

DVD ДАРОМ!



UBUNTU 17.10 64-БИТ

LINUX FORMAT

Главное в мире Linux

Декабрь 2017 № 12 (231)

ДИСТРИБУТИВ-САМОДЕЛКА

Соберем openSUSE с лучшим рабочим столом и приложениями



Академия кодига: Обновим BBC BASIC + Kotlin

UBUNTU 17.10 ПЕРЕНАСТРОИМ!

Заставьте новенький Gnome плясать под вашу дудку

64

страницы
проектов
для Linux
и Pi

ЕСТЬ
ЭЛЕКТРОННАЯ
ВЕРСИЯ
ДЛЯ ВАШЕГО
ПЛАНШЕТА!



Человек из Microsoft

“ Если сможем повысить производительность на 0,05%, это оправдывает содержание инженеров, улучшающих ядро ”

Мартин Вудвард входит в ров со львами для разговора о FOSS

Плюс: Для Pi

9 страниц в копилку Raspberry Pi

- » Собираем 4-колесного робота
- » Рулим из Node-RED через Web
- » Организуем доступ с RFID
- » Ставим Wi-Fi на карту



Сисадминам

Новые рецепты

» Мониторим серверы секретным оружием нашего админа



Сравнение

Рабочие столы

» Не нравится Gnome? Установите свой рабочий стол



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

На сей раз мы спросили наших экспертов: мы обсудили всеобщую ярость насчет изменений Gnome 3, Unity and KDE4 — а какое изменение в Open Source или Linux накалило *вашу* ярость докрасна?



Джонни Бидвелл

В Linux меня больше не злит даже шаткая поддержка подвешивания/возобновления, которая что ни день гробит мою машину с момента выхода Kernel 4.13; не злит даже страх, что установка нового дистрибутива сотрет существующие входы Grub или UEFI. Такие штуки держат в боеготовности.



Нейт Дрейк

Как специалиста по инфобезопасности, меня пару лет назад взорвал опрометчивый шаг Canonical по включению в Dash Ubuntu Unity продуктов Amazon. А последняя LTS-версия Ubuntu включает Snap-пакеты, имеющие доступ ко всем вашим нажатиям клавиш при использовании с X11 — что не идеально.



Джон Найт

Рад сказать, что ничто из OSS меня в ярость не приводило — в худшем случае, это было легкое раздражение. А не вернуть ли дистрибутивам виртуальные рабочие столы, или пусть хотя бы упростят возврат моего виртуального переключателя рабочих столов?.. О, знаете что? Я приберег свою ярость для Nvidia: ПОЧИНИТЕ ПРОКЛЯТЫЕ ОБРЫВЫ!!!



Лес Паундер

Мой самый сильный гнев вызван редактированием видео в Linux. Есть много отличных видеоредакторов, таких как OpenShot и Kdenlive, но всё, что я пробую, являет ошибки и частые сбои, так что мои правки иногда теряются. Редактирование видео — вот что надо улучшить в Linux.



Шашанк Шарма

Я особо приветствую изменения, наступающие после многочисленных дискуссий с сообществом пользователей, как это было в случае с GPLv3. Однако первоначальное объявление Gnome 3 и даже Unity разочаровывали. С тех пор я снова полюбил Gnome, но не могу принять небрежный отказ Ubuntu от Unity.



Валентин Синецын

Как, никто не назвал Vim? Видимо, коллегам неловко сознаваться, что они *тоже* поначалу выходили из него, нажав на Reset (да, эта кнопка в ходу не только у ветеранов Windows, но и новичков в Linux). Я довольно быстро запомнил `qall!`, но реально взялся за Vim только после удачного сравнения с карандашом из Рунета. А Emacs пока подбешивает...



