сокращение названий месяцев: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. Сентябрь также имеет специальное сокращение: Sept. В следующем списке показаны правильные способы ввода времени.

- » 20:02:00.000000 » 8:02pm
- » 20:02:00 » 20:02-0500
- » 20:02

Теперь, на основе всего вышесказанного, выведем на экране следующее время: 16 Август 2015 г, 8 ч 30 мин.

```
date -d "2015-08-16 08:30" +"%d %B %Y г, %l ч %M мин"
```

Наконец, время можно задавать не только через параметры, но и в файле. Давайте создадим файл с именем `datetime`. Забегая вперед, мы используем здесь утилиту `cat` из того же пакета GNU Core Utilities. Подробнее эту утилиту разберем в разделе работы с файлами. Содержимое этого файла будет следующим: 2015-07-02 9:45.

```
cat > datetime << EOF
2015-07-02 9:45
EOF
```

Отобразим дату и время из созданного нами файла. Для этого используется ключ `-f`.

```
date -f datetime
Чт июл 2 09:45:00 MSK 2015
```

Можно отобразить дату и время из файла с указанием формата.

```
date -f datetime +"%d %B %Y г, %l ч %M мин"
```

Третья функция утилиты `date` — показать время последнего изменения файла. Просмотрим время последнего изменения только что созданного нами файла, воспользовавшись ключом `-f`:

```
date -r datetime
```

Или так:

```
date -r datetime +"%d %B %Y г, %l ч %M мин"
```

Файл `datetime` нам больше не пригодится. Поэтому давайте приберем за собой. Удалим этот файл:

```
rm datetime
```

Четвертая функция утилиты `date` — задание системного времени. Для этого нужно ввести команду `date` с параметром, представляющим собой дату и время в формате `[MM]Ддчмм[ВВ][ГГ][сс]`. Здесь `MM` — номер месяца, `Дд` — число, `чч` — часы, `мм` — минуты, `сс` — сотые доли минуты. Элементы в квадратных скобках можно опустить. Например, год можно указать как двумя цифрами (`ГГ`), так и четырьмя (`ВВГГ`). Запись `010203042014.05` означает «2 января 2014 года 3 часа 4 минуты 5 секунд». Запись `0102030414` — «2 января 2014 года 3 часа 4 минуты 00 секунд».

Установим текущее системное время в 21 марта 1986 года 10 часов 30 минут 5 секунд (для этого нам, естественно, понадобятся права суперпользователя).

```
su
date 032110301986.05
exit
```

Посмотрим, что получилось.

```
date
Пт мар 21 10:31:29 MSK 1986
```

Чуть больше, чем мы установили. Но часы-то идут! Так что всё нормально.

Системное время можно также установить с ключом `-u` (или, в полных версиях, `--utc` или `--universal`). Давайте выясним, что из этого выходит.

```
su
date -u 032110301986.05
exit
```

Проверим:

```
date
Пт мар 21 13:30:10 MSK 1986
```

На три часа больше. Проверяем с ключом `-u`:

```
date -u
Пт мар 21 10:32:06 UTC 1986
```

Итак, установка системного времени с ключом `-u` требует указания времени в параметре не по текущему часовому поясу, а по всемирному координированному времени.

На нашем уроке мы рассмотрели программы, входящие в пакет GNU Core Utilities, отвечающие за отображение информации о пользователе и о системе. В следующий раз мы продолжим знакомство с этим пакетом. **LXF**

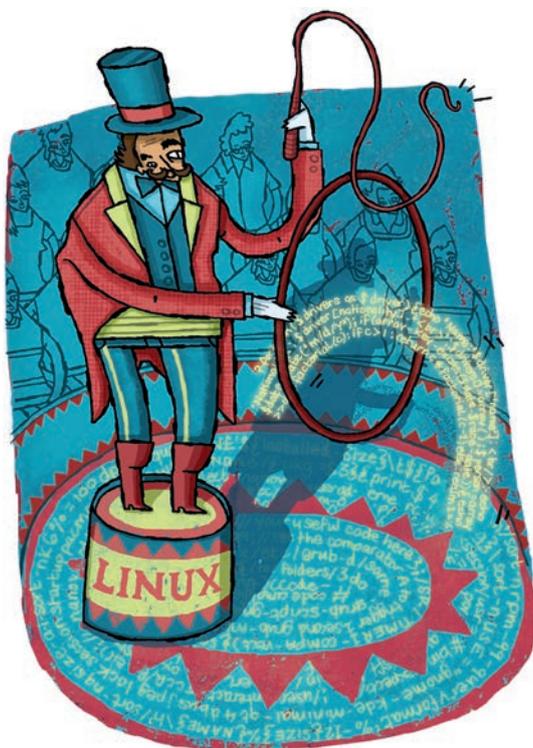
ZeroTier: VPN быстро и просто

Нейл Ботвик отвлекается от настройки *OpenVPN*, чтобы показать вам простую альтернативу с настройкой безопасной сети за считанные минуты.



Наш эксперт

У **Нейла Ботвика** огромный опыт загрузки, ведь у него по компьютеру в каждой комнате, однако его опыт перезагрузки не так велик, ведь он перешел на Linux.



VPN считается наилучшим вариантом безопасного подключения к сети извне. Нам уже случалось рассматривать настройку *OpenVPN*, но это задача нетривиальная. Как первоначальная настройка VPN-сервера, так и управление изменениями при добавлении и удалении устройств требуют времени. Вот было бы здорово, если бы кто-нибудь предложил альтернативу, простую и удобную в настройке, желателен с простым web-интерфейсом... и такая альтернатива есть! *ZeroTier* позволяет создать виртуальную сеть меньше чем за минуту и так же легко добавить в нее компьютеры. *ZeroTier* использует одноранговый протокол, в котором регистрация и обнаружение устройств проходят

через центральные серверы *ZeroTier*, а весь остальной обмен данными происходит непосредственно между компьютерами виртуальной сети. Заметьте, что если оба компьютера находятся в одной сети, то данные не выходят за пределы локальной сети. Это можно считать подобием BitTorrent, в котором трекер сообщает, у кого есть файлы, которые вам нужны, но сама загрузка и выгрузка файлов с трекером никак не связаны.

Первым делом установите программу *ZeroTier* на компьютер. Вы вряд ли найдете ее в репозиториях своего дистрибутива, потому что она достаточно новая, но пакеты для различных дистрибутивов, а также исходные коды можно загрузить с официального сайта (www.zerotier.com). Если пакета для вашего дистрибутива нет, загрузите общий установщик Linux и установите его командой

```
$ chmod +x ZeroTierOneInstaller-linux-x64-1_0_3
$ ./ZeroTierOneInstaller-linux-x64-1_0_3
```

Создание сети

Эта текущая 64-битная версия, и в вашем случае имя файла может немного отличаться. Теперь у вас есть программа для подключения к сети, но сначала нужно создать сеть. Зайдите на www.zerotier.com/admin/admin.html и создайте учетную запись [account], который потребуется при создании сети. (Кто хочет подключиться к существующей сети, может пропустить этот раздел.) Зарегистрируйте свой адрес электронной почты и пароль, и вы перейдете прямо к списку сетей, который будет пуст. Задайте имя сети и нажмите Create Network [Создать сеть]. Новая сеть сразу же появится на странице администрирования. Идентификатор сети [Network ID] — уникальный идентификатор сети, который понадобится для подключения к этой сети. Другие важные настройки: управление доступом [Access Control], которое нужно оставить включенным, если вы не хотите, чтобы к вашей сети мог подключиться кто угодно, и настройки адресации IPv4, которые нужно изменить так, чтобы при подключении каждого устройства *ZeroTier* выдавал ему IP-адрес. Затем вы сможете выбрать блок адресов, из которого будут выбираться адреса. Эти адреса предназначены только для приватной сети и не будут доступны публично. Выбор блока не имеет особого значения, пока он не пересекается с любой из сетей, к которым вы будете подключаться. Ниже вы увидите список устройств, подключенных к сети. Он пуст, поэтому давайте добавим устройство.

Клиент *ZeroTier* в Linux есть только для командной строки, но прежде чем воспользоваться им, следует запустить службу *zerotier-one*. Файлы запуска для *SysVinit* и *Systemd* входят в состав программы; пользователи *Systemd* могут запустить службу командой `$ sudo systemctl start zerotier-one.service`. Если вы хотите, чтобы сеть была доступна всегда, настройте запуск *ZeroTier* при загрузке системы. Теперь откройте терминал и выполните команду

```
$ sudo zerotier-cli join <NETWORKID>
```

Другие сервисы VPN

Быстрый поиск в сети выдаст десятки компаний, предлагающих услуги VPN (Virtual Private Network — виртуальная частная сеть), так чем же они плохи? Некоторые из них представляют собой шлюзы к Интернету в другой стране, что может

использоваться для обхода ограничений, но не предоставляют защищенного подключения к приватной сети. Другие VPN-сервисы нормальные, но нацелены на корпоративных пользователей, стоят довольно дорого и используют закрытое ПО.

Содержите на дому

Сейчас вы, вероятно, размышляете о серверах *ZeroTier* — есть ли какие-либо недостатки в использовании системы, имеющей публично доступный сервер? Хотя его серверы могут быть скомпрометированы, код доступен, и отсюда следует, что только вам решать, насколько вы в состоянии ему довериться.

Сервис бесплатен, но ограничен десятью компьютерами на сеть; количество сетей, похоже, не ограничено. Если вы хотите, чтобы в вашей сети было больше десяти пользователей, можете платить \$4 в месяц или запустить свой сервер, что также

означает, что у вас будет полный контроль над вашими данными.

Двоичные пакеты не включают сервер, так что скачайте и распакуйте исходник и запустите

```
$ make ZT_ENABLE_NETWORK_CONTROLLER=1 installer
```

Эта команда скомпилирует программу *zerotier-one* с поддержкой сетевого контроллера и затем соберет пакет установщика, который нужно запустить как обычный установщик. После этого можете командовать

```
$ /var/lib/zerotier-one/zerotier-one -d
```

Запустится сетевой контроллер. Он слушает UDP-порт 9993, поэтому убедитесь, что роутер перенаправляет его на компьютер, работающий в качестве контроллера. Прочтите файл **service/README.md** в дереве исходника для подробностей запуска собственного контроллера. Тогда вы утрачиваете некоторые достоинства *ZeroTier* и получаете кое-какую лишнюю работу, но для более крупных сетей или для сетей, где крайне важна безопасность, эта программа может быть хорошей альтернативой размещенному на своем сервере *OpenVPN*.

Затем обновите web-страницу и подождите несколько секунд, и вы увидите свое устройство в списке устройств.

Однако, запустив `$ sudo zerotier-cli listnetworks` в своем терминале, вы увидите, что доступ к этой сети запрещен. Это часть безопасности *ZeroTier*: пользователь не может присоединиться к вашей сети без вашего согласия. Вернитесь на свою страницу и отметьте окошко **Authorize**. Вскоре страница обновится, чтобы показать IP-адрес для этого устройства, и `zerotier-cli listnetworks` покажет то же самое.

Хотя вы задали имя сети, оно предназначено только для вашего удобства, поскольку для сетей и устройств *ZeroTier* использует идентификаторы. Идентификаторы сети — строки длиной 16 символов, которые мы уже использовали для добавления устройства в сеть. Идентификаторы устройств — строки длиной 10 символов. Их можно увидеть в списке устройств и в выводе команды `$ zerotier-cli info`, которая также сообщит ваш сетевой статус и версию программы. Пока нам приходилось запускать `zerotier-cli` через `sudo` или от имени `root`. Чтобы разрешить ее использование другим, выполните эту команду от имени пользователя, которому хотите дать разрешение:

```
$ sudo cat /var/lib/zerotier-one/authtoken.secret >~/zerotierOneAuthToken
```

Чтобы сделать это для пользователя без прав `sudo`, скопируйте файл **authtoken.secret**, как сделано выше, и смените владельца файла командой `show`, чтобы они принадлежали этому пользователю.

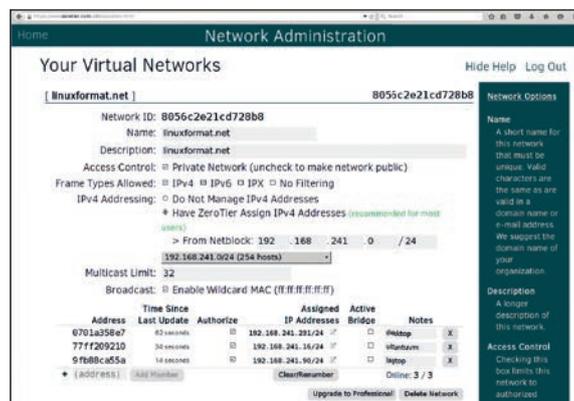
Управление адресами

Вернувшись на страницу администрирования, вы увидите, что вашему устройству выделен IP-адрес из выбранного блока. Желая назначить другой адрес (я удобства ради использую то же конечное число, что и сетевой адрес устройства), щелкните по устройству, чтобы добавить новый адрес и удалить прежний. Можно также задать каждому устройству имя, но это не имя хоста, а лишь ярлык, помогающий идентифицировать устройство. Если вы хотите обращаться к компьютерам по имени, а не адресу, добавьте к **/etc/hosts** по такой строке для каждого из них:

```
192.168.241.231 desktop
```

```
192.168.241.90 laptop
```

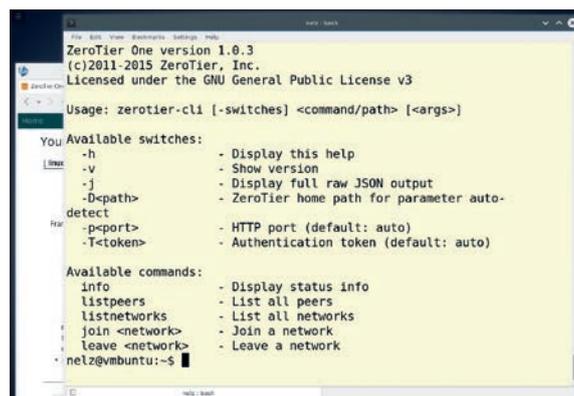
Весь трафик в виртуальной сети идет по кратчайшему маршруту. Подключив к сети свой ноутбук, вы сможете продолжать пользоваться адресами *ZeroTier* для него и других компьютеров, и весь трафик останется внутри вашей локальной сети. Это значит, что вам не придется запускать и останавливать VPN-подключение в зависимости от своего местонахождения — вы можете держать его открытым всегда. Если у вас нет интернет-соединения, интерфейс *ZeroTier* будет ждать до тех пор, пока не сможет подключиться. При использовании *ZeroTier* для подключения по локальной



► Страница администрирования *ZeroTier*, где вы управляете виртуальными сетями. Обратите внимание на удобную подсказку сбоку.

сети производительность может немного снижаться по сравнению с прямым подключением, так как *ZeroTier* шифрует весь трафик.

Разумеется, при доступе к приватной сети извне нужно обратить внимание на вопросы безопасности. Исходный код *ZeroTier* открыт, и это особенно важно для криптографической программы, поскольку вам надо быть уверенными в том, что алгоритмы и код программы безопасны. Безопасность обеспечивается надежным ключом, а не путем усложнения кода — *ZeroTier* использует 256-битное шифрование. Вам ни к чему, чтобы к вашей приватной сети подключался кто попало (в противном случае вы оставили бы галочку **Приватности** снятой), но механизм подключения и авторизации гарантирует, что никто не сможет подключиться к вашей сети в обход страницы администрирования, и вы не можете подключить чужое устройство к своей сети, не обратившись к этому устройству. Для настройки подключения необходим доступ к обоим концам соединения. Надо ли говорить, что для учетной записи *ZeroTier* следует придумать надежный, желательно сгенерированный случайным образом пароль. **LXF**



► Команда `zerotier-cli` проста в использовании разве что с несколькими командами. Помимо случаев подключения/отключения сети, она вряд ли вам понадобится.

Сеть: Основные утилиты Linux

Михалис Цукалос описывает утилиты и методы, необходимые для того, чтобы приступить к решению проблем с сетью.



Наш эксперт

Михалис Цукалос — программист, математик и администратор баз данных и Unix. А когда он не занимается ничем из вышеперечисленного, он — заядлый «любитель сделать что-то новое».



В Linux есть огромное количество утилит командной строки для работы с сетью. Мы расскажем о наиболее важных, которые помогут вам понять и решить проблемы, связанные с сетью. Но общий принцип таков: всегда пользуйтесь той утилитой, которую знаете лучше всего — если, конечно, она подходит для решения вашей проблемы. Обычно сложнее всего понять, в чем именно состоит проблема, а после выявления основной причины решить проблему сравнительно легко.

Не забывайте проверять лог-файлы на наличие ошибок или предупреждений, связанных с вашей проблемой. Это особенно полезно при проблемах на уровне приложения или проблемах с оборудованием. При необходимости повысьте уровень журналирования для получения более подробной информации. Решив проблему, не забудьте понизить этот уровень, так как лог-файлы могут слишком разрастись.

Самая полезная практика — *документировать всё*. Запишите номера портов и настройки TCP/IP всех сетевых устройств, даже компьютеров с Linux и сетевых принтеров, составьте схему их соединения вручную или с помощью утилиты вроде *Graphviz* (www.graphviz.org) и запишите сведения о них. Второй важнейший совет — если у вас есть необходимые ресурсы, создайте стратегию резервного копирования для всего в вашей сети, даже если для этого потребуются дополнительные сетевые устройства и серверы.

Компьютерные сети выходят из строя по многим причинам, но эти причины можно разделить на три основные категории: проблемы с программами, проблемы с устройствами и сочетание того и другого. Проблемы, вовлекающие и устройства, и программы, решить сложнее всего.

Однако самые досадные проблемы связаны с DNS (Domain Name System — система доменных имен). Некорректно работающий сервер DNS может вызвать столько необычных и неожиданных проблем, что вы удивитесь — особенно когда проблемный DNS-сервер получает большой объем трафика и предоставляет информацию DNS многим другим серверам DNS. Например, недавно нам попался неправильно настроенный DNS-сервер, которому удавалось заполнить все дисковое пространство на компьютере Linux, просто генерируя гигабайты сообщений в лог-файлах. Вторая по сложности группа проблем связана с маршрутизацией, поскольку хотя большинство действий могут выполняться нормально, некоторые все же могут не работать, как должны.

Время проверки

Обычно первое, что нужно сделать — попробовать подключиться к удаленному компьютеру. Это поможет выяснить три фундаментальные вещи: (а) затрагивает ли проблема все аналогичные устройства в одной сети, (б) затрагивает ли проблема только конкретное устройство и (в) связана ли проблема с удаленным компьютером — она ведь не всегда на вашей стороне.

Чтобы узнать, могут ли сетевые устройства связаться друг с другом, проще всего воспользоваться утилитой *ping*. Но дело в том, что *ping* использует ICMP-запросы, а сегодня большинство роутеров и брандмауэров блокируют такие запросы к конкретным хостам. Поэтому, даже если запрос *ping* завершается неудачно, вы не можете сказать наверняка, доступно устройство или нет.

Если вы обнаружили, что существует проблема подключения между вашей сетью и удаленным хостом, сузить проблему поможет *traceroute*. Недостаток *traceroute* в том, что она тоже использует пакеты ICMP; символ ***** в выводе команды означает, что роутер на пути прохождения запроса не возвращает сообщения ICMP. Для проверки TCP-подключения также можно воспользоваться *telnet*. Основное преимущество последней — взаимодействие с сервером вручную, в результате которого вам доступны «сырые» данные подключения. К сожалению, с помощью *telnet* можно проверить только подключения TCP.

Если вы хотите проверить соединение UDP, потребуется выбрать другую утилиту, такую как *netcat* (также известна как *nc*), или *Isof*. Команда *Isof -i UDP* покажет все открытые подключения UDP. Аналогично, команда *netcat -vv -u 8.8.8.8 53* проверяет подключения к DNS-серверу 8.8.8.8, который использует UDP. Команда *nc -vz -u 127.0.0.1 1-65535* сканирует и выводит список всех открытых портов UDP на локальном компьютере.

Скорая помощь



Ntop — сетевая утилита, отображающая сведения об использовании сети так же, как *top* делает это для процессов Linux. Также есть *ntopng*, следующее поколение исходной утилиты *ntop*. Обе утилиты очень удобны для исследования аномалий, возникающих в сетевом трафике.

Визуализация сетевых данных

Все системные и сетевые администраторы знают, как важно иметь высокоуровневое представление о сетевом трафике, как для безопасности, так и для мониторинга сети. С ростом потока сетевых данных очень сложно наблюдать за всеми данными все время. Следовательно, нам нужен способ быстро взглянуть на данные сети. Мы построили

график (см. стр. 71 внизу) в R с помощью следующих команд:

```
> timeRel <- read.table("~/Desktop/timeRel.tshark",
header=TRUE)
> plot(timeRel$frame.time_relative, main="Plotting
frame.time_relative field", col="red")
> grid()
```

На графике визуализируется значение `frame.time_relative`, это смещение времени для первого пакета. Это дает хорошее представление о потоке сетевого трафика. Статистика может очень пригодиться, когда приходится работать с большим объемом данных, но это, пожалуй, тема отдельной статьи.

Иногда вам хочется взглянуть на сами сетевые пакеты (см. рис. внизу). Перед исследованием сетевого трафика его нужно перехватить. Лучшие утилиты командной строки для перехвата сетевого трафика — *tcpdump* и *tshark*. Обе утилиты позволяют применять фильтры при захвате трафика, что снижает объем захватываемых данных. Например, *tcpdump host 192.168.2.3* перехватывает трафик, поступающий с IP-адреса `192.168.2.3` или на этот IP-адрес, а *tcpdump port 53* перехватывает только трафик DNS, что очень удобно, когда вы хотите исследовать поведение определенного сервиса.

Утилиты DNS

DNS — это то, что преобразует удобный для человека адрес сайта в IP-адрес. Эта служба так важна, что если она перестанет работать, вы не сможете открыть сайт, получить почту и т.д. Проблемы, связанные с DNS, обычно неприятны, сложны и устраняются медленно, поскольку данные DNS быстро не обновить. Основных утилит для получения информации DNS три: *host*, *nslookup* и *dig*. Все три предоставляют более или менее одинаковую информацию, если указать соответствующие ключи командной строки. *Host* хорошо справляется с простыми задачами, тогда как *dig* лучше подходит для более сложных запросов, а *nslookup* при запуске без аргументов предлагает собственную интерактивную оболочку. Помните, что любой ответ DNS-сервера нужно считать авторитетным, так как у сервера есть полная информация о зоне для домена.

Следующие команды предоставляют одну и ту же информацию DNS-серверов для домена linuxformat.com, запрашивая ее

у публичного DNS-сервера `8.8.8.8`, который запустил Google (пример вывода команд см. на стр. 80, рис. слева):

```
$ host -t ns linuxformat.com 8.8.8.8
$ dig @8.8.8.8 linuxformat.com ns
$ nslookup -query=NS linuxformat.com 8.8.8.8
```

Уменьшить объем информации поможет параметр `+short`. Если разные DNS-серверы предоставляют разную информацию для одного и того же типа запроса, значит, что-то не так. Если на компьютере с Linux не удается получить ответ на DNS-запрос, это означает, что он не настроен должным образом.

Утилита командной строки UNIX, которая считывает и записывает данные для нескольких сетевых соединений — *netcat*, и она использует протоколы TCP или UDP. С ее помощью можно выполнять различные действия, включая создание web- или чат-сервера и сканирование портов. Для большинства параметров *netcat* привилегии `root` не нужны; это в основном зависит от номера порта при создании сервера TCP/IP, так как порты 0–1024 доступны только пользователю `root`. Поведение *netcat* по умолчанию похоже на имитацию сетевого взаимодействия, которая используется *telnet*, и это означает две вещи: сетевое подключение не шифруется, и по умолчанию используется протокол TCP.

Команда *nc machine_name_or_IP 25* заставляет *netcat* взаимодействовать с сервером SMTP, при условии, что сервер SMTP использует порт SMTP по умолчанию. Команда *netcat -l 1234* заставляет *netcat* слушать порт 1234 на предмет входящих подключений TCP; следовательно, она выступает как серверный процесс. Обе операции удобны для проверки, способны ли два компьютера



WireShark — лучшая утилита для анализа сетевого трафика, которой также можно воспользоваться для перехвата трафика. Ее основной недостаток — графический интерфейс; поэтому программой нельзя воспользоваться удаленно. Однако она очень удобна при работе со сложными сетевыми данными. Если у вас есть время только на одну утилиту, изучите *WireShark*. См. [LXF191/192](#).



» На этой схеме показан формат TCP- и IP-пакетов. Да, Алиса, глубока кроличья нора.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

Скорая помощь

Наличие нескольких свободных кабелей и второго роутера ADSL или неиспользуемого коммутатора бывает бесценным для устранения неполадок. Ноутбук или планшет со всеми установленными утилитами может помочь вам, если что-то пошло не так. Наконец, вы должны научиться устранять неполадки с помощью ОС вашего роутера, потому что роутер находится на более низком уровне и может предоставить вам другой взгляд на ситуацию.

взаимодействовать друг с другом. Если способны, это обычно означает, что аппаратной проблемы нет и основная сеть работает хорошо. Тогда проверьте брандмауэр или другие процессы. Это легко сделать, взглянув на сетевой трафик.

Утилита *lsof* используется для вывода списка открытых файлов, и так как каждое сетевое устройство — это файл, с помощью данной утилиты также можно получить и информацию о сети. Для запуска *lsof* потребуются привилегии администратора, из-за необходимости прав доступа к сетевым устройствам.

Команда *lsof -i4* выведет информацию о подключениях IPv4, а *lsof -i6* — только о подключениях IPv6. Команда *lsof -u www-data | grep -i ESTABL* выведет список всех установленных подключений, владельцем которых является пользователь *www-data*, и с помощью этой команды удобно устранять неполадки некорректно работающего web-сервера. Команда *lsof -nP -iTCP -sTCP:LISTEN* показывает, какая программа слушает порт TCP, и владельца процесса.

Для вывода информации о сетевой подсистеме Linux пригодится *netstat*. Это утилита, которая отображает информацию о состоянии сети, не требуя привилегий администратора. Она очень мощная и работает на уровне сокетов, TCP, UDP, IP и Ethernet. Ее основной недостаток — работа только на локальном компьютере, тогда как такие утилиты, как *ntop* и *tcpdump*, умеют также отображать информацию о других компьютерах. Команда *netstat -r* отображает информацию о маршрутизации с локального компьютера; неправильно настроенная маршрутизация — частая причина проблем с сетью, особенно на компьютерах с несколькими сетевыми интерфейсами.

Иногда вам хочется проверить не только текущий трафик, но и более старый сетевой трафик. Это сложно, потому что надо найти способ хранить более старые данные. Лучший вариант — воспользоваться базой данных, потому что она бесплатно предоставляет вам язык для опроса данных. Следующая команда

```
mysql> select count(protocol), protocol FROM netData GROUP BY protocol limit 3;
+-----+-----+
| count(protocol) | protocol |
+-----+-----+
| 11 | DATA-TEXT-LINES |
| 19 | DNS |
| 3106 | |
+-----+-----+
3 rows in set (0.00 sec)

mysql> select count(*) as TOTAL, SourceIP from netData group by sourceIP order by TOTAL desc limit 10;
+-----+-----+
| TOTAL | SourceIP |
+-----+-----+
| 2898 | 192.168.1.100 |
| 996 | 192.168.1.101 |
| 43 | 192.168.1.102 |
| 38 | 192.168.1.103 |
| 33 | 192.168.1.104 |
| 31 | 192.168.1.105 |
| 27 | 192.168.1.106 |
| 26 | 192.168.1.107 |
| 24 | 192.168.1.108 |
| 23 | 192.168.1.109 |
+-----+-----+
10 rows in set (0.00 sec)

mysql> select COUNT(*), protocol, avg(length) from netData GROUP BY protocol;
+-----+-----+-----+
| COUNT(*) | protocol | avg(length) |
+-----+-----+-----+
| 11 | DATA-TEXT-LINES | 0.0000 |
| 19 | DNS | 1429.6316 |
| 3106 | | 107.6700 |
+-----+-----+-----+
```

➤ База данных — отличное место для хранения сетевой информации. SQL упростит вам жизнь, так как позволяет искать данные с помощью сложных запросов.

прочтет файл *data.tcpdump* с трафиком, полученным утилитой *tcpdump*, и извлечет требуемую нам информацию в удобном формате, которую можно добавить в базу данных с помощью скрипта:

```
$ tshark -r data.tcpdump -T fields -e frame.number -e frame.time_relative -e ip.src -e ip.dst -e frame.protocols -e frame.len -E header=y -E quote=n -E occurrence=f
frame.number frame.time_relative ip.src ip.dst frame.protocols
frame.len
1 0.000000000 82.98.136.22 109.74.193.253
eth:ethertype:ip:udp:dns 108
```

Вы видите результаты трех запросов (на рисунке вверху), выполненных в базе данных *MySQL*, которая содержит сетевые данные. Первый запрос показывает соединения для протокола. Параметр *limit 3* ограничивает количества выходных строк. Второй запрос находит 10 самых популярных IP-адресов, а третий выводит среднюю длину пакета для каждого протокола и общее количество пакетов.

SQL — мощный язык для создания полезных вопросов, которые помогут найти исторические сведения о сетевых данных. Например, запрос может обнаружить, что проблема подключения клиентов к серверу была связана с большим трафиком в конкретный момент времени. Точно так же можно выяснить, какие web-серверы обורвали подключения из-за большого трафика, и добавить им оперативной памяти или установить более быстрый жесткий диск.

В реальном мире

Давайте пройдем по двум вероятным сценариям реального мира и рассмотрим процесс решения проблем. Первая: в сети установлен новый роутер ADSL для подключения сети к Интернету. В сети используются хаб и коммутатор. Сначала мы проверяем Wi-Fi-подключение к роутеру, и оно работает отлично, позволяя устройствам выйти в Интернет. Однако если хаб подключен к одному из портов роутера ADSL, то ни у кого не остается доступа к сети и к Интернету, даже с рабочим подключением Wi-Fi! Мы решаем проблему следующим образом:

➤ Отключите все кабели Ethernet от роутера и подключите ноутбук к роутеру с помощью Wi-Fi и DHCP. Проверьте Wi-Fi с помощью *ping* и *traceroute*. В нашем случае они выполняются успешно, и запрос к web-интерфейсу роутера с помощью *wget* также выполняется успешно.

➤ Подключите компьютер к одному из портов коммутатора кабелем Ethernet. Проверьте, что компьютер получил свои настройки TCP/IP через DHCP; в этом случае компьютер работает отлично. Это доказывает, что проблема связана с одним из портов концентратора.

```
mtsouk@mail:~$ dig 8.8.8.8 linuxformat.com ns
;<<> DiG 9.9.5-9-Debian <> 8.8.8.8 linuxformat.com ns
(1 server found)
;; global options: <cmd>
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 19266
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 512
;; QUESTION SECTION:
; linuxformat.com. IN NS

;; ANSWER SECTION:
linuxformat.com. 236 IN NS ns0.future.net.uk.
linuxformat.com. 236 IN NS ns1.future.net.uk.

;; Query time: 9 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Jul 03 22:32:24 EEST 2015
;; MSG SIZE rcvd: 93

mtsouk@mail:~$ host -t ns linuxformat.com 8.8.8.8
Using domain server:
Name: 8.8.8.8
Address: 8.8.8.8#53
Aliases:

linuxformat.com name server ns0.future.net.uk.
linuxformat.com name server ns1.future.net.uk.
mtsouk@mail:~$ nslookup -query=NS linuxformat.com 8.8.8.8
Server: 8.8.8.8
Address: 8.8.8.8#53
```

➤ Пользуйтесь *nslookup*, *host* и *dig*, чтобы найти информацию о серверах DNS для домена *linuxformat.com*. Какую из этих утилит вы выберете, дело личного вкуса.

Nmap

Nmap — открытая утилита, созданная Гордоном Федором Лайоном (Gordon Fyodor Lyon). Она поддерживает сканирование портов, определение ОС и ее версии.

Хотя обычные пользователи могут выполнять различные операции сканирования с *Nmap*, некоторые параметры команд требуют привилегий *root*. *Nmap* поможет вам определить тип сетевого устройства

и сервисы, которые на нем запущены. Также можно проверить, сможете ли вы обмениваться данными с удаленным устройством. Если вы хотите полинговать несколько устройств одновременно, с помощью *Nmap* это сделать легко: *nmap -sP 192.168.1.0/24*. Команда *nmap -sU aMachine* также выполнит сканирование порта UDP на компьютере с именем *aMachine*.

О маршрутизации

Маршрутизация — процесс выбора пути для отправки пакетов. Специализированные сетевые устройства обычно обращаются к роутерам, которые выполняют большую часть работы. Однако даже компьютер может выступать как роутер, и чем сложнее топология сети, тем труднее найти оптимальный путь. Как и следовало ожидать, Интернет — самая большая сеть из всех.

Таблица маршрутизации — место, где добавляются все маршруты. Маршрут — это (более или менее) правило, которое соответствует одному сетевому адресу и определяет адрес следующего роутера. Если устройство на выбранном пути неисправно, TCP/IP и протоколы маршрутизации попытаются найти другой путь к месту назначения. Если IP-адрес соответствует нескольким маршрутам, вызывается набор предопределенных правил, называемых алгоритмом маршрутизации, который разрешает неоднозначность.

Статическая маршрутизация предполагает, что администратор сети вводит все необходимые правила в таблицу маршрутизации вручную. В динамической (или адаптивной) маршрутизации необходимые маршруты обнаруживаются особыми протоколами динамической маршрутизации. Эти протоколы обмениваются особыми пакетами, называемыми обновлениями маршрутизации, которые добавляются в таблицу маршрутизации. Ваш дистрибутив Linux может использовать несколько сетевых интерфейсов (Ethernet, Wi-Fi и т.д.), но только у одного из них будет значение шлюза по умолчанию. Самый простой способ просмотра информации маршрутизации для вашего компьютера Linux — выполнить команду `netstat -nr`.

Самая популярная компания, которая производит роутеры — Cisco. Операционная система Cisco называется Cisco IOS, а справа вы увидите таблицу маршрутизации роутера ADSL Cisco 877W.

```
2. mtsouk@mail: ~ (bash)
mtsouk$ ssh linode netstat -nr
ing table
  Gateway         Genmask          Flags   MSS Window  irtt If
  109.74.193.1    0.0.0.0          UG      0  0          0 et
  0.0.0.0         255.255.255.0   U        0  0          0 et
mtsouk$ ssh cwifi show ip route
connected, S - static, R - RIP, M - mobile, B - BGP
RP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
PF NSSA external type 1, N2 - OSPF NSSA external type 2
PF external type 1, E2 - OSPF external type 2
IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2,
-IS inter area, * - candidate default, U - per-user static route,
, P - periodic downloaded static route

t resort is 0.0.0.0 to network 0.0.0.0

2 is subnetted, 1 subnets
8.149 is directly connected, Dialer1
32 is subnetted, 1 subnets
.108.71 is directly connected, Dialer1
.0/24 is directly connected, Dot11Radio0
is directly connected, Dialer1MacBook:SRE mtsouk$
```

➤ Это таблица маршрутизации сервера Linux и роутера Cisco 877W. По выводу команды понятно, насколько сложнее роутер Cisco.

» Отключите все кабели от нерабочего хаба и подключите «пустой» хаб к роутеру. Затем подключайте устройства Ethernet к хабу, пока не попадется устройство, вызывающее проблемы — это корень проблемы!

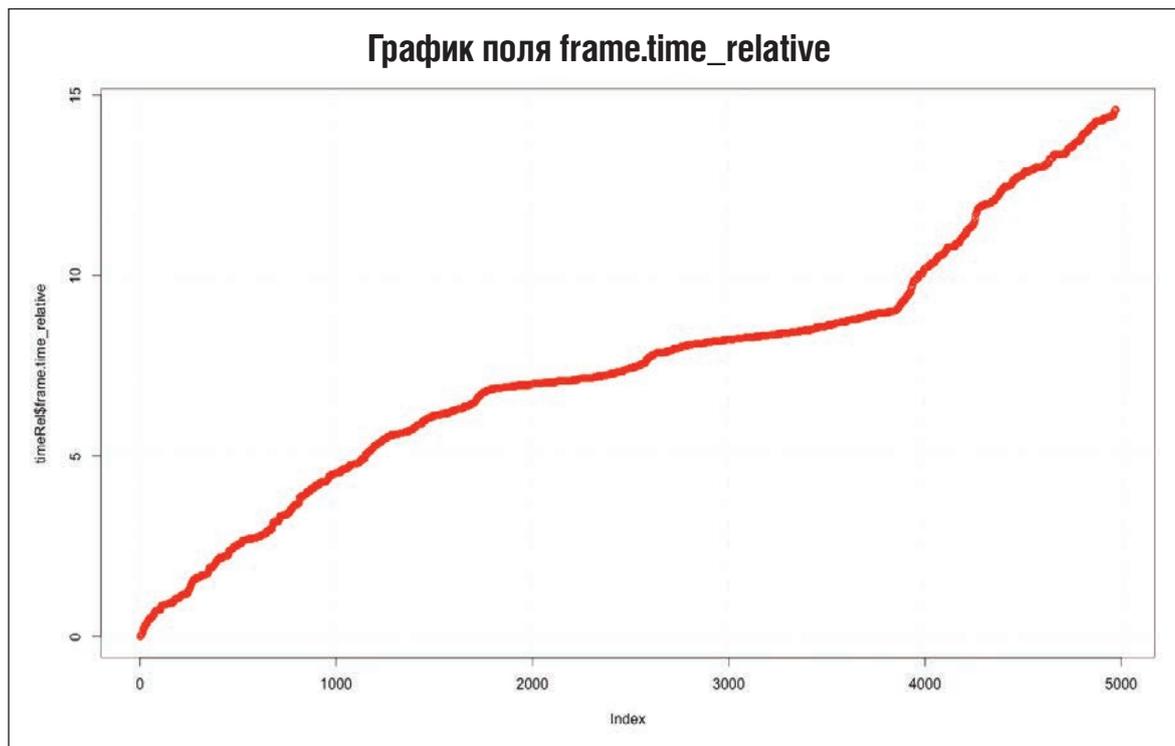
» Последний шаг — перезагрузить роутер и повторить весь процесс без использования нерабочего порта хаба. Если все работает, проблема решена!

Второй сценарий проще: компьютер Linux настроен на получение сообщений журнала от роутера Cisco, а роутер настроен на отправку сообщений журнала на компьютер Linux, но в файлах журнала на компьютере Linux ничего нет.

Компьютер Linux использует для получения сообщений сервис rsyslog. Обычно между получением данных rsyslog и их записью в файл журнала есть временная задержка.

Так как rsyslog представляет собой службу UDP, которая слушает порт 514, мы можем воспользоваться командой `tcpdump 'udp port 514'`, чтобы проверить, приходит ли на компьютер Linux какой-либо сетевой трафик от службы. Если трафик есть, он также записывается в файл журнала, что доказывает отсутствие проблем.

То же самое сделает запуск *WireShark* на удаленном компьютере, но это не всегда возможно, поэтому не стоит недооценивать утилиты командной строки! **LXF**



➤ На графике, полученном с помощью R, представлены значения `frame.time_relative`.

IP-телефония: Бэкап — это всегда умно

Прежде чем пускаться во все тяжкие, Максим Черепанов подкладывает соломки в местах возможного падения.



Наш эксперт

Максим Черепанов — заместитель директора в небольшой коммерческой фирме. По образованию железнодорожник-управленец. Linux занимается с 2008 г. Чем дольше это делает, тем дальше от Windows. За эти годы научился чистить реестр и забыл слово «антивирус».

Итак (см. Учебники LXF201), мы имеем девственно чистую систему на карточке, с доступом по SSH, с пользователем, обладающим административными привилегиями, с выходом в Интернет (кстати, вы проверили это? Если есть SSH, но нет Интернета, проверьте файл `/etc/resolv.conf`, там должен быть адрес вашего DNS-сервера или DNS-провайдера).

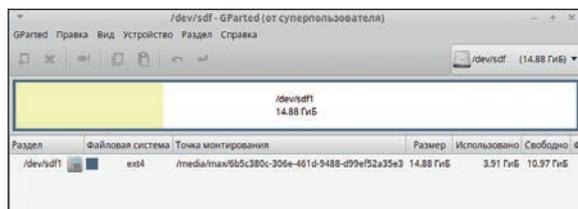
Самое время сказать: «Надо ставить Asterisk!» Но я пока этого не скажу, а предложу вам подложить себе несколько словесных соломок: давайте «полуавтоматизируем» процесс изготовления резервных копий [backup]. Зачем? А затем, что все пользователи компьютеров делятся на две категории: те, кто уже пережил отказ жесткого диска, и те, кому это еще предстоит. Карта памяти в этом отношении еще хуже, несмотря на то, что в ней нет движущихся частей. И потому мы можем только предвосхитить отказ и минимизировать его последствия, но не предотвратить.

Займемся же делом. Выключим Кубик, извлечем карту памяти, вставим ее в картридер вашего ПК. Запустим *GParted* (как, у вас его еще нет? Немедленно поставьте! Для Mint'оводов — `sudo apt install gparted`, остальные гуглят и делают аналогичное), ищем свою карту (у меня она опять `/dev/sdf`) и видим нечто вроде изображенного на рис. 1.

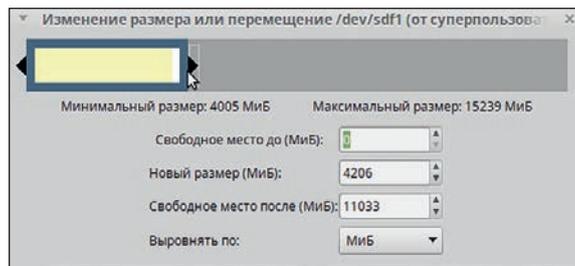
Всё верно, при первом запуске умный скрипт раздвинул границу раздела до упора, чтобы у системы было свободное место. Но бэкапить пустоту — не путь джедая. Выделяем раздел, идем в Раздел > Отмонтировать, снова его выделяем — и в Раздел > Изменить размер/Переместить (рис. 2), хватаемся мышью за правый край раздела и тянем его влево до упора. Уперлись в границу занятого места? Отмотайте обратно 250–300 МБ и так и оставьте.

Если вы будете распаковывать бэкап и забудете его растянуть, то система хотя бы запустится. Жмем Изменить раздел или переместить и применяем все операции (рис. 3).

Ждем, пока обрабатывает *GParted*, и два раза щелкаем по новому разделу. Запишите число после слов «Последний сектор»;



» Рис. 1. Стартовое окно *GParted*.



» Рис. 2. Меняем размер раздела.

у меня это 8615935 (я делаю сейчас бэкап с уже настроенной системы). Закроем *GParted*; мы готовы делать образ. Считаем блоки для `dcfldd`: писать будем блоком по 1024 байта, а размер сектора на карте 512 байт. Поэтому просто разделите число на два: $8615935/2=4307967,5$. Округляем до большего целого, и именно это число впишем в скрипт:

```
# dcfldd bs=1024 count=4307968 if=/dev/sdf | gzip -c > /home/user/cubie_aster-$(date +%H:%M-%d_%B_20%y).img.gz
```

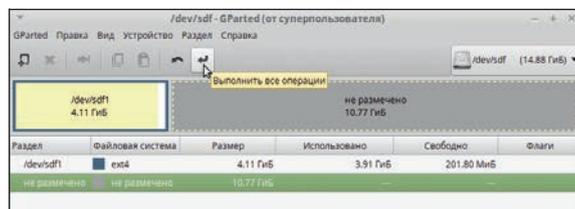
Дождитесь, пока закончится копирование, и после этого вы получите в домашней директории упакованный образ вашей системы с пометкой, когда он был создан. Хотите — добавьте свои примечания.

Если вам надо вернуться на какой-либо из образов (активировать соломку), то точно так же вставьте карту, отмонтируйте раздел и залейте на него образ из архива:

```
# gunzip -c /home/user/cubie_aster_10:28-07_Октябрь_2015.img.gz | dcfldd bs=1024 of=/dev/sdf
```

Как видно, образ создавался 7 октября 2015 года в 10:28, и понятно, что это Asterisk для Кубика.

После заливки образа откройте *GParted* и растяните границу раздела до упора вправо.



» Рис. 3. Сохраняем изменения.

Я специально остановился так подробно на процессах копирования и разворачивания образа. Не вздумайте считать себя идеальными и терпеть себя мыслью, что уж вы-то ошибок не сделаете. А некоторые ошибки бывают такими, что система просто перестает загружаться. Тогда вас и спасет соломка, которую надо стелить каждый раз, когда вы делаете серьезные изменения в системе. Лишние образы всегда можно стереть; и по нынешним объемам HDD они совсем не велики. Кроме того, данный способ пригоден для любых загрузочных и незагрузочных устройств: флешек, жестких дисков, при желании его можно использовать и с лазерными дисками; но это уже тема отдельного разговора. А вот теперь...

Устанавливаем Asterisk

Итак, карточка в Кубике, он запущен и подключен к Интернету. Выполняем:

```
# apt-get install -y build-essential apache2 mysql-server mysql-client bison flex php5 php5-curl php5-cli php5-mysql php-pear php-db php5-gd curl sox libncurses5-dev libssl-dev libmysqlclient-dev mpg123 libxml2-dev libnewt-dev sqlite3 libsqlite3-dev pkg-config automake libtool autoconf git subversion unixodbc-dev uuid uuid-dev libasound2-dev libogg-dev libvorbis-dev libcurl4-openssl-dev libical-dev libneon27-dev libsrtp0-dev libspandsp-dev libiksemel-dev libiksemel-utils libiksemel3 && pear uninstall db && pear install db-1.7.14 && reboot
```

Кратко: обновляем систему, ставим все зависимости, которые нужны для Asterisk'a и web-GUI, при установке MySQL задаем пароль (и запоминаем; я ввел для примера 123), меняем версию БД pear на стабильную (пока что в репозитории она и есть, но всё может измениться) и перезагружаем систему. Замечу, что заголовки ядра мы не ставим, уважаемый Igor Ребовник включает их сразу в сборки. Качаем исходники:

```
# cd /usr/src && wget http://downloads.asterisk.org/pub/telephony/dahti-linux-complete/dahti-linux-complete-current.tar.gz && wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-11-current.tar.gz && git clone https://github.com/akheron/jansson.git
Собираем dahti и jansson:
```

```
# tar xvfz dahti-linux-complete-current.tar.gz && cd dahti-linux-complete-* && make all && make install && make config && cd /usr/src/jansson && autoreconf -i && ./configure && make && make install
```

Собираем Asterisk:

```
# cd /usr/src && tar xvfz asterisk-11-current.tar.gz && cd asterisk-11.* && ./configure && contrib/scripts/get_mp3_source.sh && make menuselect
```

В появившемся псевдографическом меню (рис. 4) выбираем те пункты, что указаны, и жмем Save & Exit.

Это мы поставили зависимости, собрали и установили необходимые программы, выбрали для компиляции модули. Устанавливаем собственно Asterisk:

```
# make && make install && make config && ldconfig
```

Процесс не быстрый, тем более на Кубике: примерно 40 минут. Заварите себе чай, проверьте почту, послушайте немного музыки — дайте отдых голове. Дождавшись приглашения ОС, продолжим. Установим дополнительные звуки:

```
# mkdir -p /var/lib/asterisk/sounds && cd /var/lib/asterisk/sounds && wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-extra-sounds-en-wav-current.tar.gz && tar xzf asterisk-extra-sounds-en-wav-current.tar.gz && rm -f asterisk-extra-sounds-en-wav-current.tar.gz && wget http://downloads.asterisk.org/pub/telephony/sounds/asterisk-extra-sounds-en-g722-current.tar.gz && tar xzf asterisk-extra-sounds-en-g722-current.tar.gz && rm -f asterisk-extra-sounds-en-g722-current.tar.gz
```

Скачаем web-GUI, в нашем случае Free-PBX:

```
# cd /usr/src && wget http://mirror.freepbx.org/modules/packages/freepbx/freepbx-12.0-latest.tgz && tar xvfz freepbx-12.0-latest.tgz
```

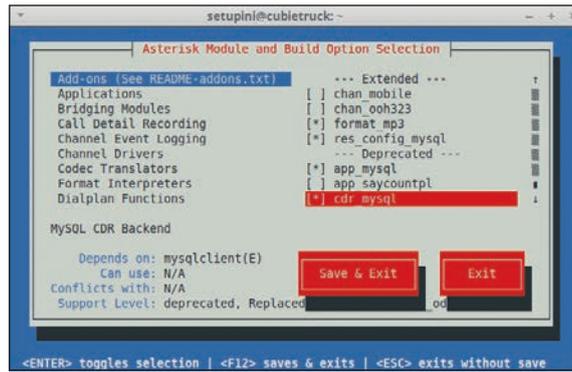


Рис. 4. Меню настроек компиляции Asterisk.

Создаем нового пользователя Asterisk:

```
# useradd -m asterisk && chown asterisk /var/run/asterisk && chown -R asterisk /etc/asterisk && chown -R asterisk /var/{lib,log,spool}/asterisk && chown -R asterisk /usr/lib/asterisk && rm -rf /var/www/html
```

Вносим изменения в конфиг Apache:

```
# sed -i 's/^(upload_max_filesize =).*$/^120M/' /etc/php5/apache2/php.ini && cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf_orig && sed -i 's/^(User|Group).*$/^1 asterisk/' /etc/apache2/apache2.conf && service apache2 restart
```

Настраиваем базу данных MySQL (там, где означено 123, подставьте свой пароль от MySQL!):

```
# mysqladmin -u root -p create asterisk && mysqladmin -u root -p create asteriskcdrdb && mysql -u root -p -e "GRANT ALL PRIVILEGES ON asterisk.* TO asteriskuser@localhost IDENTIFIED BY '123';" && mysql -u root -p -e "GRANT ALL PRIVILEGES ON asteriskcdrdb.* TO asteriskuser@localhost IDENTIFIED BY '123';" && mysql -u root -p -e "flush privileges;"
```

Вводите пароль от MySQL еще 5 раз.

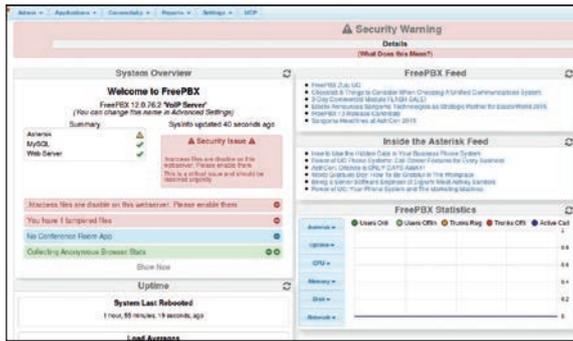
Всё готово для сборки и установки FreePBX, благо есть установочный скрипт, но для установки должен быть запущен Asterisk (про 123 помните?):

```
/usr/src/freepbx/start_asterisk start && cd /usr/src/freepbx && ./install_amp --installdb --username=asteriskuser --password=123
```

Отвечаем на заданные вопросы, если это можно так назвать: категорически не рекомендую ничего менять, кроме адреса сервера, в нашем случае это 192.168.0.100. Я неоднократно наступал на эти грабли, думал — «А дай-ка я сразу задам пользователя интерфейса здесь, чтобы потом не менять». Не стоит. Потом замучаетесь ручками в конфигурационных файлах всё выправлять. Установочный скрипт всё возьмет оттуда, откуда надо, чтобы всё работало:

```
Enter your USERNAME to connect to the 'asterisk' database:
[asteriskuser]
Enter your PASSWORD to connect to the 'asterisk' database:
[123]
Enter the hostname of the 'asterisk' database:
[localhost]
Enter a USERNAME to connect to the Asterisk Manager interface:
[admin]
Enter a PASSWORD to connect to the Asterisk Manager interface:
[amp111]
Enter the path to use for your AMP web root:
[/var/www/html]
Created /var/www/html
Enter the IP ADDRESS or hostname used to access the AMP web-admin:
[192.168.1.1] 192.168.0.100
Use simple Extensions [extensions] admin or separate Devices and Users [deviceanduser]?
[extensions]
Enter directory in which to store AMP executable scripts:
»
```

► Рис. 5. Стартовое окно FreePBX.



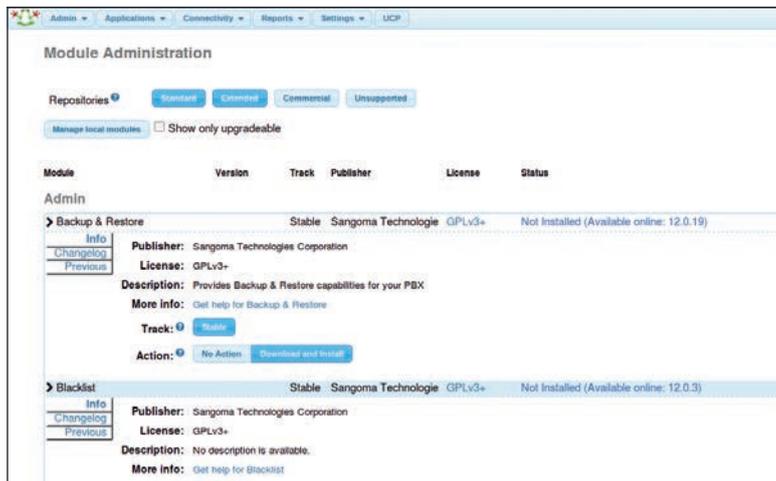
```
[/var/lib/asterisk/bin]
Created /var/lib/asterisk/bin
Enter directory in which to store super-user scripts:
[/usr/local/sbin]
/etc/amportal.conf written Assuming new install, --install-moh
added to command line
OK
Reading /etc/amportal.conf..parsed amp_conf variables from /etc/
amportal.conf:
amp_conf [AMPDBHOST] => [localhost]
amp_conf [AMPDBENGINE] => [mysql]
amp_conf [AMPENGINE] => [asterisk]
amp_conf [AMPMGRUSER] => [admin]
amp_conf [AMPMGRPASS] => [amp111]
amp_conf [AMPBIN] => [/var/lib/asterisk/bin]
amp_conf [AMPSBIN] => [/usr/local/sbin]
amp_conf [AMPWEBROOT] => [/var/www/html]
amp_conf [AMPCGIBIN] => [/var/www/cgi-bin ]
amp_conf [FOPWEBROOT] => [/var/www/html/panel]
amp_conf [FOPPASSWORD] => [passw0rd]
amp_conf [AUTHTYPE] => [database]
amp_conf [AMPEXTENSIONS] => [extensions]
amp_conf [AMPDBUSER] => [asteriskuser]
amp_conf [AMPDBPASS] => [123]
amp_conf [AMPWEBADDRESS] => [192.168.0.100]
amp_conf [AMPDBNAME] => [asterisk]
OK
```

Опять немного ждем, около 5 минут. Если видим

```
*****
* At This Time Please Restart Asterisk *
[Теперь, пожалуйста, перезагрузите Asterisk]
*****
```

то всё прошло удачно. Если нет — проверяем, что сделали не так, и исправляем.

► Рис. 6. Загрузка модулей FreePBX.



```
# amportal chown
```

Затем:

```
# amportal a ma installall
```

Какое-то время модули будут качаться и распаковываться, затем появится ошибка:

```
Module arimanager successfully downloaded
```

```
Unable to install module arimanager:
```

```
- Requires engine asterisk (>= 12), you have: asterisk 11.19.0
```

Всё верно, эта версия модуля требует Asterisk от 12-й версии и выше, а у нас 11-я. Это сделано намеренно, потому что в 11-й версии Asterisk порт SIP установлен в 5060, а в последующих в 5061, а 5060 отдан другому драйверу — PJSIP (более новая, чем SIP, библиотека мультимедиа с открытым кодом). Для наших нужд она без надобности. Будем исправлять ошибку?

Создадим ссылку для MP3:

```
# In -s /var/lib/asterisk/moh /var/lib/asterisk/mohmp3
```

Перезагрузим Free-PBX:

```
# amportal restart
```

Не закрывая терминал, зайдите в web-панель управления (в нашем случае, по адресу 192.168.0.100/html), задать пользователя для web-GUI, нажать красную кнопку Apply config вверху справа и полностью выйти из Free-PBX (из пользователя, по синей кнопке Logout вверху справа), закрыть web-панель в браузере и затем выполнить в консоли в директории /usr/src/freepbx поочередно такие команды:

```
# amportal chown
```

```
# amportal a ma refreshsignatures
```

```
# amportal restart
```

После всех этих манипуляций заходим снова на 192.168.0.100/html в вашем браузере, вводим логин-пароль и видим такую вот прекрасную картинку, как на рис. 5. Поздравляю! Ваша персональная самосборная экономичная АТС готова. До окончательной базовой настройки осталось немного. Соберем остатки сил и приступим:

Кнопочка Admin > Module admin, жмем Check Online, после проверки появятся кнопки, Standard уже отмечена. Выбираем еще Extended, нам достаточно. И жмем по очереди Download all и Upgrade all, затем Process и в следующем окне Confirm. Оболочка начнет подгружать свеженькие модули; для этого тоже требуется время, зависимое от скорости вашего Интернета и скорости системы (если вдруг вы ставите не на Кубик — инструкция по установке достаточно универсальна).

Обратите внимание, что некоторые модули требуют в виде зависимости другие модули, которые еще не подгружены. Из-за этого не всё встанет сразу. Кошерно — посмотреть по зависимостям, кто должен быть подгружен первым. Некошерно — обновляться повторно, пока не встанут все, обязательно в промежутках нажимая красную кнопку Apply config. Вот так мы и поступим, благо всё это надо будет сделать всего 2 раза. В последний прием загрузятся всего 2 модуля. Когда система ответит вам, что нет модулей для загрузки — всё готово.

Самое время создать двух пользователей и позвонить — я уверен, что вам не терпится это сделать! Это своеобразный “Hello, world!” для Asterisk’a, хоть на профильных форумах и не любят отвечать на вопрос «Как настроить Asterisk и просто позвонить с одного компьютера на другой?»

Итак, в web-GUI Application > Extention (первый раз вас вывалитесь табличка с предложением установить “SIPStation Free Trial” — отказывайтесь! Повесит всю систему: это не для Кубика), и создаем новое устройство Generic CHAN SIP device (рис. 7).

Пишем в первых двух окошках что-то произвольное, типа 1003, ищем ниже окошко Secret, это пароль. Для теста сойдет и test1003. Больше ничего не меняем и жмем в самом низу кнопку Submit, а после создания пользователя — Apply config. Вообще приучайтесь нажимать ее всегда, когда она появляется. Таким же способом создаем пользователя 1004. Хочу успокоить адептов безопасности:

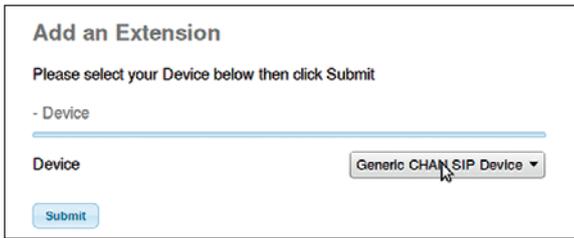


Рис. 7. Создание пользователя (устройства).

это всё только для теста. Потом их, конечно, надо удалить или задать пароль посерьезнее.

Настало время поговорить о клиентах, коими могут являться большое количество физических устройств и программ. Первые — это софтофоны, они позволят вам общаться с компьютера, имеющего микрофон и динамик. Вторые — SIP-телефоны: внешне — обычные телефонные аппараты, но подключаемые в компьютерную сеть. Третьи — VoIP-шлюзы, позволяющие подключить обычный аналоговый телефонный аппарат. Четвертые — программы на смартфонах. У каждого варианта свои достоинства и недостатки. Давайте просто настроим пару клиентов для теста нашей АТС.

К сожалению, вынужден признать, что лучшим (на мой субъективный взгляд) софтофоном является бесплатный клиент для Windows *3CXPhone*; в настоящее время я пользуюсь 6-й версией. Давайте и вы попробуете его — это ни в коем случае не реклама, есть много других вариантов, я их тоже пробовал. Но цель нашего урока — не разбор софтофонов.

Итак, ставим на ПК с Windows упомянутый клиент, но пока его не запускаем. В консоли нашего сервера от имени суперпользователя набираем

```
# asterisk -rvvvv
```

Система ответит:

```
Asterisk 11.20.0, Copyright (C) 1999 - 2013 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
```

```
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core
show warranty' for details.
```

```
This is free software, with components licensed under the GNU
General Public
```

```
License version 2 and other licenses; you are welcome to
redistribute it under
```

```
certain conditions. Type 'core show license' for details.
```

```
=====
Connected to Asterisk 11.20.0 currently running on cubietruck (pid
= 6971)
```

```
cubietruck*CLI>
```

Вы вошли в консоль управления *Asterisk*. Здесь существуют свои правила, свои команды, но важно то, что сюда выводятся системные сообщения.

А вот теперь запустите *3CXPhone* и настройте его по скриншоту (рис. 8), предварительно указав в настройках русский язык (возможность такая есть, и это очень приятно). То есть вам надо просто ввести в настройки данные о том пользователе, который создан вами в вашей АТС.

Нажимаем ОК и смотрим в окно вашего терминала *Asterisk*:

```
-- Registered SIP '1003' at 10.0.2.15:1041
```

```
> Saved useragent "3CXPhone 6.0.18815.0" for peer 1003
```

```
-- Unregistered SIP '1003'
```

```
-- Registered SIP '1003' at 10.0.2.15:38634
```

```
[2015-10-15 16:11:17] NOTICE[6987]: chan_sip.c:28112 handle_
request_subscribe: Received SIP subscribe for peer without
mailbox: 1003
```

```
[2015-10-15 16:11:20] NOTICE[6987]: chan_sip.c:29745 sip_poke_
noanswer: Peer '1003' is now UNREACHABLE! Last qualify: 0
```



Рис. 8. Настройка 3CXPhone.

Не обращайте внимания на IP 10.0.2.15 — у меня Windows для теста запущена на виртуальной машине за NAT'ом.

В нашем родном Linux'е мы поставим *Linphone* (<http://www.linphone.org/>), всё как обычно, он есть в репозиториях всех дистрибутивов, которыми я пользуюсь:

```
$ sudo apt install linphone
```

Позволю себе перестать сомневаться в вашей компетентности и дам возможность настроить его самостоятельно, тем более что это очень-очень просто; но для подсказки покажу скриншот настройки тестовой учетной записи — на рис. 9.

При нажатии ОК смотрим в консоль *Asterisk*:

```
-- Registered SIP '1004' at 192.168.0.6:5060
```

```
[2015-10-15 16:35:43] NOTICE[6987]: chan_sip.c:23739 handle_
response_peerpoke: Peer '1004' is now Reachable. (27ms / 2000ms)
cubietruck*CLI>
```

192.168.0.6 — адрес моего основного ПК, на котором я это пишу.

А теперь, наконец, звоним! Но только имейте в виду, что в описываемом примере я смогу позвонить только на реальный IP: 192.168.0.6, а клиент Windows, поскольку находится за NAT, может только звонить, но не принимать звонки.