Командирское решение

» Сообщение газеты «Известия» о том, что все компьютеры Минобороны РФ переведут с Windows на AstraLinux, мелькнуло в лентах новостей, но осталось почти незамеченным за ежедневными «страшилками» о кибервойне. Мои знакомые, далекие от ИТ, удивлялись: у них после всех разоблачений Сноудена еще «Винда» где-то осталась? А оказалось, что да.

Не очень известен широкой публике и аналогичный проект нашего, как говорили во времена моей учебы на военной кафедре, «главного противника». Trusted End Node Security (TENS), ранее известный как Lightweight Portable Security (LPS), собирается в виде live CD или загружается с флешки. И в том, и в другом названии с военной прямоотой обозначено главное: безопасность. Не доверяют, значит, своему американскому «лидеру индустрии» с западного побережья.

А про распоряжение Правительства РФ №2299-р, содержащее план перевода органов государственной власти на свободное ПО, через семь лет никто уже и не вспоминает. Насколько мне известно, единственным ведомством, которое его выполнило, стала Федеральная служба судебных приставов. Люди там служат гражданские, но тоже с погонами на плечах...

Все три истории лишней раз доказывают, что широкому распространению Linux на рабочих станциях (с серверами давно всё ясно) мешают не технические или кадровые проблемы, а лишь отсутствие «командирского решения».

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

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

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

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

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

«Если фашизм когда-либо явится в США, он будет драпирован американским флагом.» Аноним

Обзоры

CentOS 1708 12

ОС корпоративного уровня, без ограничений на распространение и с долгосрочной поддержкой, послужит для бизнеса, малого и большого.

Knoppix 8.1 13

Первенец семьи live-дистрибутивов по-прежнему предлагает отличную коллекцию приложений — вставляйте диск и удивляйте друзей!

Korora 26 Gnome 14

Прислушавшись к своему сообществу, проект внес ряд изменений, включая разработку документации.

KaOS 2017.09 15

Поскольку это дистрибутив с возобновляемым релизом, спешите читать, не то инновации устареют. А их появилось немало.

Intel Core i5 8400 16

Возможно, этот чип попадет на ваш следующий Linux-ПК. Узнайте, почему — испейте Coffee Lake.

Intel Core i9 7960X 17

Если лишние £2000 жгут вам карман, ваши страдания мигом облегчит линейка многоядерных процессоров Skylake-X.

Google Pixel 2 18

Не возродит ли обновление гаджета от Google нашу неприязнь к устройствам Pixel? А вот вряд ли. Его выручает камера — лучшая в данном классе.

Tasoma 19

Опустевший орбитальный форпост, заброшенный в итоге катастрофы... но хватит о Башнях Linux Format, как насчет новой космической бродилки от Fullbright?



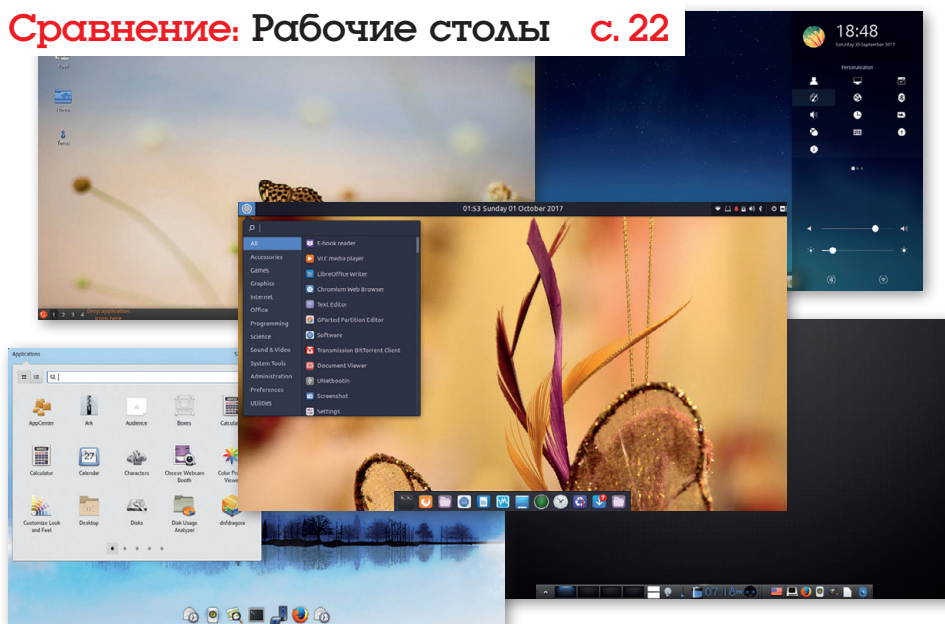
➤ Добавленная реальность превращает Башни LXF в почти терпимую зону.

ПЕРЕДЕЛАЕМ UBUNTU 17.10

Прибыл новый рабочий стол Gnome 3!
Настроим его индивидуально и раскучрим,
чтобы исследовать захватывающие новые
технологии внутри 17.10. См. с. 28



Сравнение: Рабочие столы с. 22



Интервью



« В Microsoft вообще сейчас рулят инженеры, и это здорово! »

Мартин Вудвард о том, как Microsoft примирился с OSS с. 38

На вашем бесплатном DVD



Ubuntu 17.10 Remix 64-битный
Lubuntu 17.10 32-битный
Q4OS 2.4 64-битный

» Каждый месяц — только лучшие дистрибутивы

ПЛЮС: HotPicks, код и библиотека **с. 106**

Побалуйте себя и любимых подпиской на LXF!



Доступно в AppStore!

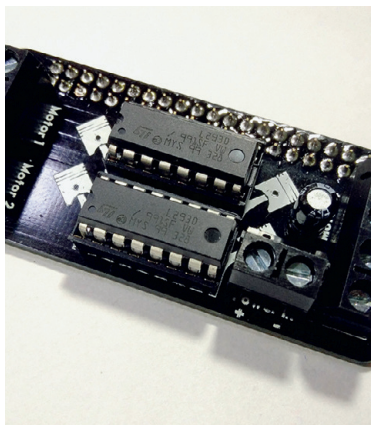


www.linuxformat.ru/subscribe

Пользователям Raspberry Pi



Новости Pi	90
Raspberry Pi продолжает нести миру образование, а также помогает слепым читать онлайн-новости. И немного про недавние сборки.	
MotoZero	91
Пользователям начального уровня: рассмотрим контроллер мотора для четырехколесного привода.	
GPIO и Node-RED	92
Создаем несложный web-интерфейс для управления контактами GPIO.	
Система доступа на RFID	94
Не оказаться бы нам запертыми в Башнях LXF при попытке внедрить магнитный замок, управляемый Pi и читалкой RFID...	
Сканируем Wi-Fi	96
С помощью Raspberry Pi 3 нанесем на карту точки доступа к локальным сетям, разъезжая по окрестностям.	



Ищите в номере

Самовыражение **44**
 Соберите пакеты, настройте их, выберите себе рабочий стол — и вот вам ваш личный дистрибутив. В мире он такой единственный.



Академия кодинга

BBC Basic V	78
Быть может, ностальгический язык BASIC рано списывать со счетов. Он еще повоюет.	
Типы данных Kotlin	82
Массивы, списки, циклы for и while, блоки, диапазоны... наверное, это и все? Больше ведь не бывает?	
Системные утилиты на Kotlin	86
Если вашей голубой мечтой всегда было системное программирование, создание потоков и генерация случайных чисел, то с этим номером журнала вам повезло.	

```

10 for (i in args) {
11     if (i == "-i")
12         iOption = true
13     else if (i == "-c")
14         cOption = true
15     else
16         break
17 }
18
19 if (iOption)
20     println("-i is set!")
21 if (cOption)
22     println("-c is set!")
    
```