Итак, набираем в *3CXPhone* (с абонента 1003) номер 1004; в *Linphone* начинает звенеть звоночек, и в консоли АТС появляется простыня сообщений. Нам всего интереснее последние три:

```
-- Called SIP/1004
```

```
-- Connected line update to SIP/1003-00000013 prevented.
```

```
-- SIP/1004-00000014 is ringing
```

```
cubietruck*CLI>
```

Вы просто молодцы сегодня! У вас всё получилось, но впереди еще ждет невероятно много любопытного. LXF

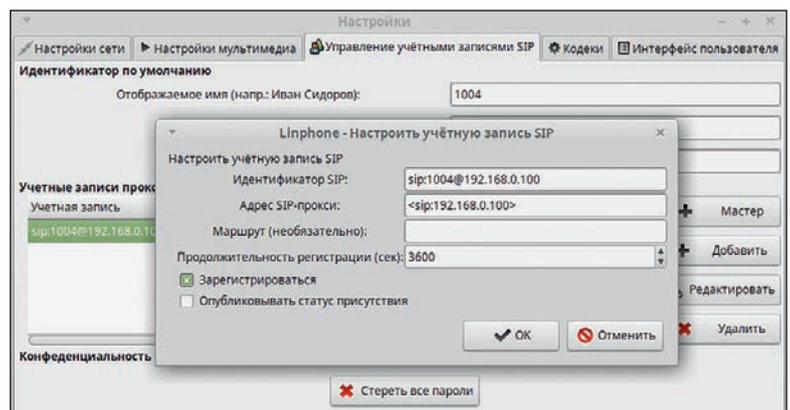


Рис. 9. Настройка Linphone.

M4: Освоим макропроцессор

Дмитрий Кузнецов — ярый сторонник автоматизации: без нее — чего стоят достижения современного прогресса?



Наш эксперт

Дмитрий Кузнецов более 10 лет ведет неравный бой с монстрами сложности, хотя некоторые почему-то считают их ветряными мельницами.

Письменность — основа человеческой цивилизации. Благодаря ей люди знают об открытиях, произошедших сотни лет назад, а каждое последующее поколение может использовать и развивать достижения предыдущих. Однако рисовать тысячи витиеватых закорючек — весьма мутное занятие, от которого человек всегда искал способ избавиться. Первым достижением на этом нелегком пути стало книгопечатание, давшее возможность быстро тиражировать один раз созданный текст. Тем не менее, пылливый человеческий разум упорно ищет способы автоматизировать и создание первого экземпляра. Конечно, машинное написание романов пока остается несбыточной мечтой, но частично автоматизировать набор текста реально уже сегодня с помощью макропроцессоров. В этой статье речь пойдет об одном из них — компактном и мощном макропроцессоре *m4* (рис. 1), доступном в любом дистрибутиве Linux.

Принцип работы

Не будет далеким от истины представлять *m4* в виде черного ящика, преобразующего текст. Это преобразование заключается в выполнении макроподстановок, то есть в поиске во входном тексте макровывозов вида NAME(P1, P2, ... PN) и их замене соответствующим макросу NAME значением. Фрагменты входного текста,

не являющиеся макровыводами, остаются без изменений: они просто копируются в выходной текст. То же происходит с любыми последовательностями символов, заключенными между правой и левой одиночными кавычками ('). Результат макроподстановки снова попадает на вход макропроцессора. Эта незначительная, на первый взгляд, особенность многократно расширяет возможности *m4*, позволяя при соответствующих навыках функционального программирования создавать довольно выразительные и эффективные рекурсивные конструкции.

Макросы бывают двух видов: встроенные и определяемые пользователем. Первые уже запрограммированы в *m4* и всегда доступны. Например, eval, служащий для преобразования арифметических выражений в их значение (eval('5+4') даст 9). Вторые должен определять пользователь на специальном языке. Встречая такие определения во входном тексте, *m4* добавляет их в своей арсенал и дальше использует аналогично встроенным. Уверенное знание синтаксиса этого языка и владение наиболее распространенными встроенными макросами — залог успешного использования *m4*. Этому и посвящена большая часть данной статьи.

Приступаем к работе

Самый простой способ попробовать *m4* в деле — просто запустить его командой *m4*. В таком режиме макропроцессор ждет ввода входного текста с клавиатуры и, тут же обрабатывая каждую введенную строку, выдает результат в стандартный поток вывода. Например, введя *5+4=eval(5+4)*, получим *5+4=9*. Очевидно, это может быть полезно лишь для быстрой проверки синтаксических конструкций языка *m4* или в качестве экстравагантного калькулятора. Гораздо практичнее заставить *m4* считывать входной текст из файла:

```
# echo '5+4=eval(5+4)' > ./test.m4
# m4 ./test.m4
```

» Рис. 1. Различные варианты вызова *m4*.

```
root@dmitry-MS-7519: /home/work/m4
root@dmitry-MS-7519: /home/work/m4# m4
5+4=eval(5+4)
5+4=9
^C
root@dmitry-MS-7519: /home/work/m4# echo '5+4=eval(5+4)' > ./test.m4
root@dmitry-MS-7519: /home/work/m4# cat ./test.m4
5+4=eval(5+4)
root@dmitry-MS-7519: /home/work/m4# m4 ./test.m4
5+4=9
root@dmitry-MS-7519: /home/work/m4#
```

Элементы языка

Определяем новые макросы

Возможность создавать новые макросы — самое главное, что делает макропроцессор универсальным инструментом. С этим стоит разобраться в первую очередь.

Синтаксис макроопределения очень прост: define(NAME, BODY), где NAME — имя нового макроса, а BODY — выражение, описывающее макроподстановку. В общем случае BODY может быть весьма хитроумной смесью конструкций языка *m4*, в простейшем — обычной строкой, которой будут заменены все последующие вызовы NAME.

Макросы могут иметь параметры. При вызове они указываются в круглых скобках через запятую, а в BODY к ним можно обращаться с помощью специальных переменных \$1, \$2, ..., \$N. Кроме того, также доступны:

- » \$0 Имя макроса
- » \$# Количество параметров
- » \$* Весь список параметров

В качестве примера напишем традиционный "hello, world" — простейшее макроопределение без параметров, предписывающее *m4* раскрывать короткий макровывод *hw* в полный текст приветствия:

```
define(hw, 'Hello, world.')
```

Следует отметить, что в макровыводах без параметров отсутствуют скобки, хотя некоторым может показаться логичнее поставить пустые. Тем не менее, *m4* другого мнения на этот счет: он толкует `hw()` как `hw(“”)`, что, впрочем, не влечет за собой негативных последствий, если в теле макроопределения не используется переменная `$#`.

Понятно, что применимость таких макросов сильно ограничена. Однако, чтобы добавить параметры, достаточно лишь передать их при вызове. Синтаксис макроопределения остается неизменным. Для примера усовершенствуем `hw`, реализовав возможность задавать имя приветствуемого субъекта. Назовем новый макрос `hx`:

```
define(hx, 'Hello, $1.')
```

Так, например, можно поприветствовать нашу любимую ОС:

```
hx('Linux')
```

Укрощаем пустые строки

Вероятно, некоторые уже заметили на рис. 2 небольшую проблему. Кроме ожидаемых результатов работы `hw` и `hx`, видны три пустые строки. Немного поразмыслив, легко понять причину: *m4* часто обрабатывает все символы перевода строки (`\n`) во входном тексте. В нашем случае проблема вызвана попаданием на выход двух символов `\n`, стоящих после макроопределений, и одного из пустой строки, добавленной для удобства чтения кода. Разумеется, такое поведение не всегда желательно, поэтому важно уметь им управлять.

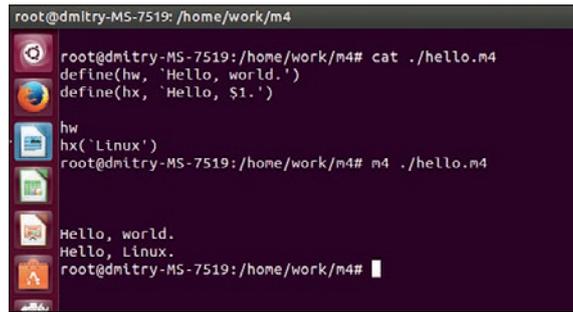
Самый простой способ — воспользоваться специальным встроенным макросом `dnl` (англ. Discard to Next Line). Как нетрудно догадаться по названию, он подавляет обработку всего последующего текста до первого символа `\n`, включая его. Таким образом, чтобы исправить наш пример, нужно в конце первых трех строк файла `hello.m4` дописать “`dnl`”.

Но достаточно написать пару десятков нетривиальных макроопределений, постоянно чертыхаясь и забывая в нужных местах вставлять `dnl`, чтобы по-настоящему почувствовать необходимость более удобного решения. Кроме того, код, пестрящий одной и той же назойливой трехбуквенной «мантрой», не всем нравится, хотя это, конечно же, дело вкуса. К счастью, в *m4* можно подавить вывод результатов обработки целого фрагмента входного текста, заключив его между `divert(-1)` и `divert()dnl`. Этот трюк основан на работе с потоками вывода *m4*, механизм работы которых подробно разбирается в следующем разделе. Следует отметить существенное отличие этого метода от использования `dnl`: он подавляет вывод результатов обработки, а не саму обработку. Поэтому определения макросов, встречающиеся в оформленном таким образом фрагменте, нормально работают, а вот `dnl define(hw, 'Hello world.')` не делает абсолютно ничего полезного.

Потоки вывода

В программировании идея потоков не нова и весьма популярна. Она так прочно укоренилась в сознании программистов, что едва представив заголовок этого раздела, многие наверняка уже невольно представили его вероятное содержание и, скорее всего, почти во всем угадали.

Потоки вывода в *m4* представляют собой набор буферов памяти, куда помещается текст после раскрытия макросов и из содержимого которых в итоге формируется выходной текст. Они обозначаются целыми неотрицательными числами, а их количество зависит от реализации. Некоторые поддерживают только 10 штук (от 0 до 9), однако в GNU-версии оно ограничено лишь размером целочисленного типа данных, что, очевидно, гораздо больше, чем может понадобиться для любого разумного применения.



» Рис. 2. Макросы `hw` и `hx` в действии.

Одновременный вывод в несколько потоков невозможен: в любой момент времени активен только один. Для его переключения служит директива `divert('N')`, где `N` — номер потока. Нулевой поток имеет особый статус. Во-первых, на него можно переключиться с помощью `divert` без параметров (эквивалент `divert('0')`). Во-вторых, он является потоком вывода по умолчанию, что позволяет использовать *m4*, вообще ничего не зная о потоках вывода: макропроцессор будет молча работать с нулевым потоком и выдаст его содержимое в качестве результата. Также допустимы отрицательные значения `N`. В этом случае выводимый текст просто отбрасывается, чем мы и воспользовались в предыдущем разделе. Итоговый выходной текст получается простой конкатенацией всех потоков в порядке возрастания их номеров. С помощью `undivert('N', ..., 'M')` можно вызвать немедленный вывод содержимого указанных потоков в текущий. Их содержимое при этом очищается, а попытки вывода потока в самого себя игнорируются. Вызов `undivert` без параметров означает вывод всех потоков.

- Проиллюстрируем сказанное на простом примере:
- » Понятно, что код `t1 t2 t3 dnl` выведет `t1 t2 t3`, поскольку вывод осуществляется в нулевой поток (по умолчанию), а все остальные потоки пусты. Следовательно, результатом работы будет содержимое нулевого потока.
 - » Код `divert('1')t1 divert('2')t2 divert('3')t3 dnl` выведет то же самое, поскольку нулевой поток остается пуст, а `t1`, `t2` и `t3` выводятся в первые три потока в порядке возрастания их номеров.
 - » Чтобы изменить порядок вывода текста, достаточно подправить аргументы `divert`. Например, `divert('3')t1 divert('2')t2 divert('1')t3` переворачивает результат: `t3 t2 t1`.
 - » Перебросив содержимое третьего и второго потоков в нулевой, можно вернуться к первоначальному порядку: `divert('3')t1 divert('2')t2 divert('1')t3 divert('0')undivert('3', '2')dnl`. В результате при формировании итогового выходного текста в нулевом потоке будет `t1 t2`, в первом — `t3`, а все остальные пустые. На выходе получаем их конкатенацию — `t1 t2 t3`.



» Рис. 3. Методы борьбы с пустыми строками.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

Алгоритмические конструкции

Логика работы приводимых до сих пор примеров сводилась к замене макровыводов готовой строкой. Разумеется, для скольких-нибудь содержательных задач требуются более развитые средства описания алгоритма генерации текста. В языках программирования эта проблема традиционно решается с помощью различных условных и циклических конструкций. В этом отношении *m4* не отличается оригинальностью, хотя и довольно аскетичен. Вместо многочисленных `if`, `ifelse`, `switch`, `do`, `while`, `until`, `for`, `foreach` доступны лишь две условные конструкции: `ifdef` и `ifelse`.

Первая используется исключительно для проверки существования макроопределения. Синтаксис `ifdef` в общем виде выглядит так:

```
ifdef (name, s1, [s2])
```

Здесь `name` — имя макроса. Если он существует, все выражение `ifdef` раскрывается в строку `s1`, иначе — в `s2`. Третий аргумент `ifdef` может быть опущен. В таком случае считается, что он равен пустой строке.

Вторая, в зависимости от количества аргументов, может быть трех видов:

» **ifelse(comment)** Версия с одним аргументом. Она вообще не имеет никакого отношения к условному выполнению. Это — средство оформления многострочного комментария. Аналогичный эффект дает макровывод `dnl` в начале каждой строки `comment`.

» **ifelse(s1, s2, equal, [not-equal])** Версия с тремя или четырьмя аргументами. Это классический условный оператор. Если строки `s1` и `s2` равны, всё выражение `ifelse` раскрывается в строку `equal`, иначе — в `not-equal`. Если четвертый аргумент опущен, считается, что он равен пустой строке.

» **ifelse(s1, s2, equal1, s3, s4, equal2, ..., [not-equal])** Версия с более чем четырьмя аргументами. Это конструкция множественного выбора, аналог `switch` в языке C. Если строки `s1` и `s2` равны, все выражение `ifelse` раскрывается в строку `equal1`, иначе аналогично обрабатывается следующая тройка аргументов, и т. д.

Циклов в *m4* нет, а выполнение повторяющихся действий достигается исключительно средствами рекурсии, которая теоретически ничем не хуже, хотя и не так привычна для большинства программистов. В качестве примера напомним макрос `seq(n, m)`, генерирующий последовательность целых чисел от `n` до `m` (например, вызов `seq('5', '10')` должен раскрываться в `5 6 7 8 9`). Конечно же, первая мысль — цикл `for`. Однако рекурсивное решение тоже довольно простое. Нетрудно заметить, что исходную задачу легко разбить на две: напечатать `n` и решить задачу, аналогичную исходной, но меньшего размера — `seq(n+1, m)`, которая также может быть решена подобным образом. Последовательное возрастание первого аргумента `seq` рано или поздно приведет к вызову `seq(m, m)`, что будет означать получение решения исходной задачи. Остается лишь записать все это на языке *m4* (см. рис. 4):

```
define('seq', 'ifelse('$1', '$2', '', '$1 seq(eval('$1'+1), '$2)'))
```

Говоря о рекурсии, нельзя не упомянуть простой, но чрезвычайно полезный встроенный макрос `shift`, принимающий неограниченный список аргументов и возвращающий его «хвост» (список без первого элемента). Например, `shift('1', '2', '3')` раскрывается в `2 3`. Любители LISP уже подметили сходство с функцией `cdr` (возвращает «хвост списка»), и, конечно же, знают, каким мощным средством обработки данных она является в тандеме с функцией `car`, возвращающей «голову списка» (первый элемент). Оказывается, в *m4* подобные трюки можно вытворять со списком аргументов макроса, вместо `car` и `cdr` используя `$1` и `shift($@)` соответственно. Для примера покажем, как поэлементно пройтись по списку аргументов, выполняя над каждым элементом произвольное действие. Пусть в нашем случае это

будет простая печать в фигурных скобках, хотя ничего не мешает его сколь угодно усложнить.

```
define('print_lst', 'ifelse('$#, '1', '{ $1}', '{ $1} print_lst(shift($@))')
```

Арифметика

Поскольку *m4* не предназначен для решения вычислительных задач, его математические возможности ограничены всего тремя встроенными макросами, которые работают исключительно с целыми числами. Это два простейших макроса `incr(n)` и `decr(n)`, реализующих инкремент и декремент соответственно, а также уже встречавшийся выше `eval`. Про первые два добавить больше нечего, а вот об `eval` стоит рассказать подробнее. Во-первых, кроме вычисляемого выражения, у него есть еще два необязательных параметра, задающих формат результата:

» **radix** Определяет систему счисления; по умолчанию он равен '10'.

» **width** Количество символов в результате; если число получается коротко, при выводе оно будет дополнено ведущими нулями.

Во-вторых, кроме четырех арифметических операций, поддерживаются еще и:

» деление по модулю (%).

» возведение в степень (**).

» поразрядные сдвиги (>>, <<).

» побитовые И, ИЛИ, исключающее ИЛИ, НЕ (&, |, ^, ~).

» логические И, ИЛИ, НЕ (&&, ||, !).

» операторы сравнения (==, !=, >, <, >=, <=).

» скобки для изменения приоритета вычислений.

Работа со строками

Средства обработки строк побогаче — семь макросов. Пять из них довольно простые:

» **len(s)** Вычисляет длину `s`. Например, `len('Hello')` дает 5.

» **format(s, ...)** Аналог известной функции `printf` из стандартной библиотеки C. Например, `format(' %s %d', '2+3=', eval('2+3'))` даст `2+3=5`.

» **index(s1, s2)** Ищет первое вхождение `s2` в `s1`. В случае успеха возвращает позицию первого символа `s2` в `s1` (нумерация с 0), иначе "-1". Например, `index('Hello', 'llo')` даст 2.

» **substr(s, pos, [len])** Возвращает подстроку `s`, начиная с позиции `pos`, длины `len`. Если параметр `len` опущен, результат ограничивается концом `s`. Например, `substr('Hello', 2, 2)` даст `ll`.

» **translit(s1, s2, s3)** Возвращает строку, полученную из `s1` заменой символов `s2[i]` на `s3[i]`. Если в `s3` нет соответствующего символа (из-за разности длин `s2` и `s3`), то `s2[i]` из `s1` просто удаляется. Например, `translit('Hello', 'He', 'A')` даст `Allo`.

Сложность оставшихся двух — лишь в использовании регулярных выражений, которые они принимают в качестве параметров. Это существенно увеличивает их возможности, а потому в *m4* нет недостатка в средствах обработки строк. Регулярные выражения — отдельная и довольно известная тема. Они активно используются не только в *m4*, но и во многих других утилитах, библиотеках и языках программирования, поэтому не будем останавливаться на описании их синтаксиса, а лишь бегло рассмотрим упоминаемые выше макросы:

» **regex(s, rexp, [replacement])** Реализует поиск по регулярно-му выражению. В зависимости от наличия третьего (необязательного) параметра может работать в одном из двух режимов. Если `replacement` опущен, `rexp` превращается в продвинутый аналог `index`, допускающий в качестве второго параметра не только простую подстроку, но и сколь угодно сложное регулярное выражение. Например, `regex('Hello', 'llo')` даст 2. В противном случае `replacement` описывает возвращаемую строку, если поиск увенчался успехом, иначе результатом `regex` будет пустая строка. В `replacement`

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

можно использовать специальные символы `\1–\9`, обозначающие фрагменты найденного текста, соответствующие заключенным в скобки частям регулярного выражения; `\&` — соответствует всему совпадающему с регулярным выражением тексту. Например, `regexp('Hello, world', '\\(Hello, \\)(world\\)', '\\1Linux')` даст `Hello, Linux`.

» **patsubst(s, rexp, [replacement])** Выполняет в **s** глобальную замену найденных по регулярному выражению **rexp** фрагментов на **replacement**, после чего возвращает **s**. Если третий параметр опущен, считается, что он равен `!`. Следовательно, найденные фрагменты текста просто удаляются. Как и в случае с `regexp`, в `replacement` допустимы специальные символы `\&` и `\1–\9`. Например, `patsubst('Hello, world', 'world', 'Linux-&')` даст `Hello, Linux-world`.

Модульность

С ростом проекта размещение исходного кода в одном файле становится очень неудобным. По этой причине *m4* позволяет разбивать код проекта на несколько файлов и включать их друг в друга с помощью директивы `include(file)`. Возвращаясь к приведенному примеру (рис. 3), можно было бы макроопределения `hw` и `hx` вынести в отдельный файл `lib.m4`, а в `test.m4` заменить их на `include('./lib.m4')dnl`.

Если что-то пошло не так (файл не найден), `include` выводит ошибку. Для тех, кого не устраивает такое ворчливое поведение, есть аналогичный макрос — `sinclude`, молча переносящий любые неудачи.

Вызов внешних программ

Будучи инструментом в стиле `*nix`, *m4* не годится на все случаи жизни. Его дело — макрогенерация. И это его единственное предназначение. Для всех остальных задач он даже не пытается предлагать посредственных решений, предоставляя вместо этого средства кооперации с другими программами.

Кроме обычной для всех утилит командной строки возможности работать в составе конвейера, *m4* позволяет выполнять команды оболочки `[shell]` в процессе своей работы.

Для этого используются два встроенных макроса:

- » **syscmd(sh-cmd)**
- » **esyscmd(sh-cmd)**

Они очень похожи: оба выполняют переданную в параметре `shell`-команду. Разница лишь в том, что результат макроподстановки

первого — пустой текст, а второго — вывод выполненной команды. Узнать код возврата последнего вызова можно при посредстве `sysval`.

Некоторым `shell`-командам может потребоваться файл для вывода результата или каких-то других целей. Для таких случаев в *m4* есть макрос, который позволяет создать временный файл: `mkstemp(template)`.

Параметр `template` — это шаблон, из которого путем добавления шести случайных символов будет сгенерировано имя создаваемого временного файла. Оно же является результатом макроподстановки `mkstemp`.

Например, вызов `mkstemp('/tmp/file')` сгенерирует нечто похожее на `/tmp/fileRb67Uw`.

Средства отладки

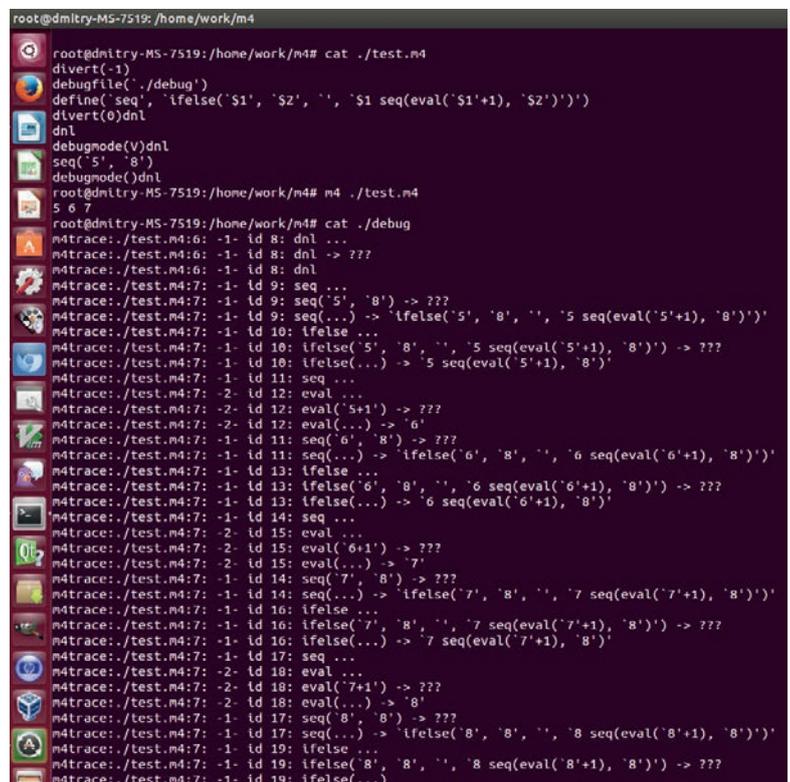
Программированию присущи ошибки. Даже ценой огромных затрат удастся лишь сдерживать их натиск. Не удивительно, что развитые средства отладки крайне важны для любой системы программирования. К счастью, обладая встроенной системой трассировки кода с внушительным набором настроек, *m4* в этом отношении весьма неплох. Не будем останавливаться на ее детальном описании, поскольку в большинстве случаев достаточно выполнить трассировку небольшого подозрительного фрагмента кода с максимально полным выводом отладочной информации. Этого легко добиться, окружив его макровыводами `debugmode(V)` и `debugmode`.

В больших проектах также может быть полезен встроенный макрос `dumpdef([names...])`, выводящий список макроопределений, имена которых указаны в параметрах (вызов без параметров дает полный список).

Отдельно стоит сказать о месте вывода отладочной информации. Оно задается с помощью встроенного макроса `debugfile(file-name)` или одноименным параметром командной строки при вызове *m4*. По умолчанию вывод осуществляется в `stderr`, который обычно подключен к `stdout`, поэтому отладочная информация и вывод макропроцессора смешиваются. Избежать этого можно перенаправлением `stderr` в файл:

```
# m4 ./test.m4 2>./debug
```

Кроме того, с помощью встроенного макроса `errprint(msg,...)` есть возможность вручную выводить отладочные сообщения в `stderr`. Это особенно полезно, поскольку доступны `__program__`, `__file__` и `__line__`, позволяющие включить в сообщение точное место возникновения ошибки.



» Рис. 4. Трассировка `seq`.

Заключение

Данная статья, конечно же, не претендует на исчерпывающее описание *m4*, однако ее должно быть достаточно, чтобы начать использовать этот замечательный инструмент для автоматизации своих

повседневных задач. Надеюсь, за время прочтения некоторые уже успели пожалеть, что не обратили на него внимание раньше, а многие часы были потрачены на рутинную работу, которой так легко было избежать. **LXF**

Система: Типы сигналов

Крис Браун пытается привлечь ваше внимание, отправляя вам сигналы «убить».



Наш эксперт

Доктор **Крис Браун** обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

В этом месяце мы займемся сигналами. Мы узнаем что это такое, и познакомимся с несколькими типами сигналов и с тем, как они используются. Мы посмотрим, как отправить сигнал из нашей программы и как написать обработчик сигнала, который управляет реакцией программы на полученный сигнал. Наконец, мы рассмотрим семь возможных способов использования сигналов.

Я собирался начать с определения слова «сигнал» в Оксфордском словаре английского языка, но отказался от этой затеи, найдя, что оно занимает почти полстраницы. Сигнал — слово весьма многозначное. В системном программировании сигнал — это событие, которое доставляется работающему процессу ядром. Сигнал говорит: «Произошло нечто (обычно вне самого процесса), возможно, заслуживающее твоей реакции». Существует несколько типов сигналов, с которыми мы вскоре познакомимся. Для реагирования на сигнал программа предоставляет обработчик сигнала. В терминах языка программирования C это функция, которая будет вызвана при получении сигнала. Если в программе не определен обработчик сигнала для какого-то типа сигнала и приходит сигнал этого типа, предпринимаются «действия по умолчанию». Обычно это завершение программы, но есть несколько типов сигналов, которые по умолчанию игнорируются.

Вывести список всех типов сигналов из командной строки можно так (я показал только первую строку вывода):

```
$ kill -l
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL 5) SIGTRAP
```

Вы видите, что у каждого типа сигнала есть имя (такое как SIGINT или просто INT) и номер (в данном случае, 2). Из всех 64 типов сигналов, которые вывела `kill -l`, вам скорее всего потребуются только десяток или около того, поэтому мы привели самые важные типы в таблице (внизу). В столбце «Действие по умолчанию»

показано, как процесс отреагирует на сигнал, если обработчик этого сигнала не задан. Оба варианта Term [Завершение] и Core [Ядро] означают, что процесс будет завершен. Различие в том, что для варианта Core также генерируется дамп ядра — файл, содержащий образ памяти процесса на момент завершения (свое название он получил от «памяти на магнитных сердечниках» ушедших дней), который бывает полезен при отладке.

Поясню содержимое нашей таблицы. Сигналы SIGTERM, SIGINT и SIGKILL связаны с завершением программ. SIGTERM часто интерпретируется как вежливый запрос «пожалуйста, прибережи за собой и завершись», и хорошо написанный демон обычно ответит на него, освободив все временно занятые ресурсы (например, рабочие файлы) и завершив работу. SIGINT — это сигнал, который отправляется драйвером терминала всем процессам, подключенным к терминалу, когда пользователь нажимает ^C. Обычная реакция — опять же завершение процесса, но некоторые программы SIGINT игнорируют. (Об этом чуть ниже.) Наконец, у нас есть брутальный сигнал SIGKILL, который процессы не могут обрабатывать или игнорировать. Получение сигнала SIGKILL означает верную и мгновенную смерть процесса, как от пули в лоб. Отправка сигнала SIGKILL должна применяться только в крайнем случае, когда более мягкие методы убеждения не дали результата.

SIGHUP нередко интерпретируется демонами как «Я только что изменил твой файл настройки — пожалуйста, перечитай его», хотя сначала его назначение было другим. (См. врезку «История SIGHUP», стр. 81.) Наконец, SIGALRM — это сигнал, который генерируется по истечении таймера интервала, запущенного системным вызовом `alarm()`. Позже мы рассмотрим пару возможных вариантов использования этого сигнала.

Большинство сигналов формируются вне процесса и возникают асинхронно, т.е. они напрямую не связаны с деятельностью процесса. Зато сигналы SIGSEGV и SIGFPE формируются внутри процесса, и оба обычно означают ошибку в коде. SIGSEGV («нарушение сегментации») означает, что программа попыталась обратиться к адресу памяти, находящемуся вне адресного пространства. Как многие программисты знают по личному опыту, в C это довольно легко устроить, если выполнить ссылку через неинициализированную точку. SIGFPE («исключение с плавающей точкой») чаще всего возникает при попытке целочисленного деления на ноль. Как ни странно, при делении на ноль числа с плавающей точкой SIGFPE не возникает, потому что в стандарте IEEE для представления чисел с плавающей точкой правомочно представление бесконечности.

Отправка сигналов

Сигналы формируются различными приложениями и возникают в разных частях системы. Также можно отправить собственные сигналы с командной строки (или из скрипта оболочки):

```
$ kill -TERM 7451
```

или из программы на C:

```
kill(7451, SIGTERM);
```

➤ В Posix определено 64 типа сигналов, но интерес для системных программистов представляют только некоторые из них, и вот их список. Не за что.

Типы сигналов

Название сигнала	Номер	Действие по умолчанию	Описание
SIGHUP	1	Term	Некоторые демоны интерпретируют этот сигнал как «перечитайте свой файл настройки»
SIGINT	2	Term	Сигнал, отправляемый при нажатии ^C в терминале
SIGTRAP	5	Core	Прерывание трассировки/конечной точки
SIGFPE	8	Core	Арифметическая ошибка, например, деление на ноль
SIGKILL	9	Term	Смертельный сигнал, который нельзя перехватить или игнорировать
SIGSEGV	11	Core	Неправильное обращение к памяти
SIGALRM	14	Term	Таймер оповещения истек
SIGTERM	15	Term	Вежливый сигнал «пожалуйста, завершите работу»
SIGCHLD	17	Игнорировать	Дочерний процесс завершился

История SIGHUP

«HUP» в SIGHUP означает «повесить трубку [hang up]» и заслуживает объяснения. Раньше у телефонов (так давно, что даже я этого не помню) имелся рычаг, и чтобы завершить звонок, вы вешали трубку обратно на этот рычаг. Мы продолжаем говорить «Он повесил трубку», хотя вешать ее уже давным-давно не на что.

Так или иначе, предположим, что Кен вошел в Unix с помощью модема и телефонной линии (обычный способ удаленного подключения до появления IP-адресов). У него была запущена оболочка, когда по какой-то причине он разорвал соединение («повесил трубку»). Позже Деннис подключается к тому же модему и обнаруживает, что попал

в оболочку Кена. Нехорошо. Чтобы избежать такой ситуации, модем и его драйвер устройства договариваются, что модем, обнаружив завершение звонка, освобождает управляющую линию «data carrier detect [несущая обнаружена]», и драйвер отвечает на это, отправляя сигнал «HUP», который закрывает оболочку Кена. Проблема решена!

Честно говоря, `kill` [убить] — не самое удачное название для функции. Ее нужно было назвать `throw` [отбросить] или `send-signal` [отправить сигнал]. Имя объясняется, думаю, тем, что во многих случаях сигнал приводит к кончине процесса. Так или иначе, 7451 — это идентификатор целевого процесса. И да, аргументы в двух примерах записываются в обратном порядке. (Ребята, это Linux — чего вы ожидали? Постоянства?) Разумеется, есть и правила. Сигналы вы можете посылать только собственным процессам, если вы не `root`; ну, а `root` подвластно всё. Системный вызов `kill` возвращает 0 в случае успеха и -1 при неудаче, и вот приемчик, которым можно пользоваться: отправив сигнал с номером 0, можно узнать, существует ли процесс (сам сигнал при этом процессу не доставляется):

```
if (kill(7451,0) == 0)
/* Процесс существует */
else
/* Процессы не существует */
Процесс также может отправить сигнал самому себе, вызвав raise(SIGTERM);
```

Ответ на сигналы

Итак, в отправке сигналов трудности никакой. Ответить на них в своем коде чуть сложнее. Во-первых, надо написать обработчик сигнала — это просто функция с особым параметром:

```
void myhandler(int signum)
{ ... }
```

Передаваемое значение говорит о типе сигнала; часто оно не слишком интересно, поскольку обычно обработчик создается для обработки одного конкретного типа сигнала, и вы заранее знаете, какой тип сигнала получили. Следующий шаг — «установить» обработчик. Сделать это можно несколькими способами. Начнем с более старого: он проще. Вызов функции `signal(SIGHUP, myhandler)`; устанавливает вызов обработчика `myhandler` каждый раз при поступлении сигнала SIGHUP. В качестве второго аргумента также можно передать два специальных значения. Первое — `SIG_IGN`. Вызов `signal(SIGHUP, SIG_IGN)`; приводит к тому, что сигнал SIGHUP будет игнорироваться (это очень распространенный вариант использования). Второе — `SIG_DFL`. Вызов `signal(SIGHUP, SIG_DFL)`; восстанавливает схему обработки по умолчанию.

Хотя сами вызовы функции `signal()` выглядят просто, от того, что происходит внутри определенных типов, у вас глаза заслезятся. Второй аргумент — это «указатель на функцию, которая принимает `int` и возвращает `void`». В качестве определения типа мы можем записать это так:

```
typedef void(*sighandler_t)(int);
```

Функция `signal` также возвращает старый обработчик (на случай, если вам понадобится его восстановить). Поэтому прототип самой функции `signal` таков:

```
sighandler_t signal(int signum, sighandler_t handler);
```

В нескольких примерах ниже функция `signal()` используется для установки разработчика; однако у такого вызова есть некоторые недостатки, и мы рекомендуем заменить ее более сложной функцией `sigaction()`. Наряду с одинаковым синтаксисом во всех

версиях Linux и Unix (чего у функции `signal()` нет), `sigaction()` обладает дополнительной гибкостью. Во-первых, можно определить набор сигналов, блокируемых на время действий обработчика. Во-вторых, можно управлять автоматическим перезапуском любых системных вызовов, прерванных поступлением сигнала.

Функция `sigaction()` принимает указатель на структуру `sigaction`, параметры которой включают следующее:

```
struct sigaction {
void (*sa_handler)(int);
sigset_t sa_mask;
int sa_flags;
};
```

Здесь `sa_handler` — указатель на функцию обработки сигнала, который мы уже видели в старом вызове `signal()`; `sa_mask` задает сигналы, которые будут блокироваться во время выполнения обработчика, а `sa_flag` может включать флаг, к которому нужно применить побитовое «ИЛИ». Единственный флаг, который я хочу упомянуть — `SA_RESTART`. Он означает, что системные вызовы необходимо автоматически перезапустить, если они прерваны сигналом. (Это поведение по умолчанию для старой функции `signal()` в Linux, но не в ранних версиях Unix.) Также есть флаг `SA_SIGINFO`, с которым обработчику сигнала передается гораздо больше информации, но я не буду углубляться в подробности. Сведем все это вместе — и вот типичный код для установки обработчика сигнала с помощью `sigaction()`:

```
void handler(int sigtype)
{ ... }
main()
{ struct sigaction action;
action.sa_handler = handler;
sigemptyset(&action.sa_mask);
action.sa_flags = SA_RESTART;
sigaction(SIGINT, &action, NULL);
...
}
```

Вызов `sigemptyset()`, который вы видите выше — это один из нескольких макросов, манипулирующих маской сигнала, которая является лишь одним из способов представления набора типа сигналов. Другие макросы — `sigaddset()` и `sigdelset()`, которые добавляют и удаляют указанные сигналы из набора, и `sigfillset()`, который инициализирует набор сигналов, включая в него все сигналы. Подробности вы узнаете на map-страницах. Мы используем набор сигналов в `sigaction`, чтобы блокировать доставку определенных сигналов во время выполнения обработчика сигнала. В примере выше мы этого не делали, передав пустой набор сигналов. Также учтите, что сигнал, вызвавший срабатывание обработчика, будет автоматически заблокирован на время выполнения обработки.

Если программа хочет отсрочить обработку сигналов во время выполнения определенных «важных участков» кода, можно установить маску сигналов, которая определяет, какие сигналы будут блокироваться. Для этого используется функция `sigprocmask()`. Приведенный ниже код не сможет быть прерван никакими сигналами (но помните, что от SIGKILL скрыться невозможно):

Скорая помощь

Если у вас есть опыт программирования для .NET, то может возникнуть соблазн сравнить сигналы с событиями .NET. Это сравнение не вполне оправдано. События в .NET возникают чаще и «на более мелком уровне» — например, событие формируется при нажатии левой кнопки мыши. А сигналы в Linux возникают реже и сильнее влияют на жизнь процесса. Также события в .NET — это в целом «хорошие новости», а сигналы — скорее «плохие».

Скорая помощь



Блокирование сигналов с помощью `sigprocmask()` — не то же самое, что игнорирование сигналов. При игнорировании сигналы просто отбрасываются. При блокировании они удерживаются до тех пор, пока не будут разблокированы. Но несколько заблокированных сигналов одного типа не объединяются в очередь — при снятии блокировки будет доставлен только один сигнал каждого типа.

```
sigset_t set;
sigfillset(&set);
sigprocmask(SIG_SETMASK, &set, NULL);
/* Код, который нельзя прерывать сигналами */
sigprocmask(SIG_UNBLOCK, &set, NULL);
```

Полезные вещи, которые можно делать с помощью сигналов

Так, на этом теория закончилась. В завершение этой статьи — семь полезных вещей, которые можно делать с помощью сигналов.

» **№ 1: Игнорировать их** Хорошо, но не обман ли это. «Полезно» ли игнорировать сигналы? Определенно да! Умение писать надежные программы, которые не «падают» сразу же при получении сигнала — одна из самых важных причин уметь игнорировать сигналы. Проще всего сделать это с помощью следующих вызовов:

```
signal(SIGINT, SIG_IGN);
signal(SIGQUIT, SIG_IGN);
```

И так далее. Помните, что сигнал `SIGKILL` проигнорировать нельзя. Также можно просто заблокировать все сигналы на постоянной основе.

» **№ 2: Прибраться и завершить работу** `SIGTERM` обычно используется как запрос «пожалуйста, завершите работу», и хорошо написанный демон должен ответить на него, сбросив все данные из памяти в файл и освободив все временные ресурсы, такие как файлы, именованные каналы, области общей памяти или семафоры, которые в противном случае переживут процесс. Вот образец кода для программы, которая создает рабочий файл и процесс и освобождает их при получении сигнала `SIGTERM`:

```
int child_pid;
void cleanup(int sigtype)
{
    unlink("/tmp/workfile");
    kill(child_pid, SIGTERM);
    wait(NULL);
    exit(1);
}
main()
{
    signal(SIGTERM, cleanup);
    open("/tmp/workfile", O_RDWR|O_CREAT, 0600);
    child_pid = fork() ... ;
    /* Продолжаем работу */
}
```

» **№ 3: Перенастройка на лету** `SIGHUP` традиционно используется для того, чтобы попросить демона перечитать свою конфигурацию на лету. Это менее «разрушительно», чем полный перезапуск демона — например, в таком случае существующие соединения с клиентами не будут разорваны.

```
int config_info;
void read_config_file(int signum)
{
    /* Чтение конфигурации из файла */
}
main()
{
    signal(SIGHUP, read_config_file);
    raise(SIGHUP);
    while (1) // Главный цикл обслуживания
    {
        // Принятие подключения от клиента
        // Выполнение действий на основе конфигурации
    }
}
```

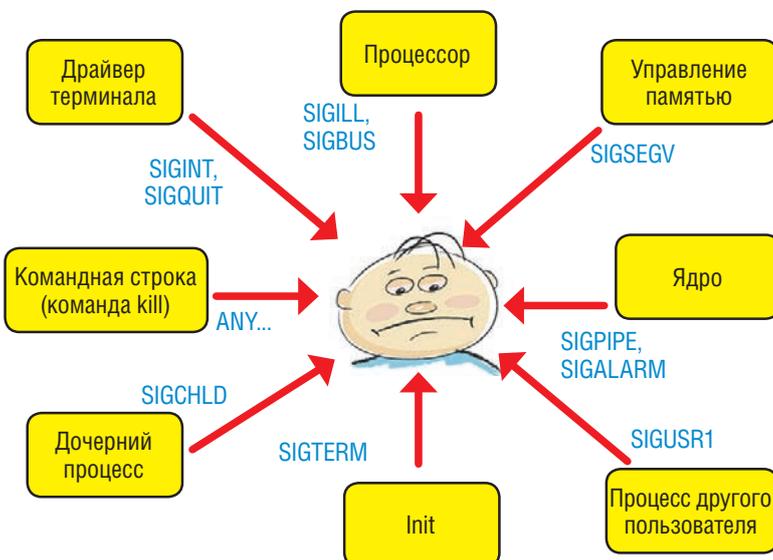
Обратите внимание на глобальную переменную `config_info`. Она символизирует конфигурацию, которую может иметь демон в реальном мире. Также обратите внимание на `raise()`, вызов принудительного считывания конфигурации в начале.

» **№ 4: Динамическое информирование о состоянии программы** Идея состоит в том, чтобы предоставить способ, с помощью которого долго работающая программа могла бы по требованию сообщать информацию о своем внутреннем состоянии — такую как суммарная информация о трафике, счетчики использования или таблицы состояния клиента. Отчет можно вывести в терминал (если есть открытый терминал) или, вероятнее, в файл. Отображаемая информация о состоянии должна быть глобальной, чтобы она была доступна обработчику сигнала. Вот вымышленная долгая программа копирования на ленту. Мы отправляем сигнал `SIGUSR1`, чтобы спросить, сколько данных скопировано:

```
int count; /* Глобальная информация о состоянии */
void print_state_info(int sig)
{ printf("%d blocks copied\n", count);
}
void main()
{
    signal(SIGUSR1, print_state_info);
    for (count = 0; count < A_BIG_NUMBER; count++) {
        read_block_from_input_tape();
        write_block_to_output_tape();
    }
}
```

» **№ 5: Включение и выключение отладки** Несмотря на наличие мощных сред разработки и отладчиков, популярным способом отладки остается сопровождение кода вызовами `printf()`. Когда они больше не нужны, их можно либо просто удалить, либо, если они могут опять понадобиться позже, отключить их во время компиляции с помощью директивы `#ifdef`. Однако более динамичный подход — включать и выключать их во время выполнения по сигналу. В этом примере отладка включается и выключается с помощью `SIGHUP`:

```
int debug_on = 0;
void toggle_debug_flag(int sig)
{
    debug_on ^= 1; // Сброс флажка
}
main()
{
    signal(SIGHUP, toggle_debug_flag);
    /* Затем позже в коде... */
}
```



» **Мамочка, а откуда берутся сигналы?**

Как завершился этот процесс?

Если вы следили за нашей серией, то знаете, что программа, которая завершается добровольно, возвращает своему родителю статус выхода — малое целое число. Ноль означает нормальное, успешное завершение работы, а ненулевые значения — какую-то ошибку. Но процесс, завершившийся сигналом, не может указать статус выхода. В этом случае номер сигнала, вызвавшего останов программы, будет содержаться в младшем байте статуса, полученного от функции `wait()`.

Следующий код иллюстрирует использование четырех макросов:

```
main()
{
    int status;
    int a, b, c;
```

```
    if (fork())
    {
        /* Родительский процесс */
        wait (&status);
        if (WIFEXITED(status))
            printf("Дочерний процесс завершен нормально
со статусом %d\n", WEXITSTATUS(status));
        if (WIFSIGNALED(status))
            printf("Дочерний процесс вышел по
сигналу %d: %s\n", WTERMSIG(status),
strsignal(WTERMSIG(status)));
    }
    else
    {
        /* Дочерний процесс */
        printf("PID дочернего процесса = %d\n", getpid());
```

```
        *(int *)0 = 99; // Формируем SIGSEGV, или...
        a = 1; b = 0; c = a/b // Формируем SIGFPE
        sleep(3);
        exit(5);
    }
}
```

Здесь мы видим, что если дочерний процесс завершился нормально, `WIFEXITED` возвращает `true` и `WEXITSTATUS` возвращает статус выхода процесса; если дочерний процесс был завершен сигналом, `WIFSIGNALED` возвращает `true`, а `WTERMSIG` возвращает номер сигнала. Пример также иллюстрирует возможность намеренного формирования сигналов `SIGSEGV` и `SIGFPE`, для цели демонстрации. Вспомогательная функция `strsignal()` также предоставляет текстовое описание каждого типа сигнала.

```
if (debug_on)
    printf(«нечто полезное для отладки»);
}
```

» **№ 6: Реализация таймаута** Идея состоит в том, чтобы сгенерировать сигнал `SIGALRM` с помощью таймера интервала, с целью заставить системный вызов, потенциально способный заблокировать систему, например, `read()`, завершиться по таймауту, если данные не получены.

```
1 /* Применим SIGALRM для создания timeout */
2
```

```
3 #include <stdio.h>
4 #include <setjmp.h>
5 #include <signal.h>
6 #include <errno.h>
7
```

```
8 void timeout_handler(int unused)
```

```
9 {
10 /* Делать нечего */
11 }
12
```

```
13 int t_getnum(int timeout)
```

```
14 {
15     int n;
16     char line[100];
17     struct sigaction action;
18
19     action.sa_handler = timeout_handler;
20     sigemptyset(&action.sa_mask);
21     action.sa_flags = 0; /* Важно! */
22     sigaction(SIGALRM, &action, NULL);
23
24     alarm(timeout);
25     n = read(0, line, 100);
26     alarm(0); /* Снять alarm */
27     if (n == -1 && errno == EINTR) return -1;
28     n = atoi(line);
29     return n;
30 }
31
```

```
32 void main()
```

```
33 {
34     int num;
35
36     while(1) {
37         printf("введите число: "); fflush(stdout);
```

```
38     if ((num = t_getnum(5)) == -1)
39         printf("Таймаут!\n");
40     else
41         printf("Вы ввели %d\n", num);
42 }
43 }
```

Это пример подлиннее и заслуживает разбора. Сначала в строках 8–11 определяется пустой обработчик сигнала. Начиная со строки 13, мы определяем функцию `t_getnum()`, которая считывает целочисленное значение из стандартного потока ввода с таймаутом, указанным в секундах. Функция возвращает `-1`, если в течение таймаута не вводилось данных. Строки 19–22 связывают обработчик с сигналом `SIGALRM`. В строке 21 поле `sa_flags` устанавливается в ноль, чтобы система не перезагрузилась автоматически при возникновении сигнала. (В Linux сделать это с помощью более простого вызова `signal()` нельзя.) В строке 24 запускается интервальный таймер, который сгенерирует сигнал `SIGALRM` по истечении указанного количества секунд. Теперь все готово, поэтому в строке 25 мы вызываем блокировку, считывая строку из стандартного потока ввода. (Да-да, я знаю, что здесь должна быть проверка переполнения буфера!) При возврате из функции `read()` мы отменяем таймер оповещения. Проверка в строке 27 задает вопрос: «Возврат из `read` произошел нормально или в результате получения сигнала?» В последнем случае `errno` будет установлен в `EINTR` («прерванный системный вызов»), и мы вернем `-1`; в противном случае (строки 28 и 29) мы преобразуем входную строку в целое число и возвращаем ее. Можно предвидеть, что у такой реализации будут проблемы, если пользователь введет само число `-1`.

Функция `main()` просто входит в цикл (строка 36), повторно вызывая `t_getnum()` с таймаутом в 5 секунд и сообщая, истек ли таймаут или были получены данные. Эту программу на самом деле надо бы продемонстрировать вживую, но даже *Linux Format* не сумеет этого сделать в печатном журнале!

» **№ 7: Планирование периодических действий** Идея состоит в том, чтобы вызывать повторение действия с регулярным интервалом с помощью сигналов `SIGALRM`. Прием состоит в том, чтобы каждый раз повторно активировать вызов сигнала в обработчике сигнала. В отличие от предыдущего использования `SIGALRM`, в этом случае мы определенно хотим установить обработчик, чтобы прерванные системные вызовы были автоматически перезапущены. Пример — возможность автоматического сохранения в текстовом редакторе, в котором содержимое текстового буфера записывается на диск каждые 30 секунд.

В следующем месяце мы обратимся к сети и узнаем, как написать клиент и сервер с использованием TCP/IP и сокетов API. **LXF**

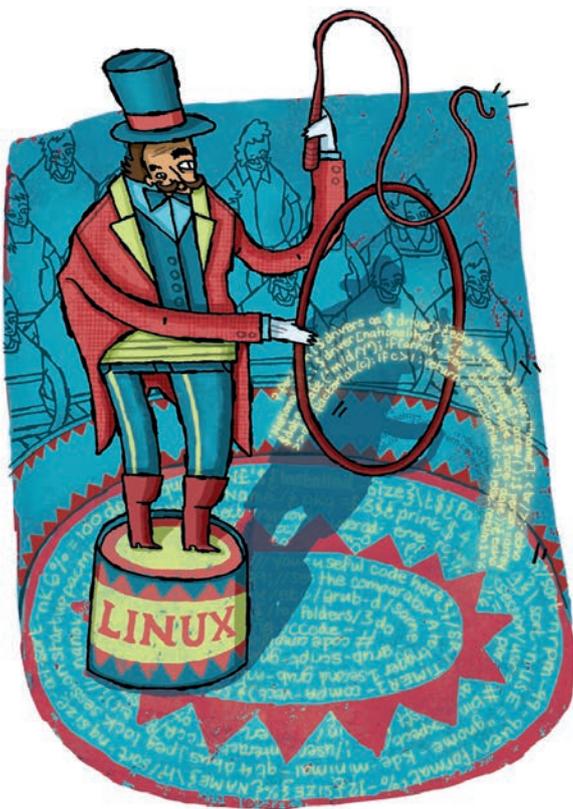
RHR: Создаем агрегатор лент

Кент Ельчук описывает, как творчески объединить ваши любимые ленты RSS.



Наш эксперт

Кент Ельчук — канадский web-разработчик, живет в Ванкувере. Помимо создания индивидуальных сайтов, он также является автором *RHR/MySQL Fundamentals and Snippets* и сотрудником CodeCanyon Marketplace.



Вы никогда не задумывались, как одни сайты отображают сводные данные с других сайтов? Этот трюк можно выполнить несколькими способами, в том числе за счет привилегий на доступ к базам данных источников, web-скрейпинга или употребления лент RSS — да и совместив какие-то из них. На нашем уроке мы рассмотрим, как собирать и комбинировать данные с помощью лент RSS. RSS, что расшифровывается как Rich Site Summary (а позже — Really Simple Summary), существует уже давно — и берет свое начало со времени работы Раманатхана В. Гухи [Ramanathan V. Guha] в Apple Computer's Advanced Technology Group в середине 1990-х — и вы, вероятно, используете их каждый день для отображения новостных лент среди прочей полезной информации.

RSS — это, как правило, XML-файлы, которые содержат данные внутри структурных тэгов; допустим, если вы зайдете в Craigslist и просмотрите регулярные записи, то увидите обычный список пунктов. Это исходный код HTML. Но в Craigslist RSS Feed те же элементы будут отображаться не в виде HTML, а в форме XML-файла, который хранит каждый список внутри родительского тэга «элемент [item]». Каждый тэг элемента содержит информацию о записи, в том числе название, дату и описание. Захват и объединение RSS-лент с помощью простого скрипта PHP — это

быстрый и легкий способ получить желаемые данные и вывести на экран нужные вам результаты. Источниками информации для RSS-лент служат многие крупные рекламные и аукционные сайты (такие как eBay, eBay Classified, Monster, Craigslist и т. д.) — и многие другие, в том числе сайты трудоустройства (например, CareerBuilder и Indeed позволяют получить XML-ленты, которые также можно синтаксически проанализировать).

Во многих случаях для доступа к лентам RSS достаточно обратиться к определенному URL, а иногда необходим API с указанным ключом издателя. Одним из примеров сайтов, которые требуют ключ издателя для получения API, является популярный сайт поиска работы, indeed.com.

Поскольку нам нужна была тема для примера, мы расскажем, как группировать RSS из различных источников, упомянутых выше. Наши примеры покажут, как сузить круг поиска работы по местоположению и должности, или же запросо сделать себе личный RSS и XML-агрегатор для отображения элементов сделки, которую вы хотите совершить через eBay, Craigslist и т. д.

Кроме того, вы сможете сделать так, чтобы результат загружался в вашем браузере на локальном web-сервере, например, LAMP в дистрибутиве Ubuntu. Установка LAMP займет всего пару минут, но мы также расскажем, как это делается. Помимо отображения результатов в web-браузере, будет ссылка на исходное сообщение, и вы сможете вернуться туда, чтобы выполнить любые действия, например, узнать о вакансии и т. д.

С чего начать

Итак, закатываем рукава и начнем разбираться, как быть в курсе всего. Первое, что вам нужно для агрегирования и анализа RSS-лент — это рабочий web-сервер с несколькими дополнительными пакетами, такими как PHP. Поскольку дело это минутное, инструкции по его установке мы приведем потом. Те, кто уже думают «О Боже, опять установка web-сервера», могут это пропустить. Для установки сервера Apache выполните следующий код:

```
sudo apt-get update
sudo apt-get install apache2
```

Затем установите пакеты PHP:

```
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

На нашем уроке для анализа лент RSS мы скачаем готовый скрипт, а также приведем методику, как сделать это вручную. В роли парсера выступит *Magpie RSS*. Примечание: расположен он на Sourceforge (<http://bit.ly/MagpieRSS>), но существуют и альтернативы без использования этого сайта, например, *Simple Pie* (<http://simplepie.org/downloads>).

Скачав *Magpie*, сохраните его в папке **www** или **html**; в новых версиях Ubuntu, к примеру, сетевые папки Apache находятся в **/var/www/html**, в более старых, такие, как Ubuntu 12.04, корневой сетевой папкой является **/var/www**.

Независимо от корневой папки, теперь вам нужно создать еще одну, с именем **rss**. Туда вы сохраните извлеченную папку *Magpie*, с именем **magpierss-0.72**. Для простоты переименуйте

Скорая помощь

Вы всегда можете использовать *curl*, чтобы открыть ленты RSS и сохранить их в виде файлов XML. Затем их можно использовать вместо URL.

Придаем стиль лентам и ссылкам

Поскольку вывод отображается в обычной таблице, пока что он не выглядит не особо стильно, но легко добавить плагин, который сделает его круче.

Набрав в web-поиске «лучшие быстрые плагины для таблиц загрузки», вы увидите, что есть много вариантов, равно как и учебников для добавления стилей, но мы быстро объясним, как сделать это с помощью плагина под названием Datatables, который можно скачать с <https://datatables.net/download/download>.

Помимо того, что Datatables представляет таблицу изменения размера для многих устройств и имеет функцию поиска, можно еще и выбрать, сколько записей вы хотите видеть на странице.

Хотя на **LXF DVD** мы разместили файл с именем **index_styled.php**, я скорей всего объясню, как легко

добавить этот плагин. Для начала зайдите в файл и смените тип документа на html 5. Так, самый первый тэг <html> может быть заменен на <!DOCTYPE HTML>. После этого добавьте заголовочную часть со ссылками на необходимые файлы.

Верхняя часть данных представлена ниже:

```
<head>
<link rel="stylesheet" type="text/css" href="//cdn.datatables.net/1.10.7/css/jquery.dataTables.css">
<script type="text/javascript" charset="utf8" src="//code.jquery.com/jquery-1.10.2.min.js"></script>
<script type="text/javascript" charset="utf8" src="//cdn.datatables.net/1.10.7/js/jquery.dataTables.js"></script>
<script type="text/javascript">
$(document).ready( function () {
```

```
$('#example').DataTable();
});</script>
</head>
```

Обратите внимание, что вы можете использовать файлы, загруженные с сайта Datatables, или ссылки из CDN (Content Delivery Network).

Осталось только добавить два атрибута в тэг таблицы:

```
id="example" class="display"
```

Теперь, загрузив страницу в web-браузере, вы увидите, что все выглядит намного элегантнее.

Те, у кого больше опыта кодирования, могут пойти немного дальше — например, использовать вспомогательные медиа-запросы и классы, чтобы менять размер текста и удалять столбцы, при изменении ширины используемого экрана.

ее по правому щелчку мыши в **maggie**. В дополнение к папке **maggie**, у вас есть два файла: **index.php** и **customs.php**. Файл **index.php** нужен для отображения содержимого и чтобы открывать его в браузере и **customs.php**. А **php** включает функции приема URL и сбора данных из источников RSS (Craigslist, Ebay Classified, [Monster.com](#), [Careerbuilder.com](#) и [Indeed.com](#)). Примечание: большинство лент в образце кодирования обрабатываются с помощью **Maggie**, за исключением Indeed и Careerbuilder. Их анализ осуществляется с помощью встроенного класса PHP — **DOMDocument()**.

Код, используемый на нашем уроке, выложен на **LXF DVD** и на сайт, то есть отредактировать вам придется только строки, касающиеся фактических адресов, которые находятся в районе строки 3 в файле **index.php**. Это ваш исходный список URL-адресов для RSS-лент, которые вы хотите объединить. Тем не менее, возможно, вы также захотите кое-что изменить в пользовательских настройках анализатора, в районе строки 108 в файле **customs.php**. В настоящее время пользовательские URL-адреса настроены на Indeed и Careerbuilder.

Вам нужно раскомментировать адреса для CareerBuilder и Indeed, поскольку оба они требуют ключа издателя. Если вы зарегистрируетесь и получите ключи разработчика, все, что вам нужно сделать, это заменить **YOUR_KEY_HERE** на соответствующие ключи.

Indeed Publisher Program доступна на www.indeed.com/publisher, а информацию о том, как собрать результаты поиска для сайта CareerBuilder, можно найти на <http://bit.ly/CareerBuilderAPI>.

Теперь, разобравшись с основами, самое время пройтись по коду и объяснить, как это работает. Для начала откройте файл **index.php** в web-браузере. URL-адрес на локальном компьютере будет <http://localhost/rss> или <http://localhost/rss/index.php>. В верхней части файла, создается массив с именем **\$urls**, содержащий ленты, которые вы хотите агрегировать.

Получить RSS-ленты с различных источников довольно просто, но у каждого поставщика будет свой способ их поиска. Безусловно, какие-то найти будет легче, какие-то труднее. Давайте начнем с Craigslist. Если перейти на <http://craigslist.co.uk> и посмотреть вакансии в разделе квалифицированного труда/ремесла в Лондоне, вы увидите множество записей. По нажатию на кнопку RSS в правом нижнем углу экрана вы получите эту ленту, или можно сначала сузить область поиска, а затем добавить ее. В верхней левой части экрана нужно проверить окно «контракт». Затем нажмите на желтый значок RSS в правом нижнем углу, и вы увидите стра-

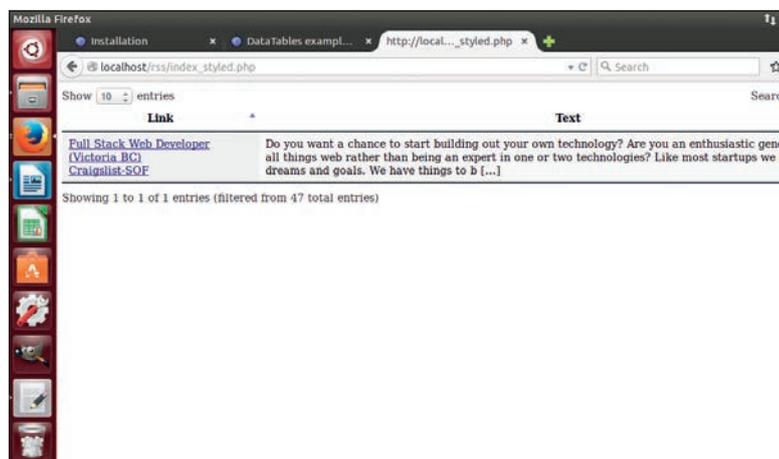
ницу ленты. URL-адрес в браузере является примером того, что вы могли бы добавить в ваши URL-адреса лент в верхней части страницы **index.php**.

Несколько источников

Другой пример будет посвящен сбору RSS-лент с eBay, и здесь все не столь очевидно. Но вы можете сузить поиск до того, что вам нужно, добавив **&_rss=1** в конце URL в web-браузере, чтобы увидеть новую страницу с нужной лентой.

Следующие две строки указывают файл, который **Maggie** будет использовать для анализа и файл **customs.php**, который использует пользовательские функции и проверки. Дальше немного сложнее. В районе строки 108 в **customs.php** вы увидите функцию, которая проверяет, существует ли массив **\$urls**. Получив подтверждение, она проходит через цикл по каждому элементу для каждого URL.

Если первый набор условий выполнен, проверяется, содержит ли часть строки URL текст 'indeed' или 'careerbuilder'. Встроенная функция PHP **strpos()** определяет, есть ли определенное слово в URL. Если 'indeed' или 'careerbuilder' в URL нет, запустится функция **custom_url(\$urls)** с соответствующим URL. Эта функция находится в верхней части **customs.php**.



» Есть много плагинов, таких как Datatables (на рисунке), благодаря которым ваш вывод можно сделать элегантным.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!

После обработки URL функцией `custom_url()`, URL передается в функцию `fetch_rss()`, которая является функцией *Magpie* и находится в файле `magpie/rss_fetch.inc`.

Вкратце, *Magpie* захватывает URL и возвращает значение `$rss_items`. При этом элементы массива, содержащие название, описание и ссылку, разделяются на отдельные переменные с именами `$title`, `$description` и `$date`.

Следует учесть, что на Craigslist используется формат даты `$item['dc']['date']`, тогда как другие источники, как правило, используют `$item['pubDate']`. Этот пример кода позволяет не только легко проверить ленту, но и при необходимости посмотреть тэг и добавить условие.

Когда переменные потока установлены, есть условия, которые создают на основе текста URL переменную с именем `$origin`. Таким образом, вы можете просматривать источник в браузере, запуская скрипт.

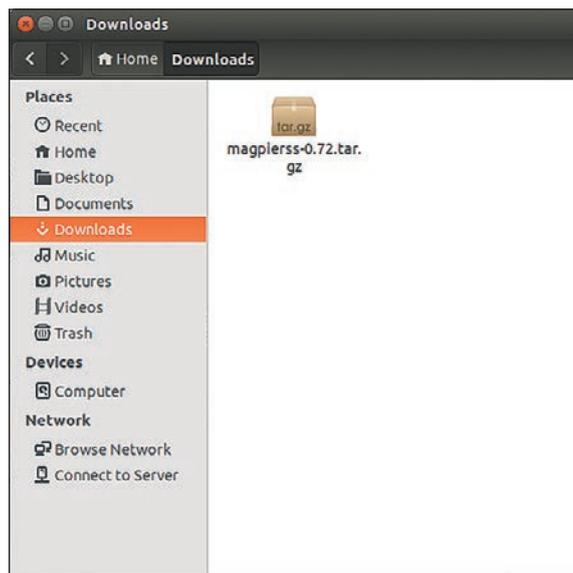
Наконец, в конце этой функции собран массив `$tot_array` со всеми адресами, которые обрабатывает эта функция. `$tot_array` содержит все переменные, включая ранее созданные дату и название. Массив включает строки, содержащие все фрагменты данных через запятую.

Функции и условия

Вернемся немного назад: помните, функция `custom_url()` не работает с адресами `Indeed.com` или `Careerbuilder`? А что, если вам нужны именно они? Пора ответить на этот вопрос. Так как `Indeed.com` и `Careerbuilder` имеют разные тэги XML, у каждого из них будет свой метод анализа. На самом деле, это стоит запомнить, так как вы можете применить те же методы с другими XML-файлами, которых нет в этом примере кодирования — просто следовать схеме, делая необходимые поправки.

Сначала код проверяет, относится ли URL к `Indeed.com`. Если это не так, то проверка будет выполняться для `CareerBuilder`.

Далее будем предполагать, что вы находитесь в своем пользовательском API от `Indeed.com`, который вы получите бесплатно, если зарегистрируетесь.



➤ *Magpie RSS* можно скачать с Sourceforge (Не переживайте, это всего лишь TAR, а не установщик)

Во-первых, создается новый DOMDocument, в кодировке UTF-8. После этого применяется метод `$dom->load($url)` и производится загрузка URL. Теперь у вас выводится список. В XML-файле от `Indeed` каждый результат обозначается как 'result', а в `CareerBuilder` каждая запись будет 'JobSearchResult'.

В остатке этого примера мы ограничимся `Indeed.com` и его тэгами. Для показа, что будет интерпретироваться, можете просто взять URL своего API от `Indeed.com` и загрузить его в браузере. Это будет XML-файл, где есть все тэги и данные для каждой записи.

Затем для анализа каждого результата будет использоваться цикл по каждому элементу. Как вы можете увидеть из кода (на `LXF DVD`), для получения желаемых значений применяется метод `getElementsByTagName()` и некоторые другие способы кодирования; аналогично тому, как *Magpie* работает с другими RSS-лентами.

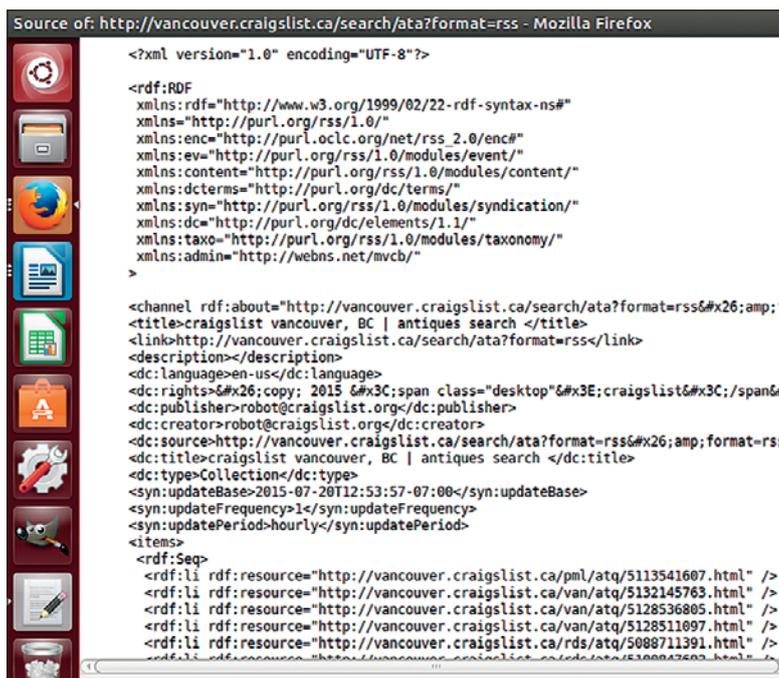
Вы заметите такие методы кодирования и такие функции, как `htmlspecialchars_decode()`, которые используются при создании значения переменных, наподобие `$jobtitle_value`. Декодирование по сути преобразует символичный код в требуемый символ. В случае с `Indeed`, вы также увидите, что, если есть `$jobkey_value`, то создается новая `$myurl_value` — потому, что она будет обеспечивать связь с записью компании-источника, а не с постом на `Indeed`.

Отметим, что с каждым циклом по элементу `$tot_array` всё удлиняется и по завершении цикла включает все предложения. В сущности, `$tot_array` — это все из *Magpie* и из анализа пользовательских XML вместе взятые. Теперь можно вернуться к `index.php` и продолжить с того момента, где мы остановились, то есть в районе строки 9.

Если мы получили записи из всех наших источников, у нас есть `$tot_array`. Что касается кода, если массив существует, то он выполняется с помощью функции `usort()`, которая вызывает функцию `cmp()`. Она используется для обработки массива, чтобы самые новые объявления были в начале.

Когда эта функция завершит свою работу, HTML-код отобразит таблицу результатов. С левой стороны будет название и ссылка на пост, посередине будет описание, в конце — дата.

Чтобы лучше понять, как отображается каждая запись, обратите внимание на строку 28. Как видите, чтобы наши результаты были уникальными и не имели дубликатов, используется



➤ Исходный код ленты, содержащий структурированные записи.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Скрейпинг данных

Если RSS-лента недоступна, альтернативным методом является скрейпинг [scraping — англ. «выскребание»]. Это техника анализа исходного кода web-страницы на наличие образцов данных, которые могут вас заинтересовать.

Возможно, некоторые из ваших источников не имеют RSS-лент. Но поскольку они, вероятно, содержат некие записи в форме перечня web-страниц HTML, таблиц или обязательных тэгов `<div>`, вы всегда можете посмотреть исходный код, найти соответствующий образец, а затем наскрести оттуда

данные. Зачастую, записи такого рода будут иметь уникальный класс или какой-либо другой набор атрибутов, позволяющих легко понять, где начинается и где заканчивается нужная вам информация.

Чтобы произвести скрейпинг страницы с PHP, вы можете использовать одну из своих функций, таких как `file_get_contents()` или `curl()`, чтобы получить ее исходный код. `Curl` имеет гораздо больше преимуществ, включая возможность использовать прокси-сервер, например, *Tor*. Тем не менее, при скрейпинге данных следует проявлять

осторожность. Возможно, стоит связаться с обладателем авторских прав и сохранить контент, не публикуя, чтобы с него не проникали.

Преобразовав контент в строку, можно использовать регулярные выражения и функции PHP, такие как `preg_match_all()` и `preg_match()`, чтобы разбить файл на записи. В конце концов вы получите те же данные, что и с RSS-лент. Если вы хотите, чтобы все ваши данные хранились в одном месте, то добавлением всего нескольких строк кода можно объединить метод скрейпинга с лентами RSS.

функция `array_unique()`. Затем, в районе строки 33, используется цикл по всем элементам для разбора `$tot_array`.

Для каждой записи в массиве создаются новые переменные заголовка, даты, URL и описания записи. Поскольку каждый элемент в `$tot_array`, также известный как `$tot`, представлен как строка переменных, разделенных запятыми, применяется также функция `explode()`, для создания еще одного массива с именем `$all`.

Массив `$all` содержит фактически переменные. Поскольку дата является первым пунктом в массиве, она отображается как `$all[0]`. Можно добавить, что вторым элементом было название, поэтому оно имеет значение `$all[1]`. Если вы не особо разбираетесь в массивах, имейте в виду, что все индексированные массивы начинаются с 0, а не с 1, как вы могли бы ожидать.

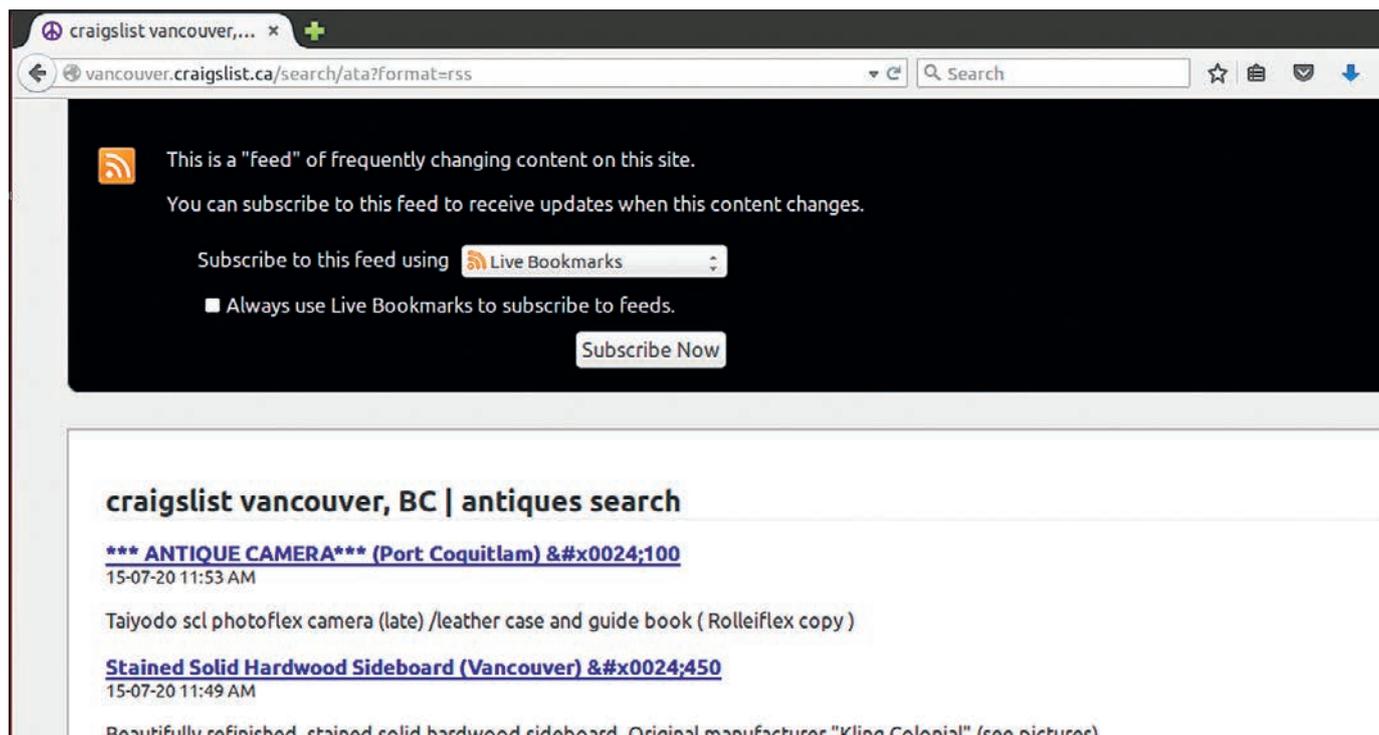
Доводим до ума агрегацию

Возвращаясь к последней переменной даты, вы увидите, что она содержит другой код. Дело в том, что функция даты используется для отображения результатов с указанием года, месяца и дня. Кроме того, есть функция `strtotime()`, чтобы получить дату из цикла, находящегося внутри элемента `$all[0]`, и конвертировать ее в метку времени Unix.

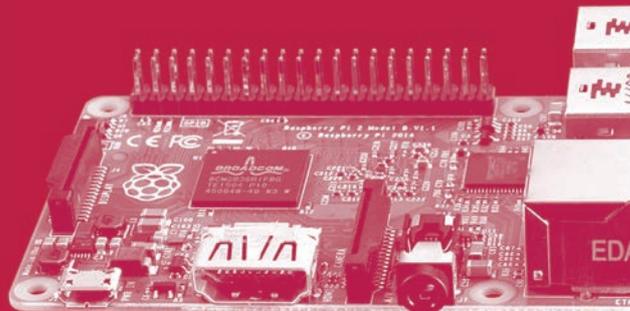
Взглянув на исходный код в файле `index.php`, вы увидите, что каждая запись отображается в отдельной строке таблицы, и каждое значение указано внутри тэгов `<td></td>`. В нижней части файла вы также увидите закрывающий тэг `</body>` и закрывающий тэг `</html>`. Если все так, у вас есть рабочий RSS-агрегатор и представление о том, как функционирует наш простенький скрипт. Но пока что он выглядит немного старомодно, и надо малость навести лоск. Для удобства использования можно добавить URL из базы данных, чтобы у вас под рукой были разные объединенные источники.

Также можно добавить функцию управления записями. Так как у каждой записи будет свой URL, можно сделать кнопки для вставки данных в базу данных: например, одна кнопка — чтобы добавить адрес в избранное, а другая — чтобы пометить как ненужный, и в следующий раз, когда вы открываете страницу, его не было бы видно.

Вот и все. Теперь у вас есть свой RSS-агрегатор лент, который будет отображать источник и описание вакансий по дате. Ищете ли вы работу или сравниваете предложения, пытаетесь совершить сделку, или просто хотите быть в курсе новостей из нескольких источников, небольшой PHP-скрипт позволит вам достичь желаемого. **LXF**



► Пример отображения RSS-ленты в браузере.



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

**ЭНДРЮ
МАЛХОЛЛАНД,**
создатель PiNet,
откроет, с чего все
началось



Привет...

Два года назад я вел вводный курс по Raspberry Pi в местной начальной школе. Даже с небольшой группой учеников это был логистический кошмар. Я часами создавал идеальный SD-образ со всем необходимым ПО и переносить его на SD-карту. Однажды карта полетела, и пришлось просить всех учеников установить какую-то одну программу — не помню, какую, потому что я несколько раз путал SD-карты.

Этот опыт навел меня на мысль, что должен быть способ получше.

Одно время я баловался с LTSP (Linux Terminal Server Project), созданным для поддержки компьютерного класса с машинами на Linux. Я быстро понял, что с Raspbian тут несколькими командами не обойтись. И я начал работать над скриптом, который сперва назывался RaspberryPi-LTSP, а теперь стал PiNet.

Как он работает? Сначала каждый Pi через проводную сеть загружает свою ОС с центрального сервера PiNet. Затем перед учеником появляется графический экран входа в систему. Поскольку все пользователи хранятся на сервере, любой ученик имеет доступ к любому Pi. При загрузке с сервера, желая внести изменения в ОС (скажем, добавить новый пакет), вы просто ставите его на сервер и перезагружаете все Pi.

Помимо этих двух основных особенностей, он невероятно прост в настройке и включает возможности общего доступа к папкам, автоматическое резервное копирование, систему, которая позволяет собирать и раздавать работы, и многое другое. В настоящее время PiNet используют тысячи учеников в сотнях школ по всему миру.

Picademy в Бирмингеме

Обучение в стиле Pi теперь и в сердце Англии.

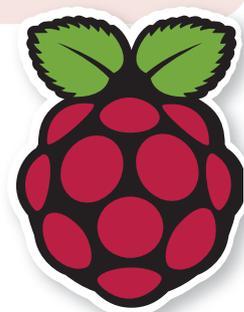
Мы уже сообщали о возможностях Raspberry Pi Foundation расширить свою схему сертификации учителей через Google Digital Garage в Лидсе. Это оказалось настолько популярным, что Google предложил фонду доступ к Digital Garage в Бирмингеме.

Picademy@Google предназначен для преподавателей любого предмета начальной, средней или старшей школы. В Лидсе курсы и мастер-классы ведет некто Лес Паундер, который посвящает массу времени, помогая взрослым и детям создавать чудные и чудесные проекты.

Новая Picademy@Google в Бирмингеме предложит не менее классные вещи: па-рень по имени Мартин О'Хэнлон (Martin O'Hanlon), известный своей работой, связанной со stuffaboutcode.com и мастерством в *Minecraft*, о котором мы уже рассказывали на этих страницах.

Если вам интересно принять участие и стать сертифицированным преподавателем Raspberry Pi, заполните форму на сайте, указанном ниже. Доступны следующие даты: 1–2 октября, 2–3 ноября, 7–8 декабря.

www.raspberrypi.org/picademy/google



Pidrophone

Поможет спасти китов.

Pi помогает людям по всему миру, размещая крошечный ПК там, куда никто никогда не добирался. Один проект использовал Pi для создания дешевого гидрофона (подводного микрофона), чтобы записывать и отслеживать китообразных в океанах. Коммерческие аналоги запрети-тельно дороги и, как правило, не поддаются модификации.

Команда проекта протестировала свои автономные записывающие устройства у побережья Бразилии, наблюдая морское движение и популяцию китов и дельфинов. Имея до 5 блоков резервного питания, устройство способно вести непрерывную запись до двух недель.
<http://bit.ly/LXHydrophone>

» ИиУооуу!
(Что по-китовьи:
«Как же круто,
что столько можно
сделать с Pi!»)



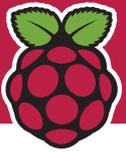
Picade жив!

Крошечный (дорогой) игровой автомат.

Всем любимому сообществу-поставщику Pi, Pimoroni, в августе исполнилось 3 года. Они переехали в более просторный офис и расширили штат до 20 человек. В честь этого в Pimoroni решили смахнуть пыль с проекта Picade Kickstarter и придать ему нового лоску. Стильный, самостоятельно собираемый игровой автомат в стиле ретро для вашего Pi включает новую макетную плату (совместимую с Arduino, оснащенную усилителем в 2,8 Вт), настоящий игровой джойстик и 12 миниатюрных кнопок-переключателей. При цене в £180, он дороговат, но есть версия Console, поменьше, за £90, с разъемом HDMI.
<http://shop.pimoroni.com/products/picade>

» Создайте из своего Pi настольный игровой автомат для ретро-развлечений.





Agobo 2

Лес Паундер, в личине Франкенштейна, создает в своей лаборатории нечто прекрасное — правда, больше из металлических, а не органических частей.

Вкратце

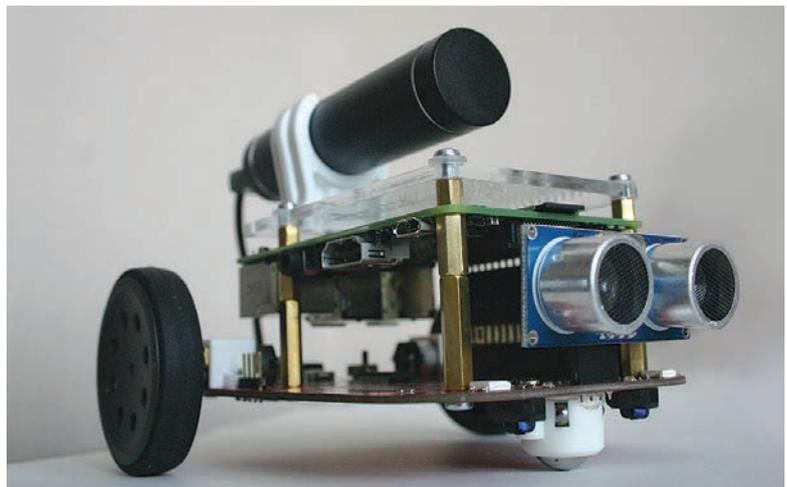
» Простой робот с широкими возможностями, спроектированный под A+, B+ и Raspberry Pi 2 и ориентированный на новичков и средних пользователей. Имеет надежный дизайн на основе печатной платы и простой в сборке комплект деталей. Управляется грамотно подобранной библиотекой Python, позволяющей без особого труда попробовать себя в робототехнике.

Любительские проекты в области робототехники бывают очень индивидуальны. Мы создаем корпус, формируя тело робота, добавляем датчики, которые станут его глазами и ушами, программируем его ум. Но начинающим робототехника готовит много подводных, поэтому такие компании, как 4tronix, выпускают им в помощь готовые наборы.

Первый Agobo вышел в конце 2014 г. и предоставлял печатную плату (Printed Circuit Board, PCB) с припаянными к ней компонентами, где пользователь мог разместить Raspberry Pi A+. Одним из пунктов критики этого Agobo было как раз использование A+, ведь ее чаще всего приходилось приобретать специально для этого проекта. Компания 4tronix учла этот момент и выпустила Agobo 2, который работает с A+, B+ и Raspberry Pi 2.

Контакты GPIO теперь повернуты на 90° против часовой стрелки так, что порты USB находятся сзади, а большая по размеру плата B+/Pi 2 расположена над задним мостом робота за счет простого набора дополнительных GPIO. Компания известна своим использованием печатной платы для шасси, что дает роботу прочную платформу и всю необходимую электронику в одном пакете.

Для приведения робота в действие Agobo 2 использует металлические редукторные моторчики и, благодаря встроенному контроллеру, обеспечивает точный двунаправленный контроль двигателей. Входы поступают на ультразвуковой датчик HC-SR04 — аналог парковочного —



» Agobo 2 объединяет все компоненты в прочный PCB-корпус, создав компактного, надежного робота с продуманным дизайном.

и два датчика линии, способных обнаруживать нанесенную траекторию движения робота. Поскольку робототехнические проекты, как правило, не предусматривают экрана, вам придется наладить SSH-доступ через Wi-Fi или Ethernet. В Agobo 2 есть специальный последовательный интерфейс для подключения к TTL по USB, обеспечивающий последовательное соединение с роботом прямо с вашего ноутбука. Agobo 2 также снабжен небольшим быстродействующим переключателем, который поможет вдохнуть жизнь в проект благодаря библиотеке Python (подробнее об этом далее). Agobo 2 подпитывается от небольшой батареи мобильного телефона, подключенной к плате, чтобы обеспечить постоянное энергоснабжение компонентов и Raspberry Pi.

позволяет роботу вращаться вокруг своей оси и грациозно поворачиваться. Функции можно освоить независимо от возраста и уровня способностей, и за 14 строк кода мы смогли запрограммировать робота, обходящего любые препятствия в пределах 50 см от ультразвуковых датчиков.

Agobo 2 усовершенствовал исходную платформу, сохранив простоту использования оригинала, как в программном, так и аппаратном отношении. Чтобы собрать комплект, потребуется только отвертка, а библиотека Python подружит с робототехникой даже новичка. Есть и другие комплекты для создания роботов, но экономически эффективным решением является Agobo 2. **LXF**



Свойства навскидку



Корпус PCB

Agobo 2 предпочел разместить Raspberry Pi, аккумуляторы и двигатели на одной прочной печатной плате.



Последовательное соединение

Роботы подвижны, так что смысл в параллельном подключении через USB к TTL есть.

Умный робот

Однако оборудование — это еще не все. Хорошему роботу необходимо подходящее ПО, и 4tronix собрали великолепную библиотеку Python, обеспечивающую функциональность робота. Для ультразвуковых датчиков создана собственная функция `getDistance()`, которая выполняет сложные расчеты расстояния от объекта, определяемое ультразвуком. Свою функцию имеют и датчики следования — она сообщает текущее состояние каждого датчика. Передний и задний ход точно контролируются благодаря ряду функций, обрабатывающих сигналы всех двигателей вместе. Это

LINUX FORMAT Вердикт

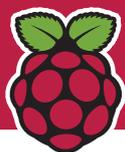
Agobo 2

Разработчик: 4tronix
Сайт: www.4tronix.co.uk/store
Цена: £ 32,45

Функциональность	9/10
Производительность	10/10
Удобство в работе	8/10
Документация	9/10

» Прочный корпус и несложная в сборке и кодировании платформа позволят любому шагнуть в мир робототехники.

Рейтинг **9/10**



Raspbian: Ставим и настраиваем

Лес Паундер расскажет, как установить дистрибутив Raspbian на ваш Raspberry Pi и выжать максимум из нового интерфейса.



Наш эксперт

Лес Паундер работает с Pi уже три года и входит в число преподавателей Picademy от Raspberry Foundation. Вот его блог: <http://bigl.es>.



Raspbian был штатным дистрибутивом для Raspberry Pi с момента его появления в 2012 г. На его примере мы впервые увидели, как обычная ветка Debian развилась в полнофункциональный продукт с массой применений. На нашем уроке мы заглянем под капот и ознакомимся со свежей версией.

Простейший способ установить Raspbian — через *NOOBS* (*New Out Of The Box Software*). Для этого понадобится карта microSD объемом не менее 8 ГБ, отформатированная под файловую систему FAT32. Скачайте ZIP-архив *NOOBS* (www.raspberrypi.org/downloads) на другой компьютер и распакуйте его на микро-SD. Извлеките карту памяти из компьютера и вставьте ее в свой Raspberry Pi. Пристыкуйте клавиатуру и другие периферийные устройства, не отключая источника питания. После загрузки Raspberry Pi перед вами появится меню с разными ОС для установки. Отметьте флажок рядом с Raspbian, нажмите Установить и подождите. Установка займет около 10 минут.

По завершении установки ваш Raspberry Pi перезагрузится в среду рабочего стола Raspbian, но сначала надо будет слегка подправить настройки через *raspi-config*, который запускается автоматически при первой загрузке. На данном этапе требуется только расширить конфигурацию Raspbian, чтобы она в полной мере использовала возможности вашей SD-карты.

По завершении перейдите к кнопке Закончить, с помощью клавиш управления курсором (т.е. стрелок), и нажмите Enter для подтверждения.

По умолчанию, любой Raspberry Pi имеет одни и те же детали учетной записи:

Username



➤ Raspbian является дистрибутивом по умолчанию для Pi, и новый дизайн сделал его гораздо удобнее.

Password

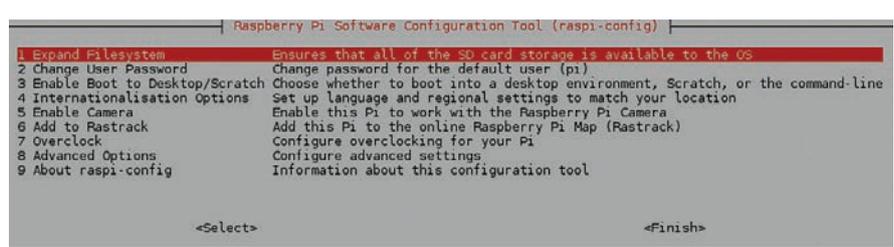
После входа в систему перед нами появляется *LXTerminal* Raspbian, предоставляющий максимум возможностей всего с помощью нескольких общих для Linux команд. Войдя в систему, мы попали в каталог **home** — туда, где хранятся все наши документы, музыка, фотографии и т.д. Чтобы понять, в какой именно части

Raspi-config

Raspberry Pi имеет собственный инструмент настройки, *raspi-config*. На нашем уроке он автоматически открывается при первой загрузке Pi. Но если вам надо в дальнейшем ввести какие-либо изменения, вы можете заново открыть его из *LXTerminal*, набрав `$ sudo raspi-config`.

С помощью *raspi-config* можно включить камеру Raspberry Pi, настроить аудио и сделать так, чтобы при загрузке запускался SSH-сервер. Можно также разогнать Pi, чтобы он работал быстрее, но переусердствовав с этим, Pi можно повредить: разогнанный процессор генерирует много тепла, а в Pi изначально нет радиатора для отвода тепла CPU.

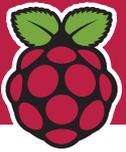
Raspi-config также может изменить распределение памяти между системой и GPU (Graphic



➤ Чтобы создать графическое меню в терминале, *Raspi-config* взамен диалога использует *whiptail*. И *newt* вместо *ncurses*, чтобы меню были дружелюбнее.

Processing Unit): например, веб-серверу без монитора много памяти не требуется, а настольному компьютеру необходимо побольше — для запуска

ПО наподобие оконного менеджера, который, как правило, и принято использовать при работе с Raspberry Pi.



Команды Linux

Raspbian основан на популярном дистрибутиве Debian, а это значит, что в нем множество команд. А если вы хотите узнать о какой-нибудь команде поподробнее? Введите `man` — это сокращением от `manual` [руководство] — команду, используемую для доступа к файлам справки для команды. Например, чтобы прочитать их для `scrot`, инструмента захвата экрана, следует ввести `$ man scrot`.

Автоматически откроется справка, которую можно просматривать с помощью клавиш курсора, и выйти из нее, нажав `q`.

Есть в Linux и другой удобный маленький трюк: автодополнение. Команды, которые вводятся в `LXTerminal/Bash`, могут автоматически дополняться нажатием клавиши `Tab`: например, если ввести `scr`, а потом нажать `Tab`, это слово автоматически

дополнится до `scrot`, а если это не та команда, которую вы ищете, можно нажать `Tab` еще раз, и Raspbian покажет вам все команды, соответствующие записанному.

Добавление еще нескольких символов поможет вам еще приблизиться к требуемой команде, а повторным нажатием `Tab` можно дополнить ее автоматически.

файловой системы мы находимся, можно воспользоваться командой для вывода местоположения, `$pwd`.

Она укажет наше расположение как `/home/pi/`; а если нам нужен другой каталог? Если он находится в директории `home`, то в первую очередь надо выяснить, какие еще каталоги там есть. Для этого вводим в командной строке `LXTerminal` команду `ls`.

Она выведет все каталоги и файлы в нашем каталоге `home`. Видя еще один, с именем `Downloads`, и желая туда перейти, меняем каталог командой `cd` и указываем расположение каталога, куда мы хотим попасть, вот так: `$ cd Downloads`. (Примечание: `$` набирать не надо. Это подсказка командной строки, которая означает, что вы находитесь в `LXTerminal/Bash`).

Теперь повтор команды `pwd` сообщит, что мы обретаемся в `/home/pi/Downloads`, а если надо вернуться в `/home/pi`, можно ввести:

```
#To go up one directory
$ cd ..