Учебники

Основы терминала Taskset	56
Достигнем высшей власти — будем управлять ядрами процессора!	
Основы терминала Каналы	58
Освежаем память о таковых в UNIX и осваиваем их в Linux.	
Офис LibreOffice Calc	60
Как насчет создания сводных таблиц или построения столбчатых диаграмм? Превращаем вас в индзя-ячейкоборца.	
Сети Ethernet на 10 Гб/с	64
Если Gigabit Ethernet тормозит, пора переходить на новый стандарт. Мы вам поможем это сделать.	
Конфиденциальность OpenVPN	66
Проведем вас за ручку по зашифрованному тоннелю, чтобы на ваши данные никто не смел покушаться.	
Безопасность Сертификаты SSL	70
Заглядываем в зазеркалье безопасности сертификатов, от реализации SSL/TLS в командной строке до системы Let's Encrypt.	
Интернет вещей Гори ясно	74
Разместим на web-панели Послушного Дома слайдер для регулировки яркости светодиода.	

Постоянные рубрики

Новости	6	Сравнение	22
Выпала новая ROSA, в Китае вышел клон Raspberry Pi, Linux занял весь Top-500, новый дистрибутив автоматизирует производство, DNS блокирует вредоносные домены, анонсированы системы высокопроизводительных вычислений, Dell сделала ПК с Ubuntu, а емкость NVDIMM удвоилась.		На слуху, конечно, «большая двойка», но ведь и другие неплохи: Budgie, Deepin, Enlightenment, LXQt, Pantheon.	
Вести мобильных ОС	20	Интервью LXF	38
Народилось семейство 5G-модемов, IoT-данные ускоряют вызов аварийно-спасательных служб, телефоны наращивают интеллект, а «умные дома» переходят на mesh-сети.		Очевидец Мартин Вудвард свидетельствует: Microsoft более не враг!	
		Рубрика сисадмина	50
		С новым Доктором! Валентин Синицын философствует по поводу судьбы Solaris и анализирует систему.	
		HotPicks	100
		Отведаете горяченького! Лучшие в мире новинки сводного ПО: <i>BeFive</i>	

Smartball, Borg, Calibre, ChickenPeppa, Geary, Glances, Gnome Commander, Klavaro, MConnect, Storyboarder, Zrandr.

Диск Linux Format **106**
 Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... **108**
 Опускайте подъемный мост, расставляйте лучников у бойниц: обороняйте свой замок с Raspberry Pi в LXF229.

Через месяц **112**
 Linux создан ради скорости! Есть шустрые дистрибутивы, которые взбодрят ваше оборудование.



В ЭТОМ НОМЕРЕ: Роса зимой » Малина по-китайски » Linux — чемпион » Индустрия 4.0
» Ботнеты, стоп! » Суперкомпьютеры идут в народ » Ноутбуки с Linux » Память удвоилась

РОССИЙСКОЕ ПО

Освежающие росы...

Отечественные разработчики обновили версию российского RPM-дистрибутива.

НТЦ ИТ РОСА представила ROSA Fresh R10 — второй релиз Fresh на платформе 2016.1, поддержка которого, согласно политике выпусков, будет осуществляться 4 года, до окончания разработки новой платформы 2018.1. Пока доступны образы с KDE4 и Plasma5, содержащие все исправления и обновления, вышедшие с момента релиза R9, дополненный состав ПО и обновленный дизайн. Позднее сообществом будут подготовлены образы с Gnome 3, LXQt и Mate. Итак, что нового?

- » Ядро 4.9.60 LTS.
- » Mesa 17.1.6 с поддержкой OpenGL вплоть до 4.5 и нового графического API Vulkan.
- » Исправлена совместимость драйверов broadcom с новыми ядрами, вплоть до 4.13.

» Исправлены проблемы с работой многих USB WiFi-адаптеров.

» Изменены настройки утилиты dracut для подготовки initial RAM disk images (initrd): теперь в initrd добавляется больше драйверов устройств. Это позволяет избежать многих сбоев при загрузке системы, особенно с ядром 4.13 и новее.

» В фирменную утилиту настройки видеокарты XFdrake добавлена возможность настройки различных режимов ускорения на картах Intel.

» Исправлены ошибки с запуском 32-битных приложений в 64-битной среде на гибридной графике с Nvidia Optimus.

» Набор утилит для работы с CD и DVD cdrkit заменен на cdrtools.

» ROSA Fresh R10 — один из немногих современных дистрибутивов, пользователей которого, помимо новомодной Plasma 5, доступен и «теплый, ламповый» KDE4.

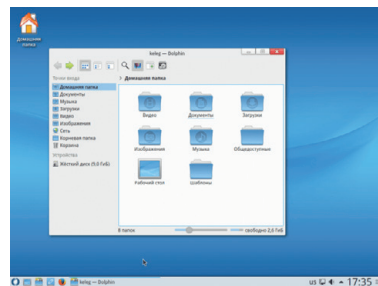
» Улучшена совместимость со сторонними пакетами для sysvinit, а не systemd.

» В репозитории добавлены библиотеки OpenSSL, libcrypto.so.10 и libssl.so.10.

» Исправлена критическая уязвимость KRACK, затрагивающая WPA2.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**



МИНИ-КОМПЬЮТЕРЫ

А кто не читит цитат — тот Ренегат!

«Малиновый» клон Libre Renegade поддерживает Android, USB 3.0 и видео 4K.

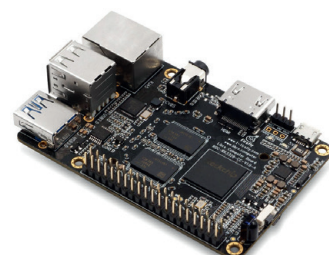
Китайская компания Libre Computer предложила свой клон Raspberry Pi Model B — Renegade, и организовала на платформе краудфандинга Indiegogo сбор средств для его выпуска. За первую декаду декабря собрано только \$625 из запланированных \$10 тыс., однако начатая в ноябре кампания по сбору средств для производства Tritium (разработанный Libre Computer клон Raspberry Pi 3) уже принесла \$37 тыс. от более 600 спонсоров.

Renegade оснащается ОЗУ 1, 2 или 4 ГБ DDR4 (Pi 3 поставляется с DDR2), процессором Rockchip RK3328 (4 ядра Cortex A53 и GPU ARM Mali-450MP2), разъемом Gigabit

Ethernet, 2-мя портами USB 2.0 Type A, портом USB 3.0 Type A (отсутствует у Pi), флэш-модулем eMMC 5.x (от 8 до 128 ГБ) и слотом для карт microSD. Выходы видео и аудио — HDMI 2.0a до 4K @ 60 Гц с поддержкой HDR10 и HLG, порт 3,5 мм AV (композитное видео + стереоаудио); видеокodeки — 4K VP9, H.265 и H.264, 1080p VC-1, MPEG-1/2/4, VP6/8. Поддерживаются ОС Ubuntu 16.04 с Rockchip Linux 4.4 Kernel и Mainline Linux 4.14 LTS Kernel, а также Android 7.1 Nougat. Размер — 85×56 мм. Производитель разработал ПО для применения Renegade в качестве медиа-центра, игрового компьютера и настольного ПК. Поставки

» Ориентированный на решения для домашних устройств мультимедиа и Интернета вещей одноплатный компьютер Libre Renegade сопоставим по цене с Raspberry Pi Model B, но имеет более высокие спецификации.

начнутся в январе 2018 г.; модель с 1 ГБ ОЗУ будет доступна спонсорам за \$35 (розничная цена — \$45), с 4 ГБ — за \$70 во время кампании, а в розницу — за \$80. За \$100 войдут еще радиатор, активный кулер, блок питания 5 В/2,5 А и 32-ГБ карта MicroSD.



TOP-500

Восхождение к сияющим вершинам

Осенью 2017 г. на лучших суперкомпьютерах мира полностью доминирует Linux.