#Use an absolute location to navigate to home
$ cd /home/pi

#A quick way to return to home is
$ cd ~
```

Теперь давайте загрузим рабочий стол Raspbian, набрав `$ startx` и нажав `Enter`.

Через несколько секунд Raspbian загрузится, и вы увидите чистый рабочий стол с довольно крупным логотипом Raspberry Pi. Саймон Лонг [Simon Long], который присоединился Фонду в конце 2014 г., усовершенствовал рабочий стол Raspbian, потратив немало времени на упрощение его интерфейса и убрав повторения приложений из предыдущих версий. Главное меню располагается в левом верхнем углу экрана, внутри него мы видим ряд подменю.

Первое подменю называется `Programming` и включает две версии Python: 2.7 и 3. Также в нашем расположении `Sonic Pi`, популярное приложение для создания музыки на базе языка программирования Ruby. В следующем подменю, `Интернет`, расположен web-браузер и быстрые ссылки на магазин приложений Pi-Shop и официальные ресурсы Raspberry Pi. В подменю `Игры` мы найдем популярную игру `Minecraft`, в которую можно войти с помощью предустановленного API Python, недавно обновленного до версии Python 3. Наконец, в подменю `Расширения` содержится коллекция полезных приложений, таких как просмотрщик PDF, менеджер архивов, текстовый редактор и файловый менеджер для графического просмотра файлов и каталогов нашего Raspbian Pi.

В верхней части экрана рядом располагаются иконки. Это ярлыки к web-браузеру, файловому менеджеру, терминалу и другим приложениям. Далее, в правом верхнем углу есть группа из четырех иконок. Первая — это инструмент настройки Wi-Fi, пришедший на смену тому, что был в главном меню. Новый инструмент настройки легко справляется с большинством приставок Wi-Fi. Просто щелкните левой кнопкой мыши на значке Wi-Fi,

подождите, пока он обнаружит доступные вам сети, выберите маршрутизатор из указанных в списке и введите в диалоговом окне пароль. Через несколько секунд ваш Raspberry Pi окажется подключенным к сети.

Перемещаясь по экрану далее, мы видим регулятор громкости; левым щелчком мыши отображается ползунок громкости, а по щелчку правой кнопки мыши вывод HDMI переключается на 3,5-миллиметровый разъем для наушников. Оставшиеся два значка показывают уровень нагрузки процессора и приложение даты/времени.

При необходимости обновить программное обеспечение, устанавливаемое с Raspbian по умолчанию, для загрузки приложений и любых зависимостей пригодится `APT (Advanced Packaging Tool)`. Для этого откроем `LXTerminal`, через ярлык в верхней части рабочего стола. Наша первая команда обновит список программ, установленных на нашем Pi, согласно списку на серверах Raspbian, обеспечив актуальное состояние нашего ПО: `$ sudo apt-get update`. Теперь мы можем обновить ПО нашего Raspberry Pi, набрав `$ sudo apt-get upgrade`. Вероятно, это приведет к тому, что в `LXTerminal` появится длинный вывод. Если вас попросят применить изменения перед продолжением обновления, убедитесь в том, что вы знаете, что именно устанавливается.

Установка ПО также осуществляется аналогичным образом. В завершение нашего урока установим инструмент захвата экрана под названием `scrot`, набрав `$ sudo apt-get install scrot`. Еще чуть-чуть, и приложение будет установлено и готово к использованию (см. *Команды Linux*, выше, о том, как получить доступ к файлам справки `scrot`).

Дело сделано! Мы установили Raspbian, настроили его и узнали, где найти и как обновить ПО — можно приступать к работе над проектом! **LXF**

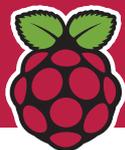


Скорая помощь

Raspbian можно использовать со всеми моделями Raspberry Pi, включая самый маломощный A+. Можно использовать его как полноценный рабочий стол, а можно — для удаленного доступа через SSH.

» В новом меню улучшен поиск по приложениям. В прежних версиях меню было перегружено.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!



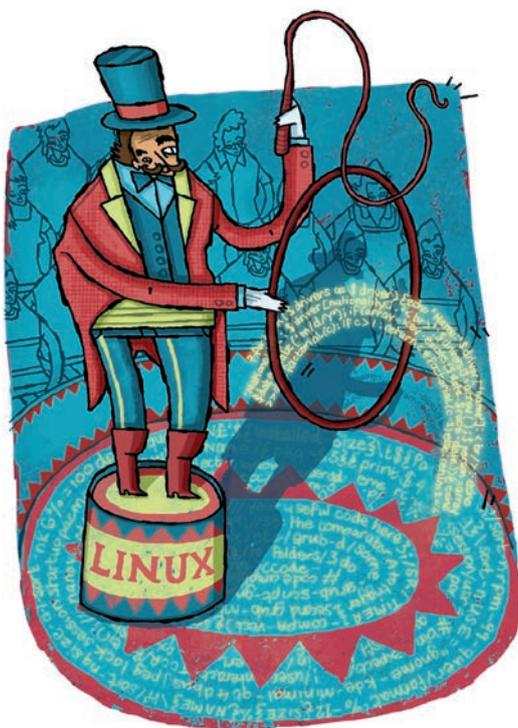
WhatsApp: PiBot ДЛЯ МОНИТОРИНГА

Маянк Шарма показывает, как взаимодействовать с Raspberry Pi и управлять им через популярную платформу обмена сообщениями *WhatsApp*.



Наш эксперт

Маянк Шарма был редактором www.linux.com и пишет для LinuxToday, Digg и PC Plus.



Вы используете Raspberry Pi в проекте без экрана, таком как медиа-проигрыватель, сервер NAS, сервер раздач [seedbox] или камера видеонаблюдения? Если да, то ваш Pi скорее всего запрятан в труднодоступном месте. Вы всегда можете подключиться к нему удаленно, но как отслеживать его в реальном времени? Как узнать, не перегрелся ли он? Или — что на нем переполнился диск? В этом проекте мы поиграем в Бога: наделим Raspberry Pi самосознанием и научим его говорить.

Выражаясь в более мирских терминах, мы поставим на Pi утилиту *sendxmpp*, которая позволит ему обмениваться данными по популярному протоколу обмена сообщениями XMPP. Pi будет отправлять нам сообщения при каждом возникновении заранее оговоренного события. Прежде всего, вам понадобится учетная запись для обмена сообщениями XMPP для Pi. Если у вас еще нет XMPP-сервера, можете зарегистрироваться на любом из публично доступных (<https://xmpp.net/directory.php>). Мы воспользуемся сервисом <https://jabber.hot-chilli.net>, который получил наивысшую оценку безопасности от <https://xmpp.net> и позволяет зарегистрировать учетную запись на самом сайте. После регистрации учетной записи для своего Raspberry Pi добавьте его в друзья к своей обычной учетной записи, на которую хотите получать оповещения.

Теперь зайдите на Pi, обновите репозитории и загрузите утилиту *sendxmpp* командой `sudo apt-get install sendxmpp`. Это скрипт Perl, и он притянет за собой необходимые зависимости Perl. После



➤ Документации для библиотеки *Yowsup* не слишком много, но с ней поставляется несколько полезных примеров приложений, которые интересно изучить.

```
pi@raspberrypi: ~/yowsup
pi@raspberrypi ~/yowsup $ yowsup-cli demos --yowsup --config config
Yowsup Cli client
=====
Type /help for available commands

[offline]:/L
Auth: Logged in!
[connected]:/message send 919968139981 "This is a test message sent from the Raspberry Pi"
[connected]:Sent: 1432009220-1
[connected]:
Iq:
ID: 2
Type: result
from: 918375972443@s.whatsapp.net

[919968139981@s.whatsapp.net(19-05-2015 04:21)]:[1432005783-1] Got it!
Message 1432005783-1: Sent delivered receipt
[connected]:
[connected]:
```



Видеочаты

Если вы используете Raspberry Pi 2 как обычный настольный ПК, можете установить на него клиент обмена сообщениями и общаться со своими друзьями в текстовом или в видеорежиме. Протокол XMPP, о котором мы рассказали в этом руководстве — один из самых популярных протоколов обмена сообщениями, и он используется такими сервисами, как Google Chat.

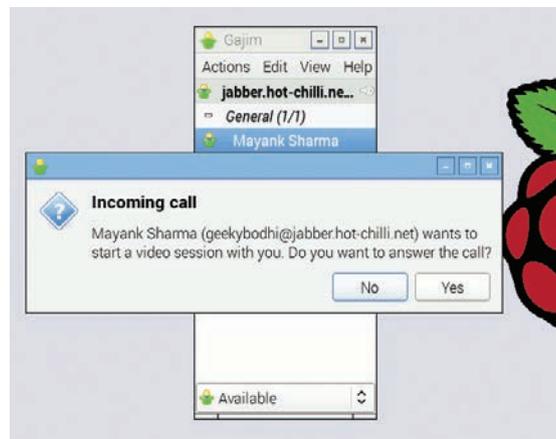
Один из лучших клиентов XMPP для Raspberry Pi, с правильным сочетанием размера и возможностей — *Gajim*. Он доступен в официальных репозиториях Raspbian, и его можно установить просто командой

```
$ sudo apt-get install gajim
```

После установки его можно привязать к учетной записи XMPP и начать обмениваться текстовыми сообщениями с друзьями. К Pi также можно

подключить видеочапу для видеозвонков с приятелями. Но для этого сначала нужно убедиться, что у вас есть пакеты *gststreamer1.0-plugins-bad* и *python-farstream*. Оба пакета доступны в официальных репозиториях Raspbian, и их можно установить с помощью менеджера пакетов *APT*. После установки *web*-камеры и необходимых компонентов в окне чата появится кнопка для запуска видеосеанса.

Если с передачей звука или видео возникают проблемы, зайдите в *Edit > Preferences > Audio/Video* [Изменить > Настройки > Звук/Видео] и убедитесь, что используется правильное устройство для вывода звука. Если у вас медленное интернет-соединение, также можно выбрать более низкое разрешение видео, в выпадающем списке *Video Size* [Размер видео].



► Для *Gajim* есть несколько интересных плагинов, например, *OTR*, который позволяет зашифровать весь обмен сообщениями.

установки утилиты создайте файл под названием *.sendxmpprc* в каталоге *home* с данными учетной записи Pi в XMPP, например, так:

```
$ nano ~/.sendxmpprc
rpi@pi@jabber.hot-chilli.net my-secret-password
```

Не забудьте заменить имя пользователя и пароль данными учетной записи, которую вы зарегистрировали для Pi. Сохранив файл, можно отправить сообщение, командой

```
$ echo "Привет, это Pi!" | sendxmpp -t geekybodhi@jabber.hotchilli.net
```

Приведенная команда отправляет сообщение от Pi идентификатору XMPP, указанному с параметром *-t*. Замените приведенный в примере идентификатор на свой идентификатор XMPP. Если вы зайдете в свою обычную учетную запись обмена сообщениями, то получите уведомление в виде обычного сообщения от учетной записи XMPP Pi.

Также можно передавать вывод команд *Bash*, например:

```
$ echo "Сегодня" $(date) | sendxmpp -t geekybodhi@jabber.hotchilli.net
```

Эта команда отправляет вывод команды *date*. Вот еще один, чуть более полезный пример:

```
$ echo $(/opt/vc/bin/vcgencmd measure_temp) | sendxmpp -t geekybodhi@jabber.hot-chilli.net
```

Эта команда опрашивает датчики температуры Pi с помощью утилит, входящих в пакет *raspberrypi-firmware-tools*. Результаты опроса мы отправляем своему обычному пользователю клиента обмена сообщениями. С помощью этой команды можно отслеживать Pi и отправлять оповещения при превышении температуры указанного порога. Скопируйте содержимое листинга 1 (см. стр. 95) или загрузите его с <http://pastebin.com/NdQw5frr> в файл *status.sh*. Затем настройте запись *crontab*, выполнив команду *crontab -e* и введя следующую строку:

```
*/5 * * * * ~/status.sh
```

Здесь мы просим Pi запускать скрипт *status.sh* каждые пять минут. Не забудьте сменить расположение файла *status.sh* на расположение в вашем Pi. Но что же происходит в скрипте *status.sh*?

Этот скрипт сохраняет температуру Pi в переменную *temp*, отделив от значения текстовый довесок и десятичную часть, так как *Bash* умеет обрабатывать только целые числа. Затем скрипт проверяет, не превышает ли температура 40°C, и если да, отправляет нам сообщение. Этот скрипт можно расширить, чтобы следить за тем, что происходит на Raspberry Pi: например, попросить отправлять сообщение при каждом появлении сообщения в лог-файле или при каждом изменении состояния демона. Скрипт *sendxmpp* помогает отслеживать, что происходит на Pi — но с этим ничего не поделаешь, пока вы не зашли на Pi. А если зайти нельзя? Если вы получили предупреждение от домашнего Pi, находясь на работе? Не лучше ли было бы также и управлять Pi с помощью сообщений?

Ваш лучший друг

WhatsApp — одно из самых по-

пулярных приложений для обмена сообщениями, и им можно воспользоваться с Raspberry Pi. Библиотека *Yowsup* для Python позволяет использовать свою учетную запись *WhatsApp* для обмена сообщениями с контактами. Когда новизна отправки сообщений от Pi своим друзьям поблекнет, вы сможете с помощью этой библиотеки отслеживать и управлять Pi, отсылая сообщения через *WhatsApp*.

Прежде чем устанавливать *Yowsup*, установите ее зависимости, скомандовав

```
$ sudo apt-get install git python-dev libncurses5-dev
```

Затем воспользуйтесь командой

```
$ git clone git://github.com/tgalal/yowsup.git
```

для загрузки библиотеки в текущий каталог, и установите ее:

```
$ cd yowsup
```

```
$ sudo python setup.py install
```

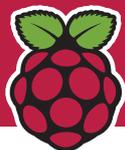
После установки библиотеки пора зарегистрировать ваш мобильный телефон в *WhatsApp*. В каталоге *yowsup* создайте файл с именем *mydetails* и следующим содержимым:

```
$ nano mydetails
cc=44
phone=447712345678
```



Вы сможете управлять Pi, отсылая сообщения через WhatsApp.

» Подпишитесь на печатную или электронную версии на www.linuxformat.ru/subscribe!



Параметр `cc` указывает на код страны, для Великобритании это 44. Замените его и номер телефона на свои код страны и номер телефона. Символ `+` указывать не надо. Затем сохраните файл и воспользуйтесь следующей командой, чтобы запросить у WhatsApp регистрационный код:

```
$ python yowsup-cli registration --config mydetails --requestcode sms
```

Через несколько секунд вы должны получить SMS на телефон, номер которого вы указали в файле `mydetails`. Сообщение будет содержать шестизначный код. Воспользуйтесь им для регистрации номера телефона в *WhatsApp*:

```
$ python yowsup-cli registration --config mydetails --register xxx-xxx
```

Замените `xxx-xxx` на свой код. Через секунду или две вы получите ответ от *WhatsApp*, который будет выглядеть примерно так:

```
status: ok
kind: free
pw: jK0zdPJ9zz0BBC3CwmnLqmxuhBk=
price: 0.89
price_expiration: 1434674993
currency: EUR
cost: 0.89
expiration: 1463544490
```

```
login: 448375972334
```

```
type: new
```

Единственное, что нам здесь интересно — это пароль в переменной `pw`. Скопируйте и вставьте его в файл `mydetails`, который теперь должен выглядеть так:

```
cc=44
phone=447712345678
password=jK0zdPJ9zz0M8G3CwmnLqmxuhBk=
```

На этом все. В библиотеке *Yowsup* есть демонстрационное приложение, которым можно воспользоваться для отправки и приема сообщений. Запустите его командой

```
$ yowsup-cli demos --yowsup --config mydetails
```

Она откроет клиент командной строки *Yowsup*. Наберите `/help`, чтобы увидеть все доступные команды. Подсказка `[offline]` означает, что вы не подключены к серверам *WhatsApp*. Воспользуйтесь командой `/L`, которая получает данные аутентификации из файла `mydetails` и подключается к серверу. После этого подсказка изменится на `[connected]`.

Теперь можно отправлять сообщения другим пользователям *WhatsApp*. Чтобы отправить сообщение на номер 449988776655, введите:

```
/message send 449988776655 "Привет, я послал это с Raspberry Pi!"
```

Если получатель ответит, его сообщение появится на консоли Raspberry Pi. Чтобы завершить сеанс, воспользуйтесь командой `/disconnect` для выхода.

Что нового, Pi?

Подлинное преимущество библиотеки *Yowsup* в том, что с ее помощью можно выполнять действия на Pi: скажем, можно отправить сообщение *WhatsApp* для проверки определенных параметров Pi, например, свободного места и температуры, а затем обновить или выключить Pi. С помощью библиотеки также можно управлять выводами GPIO и любыми подключенными периферийными устройствами — например, дверью.

Для взаимодействия с Pi можно воспользоваться скриптом Python в листинге 1 (справа). Этот скрипт слушает сообщения с заданного номера, распознает определенные ключевые слова и выполняет соответствующие действия. Так, если отправить, например, 'Привет, Pi', Pi поприветствует вас в ответ. Если Pi получит сообщение, начинающееся со слова "memory", он выполнит команду `df -h` и отправит вам результаты в виде сообщения.

В этом скрипте используются классы, созданные итальянским блогером Карло Машеллани [Carlo Mascellani]. Они находятся в двух файлах — `wasend.py` и `wareceive.py`, которые можно загрузить командами

```
$ wget http://www.mascal.it/public/wasend.py
```

```
$ wget http://www.mascal.it/public/wareceive.py
```

В том же каталоге создайте файл `pitalk.py` с содержимым, приведенным в листинге 2 (справа). Теперь создайте скрипт оболочки с именем `talktome.sh`, который вызывает скрипт Python `pitalk.py`:

```
$ nano talktome.sh
#!/bin/bash
while :
do
sudo python /home/pi/yowsup/pitalk.py
done
```

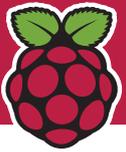
Теперь сделайте скрипт исполняемым, скомандовав `chmod +x talktome.sh`, и сделайте так, чтобы он запускался автоматически при загрузке Pi, добавив его в файл `/etc/rc.local`:

```
$ sudo nano /etc/rc.local
/home/pi/yowsup/talktome.sh
```



➤ Raspberry Pi не всегда получает сообщения, и, увы, не сделает вам мохито, как бы вы ни просили.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



Сохраните файл, перезагрузите Raspberry Pi, и скрипт запустится автоматически.

Давайте разберем скрипт, чтобы лучше его понять. Функция `credential()` вверху подключает скрипт к серверу *WhatsApp* с вашими учетными данными. Измените оба параметра этой функции. Функция `Answer()` указывает номер телефона *WhatsApp*, с которым общается наш Pi. Это важно, чтобы никто другой не мог управлять нашим Pi.

Разбор скрипта

Затем мы определяем функции, которые выполняют самую задачу опроса Pi с помощью сообщений *WhatsApp*. Например, функция `Refresh()` обновляет список репозитория, а `Restart()` перезагружает Pi. Функции `Temp()` и `Disk()` чуть сложнее. Первая получает и обрезает информацию о температуре, как описывалось ранее в этом руководстве. Функция `Disk()` аналогичным образом форматирует и перераспределяет вывод команды `df -h` для более удобного чтения.

В главной части программы (цикл `while`) скрипт ожидает сообщения, и при его получении отбрасывает исключение `MessageReceived`. Полученное сообщение начинается с номера телефона, за которым следует текст сообщения, например, "449876543210 Message".

При появлении исключения скрипт сначала преобразует всю строку в нижнем регистре с помощью метода `value.lower()`. Затем он проверяет, что сообщение пришло именно с того номера, с которого ожидалось. Если нет, скрипт добавляет сообщение в лог-файл и не отвечает на него.

А если номер телефона правильный, скрипт отрезает номер от сообщения, оставляя только текст. Затем условия `If` разбирают сообщения и решают, как на них ответить.

Чтобы вы получили представление о возможных вариантах, мы применили пять типов сравнений. Первые два ищут символы в начале текста, например, `if received[:4]=="Привет": Answer("Привет, друг!")` срабатывает, если первые четыре символа сообщения `П, р, и и в,` и отвечает "Привет, друг!". Это срабатывает, даже если получено сообщение: "Привет, Raspberry Pi, ты онлайн?" Второе условие тоже ищет символы в начале сообщения, но срабатывает при обнаружении любой из двух строк `restart` или `reboot`. Три следующих условия выполняют проверку другого типа. Они срабатывают, если соответствующий текст находится в любой части со-

общения, а не только в начале. Так, если отправить сообщение "What's the status of your disk? [Каково состояние диска?]", скрипт обнаруживает ключевое слово "disk" и вызывает функцию `Disk()`. Аналогично, если

отправить сообщение "You're not running too hot, are you? [Ты не перегрелся, а?]", скрипт обнаружит ключевое слово "hot" и ответит значением температуры с датчика. Если скрипт не обнаружит ключевых слов, он просто ответит "Eh? What was that? [Что-что?]".

Скрипт можно расширить, добавив в него огромное количество задач по автоматизации дома. Можно даже взять модуль камеры Pi и воспользоваться библиотекой Python *Picam* для создания фотографий и видео и их отправки в сообщении *WhatsApp*. Несколько примеров развертывания скрипта в полезные приложения приведены на вики-странице библиотеки *Yowsup* (<https://github.com/tgalal/yowsup/wiki>). **LXF**

Скрипт можно расширить, добавив задачи автоматизации дома.

Листинги

На нашем уроке мы используем два скрипта, которые можно найти на **LXF DVD** или по ссылке <http://pastebin.com/NdQw5frt>.

Листинг 1: status.sh

```
#!/bin/bash
temp=$(/opt/vc/bin/vcgenclm measure_temp | cut -c6-7)
if [ "$temp" -gt 40 ]; then
    echo Whoa! My temperature is up to $(/opt/vc/bin/vcgenclm
measure_temp). Power me down for a bit! | sendxmpp -t geekybodhi@
jabber.hot-chilli.net
fi
```

Листинг 2: pitalk.py

```
import os, subprocess, yowsup, logging
from wasend import YowsupSendStack
from wareceive import YowsupReceiveStack, MessageReceived
def credential():
    return "447712345678","jK0zdPJ9zz0BBC3CwmnLqmxuhBk="
def Answer(riSP):
    try:
        stack=YowsupSendStack(credential(), [(["447668139981", riSP]])
        stack.start()
    except: pass
    return
def Refresh():
    Answer("Refreshing the repos.")
    os.system("sudo apt-get -y update")
    Answer("Repos updated.")
    return
def Restart():
    Answer("Rebooting")
    os.system("sudo reboot")
```

```
return
def Temp():
    t=float(subprocess.check_output(["/opt/vc/bin/vcgenclm measure_
temp | cut -c6-9"], shell=True):-1)
    ts=str(t)
    Answer("My temperature is "+ts+" C")
    return
def Disk():
    result=subprocess.check_output("df -h.", shell=True)
    output=result.split()
    Answer("Disk space:\nTotal: "+output[8]+" \nUsed: "+output[9]+"
("+output[11]+" \nFree: "+output[10])
    return
while True:
    try:
        stack=YowsupReceiveStack(credential())
        stack.start()
    except MessageReceived as rcvd:
        received=rcvd.value.lower()
        if received[:len("447668139981")]=="447668139981":
            received=received[len("447668139981"):]
            if received[:4]=="hiya": Answer("Hi chap!")
            elif received[:7]=="restart" or received[:6]=="reboot": Restart()
            elif "disk" in received: Disk()
            elif "hot" in received: Temp()
            elif "refresh" in received: Refresh()
            else: Answer("Eh? What was that?")
        else: #message from wrong sender with open("/home/pi/
whatsapp.log", "a") as mf:mf.write("Unauthorised access from:
"+received[:len("91996813 9981")]+"\n")
```

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 Дистрибутивы, не загружающиеся с DVD
- 2 Восстановление утерянной адресной книги электронной почты
- 3 Удаление Xfce из Ubuntu
- 4 Преобразование файлов DJVU в PDF
- 5 Восстановление неработающего загрузчика
- 6 Решение проблемы с медленно работающим диском

1 Незагружаемый Debian

Когда-то моим компьютером был 8-битный Sinclair ZX81, и увидев диск с 64-битным Debian 8 на обложке LXF199, я решил посмотреть, куда зашел прогресс. У меня Viglen Genie с двухядерным процессором AMD Athlon 64 × 2 Dual-Core Pro 4200+ с частотой 2,2 ГГц и 32-битной Windows 7, и я был уверен, что он должен быть совместим с 64-битной системой. Но он даже не взглянул на Debian. Он не смог загрузить elementary OS, хотя запустил утилиту проверки памяти. При этом, как ни странно, elementary OS установилась на мой ноутбук Advent 4489.

Я даже удалил Windows XP, которая изначально была на Genie. Так что же произошло? Думаю, что 64-битным ОС необходимо 2 ГБ ОЗУ, а у меня сейчас всего 1 ГБ — не в этом ли проблема?

Майк Уимбери [Mike Wimbury]

Процессор и в самом деле 64-битный, но причина Вашей проблемы не в нем, так как elementary OS 32-битная и загрузится и на этом процессоре. Опять же, 1 ГБ оперативной памяти — по нынешним меркам немного, но это не должно препятствовать загрузке дистрибутива. Он может работать медленно, но должен загружаться. Сложно дать конкретный совет, не зная точных симптомов проблемы, но есть несколько вещей, которые можно попробовать. Первая — будьте терпеливы: иногда загрузочные скрипты Live CD могут «подвиснуть» при попытке определения конфигурации оборудования или сети. Поэтому дайте им несколько минут, прежде чем делать печальные выводы.

Если проверка памяти прошла хотя бы в двух полных фазах, и ничего не выявила, попробуйте скопировать образ Debian или ISO-образ elementary OS на диск, желательно DVD-R, так как они более надежны по сравнению с DVD-RW. Если Вы не знаете, как записать ISO-образ на диск, обратитесь к FAQ на DVD. Так Вы исключите любые возможные проблемы, вызванные Вашим достаточно старым

```
GNU GRUB version 2.00
getparams 'Try elementary OS without installing'
linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=$isofile noefect quiet splash
initrd (loop)/casper/initrd.lz
Minimum Emacs-like screen editing is supported. TAB lists completions.
```

Редактор меню Grub не из красавцев, зато позволяет менять параметры загрузки на лету.

BIOS и двухэтапным процессом загрузки, который используется на наших DVD. Также стоит проверить, доступно ли обновление BIOS для Вашей материнской платы на сайте производителя.

Попробуйте вариант с безопасной загрузкой из меню Debian. В нем не только отключаются некоторые возможности для более безопасной (хотя и более медленной) загрузки, но также отключается и экран заставки, чтобы Вы могли увидеть загрузочные сообщения. Так Вы сможете попробовать понять, почему останавливается загрузка. Если она останавливается с ошибкой, можете скопировать сообщение в поисковую систему и узнать, что способно вызывать эту ошибку. В Elementary OS нет варианта с безопасной загрузкой, но можно отключить графический экран заставки. Выделите пункт Try elementary OS without installing [Попробовать elementary OS без установки], но не нажимайте Enter, а вместо этого нажмите e для изменения параметров загрузки. В строке, которая начинается с linux, удалите quiet splash, затем нажмите Ctrl+x или F10 для загрузки. Так Вы получите экран с подробными загрузочными сообщениями, который по меньшей мере скажет Вам, где загрузка останавливается, даже если Вы не видите явного сообщения об ошибке.

2 Утерянные адреса

Я использую Claws Mail для отправки новой рассылки благотворительной группе. Недавно я перераспределил получателей в группы по пятьдесят — это максимальное количество, которое разрешает мой провайдер. Заодно я более логично отсортировал свои личные

файлы. Перед закрытием программы я воспользовался функцией Backup моего настольного Mint для резервного копирования каталога .claws-mail на внешний жесткий диск. Когда на следующий день я запустил программу, то увидел сообщение о том, что адресная книга не может быть прочитана. Я закрыл программу и запустил ее снова. Она открылась нормально, но адресная книга была пуста. В адресной книге ~/.claws-mail/ есть сорок файлов с именами вида addrbook-00000x, где x — целое число. Одни файлы имеют расширение .xml.bak, другие — .xml. Вторые открываются в Firefox, а первые — в текстовом редакторе, и оба содержат адреса вида <person uid=""first-name=""last-name=""...etc.../>. В кавычках содержатся соответствующие адреса и имена. В резервной копии на внешнем жестком диске содержится дублирующий набор файлов. Можете сказать, как вернуть файлы обратно в адресную книгу, не копируя записи по одной (что займет не один день)? И что могло стать причиной проблемы?

Сбуфер, с форумов

Claws Mail считывает различные конфигурационные файлы и файлы данных в память при запуске и записывает обратно при выходе. Если Вы хотите выполнять любые операции в файлах в каталоге ~/.claws-mail, будь то изменение или резервное копирование, это нужно делать, когда Claws не запущен, в противном случае Ваши изменения будут перезаписаны, а резервная копия станет неактуальной.

Нумерованные файлы .xml содержат различные адресные книги, а файлы .xml.bak — резервные

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**.

копии предыдущих версий. У Вас также должен быть файл **addrbook--index.xml** — он сообщает Claws, какие адресные книги для чего используются. Если у Вас его нет, воссоздайте его следующим образом:

```
<?xml version="1.0" encoding="UTF-8" ?>
<addressbook>
<book_list>
<book name="first book name"
file="addrbook-000001.xml" />
```

```
<book name="secondbook name"
file="addrbook-000002.xml" />
...
</book_list>
</addressbook>
```

Чтобы увидеть имя каждой адресной книги, взгляните в ее файл. Имя адресной книги находится во второй строке.

Возможно, есть лучший способ сделать то, что Вы хотите. Провайдеры и почтовые шлюзы не одобряют групповой рассылки почты — она может быть спамом, особенно если происходит с IP-адреса, выданного домашнему пользователю (подумайте о компьютере с Windows, который заражен спам-ботом). А если Ваш провайдер введет еще более строгие ограничения? Хотите Вы разбивать свою адресную книгу еще раз и прилагать еще больше усилий для отправки каждой рассылки? Проще было бы настроить список рассылки, тогда Вы отправите письмо один раз, и каждый подписчик получит копию. Существует множество бесплатных сервисов почтовых рассылок, например, Группы Google. Вы можете создать список рассылки несколькими щелчками мыши и настроить рассылку так, чтобы ее могли производить только определенные пользователи (это позволит кому-то другому выполнить рассылку, если Вы будете недоступны). Список рассылки может быть открытым для всех или доступным только по приглашениям в зависимости от Ваших предпочтений.

3 Удаление Xfce

В Я установил рабочий стол Ubuntu в свежей версии Ubuntu Studio 14.04 и хотел бы удалить Xfce. Я пробовал делать это раньше, но иногда после удаления HUD не может найти ни одного установленного приложения. Как правильно удалить Xfce из Ubuntu Studio?

globoetrotterdk, с фопумов

Для различных вариантов рабочих столов Ubuntu использует метапакеты. Метапакет — это такой пакет, который не устанавливает ничего важного, но обладает зависимостями от настоящих пакетов, которые нужны для этого рабочего стола — фактически это список пакетов для установки. Поэтому сначала Вам надо установить метапакет, который для Xfce называется *xubuntu-desktop*. Пакеты можно найти в графическом менеджере пакетов или воспользоваться *apt-cache* в командной строке:

```
$ apt-cache search -n -- -desktop
```

Двойной дефис необходим, чтобы *apt-cache* не пыталась интерпретировать *-desktop* как список параметров. Для большинства команд оболочки оно означает, что команда должна перестать обрабатывать параметры и считать все остальное обычным аргументом.

Разумеется, раз пакет *xubuntu-desktop* ничего не устанавливает, то при его удалении ничего не удаляется, но системе больше не нужны его зависимости, и их можно удалить с помощью *apt-get*:

```
$ sudo apt-get --dry-run autoremove
```

Эта команда выведет список удаляемых пакетов, однако не удалит их. Проверьте, что в списке нет ничего, что Вам хотелось бы оставить, и затем запустите команду без параметра *--dry-run*. Можно также добавить параметр *--purge*, который удалит конфигурационные файлы пакетов, а также программ и документации (по умолчанию конфигурационные файлы сохраняются, чтобы в будущем Вы могли переустановить пакет и сохранить настройки).

Разрешая системе управления пакетами решить, что нужно удалять, Вы снижаете риск удаления чего-то используемого другими программами — именно это, похоже, произошло с Вами. Если Вы все же переживете собой, всегда можно переустановить *xubuntu-desktop*, который установит



Коротко про...

Разбираемся со случайными «падениями»

Ничто не выводит из себя больше, чем неожиданные «падения» компьютера. Если «падение» можно воспроизвести, у вас есть стартовая точка, которую можно исследовать; ну, а если это происходит в случайные моменты времени и с любыми программами? Если это происходит независимо от используемых программ, то проблема может быть аппаратной. При наличии запасного блока питания попробуйте заменить им свой. Блоки питания настольных компьютеров сильно различаются по качеству, и более дешевые из них способны вызывать сбои питания, которые приводят к «падению» системы. Они также способны повредить компоненты, которые стоят в несколько раз дороже, чем новый блок питания. Если у вас ноутбук, возможно, проблема

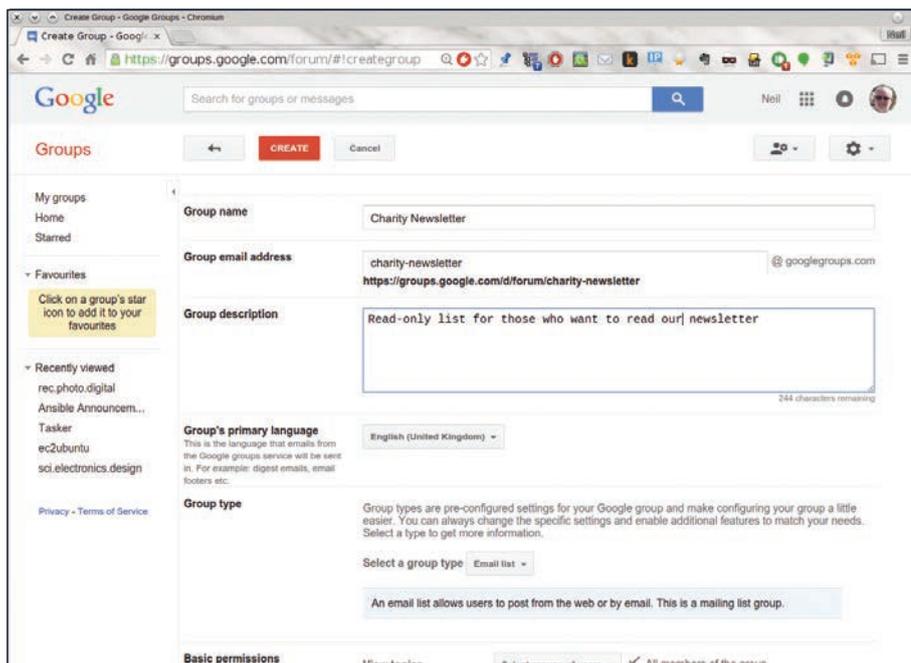
в аккумуляторе. Если ноутбук не работает нормально только от батареи, замените ее.

Другая распространенная причина — проблема с неисправной памятью. Неисправна не вся память, а всего несколько байт. Большую часть времени всё работает прекрасно, пока кто-нибудь не попытается обратиться к этим байтам, и... БАБАХ! Тестировать память на рабочей системе ненадежно, потому что вы не сможете протестировать все, что может потребоваться системе; воспользуйтесь Memtest86+. Это загружаемая система, которая задействует минимальный объем оперативной памяти, оставляя остальное для тестирования. Она есть на нашем LXF DVD в последнем пункте меню загрузки.

Memtest86+ выполняет исчерпывающий набор тестов, которые занимают определенное

время, но даже этого недостаточно. Так как проблемы с памятью могут появляться и исчезать, то за один проход их можно упустить. Поэтому выполните тестирование как минимум в два прохода и желательно ночью.

Если ни один из проходов ничего не показал, возможна проблема с дистрибутивом. Если у вас популярный дистрибутив, то проблема вряд ли в нем, если она не коснулась других пользователей. Еще одна возможная причина — дурное напряжение электросети: скачки напряжения вверх или вниз могут вызвать перезагрузку. Проблему может решить какое-нибудь устройство сглаживания напряжения, например, ИБП. Его также стоит приобрести, если вы планируете использовать компьютер для серьезной работы.



➤ Организовать рассылку для большого числа получателей проще с помощью выделенного почтового сервера.

все остальные пакеты, и система вернется в рабочее состояние. Программные зависимости бывают сложными, и хотя очистка системы от всего ненужного ПО приносит удовлетворение, иногда проще оставить все как есть и сэкономить свое время, а не несколько мегабайт дискового пространства — если только у Вас не маленький SSD и место на диске очень важно.

4 Преобразование в PDF

Несколько лет назад я прочитал в Linux Format о *gscan2pdf* и с тех пор пользуюсь им для архивации документов. Я сохранял их в формате DJVU, так как он гораздо компактнее всего, что я видел. Побольше бы только программ и систем поддерживали их... Сейчас мне надо отправить несколько файлов тому, кто не сможет их прочитать. Я знаю, что могу открыть их в *gscan2pdf* и снова сохранить в PDF, но не хочу усугублять ломоту в кистях из-за постоянного щелкания мышью. Можно ли сконvertировать все файлы группой в терминале или в графической программе? Куда можно ввести весь список файлов, чтобы он сконvertировался сразу? У меня Ubuntu 15.04, если эта информация играет какую-то роль.

Стюарт Уилсон [Stuart Wilson]

○ DJVU — это примечательный формат, и он весьма неплохо поддерживается в Linux, *Okular* считывает и обрабатывает их гораздо быстрее по сравнению с аналогичными документами в PDF. Я также нашел несколько программ просмотра для Android — я пользуюсь *AnDoc*, которая поддерживает как PDF, так и DJVU. Конvertировать файлы с командной строки очень просто. Так как Вы пользуетесь *gscan2pdf*, то у Вас уже установлены инструменты DJVU. Преобразование выполняется в два этапа:

сначала с помощью *djvups* файл преобразуется в PostScript, затем с помощью *ps2pdf* — в PDF. Это можно проделать для каждого файла:

```
$ djvups somedoc.djvu somedoc.ps
$ ps2pdf somedoc.ps somedoc.pdf
$ rm somedoc.ps
```

Однако обе эти программы работают с каналами, и можно отправить преобразованные данные от одной к другой, не создавая файл PostScript:

```
$ djvups somedoc.djvu | ps2pdf - somedoc.pdf
```

При запуске *djvups* без имени выходного файла она выводит файл в формате PostScript в стандартный вывод. Он направляется в *ps2pdf*. Дефис “-” вместо имени входного файла означает, что файл надо считать из стандартного ввода. Команда по-прежнему конвертирует файлы по одному, но с помощью небольшого волшебства оболочки мы можем сконvertировать все файлы DJVU в текущем каталоге следующим образом:

```
$ for f in *.djvu
$ do
$ djvups $f | ps2pdf - ${f%*.djvu}.pdf
$ done
```

Этот скрипт запускает *djvups* и *ps2pdf* для каждого соответствующего файла в каталоге. Фрагмент в конце команды *ps2pdf* отрежет *.djvu* от имени входного файла и заменит его на *.pdf*. Вот так Вы и получите все файлы DJVU и их эквиваленты в PDF (гораздо большего объема) в том же каталоге.

5 Поломанный загрузчик

В У меня есть старый ноутбук с Windows XP, на который я несколько лет назад установил Linux Mint 8. Да, я же сказал, что он старый! Теперь я хочу обновиться до Mint 15, установочный диск которого у меня есть. Я переформатировал старый раздел с Mint 8, готовясь его заменить. Но теперь при перезагрузке ноутбука *Grub* выдает сообщение с ошибкой 17, и похоже,

дальше мне не сдвинуться. Подозреваю, что мне не стоило удалять раздел, который был первым пунктом в списке загрузки — но как решить эту проблему? У меня нет диска восстановления для Windows XP, даже если бы механизм с двойной загрузкой позволил бы им воспользоваться, и мне хочется избежать полного форматирования жесткого диска, для которого у меня есть диски восстановления.

Дэвид Бью [David Bew]

○ Как Вы и заподозрили, Вы, видимо, удалили раздел, который содержал файлы, используемые *Grub* для загрузки компьютера. Начальный код *Grub* находится в первом блоке диска, главной загрузочной записи (MBR), и потом вызывает другой код с этого раздела, поэтому *Grub* все-таки загружается, но потом показывает ошибку. Никакой крайности удалять раздел для установки более нового Mint не было: можно было просто загрузиться в более новый установщик Mint (кстати, почему у Вас версия 15, а не текущая 17.2?) и велеть ему выполнить установку в раздел Mint 8. Но, каким бы хорошим ни было это суждение задним умом, сейчас загрузчик не работает, и его нужно починить.

Отсутствие раздела не позволяет загрузиться только с жесткого диска, поэтому Вы все еще должны загрузиться с диска Mint и установить последнюю версию, в которой будет новый загрузчик, и это должно помочь Вам обойти проблему. Другой вариант — восстановить загрузчик Windows, хотя в этом есть смысл, только если на диске нет других дистрибутивов Linux. Загрузитесь с диска аварийного восстановления, войдите в оболочку восстановления и выполните команду

```
fdisk /mbr
```

Эта команда предназначена исключительно для Windows XP — в более поздних версиях Windows используются другие команды. Если у Вас под рукой нет установочного диска Windows или Вы предпочитаете графические программы, загрузите *Rescatux* с www.supergrubdisk.org/rescatux, запишите его на диск и загрузитесь с него. Вам понадобится бета-версия (на данный момент, 0.32 beta3). После загрузки Вы увидите кнопку *Restore Windows MBR* [Восстановить загрузчик Windows] — возможно, чтобы ее увидеть, придется прокрутить окно. После этого компьютер должен загрузиться в Windows как обычно, а потом, когда у Вас будет время, Вы сможете вернуться к установке Linux.

6 Медленный диск

В Мой жесткий диск, кажется, работает медленнее, чем должен. Есть ли какая-то программа, с помощью которой я смогу измерить его производительность? Понимаю, что лучше было бы сделать это сразу при покупке компьютера, чтобы сравнить, но может ли что-то помочь?

Эрик Фостер [Eric Foster]

○ Сам жесткий диск вряд ли может замедлиться, если только в нем не возникла какая-то неисправность. Проверить это

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 устройства, которые у вас установлены.

Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://sourceforge.net/hardinfo.berlios>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму, отправляемому нам.

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 приведенные ниже 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
```

на фрагментированность. Если она сообщает, что диск фрагментирован слишком сильно [*Red:* — Не делайте этого на SSD.], запустите команду `sno`, без параметра `-c`. **LXF**

Version 1.97		Sequential Output			Sequential Input			Random	Sequential Create			Random Create													
Size	Per Char	Block	Rewrite	Per Char	Block	Seeks	Num Files	Create	Read	Delete	Create	Read	Delete												
K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	Ave	% CPU	Ave	% CPU	Ave	% CPU												
32080M	154	23	73198	3	33213	3	2605	42	60702	2	64.1	2	16	5322	11	+++++	+++	14089	41	21530	31	+++++	+++	25664	57
Latency	3698ms	40824us	4122ms	2475us	2062ms	2377ms	Latency	18279us	241us	446us	155us	15us	1809us												

Если *hdparm* не предоставляет достаточно подробных данных о производительности диска, *Bonnie++* даст вам полную информацию о работе жесткого диска и файловой системы.

можно утилитой *smartctl*, которая входит в пакет *smartmontools*, выполнив следующую команду:

```
$ sudo smartctl --test=long /dev/sda
```

Она сообщит Вам, сколько времени потребуется на выполнение теста. Переждите это время, а затем выполните команду

```
$ sudo smartctl --log=selftest /dev/sda
```

чтобы увидеть результаты. Если Вам нестерпимо, добавьте `--test=short`, но это сэкономит время только в том случае, если неисправность будет обнаружена. В противном случае Вам придется выполнить полный тест, чтобы быть уверенным в результате.

Что касается проверки скорости, базовые тесты производительности выполняет утилита *hdparm*. Ей нужно передать устройство, на котором находится файловая система Linux, например:

```
$ sudo hdparm -tT /dev/sda2
```

Для более глубокого тестирования можно воспользоваться *bonnie++*, которая должна быть в менеджере пакетов Вашего дистрибутива:

```
$ bonnie++ -d /directory/to/check
```

Указанный каталог может располагаться где угодно, но лучше всего создать пустой каталог специально для этой цели. Если Ваш дистрибутив установил программу в `/usr/sbin`, то для ее запуска может потребоваться указать полный путь: `/usr/sbin/bonnie++`. *Bonnie++* не обязана запускаться с правами root: если Вы попытаетесь это сделать, программа настаит на указании имени пользователя

с параметром `-u`. Эти тесты следует выполнять так, чтобы диск в это время не использовался никакой другой программой, в противном случае результат будет занижен. *Bonnie++* предоставляет подробные результаты тестирования, в которых есть смысл только при сравнении их с другими, например, с результатами для нового диска. Однако программа также бывает полезной, когда Вы пытаетесь улучшить производительность своего диска и хотите увидеть, реальны ли субъективные улучшения.

Если *smartctl* не показала ошибок, и в частности, при наличии хороших результатов *hdparm* — возможно, дело в замедлении файловой системы, а не диска, где она находится. Такое случается, когда файловая система фрагментирована, особенно при работе с большими файлами. Файловые системы Linux успешно поддерживают свою дефрагментацию, но если файловая система переполнена, дефрагментация почти неизбежна. Если у Вас ext4 (а в большинстве дистрибутивов она стоит по умолчанию), проверьте фрагментацию с помощью *e4defrag*:

```
$ sudo e4defrag -c /dev/sda1
```

При передаче устройства таким образом будет проверена вся файловая система. Другой вариант — указать каталог, и тогда будут проверяться только файлы в этом каталоге. Если это Ваш домашний каталог, `sudo` использовать не надо. Эта команда проверяет файловую систему



Коротко про...

Компиляция программ из исходников

В большинстве случаев все необходимые программы можно найти в репозиториях пакетов вашего дистрибутива. Даже если того, что вам нужно, нет в основных репозиториях, требуемые пакеты часто имеются во внешних репозиториях, таких как персональный архив пакетов (PPA) Ubuntu, который можно добавить к списку источников своего менеджера пакетов.

Иногда того, что вам надо, в менеджере пакетов нет — либо потому, что для нужной вам программы не было выпущено пакета для вашего дистрибутива, либо потому,

что вам нужна версия новее, чем имеющаяся в вашем дистрибутиве. В этих случаях придется заказать рукава и скомпилировать программу из исходников, что совсем не так страшно, как кажется. Вам незачем знать, как писать программы — программа уже написана. Остается лишь выполнить несколько инструкций.

Исходный код поставляется в виде архивов, файлах вида **foo-1.2.3.tar.gz**. Этот архив содержит все исходные файлы для программы *foo* версии 1.2.3. При распаковке архива обычно создается каталог **foo-1.2.3**, в котором вы найдете

все исходные файлы и некоторую документацию. Самые важные файлы обычно называются **readme** и **install** и рассказывают все, что надо знать для установки программы. Если этих или похожих файлов нет, проверьте документацию на сайте проекта. Первый шаг — прочесть эти файлы и найти любые зависимости, чтобы их можно было установить из менеджера пакетов.

Затем можно выполнить указания по компиляции, которая обычно состоит из трех этапов: настройка, компиляция и установка. С типовой системой *autotools*, используемой в большинстве ПО с открытым

исходным кодом, можно запустить команды

```
$ cd foo-1.2.3
$ ./configure
$ make
$ sudo make install
```

Если эти команды говорят об отсутствующей библиотеке, а вы установили ее из менеджера пакетов, вернитесь и установите сопутствующий пакет `-dev` или `-devel`: они необходимы для компиляции. Команда `make` скомпилирует исходный код в исполняемые файлы программы. Наконец, команда `make install` помещает эти файлы в соответствующие каталоги вашей файловой системы.



LXF HotPicks



Александр Толстой
приправляет острым пикантным соусом (бесплатным) приложения этого месяца — вашего удовольствия ради.

OpenMediaVault » gCAD3D » GNU Coreutils » Shotcut » Owncloud » Geany
» Evilwm » Minetest » OpenClonk » Fish » Calamares

Программа NAS

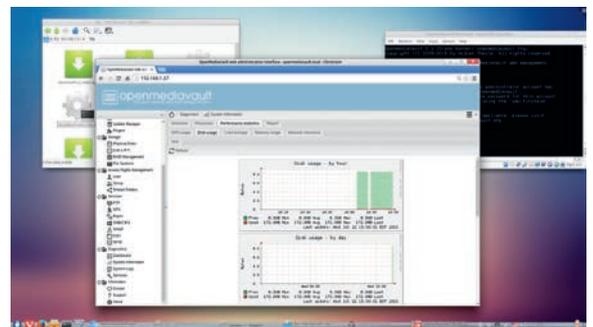
OpenMediaVault

Версия: 2.1 Сайт: <http://openmediavault.org>

Любой, кто собирает на своем жестком диске фильмы и изображения, очень скоро начнет подумывать об использовании NAS (Network Attached Storage, сетевое хранилище), и хотя оно может показаться сложным, на самом деле это не так, благодаря OpenMediaVault, специальному решению, которое легко справляется с этой задачей. OpenMediaVault — это спедистрибутив Linux на базе GNU/Debian, который вместил SSH, (S)FTP, SMB/CIFS, медиа-сервер DAAP, rsync, клиент BitTorrent и все компоненты, необходимые для превращения вашего устройства в NAS. OpenMediaVault предназначен для дома или небольшого офиса,

то есть его можно установить на старый ПК, а не на мощный компьютер. Фактически, OpenMediaVault может работать даже на виртуальной машине, если вы переместите внешнее устройства хранения из системы хоста в гостевую систему.

Развернутая система доступна из традиционного web-интерфейса, который выглядит как опрятная страница приветствия с деревом категорий слева и разбитым



» Виртуальное устройство NAS развертывается за минуты.

OpenMediaVault 2.1 обновил панель управления и виджеты.

на категории контентом справа. В OpenMediaVault имеется солидный набор опций управления хранением, в том числе управление RAID и SMART, поддержка *OwnCloud*, расширенное управление пользователями и группами и десятки других удобных опций управления. OpenMediaVault 2.1 обновил панель управления и виджеты и улучшил движок внутреннего сетевого интерфейса, поддержку VLAN и Wi-Fi, и разнообразные отладки интерфейса. Кроме того, система теперь использует локальное хранение браузера вместо cookies для кэширования пользовательских настроек интерфейса. Все это упаковано в преднастроенные образы для архитектуры x86 и x86_64, и каждый образ весит около 400 МБ.

Проект OpenMediaVault отличается множеством пересечений с FreeNAS, откуда он изначально и ответвился. Однако у него есть явные преимущества, например, программа установки дополнений и обновлений (вместо общего переписывания в FreeNAS) и более отзывчивая web-страница. Последнее достигается посредством использования технологий ExtJS и Ajax в противоположность Python+Django во FreeNAS. В реальности это означает, что вам не придется постоянно обновлять web-страницу, поскольку все изменения данных применяются мгновенно в режиме live.

Исследуем интерфейс OpenMediaVault...

Доступ к OpenMediaVault

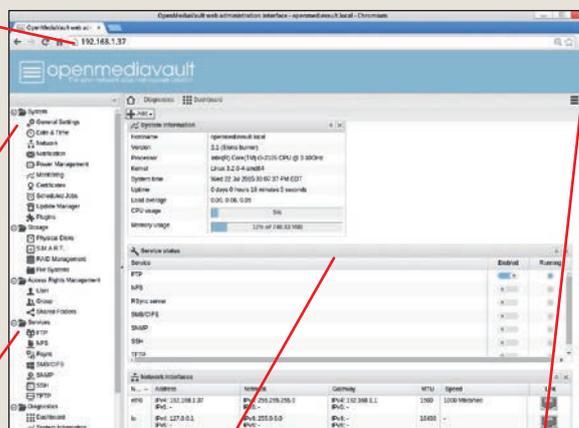
Введите верный IP-адрес, для первого входа имя пользователя admin с паролем OpenMediaVault.

Основное дерево

Навигация осуществляется через дерево категорий со множеством функций, и там вы сможете настроить практически всё.

Опции распределенного доступа

Сервисы сетевого доступа являются центральной частью подхода NAS к хранению данных.



Основная область

Панели можно перетаскивать и изменять их расположение, чтобы вместить как можно больше данных на экран.

Быстрые действия

Эта крошечная кнопка дает быстрый доступ к опциям выхода/перезагрузки и настройкам языка.

Программа CAD

gCAD3D

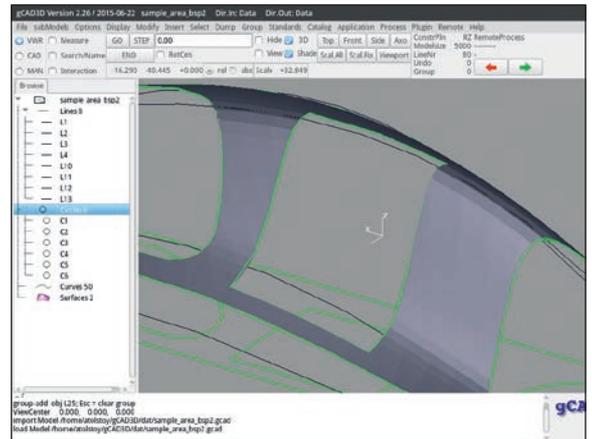
Версия: 2.26 Сайт: www.gcad3d.org

Это приложение 3D CAD/CAM (Computer-Aided Design and Manufacturing) для GNU/Linux, которое позволяет готовить, создавать, визуализировать и анализировать данные и легко взаимодействовать с моделями вызова, такими как HTML и HREF. Вы также можете использовать *gCAD3D* для быстрого взаимодействия с программы вызова, изменения параметров модели, изменения области просмотра, отображения 3D-моделей и изображений (растровых), а также 2D-чертежей. Однако важной подвигкой является выход *gCAD3D* под GPL3 и размещение ее кода на <https://github.com/gcad3d> (именно поэтому мы и включили его в HotPicks).

В интерфейсе *gCAD3D* содержится масса кнопок управления, но он не выглядит перегруженным. На левой панели — все параметры, точки, линии, кривые, плоскости и поверхности. Выбрав здесь нужный вам элемент, вы можете изменить его в области ниже.

В верхней панели — множество опций, влияющих на режим отображения и параметры рендеринга. Все это окружает основную область, где создается, масштабируется и поворачивается модель.

Чтобы приступить к работе, нужны базовые навыки работы в CAD, однако очевидно, что отправной точкой будет изучение создания каркасов и наложения на них поверхностей. Чтобы помочь вам сделать первые шаги, на официальном сайте размещена очень полезная и удобная документация. Каждый раздел являет собой небольшой урок, где вы узнаете, как работать с интерфейсом программы, потом создаете простые формы и постепенно добираетесь до более сложных конструкций. И, наконец, вы можете или сохранить результаты



➤ Теперь приложение CAD выводит под лицензией GPLv3.

в одном из множества форматов CAD, таких, как DXF, IGES, STEP, VRML, или экспортировать его в BITMAP.

Приложение предлагает двоичные Deb-пакеты для 32- и 64-битных дистрибутивов семейства Ubuntu и Debian, но для других дистрибутивов Linux нам пакетов найти не удалось, так что вам, возможно, придется компилировать приложение из исходника. В файле *readme* на странице *gCAD3D* в GitHub вы найдете инструкции.

В интерфейсе масса кнопок управления, но он не перегружен.

Основные системные утилиты

GNU Coreutils

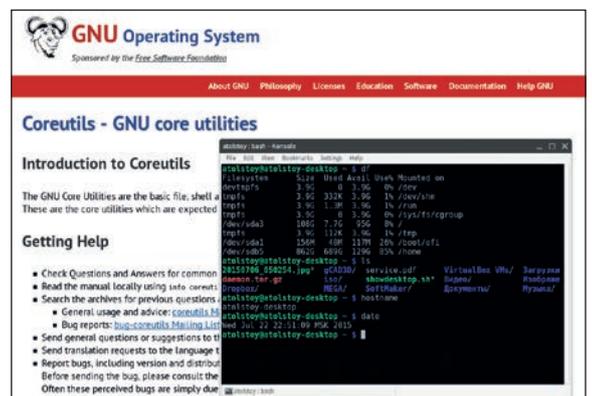
Версия: 8.24 Сайт: <http://bit.ly/GNUcoreutils>

В Unix всё является файлом, и почти каждый файл принадлежит некому пакету. Это краткое описание файловой системы, которую вы можете видеть в любом дистрибутиве. Во многих случаях можно угадать имя пакета, узнав его основные файлы, но иногда его легко забыть, когда вы настолько сжились с командами. Пакет GNU *coreutils* [Основные утилиты], например, имеется во всех дистрибутивах Linux и в большинстве Unix-подобных систем, но вы, вероятно не все, будете вспоминать имена команд, которые в нем имеются, такие, как *sort*, *cat*, *chmod*, *chown*, *chroot*, *cp*, *date*, *dd*, *echo*, *hostname*, *ln*, *ls* и т.д., все они поддерживаются под эгидой проекта GNU (см. также стр. 62).

Спустя почти год разработки, отладок и версий, вот основное из того нового, что появилось в *coreutils 8.24*. Команда *chroot* теперь поддерживает флаг *--skip-chdir*, который оставляет директорию по умолчанию нетронутой после запуска *\$ chroot*,

без перехода по умолчанию на /. Команда *dd* обрела долгожданную (в течение не одного десятка лет) функцию показа состояния записи с новым уровнем *status=progress* для печати в *stderr* статистики передачи данных приблизительно каждую секунду. Команды *cp*, *install* и *mv* ускорили действия с пустыми областями внутри файлов, и теперь они обрабатываются на низком уровне файловой системы.

Были реализованы некоторые другие изменения в поведении, с более точной обработкой жестких ссылок, лучшим копированием неполных файлов и отведением места на диске. Для тех, кто предпочитает работать с командной строкой, новые дополнения к GNU *coreutils* покажутся весьма



➤ Довольно трудно представить себе всю мощь утилит GNU, но поверьте, они потрясающие!

привлекательными. Еще один пример широкого использования — удаленное администрирование, которое часто включает *stty* и команды обработки файлов при работе через сеть. Чтобы насладиться всеми прелестями новой версии, вам всего лишь надо обновиться — или дождаться обновления. Пакет называется *coreutils* во всех дистрибутивах Linux, так что вам стоит разузнать в своем менеджере пакетов, появилась ли в вашем дистрибутиве самая последняя версия.

С точной обработкой жестких ссылок и лучшим отведением места.

Видеоредактор

Shotcut

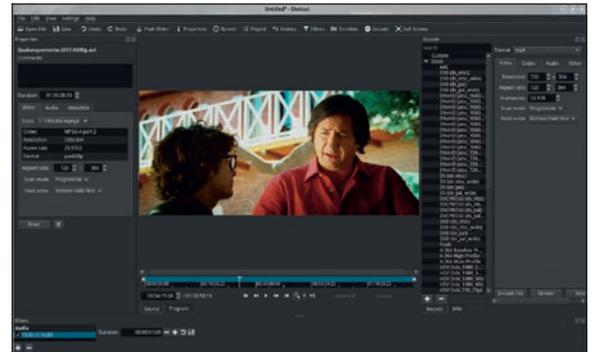
Версия: 15.07 Сайт: www.shotcut.org

В Linux множество видеоредакторов на инструментариях *GTK* и *Qt*, однако приложения на *Qt* имеют заметную тенденцию быть более продвинутыми и предлагать функции профессионального уровня. Вероятно, самым солидным видеоредактором является *Kdenlive*, однако на сей раз в HotPicks мы осветим другой видеоредактор на базе *Qt* — *Shotcut*. Это модульное приложение в темных тонах, созданное с помощью *MLT (Media Lovin' Toolkit)*, также использованного в *Kdenlive* и *OpenShot* и движка *Ffmpeg*.

У *Shotcut* столько панелей и наборов инструментов, что основная область со своими кадрами и временной шкалой занимает куда меньше места, чем обычно, и именно так всё и задумывалось. *Shotcut* создали в 2012 г. с целью добиться простоты *Kino* (покойся с миром... см. www.kinodv.org). В *Shotcut* немало хорошего — например, редактирование в режиме live (не требуется никакого импорта), обработка изображений на основе OpenGL GPU, моментальные

снимки экрана и с web-камеры, воспроизведение потока в сети (HTTP, HLS, RTMP, RTSP, MMS, UDP) и многое другое, хотя большинство этих функций вы отыщите и у его конкурентов. Однако Дэн Деннеди [Dan Dennedy], главный разработчик *Shotcut*, указывает, что его приложение более стабильное и требует меньше знаний, чем *Kdenlive* и иже с ним.

Shotcut поддерживает также высокий темп разработки и предлагает множество крутых новых функций в каждом релизе. Например, версия 15.07 предлагает поддержку разрешения 4K UHD, пять новых фильтров для имитации старых фильмов, новый эквалайзер и аудиофильтры, лучшее кодирование H.264 (с изменяемым битрейтом по умолчанию) плюс поддержку



➤ *Shotcut* поддерживает быстрый темп разработки.

копирования и вставки для временной шкалы. Этот редактор действительно стоит попробовать, что довольно легко сделать благодаря Деннеди, который терпеть не может размещать пакеты *Shotcut* в репозиториях для определенных дистрибутивов Linux. Вместо этого имеется парочка статических сборок в формате TAR.BZ2, доступных на сайте проекта. Установки не требуются: вы просто распаковываете скачанный архив и запускаете приложение.

Shotcut создали в 2012 г. с целью добиться простоты Kino.

Частное локальное облако

OwnCloud

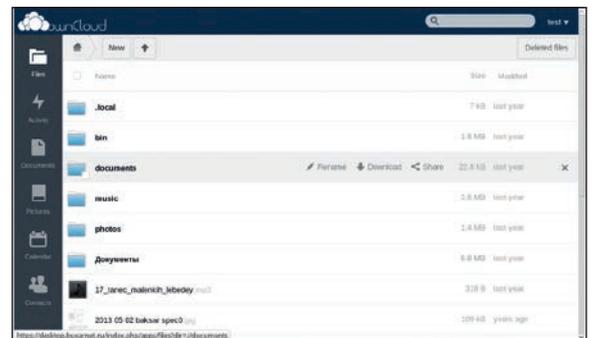
Версия: 8.1 Сайт: <https://owncloud.org>

Мы очень любим *OwnCloud*, решение для хостинга персональных файлов, которое выглядит и работает, как удаленный облачный сервис. *OwnCloud* позволяет использовать новейшую технологию без риска утратить контроль над безопасностью данных, конфиденциальностью, объемом дискового пространства и т. д.

Чтобы запустить *OwnCloud*, лучше всего иметь отдельный компьютер, который вы сможете превратить в сервер. Если лишнего компьютера у вас нет, попробуйте настроить персональное облако внутри виртуальной машины. С технической точки зрения вам нужны: установленная и работающая база данных *MySQL*; сервер NTP для синхронизации времени; web-сервер *Apache*, плюс ряд зависимостей PHP 5 для самого *OwnCloud*. Однако особых навыков все это от вас не потребует, и процесс сопровождается исчерпывающей документацией.

Когда вы его настроите, адрес `localhost/owncloud` откроет web-интерфейс *OwnCloud*, где вы можете создать учетную запись и войти. Web-интерфейс выглядит весьма стандартным CMS с деревом категорий слева и содержимым текущей категории справа.

OwnCloud поддерживает массу отличных функций, таких, как предварительный просмотр документа и движок эскизов практически для всего (включая шрифты и изображения RAW), редактирование совместных файлов, доступ к WebDAV и многое другое. *OwnCloud* также расширяется дополнительными приложениями и синхронизирован с внешними сервисами, вроде Google Drive и Dropbox, или для



➤ Храните, просматривайте и работайте со своими файлами в *OwnCloud*, собственном облаке.

воспроизведения музыки или отображения контента в сетевом менеджере файлов. Репозитории приложений весьма обширны и доступны на <https://apps.owncloud.com>. Список вариантов использования *OwnCloud* можно продолжать до бесконечности, включая доступ из Интернета, управление из приложения Android, шифрование файлов, хранение версий файлов и отмену удаления, LDAP и интеграцию Active Directory, и т. д.

Новая версия 8.1 обещает более быструю загрузку и скачивание файлов, лучшее управление шифрованием файлов и возможность редактирования офисных документов с помощью WebODF, который поддерживает аннотации и работает с большим количеством текстовых стилей.

Расширяется, и синхронизирован с внешними сервисами.

Текстовый редактор и IDE

Geany

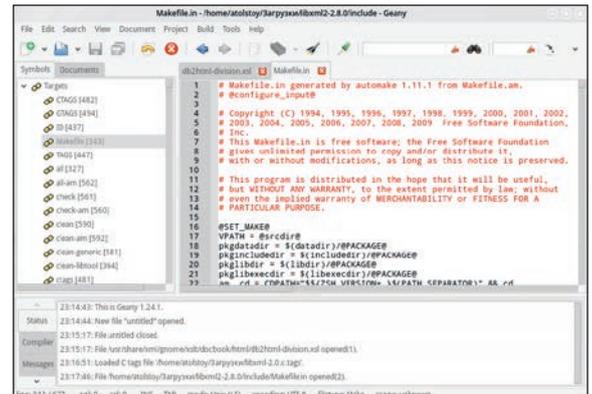
Версия: 1.25 Сайт: www.geany.org

В Linux такое количество текстовых редакторов, что порой их простиительно и перепутать. Одни редакторы нацелены на простоту, другие пытаются предложить дополнительные инструменты или занять особую нишу. *Geany* можно использовать в качестве обычного редактора, как *KWrite* или *Gedit*, однако его сильная сторона — функции интегрированной среды разработчика (Integrated Development Environment, IDE). Фактически, *Geany* находится где-то между расширенным текстовым редактором и простой IDE с очень небольшим числом внешних зависимостей помимо *GTK2* и *Vala*.

Интерфейс его тоже весьма аскетичен; здесь много вкладок, которые упаковывают множество функций в минималистскую раскладку. Символы и деревья документов слева, сообщения о состоянии [Status messages] и прочие категории вывода live размещаются внизу, а основная область показывает текущий документ и другие открытые документы в отдельных вкладках.