Согласно очередному рейтингу Top-500, на Linux работают все 500 лучших суперкомпьютеров мира. Две последние не-Linux системы, принадлежащие Китаю и управляемые AIX компьютеры IBM POWER, в ноябре 2017 г. покинули Top-500. Гонку суперкомпьютеров уверенно возглавляет Китай: ему принадлежат 202 системы, США — 144. Lomonosov 2 в новом Top500 находится на 63-м месте (59-е полгода назад), Lomonosov — на 227-м (было 164-е), Tornado на 412-м (было 297-е); полгода назад в Top-500 также входили 3 суперкомпьютера из РФ, год назад их было 5, в 2012 г. — 12.

На 1-м месте расположился китайский Sunway TaihuLight с производительностью 93 петафлоп, на втором также китайский Tianhe-2 (33,9 петафлоп), на третьем — Piz Daint швейцарского национального суперкомпьютерного центра (19,6 петафлоп),

на 4-е место попала система Gyoukou (Япония), а Titan Национальной лаборатории Оук-Ридж (США) опустился на 5-е место.

Когда в июне 1993 г. был составлен первый Top-500, Linux еще рассматрива-

На Linux работают все 500 лучших суперкомпьютеров мира.

лась едва ли не как игрушка. Суперкомпьютер на этой ОС впервые вошел в Top-500 в 1998 г.; безусловным лидером среди ОС для суперкомпьютеров долгое время оставалась UNIX. К 2003 г. в Top-500 уже проявлялась тенденция к доминированию Linux; с 2004 г. Linux — бесспорный лидер.

Linux Foundation называет две причины такого успеха.

Во-первых, большинство суперкомпьютеров созданы для решения узкоспециализированных задач, и разрабатывать отдельную пользовательскую ОС для каждой такой системы бессмысленно, поскольку исходный код Linux может быть легко модифицирован и оптимизирован под любой одноразовый проект. Например, ядро Linux 4.14 позволяет суперкомпьютерам использовать Heterogeneous Memory Management (HMM), благодаря которому GPU и CPU получают доступ к общему адресному пространству процесса; 102 системы из Top-500 используют технологии GPU accelerator/ со-processor, которые будут улучшены благодаря HMM. Во-вторых, стоимость лицензирования дистрибутива Linux не зависит от количества узлов, на которых он используется, благодаря чему затраты разработчиков остаются на уровне других ОС или даже ниже.

ДИСТРИБУТИВЫ

В новую эру!

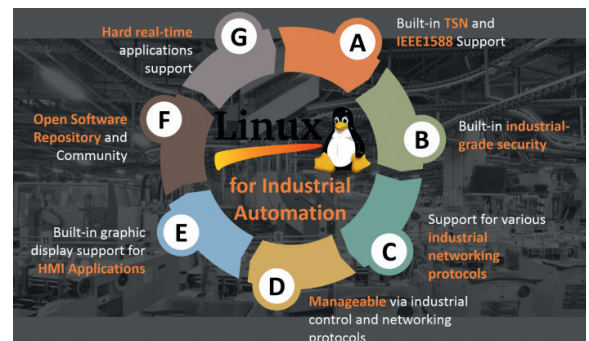
NXP Semiconductors предлагает дистрибутив для «умной» автоматизации производства.

Нидерландская компания NXP Semiconductors анонсировала Linux-дистрибутив с открытым исходным кодом Open Industrial Linux (OpenIL), предназначенный для автоматизации производства и способствующий внедрению на предприятии новых тенденций, присущих эре Индустрии 4.0 (Industry 4.0), или четвертой промышленной революции.

Пользователи OpenIL получают доступ к настоящей промышленной безопасности на основе надежных вычислений, упрощенного ПО, криптографических операций и комплексной безопасности; OpenIL поддерживает TSN (синхронизируемые по времени сети), включая ограничение по каждому потоку и формирование сетевого трафика с привязкой по времени, а также структуру файловой системы Ubuntu, что позволяет использовать стандартные возможности SELinux.

OpenIL опирается на проект buildroot (поэтому разработчики могут адаптировать свои прошивки к флэш-памяти доступного формата, добавлять или удалять приложения по мере необходимости) и содержит ПО для взаимодействия с ИТ-инфраструктурой (сетевые стеки, web-серверы), инструменты для создания сценариев и системные утилиты, имеющиеся в любом дистрибутиве Linux, а также расширения реального времени; поддерживаются сети TSN (чтобы настроить параметры TSN, требуется только обновить XML-файл), сервисы edge computing (периферийные, или граничные вычисления) и точная синхронизация устройств в режиме реального времени с использованием gPTP; есть драйверы Ethernet для сетей TSN.

В состав OpenIL входит фреймворк для разработки приложений реального времени Xenomai; расширения реального



OpenIL объединяет безопасность, поддержку TSN и граничных вычислений, а также нацеленность на Industry 4.0 в одном промышленном дистрибутиве Linux.

времени превращают Linux из ИТ-ориентированной ОС в полноценную RTOS с полным комплектом ИТ-функций, и облегчают портирование из RTOS такого ПО как VxWorks или pSOS.

КИБЕРБЕЗОПАСНОСТЬ

4 × 9 = нейтрализация угроз

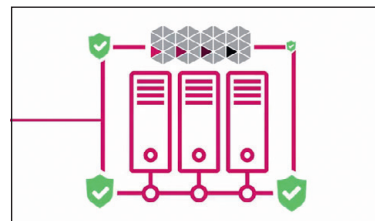
DNS-сервис Quad9 блокирует ботнеты и фишинговые атаки, используя собранную IBM базу данных.

Global Cyber Alliance (GCA) совместно с IBM и Packet Clearing House представили бесплатный публичный сервис DNS, под названием Quad9 (по его IP-адресу 9.9.9.9), для блокировки доменов и интернет-хостов, связанных с ботнетами, фишинговыми атаками и т.п. В первую очередь Quad9 ориентирован на компании, не использующие собственные фильтры DNSBL и DNSWL.

Quad9 работает аналогично другим публичным серверам DNS, однако не возвращает разрешение имен сайтов, которые специализированными сервисами ежедневно идентифицируются как «представляющие угрозу». При этом персональная информация и IP-адреса конечных пользователей Quad9 не собираются и не сохраняются; по словам представителя GCA, храниться будут только черновые данные геолокации, необходимые для отслеживания распространения запросов, связанных с конкретными вредоносными доменами.

Одним из основных источников информации Quad9 об угрозах является система IBM X-Force, база данных которой содержит более 40 млрд проанализированных веб-страниц и изображений; также используются каналы от 18 других партнеров, в числе которых Abuse.ch, Anti-Phishing Working Group, Bambenek Consulting, F-Secure, mnemonic, 360Netlab, Hybrid Analysis GmbH, Proofpoint, RiskIQ и ThreatSTOP. На основе списка «миллион наиболее запрашиваемых доменов», а также «золотого списка» безопасных провайдеров, под которыми подразумеваются сайты основных интернет-сервисов (Microsoft Azure cloud, Google, Amazon Web Services и т.п.) генерируется «белый список» никогда не блокируемых доменов. Затем список заблокированных сайтов, как и белый и «золотой» списки, преобразуются в формат Response Policy Zone (RPZ) и направляются в кластеры серверов DNS, поддерживаемые Packet Clearing House через зоны DNS по всему

► GCA выражает надежду, что DNS-платформа Quad9 будет востребована и пользователями Интернета вещей, поскольку «умная» бытовая техника зачастую лишена обновлений безопасности.



миру. Кластеры серверов DNS, каждый из которых сбалансирован по нагрузке с помощью dnsmdist, используют сочетание серверов Unbound и PowerDNS для доставки ответов.