Geany автоматически выделяет синтаксис кода для большинства языков и поддерживает тэги кода. Это интересная функция, которая соотносит символы в компилированной программе с соответствующими объявлениями и определениями в исходном файле. Хотя вы и сами можете создать тэги из любого исходного файла, здесь имеется впечатляющая библиотека готовых к использованию тэгов для десятков проектов с открытым кодом (см. список на <http://wiki.geany.org/tags/start>).

Geany предназначен для написания кода и его немедленной компиляции, как принято в более тяжелых IDE. Здесь имеется меню Build [Сборка] с ярлыками для задач компиляции, обычного *make* и *GCC*, причем их можно настроить индивидуально,



► *Geany* надежный и мощный, и при этом его GUI не захлаплен.

например, указав другой компилятор или задав местоположение сборки. Кодируя, вы получите удовольствие от множества удобных улучшений, таких, как автоматический отступ, простое выполнение *grep*, перемещение внутри файла с помощью маркеров, и т. д.

Geany 1.25 предлагает также много новых плагинов, например, *Automark*, *Geany-Ctags*, *GitChangeBar* и *Overview*, вместе с рядом отладок. Мы даже рискнем заявить, что *Geany* — достойный соперник *Notepad++*, и это — обязательное ПО для каждого разработчика Linux.

Рискнем заявить, что Geany — достойный соперник Notepad++.

Менеджер окон

Evilwm

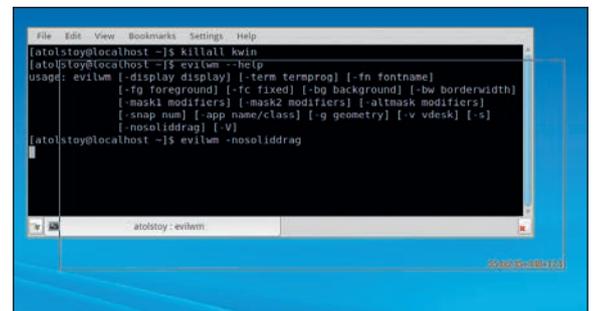
Версия: 1.1.1 Сайт: www.6809.org.uk/evilwm

Менеджеры окон часто хороши для наведения красоты на скриншоты, для прекрасных тем оформления окон, теней и композитных эффектов, однако *Evilwm* немного не такой. Это минималистский менеджер окон для X11, который разработал шотландский программист и администратор UNIX Ширан Анкомб [Ciaran Anscorb].

Проект был запущен в 2001 г., и с тех пор обновлялся раз в несколько лет, что делает появление версии 1.1.1 выдающимся событием. Менеджер окон выглядит по-прежнему и добавляет границу в 1 пиксель вокруг окон без видимых способов контроля. Работа с *Evilwm* означает, что вам придется освоить еще несколько клавиш — некоторые из них напоминают работу в *Vi*: например, клавиши *h*, *j*, *k*, *l* перемещают окна влево, вниз, вверх или вправо на 16 пикселей соответственно. Если вам надо изменить размер окна, не перемещая его, удерживайте *Shift*. Клавиши *u*, *b*, *i* и *p* перемещают

окна вверх-влево, вверх-вправо, вниз-влево или вниз-вправо соответственно, а клавиша *l* (маленькое L) отображает информацию по текущему окну.

Как легко представить, на привыкание к *Evilwm* потребуется время, но вы можете попользоваться и мышью. Граница окна в 1 пиксель, если ее захватить и перетаскивать, позволяет выполнять действия с разными кнопками мыши. Например, левая кнопка перемещает окно, правая кнопка помещает его вниз, а средняя кнопка используется для изменения размера. Помимо этого, *Evilwm* поддерживает виртуальные рабочие столы (клавиши 1–8 для переключения между ними) и разные опции файла настройки. Вы также можете



► Пытаться ухватить эту крошечную границу окна мышью — тщето, используйте клавиатуру!

передать нужные параметры просто при запуске менеджера окон: например, указать привязку к границе [snap-to-border] с помощью `$ evilwm -snap 16`, где 16 — это привязка в пикселях. Есть и другие полезные опции, например, контроль перерисовки окон при перемещении и изменении размера. Перерисовка (обновление в режиме live) используется по умолчанию, но если в силу неких причин (например, у вас низкая производительность видео) вам надо ее отключить, воспользуйтесь `$ evilwm -nosoliddrag`.

Evilwm предлагает крошечный бинарный размером всего 44к, но так как подобные мелочи давно никого не волнуют, основная причина использования *Evilwm* — чистый минимализм без малейших признаков раздутости.

Evilwm — чистый минимализм без признаков раздутости.

HotGames Развлекательные приложения

Открытый мир игры-песочницы

Minetest

Версия: 0.4.12 Сайт: <http://www.minetest.net>

Почти все слышали о *Minecraft*, исключительно популярной игре-песочнице с открытым миром, однако *Minecraft* все же проприетарная и не относится к свободному ПО, хоть и предлагает версию Linux. *Minetest* выглядит и играет очень похоже, и да, она с открытым кодом! Это также блочная игра-песочница с бесконечным миром и игровым движком, где вы можете изменять мир с помощью блоков в стиле LEGO. Вам придется исследовать, копать и строить трехмерный мир и создавать что-то из сырья. Всё как в *Minecraft*, однако умереть в *Minetest* намного сложнее, а набор текстур по умолчанию выглядит гораздо привлекательнее, чем в игре с закрытым кодом.

Minetest поставляется с базовой программой запуска игры и с несколькими шаблонами мира с блоками и предметами

по умолчанию, что соответствует песочнице без всякого сюжета. Чтобы было интереснее, добавьте моды (новые инструменты, блоки и предметы), под-игры (сценарии) и пакеты текстур (для украшения среды); все это доступно на www.minetest.net/customize. Под-игры меняют цель игры, например, выживание, строительство или матч на выживание. Установка дополнений отличается простотой: надо распаковать скачанные архивы в соответствующую поддиректорию `~/minetest`, например, `~/minetest/mods` или `~/minetest/games` или `~/minetest/textures`, после чего контент будет доступен внутри

Чтобы было интереснее, добавьте моды, под-игры и текстуры.



► Скачайте пакет текстур с высоким разрешением, они выглядят потрясающе!

программы запуска игры. Например, после установки мода запустите *Minetest*, перейдите в меню Configure мира и нажмите Enable All > Save.

Minetest имеется в пакетах для многих дистрибутивов Linux, включая Ubuntu/Mint, Fedora, openSUSE, Arch и т.д. Как всегда, если ваш пакет устарел, найдите самую свежую версию на сайте.

Многопользовательская игра

OpenClonk

Версия: 6.1 Сайт: www.openclonk.org

OpenClonk — тактическая игра для нескольких участников, в простом 2D-ландшафте в стиле муравьиной фермы. Вы управляете группами Клонков; эти маленькие гуманоиды отлично справляются как с добычей полезных ископаемых, так и с подземными стычками с другими группами. Игра продолжает традиции старых условно-бесплатных игр *Clonk* (www.clonk.de), где внимание сконцентрировано на экономике, сражениях и добыче полезных ископаемых (это часто описывается как помесь *The Settlers* и *Worms*).

Игровой движок и контент открыли код (под лицензией ISC) после завершения коммерческой марки, *Clonk Rage*, и игра значительно разрослась благодаря большому количеству модов и работе добровольцев. Основное различие между *OpenClonk* и десятками схожих аркадных скроллеров в том, что основной сюжет

OpenClonk связан с добычей и исследованиями. Вы прокладываете себе путь, копая и взрывая горную породу, и хотя можно сбрасывать бомбы, кидать гранаты и запускать ракеты и подрывать предприятия противника бочками с порохом, вы увидите, что нырять в неисследованное подземное озеро ничуть не менее интересно. Во многом *OpenClonk* является отличным руководством по горному делу и открытой разработке, поскольку там есть большая часть материалов, которыми вам надо пользоваться (кроме, пожалуй, ракетных пусковых установок). Цель игры каждый раз меняется, и приветствуется игра против других игроков.

Это часто описывается как помесь The Settlers и Worms.



► *OpenClonk* вознаграждает храбрых искателей приключений. Копайте, сражайтесь с противниками и побеждайте свою клаустрофобию!

Обычной целью является добыча ценных ресурсов из недр посредством строительства шахты или сражений друг с другом на карте в стиле арены, где допускается мародерство.

Интереснее всего играть в *OpenClonk*, соединяясь с друзьями по LAN или присоединившись к онлайн-игре (<http://league.openclonk.org>), где вы встретите ветеранов этой игры. Если вы пока чувствуете себя зеленым для большой игры, предлагаем пройти обучение, поскольку оно помогает одолеть основы.

OpenClonk распространяется в виде статической сборки, и вы найдете эту игру во многих дистрибутивах Linux.

Оболочка UNIX

Fish

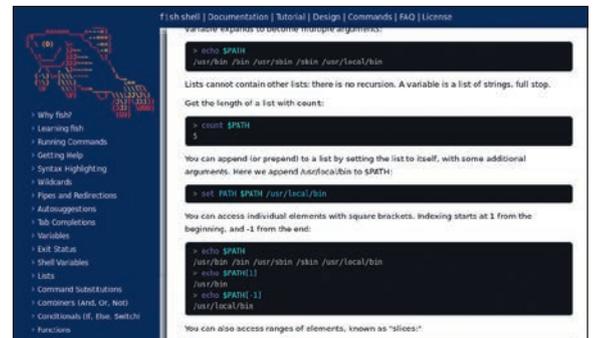
Версия: 2.2.0 Сайт: <http://fishshell.com>

Когда мы открываем эмулятор терминала в Linux, обычно это *Bash* (*Bourne Again Shell*). Однако есть и другие опции, и одной из самых дружелюбных к пользователю является *Fish* (что означает Friendly Interactive Shell).

Fish появилась в 2005 г., имея цель предоставить богатый набор мощных функций так, чтобы их было легко изучить, запомнить и использовать. Примета *Fish* — предоставление всех своих функций по умолчанию, в отличие от предыдущих оболочек, таких, как оболочки *Bourne* (*Ksh*, *Bash*, *Zsh* и т. д.) или *C* (*csh* и *tcsh*), в которых функций по умолчанию не было, ради экономии системных ресурсов.

Функции, включенные в *Fish* по умолчанию — автопредложения, основанные на вашей истории и предыдущих запросах, цветной вывод VGA благодаря поддержке *term256*, веб-настройка (подход для оболочки уникальный!) и очень здравый синтаксис для написания индивидуальных скриптов.

Некоторые из перечисленных функций найдутся и в *Bash* или *Zsh*, если индивидуально настроить их профили, но прелесть *Fish* в том, что вы получаете всё немедленно. Вместо нажатий Tab для дополнения команды или имени файла, просто начните печатать, и вы увидите, как *Fish* тут же предлагает варианты. Движок дополнения работает сказочно, поскольку распознает пути файлов, командные опции и даже вашу предысторию. Изучать *Fish* особенно легко и интересно благодаря очень удобной и детальной документации со множеством примеров и руководств, относящихся к таким областям, как переменные, алиасы команд, объединители, условия, функции, и т. д. — даже не являясь программистом, можно использовать шабло-



Движок *Fish* относится с уважением даже к вашей пользовательской истории.

ны для автоматизации многих рутинных задач. Подробнее об этом можно узнать на <http://bit.ly/FishTutorial>.

Найти *Fish* тоже очень легко: велики шансы, что она уже есть в менеджере пакетов вашего дистрибутива. Сайт *Fish* также предлагает прекомпилированные бинарники для Ubuntu, Debian, Fedora, CentOS, openSUSE, Arch и Gentoo.

Прелесть Fish в том, что вы получаете всё немедленно.

Среда программы установки

Calamares

Версия: 1.1.2 Сайт: <https://calamares.io>

Не обращайте внимания на надпись KaOS на экранном снимке: установщик *Calamares* ничуть не зависит от дистрибутивов и выглядит одинаково в любом дистрибутиве Linux. Суть *Calamares* в обеспечении современной, тонко настраиваемой и богатой функциями среды установки, что призвано несколько улучшить ситуацию, поскольку мы все знаем, что для большинства новичков именно установка Linux является самым нервным этапом.

Программа установки показывает список шагов слева и предполагает, что вы постепенно будете переходить от шага к шагу, вводя нужную информацию и нажимая Next. Говоря технически, *Calamares* создан с помощью Qt 5, C++11, Boost.Python, чуть-чуть KDE Frameworks 5 и KDE Partition Manager. Однако он никак не привязан к рабочему столу Plasma 5 и широко применяется в дистрибутивах с разными рабочими столами — на данный момент это

KaOS, Manjaro, BBQLinux, плюс идет оценка в нестабильных сборках OpenMandriva, Fedora, Tangu, Sabayon и ряда других.

Так зачем бы вам устанавливать *Calamares* на свою текущую систему? Ответ в том, что это шаг к вашему личному дистрибутиву Linux, будь то ответвление дистрибутива или образ резервной копии тонко настроенной системы с полезными вам функциями. *Calamares* позволяет переместить загруженную систему на другой раздел, настроить boot loader и сделать все остальное. Однако редкие двоичные пакеты *Calamares* от Fedora или OpenMandriva для индивидуального дистрибутива практически бессмысленны, а значит, лучше компилировать его из исходника. Предполагается,



Prepare > Install > Finish — и вуаля!

что *Calamares* компилируется с помощью *cmake* обычным способом:

```
$ mkdir build && cd build && cmake .. && make
```

Однако *cmake* выдаст жалобу на отсутствующие зависимости, если у вас нет стека *Qt5-devel* и пакетов *boost* и *python3*. После этого всё должно скомпилироваться. Сейчас у *Calamares* имеется 27 модулей (доступных на <http://bit.ly/CalamaresModules> вместе с документацией), которые вы можете объединить и создать комфортную для вас среду установки. LXF

Calamares — шаг к вашему личному дистрибутиву Linux.

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 9 ГБ качественного DVD.



Дистрибутивы

Недавно я видел передачу про собак, и в ней сказали, что все породы собак — от датского дога до молпа — произошли от одного вида волков. Меня поразило, что единственный источник всего за несколько тысяч лет произвел такое разнообразие. И я вспомнил об этом, составляя данный Linux Format DVD. У нас есть два дистрибутива на базе Ubuntu; предок у них один, но они очень разные. Третий дистрибутив, OpenMediaVault, отличается еще сильнее, а он основан на Debian, то есть у всех троих общий дед. Это доказывает истинность высказывания «Linux есть Linux»: у всех этих дистрибутивов один и тот же основной код ядра, и этим сходство не ограничивается, но они выглядят и ведут себя совершенно по-разному. Возможность изменять столько аспектов ОС согласно конкретным нуждам, сохраняя всё тот же проверенный код ядра — одно из колоссальных достоинств Linux. То, что эта гибкость свободна и бесплатна — не только преимущество, но и причина существования самого такого выбора. Неудивительно, что пользователи бываю ошарашены, впервые пробуя Linux после Windows. Почти единственное от Windows, чего нет в Linux — это подход «бери либо проваливай»!

Neil

Важно ВНИМАНИЕ!

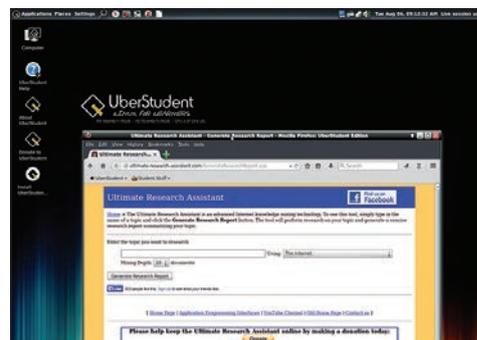
Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Linux для учащихся

UberStudent 4.1

Данный выпуск LXF имеет образовательный уклон, и может ли быть выбор главного дистрибутива лучше, чем UberStudent? Это потомок Ubuntu с рабочим столом Xfce, нацеленный прежде всего на тех, кто получает среднее или высшее образование. Важен в нем, однако, выбор программ: в UberStudent есть всё полагающееся общецелевому настольному дистрибутиву, однако приложено еще и множество программ специально для студентов. А поверх добавленных образовательных программ — несколько приятных штрихов в виде индивидуальных расширений для Chrome и Firefox.



Альтернативная работа с окнами

WattOS R9

Многие дистрибутивы заявляют о своей оптимизации подустаревшее «железо», часто за счет добавления облегченного рабочего стола. WattOS — производное Ubuntu, практикующее этот подход в релизе на базе LXDE; но его Microwatt Edition идет дальше. Оно использует мозаичный менеджер окон i3, отображающий окна без перекрытия, в отличие от обычного, плавающего вывода окон. Открытые окна мостят экран подобно плитке. Откройте одно окно — и оно заполнит весь экран; откройте второе — и оба

сожмутся, чтобы занять пол-экрана каждое. Конечно, иногда требуется приоритет какого-то из окон, так что данный подход контролируется. Основной механизм управления у i3 — клавиатура, причем до такой степени, что обоим WattOS отображают основные сочетания клавиш. Новичков может смутить отсутствие меню приложений: вместо этого, открыв менеджер файлов, перейдите в его раздел Applications из левого меню. Отсюда можно просматривать доступные программы, настраивать систему и т.д.



UberStudent 4.1 **НОВЫЙ 64-БИТНЫЙ РЕЛИЗ**

Ubuntu, пересобранный для студентов и преподавателей

wattOS R9
Легковесный дистрибутив с долгой поддержкой, который получит диск ПК с 256 МБ ОЗУ!

Эксклюзив!
16 учебных видеокурсов от Urban Penguin. Окупились в Linux с разберу!

LINUX ЛУЧШИЕ ДИСТРИБУТИВЫ И БОЛЕЕ ТОГО! ВСЕ ДЛЯ БЫСТРОГО СТАРТА В LINUX

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.

Спецдистрибутив для NAS

OpenMediaVault 2.1

OpenMediaVault — готовый файловый сервер для домашних и небольших офисных сетей. В отличие от большинства дистрибутивов нашего DVD, этот — не live, а более традиционный установочный. Скопируйте ISO-образ на CD-R или карту USB и загрузитесь с него, чтобы сделать установку. Возможно, вы захотите посмотреть OpenMediaVault, прежде чем пускать его на свой жесткий диск, и разработчики об этом подумали. Зайдите на <http://openmediavault.org> и далее по ссылке Demo; она ведет к онлайн-демо, и можно опробовать часть функций. В обзоре на стр. 100 — более подробная информация.

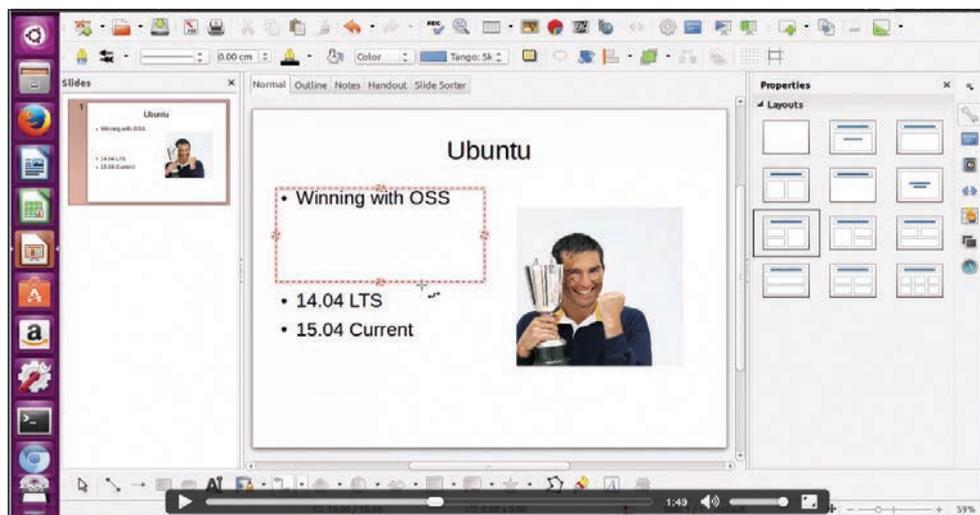


Linux для учащихся, дубль 2

Обучение Linux

Мы раздобыли кое-какие видеокурсы, чтобы помочь вам освоиться с вашим дистрибутивом Linux. Они основаны на Ubuntu 15.04, но, в русле парадигмы «Linux есть Linux», большинство из того, чему они учат, приложимо ко многим другим дистрибутивам. Видео *Using the Desktop* [Используем рабочий стол] — чисто про Ubuntu (или, точнее, Unity), а видео по управлению пакетами не очень поможет владельцам RPM-дистрибутивов типа Fedora и openSUSE, но разделы по программам и приложениям, например,

по web-браузерам и LibreOffice, пригодятся в любом дистрибутиве, лежащем в основе. Видео — в стандартном формате MP4, и вы сможете просмотреть их где угодно, а самый простой способ — щелкнуть по эскизу видео после загрузки содержимого DVD в браузер. Если вы используете Chrome/Chromium и у вас устройство Chrome, вы даже сможете их смотреть на семейном телевизоре, хотя не все семьи положительно относятся к подобному вмешательству в их привычные мыльные оперы и реалити-шоу. [LXF](#)



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 тестирования системы.

Kernel Исходный код самого последнего стабильного релиза ядра.

Mement86+ Проверьте ОЗУ на предмет сбоев.

Plop Простой менеджер загрузки для запуска ОС с CD, DVD и USB.

RaWrite Создавайте загрузочные диски в MS-DOS в Windows.

SBM Независимый от ОС менеджер загрузки с несложным интерфейсом.

WvDial Соединяйтесь с Интернетом через телефонный модем.

Поскольку этот выпуск посвящен образованию, и его цель — помочь вам в освоении Linux, мы включили 16 видеоруководств от Urban Penguin, которые ведет эксперт Linux Эндрю Моллетт [Andrew Mallett]. Эти видео познакомят вас с Linux, с рабочим столом и со стандартными инструментами разработки. Эндрю проведет вас за ручку по всему процессу, используя классические программы рабочего стола и офисные инструменты, а затем расскажет о задачах посложнее. Если у вас хватит смелости, мы рассмотрим, как приступить в Ubuntu к разработке программ, а также осветим основы программирования. Пройдя и это, мы обратимся к основам систем Linux, которые работают внутри, чтобы углубить ваши знания. Если вам по душе идея открытого кода, Linux и Ubuntu, можете узнать больше на www.theurbanpenguin.com.



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:



LXF199
Август 2015

250 руб.

- » **Сделай сам** Да, вам по силам собрать ПК!
- » **Лети с приветом** Быстрый обмен сообщениями
- » **Загадки протоколов** Мы их разгадаем
- » **Система** Дух времени

LXFDVD: Debian 8, LXLE 14.04, Elementary OS, Point Linux, ROSA, Zorin OS, 9 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_199/
PDF-версия: shop.linuxformat.ru/elxf_199/



LXF200
Сентябрь 2015

250 руб.

- » **200 полезных советов** Ради праздника!
- » **Не Linux'ом единым** Альтернативы есть
- » **Откуда берутся пакеты** Фермы по сборке
- » **Немного ностальгии** Предадимся воспоминаниям

LXFDVD: Fedora 22, Peppermint OS, Q4OS, Ubuntu, Linux Lite, Sabayon, Tails, IPFire, Sparky, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_200/
PDF-версия: shop.linuxformat.ru/elxf_200/



LXF201
Октябрь 2015

250 руб.

- » **Linux против Windows 10** Пингвин завесит Окна
- » **Навязчивый сервер** Дистрибутивы для Web
- » **Кто у нас ябеда** Electronic Frontier — о соцсайтах
- » **АТС из Кубиков** Cubieboard и Asterisk

LXFDVD: Mageia 5, 4MLinux 13, AntiX 15-V, Tiny Core, Emmabuntüs 3-1.02, GParted Live 0.24.0-2, 11 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_201/
PDF-версия: shop.linuxformat.ru/elxf_201/

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 ВКонтакте

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

» Содержание

LINUX
ФОРМАТ

Страница 1

ДИСТРИБУТИВЫ

Ubuntu 14.1 (64-битный)
Wheezy R9 (32-битный)
OpenMediaVault 2.1 (32-битный)

ВОЛУС

16 видеоуроков

НОТРІСКС

Salpares 1.1.2 Среда установки
salpares-1.0
Erlang 1.11 Менеджер окон
www.erlang.org/erlang
Fish 2.2.0 Оболочка Unix
fishshell.com

rsync 2.26 Программа SCD
www.rsync.org

Geany 1.25 Текстовый редактор и IDE
www.geany.org

GNU Coreutils 8.24 Основные системные утилиты
ftp://gnulinux.org

Minetest 0.4.12 Открытый мир игры-песочницы
www.minetest.net

OpenSick 6.1 Многопользовательская игра
www.opensick.org

OpenMediaVault 2.1 ПО для NAS
openmediavault.org

OpenShot 8.1 Частное локальное облако
openshot.org

Shotcut 5.07 Видеоредактор
www.shotcut.org

УЧЕБНИКИ

Botmoku
Pibot
Raspbian
RSS

ПОМОЩЬ

Руководство новичка
Руководства
Ответы
DVDFAQ

Окончание на обороте »

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!

UberStudent

4.1

НОВЫЙ
64-БИТНЫЙ
РЕЛИЗ

Ubuntu, пересобранный для студентов и преподавателей

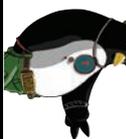
Wattos R9

Легковесный дистрибутив с долгосрочной поддержкой, который потянут даже ПК с 256 МБ ОЗУ! 32-битный



Эксклюзив!

16 учебных видеокурсов от Urban Penguin. Окунитесь в Linux с разберу!



ЛУЧШИЕ ДИСТРИБУТИВЫ И БОЛЕЕ ТОГО!
ВСЕ ДЛЯ БЫСТРОГО СТАРТА В LINUX

LINUX
ФОРМАТ

Содержание

LINUX FORMAT

Страна 2

ГЛАВНОЕ

Checkinstall
Coreutils
Hardinfo
Kernel
Memtest86+
Plop
SBM
Wvdial

The Cathedral and the Bazaar Классический текст Эрика Раймонда [Eric S Raymond] «Собор и базар»
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом 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
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
Linux Dictionary Словарь Linux, объясняющий специфическую терминологию

ДОКУМЕНТАЦИЯ: 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Advanced Bash Scripting Guide Подробное руководство по программированию на Bash
Bash Guide for Beginners Руководство по Bash для начинающих
 Bourne Shell Scripting Начальное руководство по программированию на Bash

ДИСТРИБУТИВЫ

ClearOS 7.1.0 (64-битный)
EXTIX 15.4 LXQt (64-битный)
Ubuntu Studio 15.10 (64-битный)

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 данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных в случае жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с сайта www.linuxformat.ru узнайте все о вашей копии диска.
Тираж издательства ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИПТР ВАО № 77-03.

Создание установочных дисков при помощи cdcrecord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdcrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdcrecord -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:

```
cdcrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdcrecord*, сохраните некоторые настройки в файле `/etc/default/cdcrecord`. 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

```
cdcrecord -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 нажмите на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdcrecord* 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.

infobox

Виртуальные серверы нового поколения



Серверы летают!

- ✓ 3 дата-центра в России и Европе
- ✓ Высочайшая скорость Enterprise SSD
- ✓ Тройная репликация всех данных
- ✓ Безлимитный трафик
- ✓ Докризисные цены — от 200 рублей!

Закажите прямо сейчас
infobox.ru/vps/



LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 2000 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Валерий Смирнов, Елена Толстякова

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

ООО «Линукс Формат»

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 13397

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@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] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Bothwick], Джолон Браун [Jolyn Brown], Крис Браун [Chris Brown], Алекс Кэмпбелл [Alex Campbell], Кент Эльчук [Kent Elchuk], Мэтью Хэнсон [Matthew Hanson], Дэйв Джеймс [Dave James], Кевин Ли [Kevin Lee], Кристофер Ливингстон [Christopher Livingston], Эрик Манн [Eric Mann], Нейл Мор [Neil Mohr], Эндрю Малхолланд [Andrew Mulholland], Ник Пирс [Nick Peers], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Александр Толстой [Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Евгений Балдин, Андрей Гондаренков, Дмитрий Кузнецов, Дмитрий Пантеличев, Алексей Федорчук, Максим Черепанов

Иллюстрации Шейн Коллиндж [Shane Collinge], Крис Хедли [Chris Hedley]

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 01225 442244, email: linuxformat@futurenet.com

РОССИЯ: Санкт-Петербург, пр. Медиков, 5, корп. 7

Тел. +7 (812) 309-0686, e-mail: info@linuxformat.ru

По вопросам сотрудничества, партнерства, оптовых закупок: partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права защищены. 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 ответственность за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса (Linus Torvalds). "GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.

В декабрьском номере

Чемпион-2015

Мы натравили лучшие настольные дистрибутивы друг на друга — с целью посмотреть, как этот год изменил дистрибутивный ландшафт.

Free Software Foundation

В минувшем октябре FSF стукнуло 30. Отмечаем этот факт статьями о свободе ПО и интервью.

Файловые системы Linux

Эти системы хранят ваши драгоценные файлы, но какая из них лучше? Исследуем углубленно.

Улучшаем рабочий стол

Надоела привычная среда? Узнайте, как усовершенствовать свой рабочий стол Linux.

Содержание будущих выпусков может меняться — вдруг мы не в меру увлечемся битвой дистрибутивов...



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+

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.

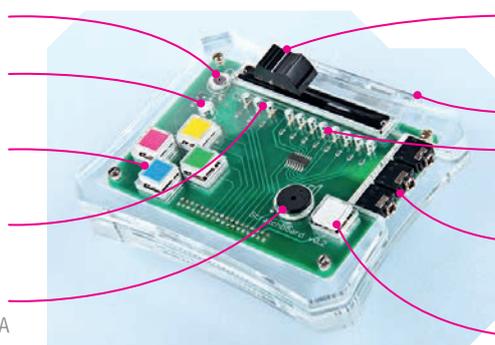
ДАТЧИК ЗВУКА*

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 **9500 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

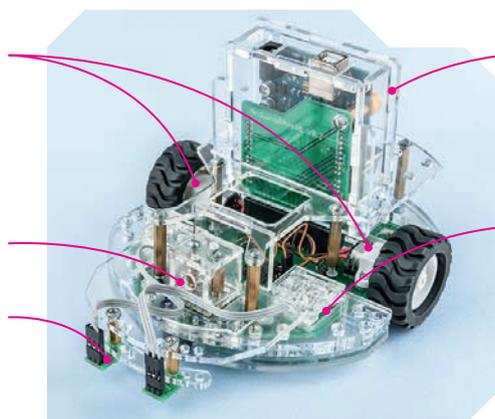
обеспечивает возможность:

- 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 **19 500 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)

HETZNER DEDICATED SERVERS

HETZNER
— ONLINE —

В ЦЕЛОСТИ И СОХРАННОСТИ!

ХРАНЕНИЕ ДАННЫХ В ГЕРМАНИИ



ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3 Quadcore Haswell
- 32 ГБ ECC RAM
- 2 x 2 ТБ 6 Гбит/с SATA 7200 об/м Enterprise класс Software-RAID 1
- 30 ТБ Трафик*
- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта

2600

рублей в месяц
+ Установка 3700 рублей



ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3 Quadcore Haswell
- 32 ГБ ECC RAM
- 2 x 480 ГБ 6 Гбит/с SATA SSD Data Center Series
- 50 ТБ Трафик*
- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта

3700

рублей в месяц
+ Установка 3700 рублей



Безопасное защищённое хранение ваших данных в собственных дата-центрах немецкой компании Hetzner Online, соответствующее высоким требованиям к стандартам защиты данных в Германии.

RU.HETZNER.COM



100%
Green Electricity
Energy-efficient
Hardware

GreenIT **2011**
Best Practice Award

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте более чистое будущее вместе с Hetzner Online!

* Нет платы за превышение. При превышении 30 ТБ/месяц (PX60), 50 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.