С момента запуска нового сервиса кластеры серверов DNS для него были сконфигурированы в 70 пунктах 40 стран мира; до конца года их количество будет доведено до 100, а в 2018 г. — до 160. IBM сообщает, что данные телеметрии от Quad9 о заблокированных доменах будут совместно использованы партнерами для улучшения ответов на угрозы для своих клиентов.

НОВИНКИ AMD

Суперкомпьютеры для всех

Представлена P47, высокопроизводительная вычислительная система на базе процессора EPYC.

На состоявшейся в ноябре в Денвере Международной конференции по высокопроизводительным вычислениям SC17 компания AMD и ее партнеры анонсировали системы высокопроизводительных вычислений (HPC) на основе CPU AMD EPYC и графических ускорителей AMD Radeon Instinct, которые в сочетании с предлагаемым AMD ПО и новой открытой платформой ROCm 1.7 с обновленными средствами разработки и библиотеками позволяют создавать готовые системы с петафлопсной производительностью (среди особенностей ROCm 1.7 — поддержка самых новых видеопроцессоров Radeon и фреймворков глубинного обучения TensorFlow и Caffe). В ходе презентации было заявлено, что AMD намерена сделать суперкомпьютеры доступными для широкого применения в корпоративных ИТ-системах и небольших центрах обработки данных.

Система P47 от компании Inventec оснащается одним CPU из серии EPYC 7000, а также четырьмя GPU Radeon Instinct MI25, каждый из которых обеспечивает производительность до 12,3 терафлоп при одинарной точности. При запуске множественных нагрузок один EPYC 7601 дает трехкратное преимущество по показателю «производительность на доллар» по сравнению с Intel Xeon Platinum 8180M. Компания AMAX разработала [SMART] Rack P47, включающее до 20 платформ P47 высокопроизводительное устройство стоечно-масштабируемой архитектуры, обеспечивающее вычислительную мощность до 1 петафлоп и поддерживающее до 10 ТБ памяти DDR4 на каждую стойку; в комплект также входит оптимизированное для задач HPC программное обеспечение [SMART]DC DCiM. «Начиная с одной платформы P47, мы можем масштабировать

► AMD принимают предварительные заказы на HPC-серверы и комплексы [SMART]Rack P47, начало поставок которых запланировано на первый квартал 2018 г.



производительность вплоть до класса суперкомпьютеров», заявила Джулия Ших [Julia Shih], вице-президент по развитию бизнеса компании AMAX.

Список OEM-производителей и системных интеграторов, предлагающих системы на базе EPYC, включает 13 компаний, в том числе ASUS, BOXX, Gigabyte, HPE, Penguin, Silicon Mechanics, Supermicro, Synnex и Tuan (примечательным является отсутствие в этом списке компании Dell).

ОБОРУДОВАНИЕ

Независимы от Windows 10

Пять новых компьютеров Dell Precision совместимы с GNU/Linux.

Компания Dell выпустила пять новых компьютеров серии Precision; все они сертифицированы Canonical, их пользователям не придется беспокоиться о совместимости. В UK цены на новые Precision варьируются в пределах от £1,2 тыс. до £3 тыс.

Рабочая станция Dell Precision 5720 оснащена 27-дюймовым дисплеем с разрешением 4K Ultra HD и графической картой AMD Radeon Pro WX, ОС — Ubuntu 16.04 LTS; предлагается выбор между процессорами Intel Xeon и Core 6-го или 7-го поколения.

Ноутбук Dell Precision 7720 имеет 17-дюймовый LED-дисплей 1600×900 и четыре конфигурации; в базовой модели установлен CPU Intel Core i5-7300HQ. Объем памяти, в зависимости от цены, от 8 до 64 ГБ. Графическая карта в максимальной ком-

плектации — Nvidia Quadro P4000 8 ГБ GDDR5.

Ноутбук Dell Precision 7520 — это компактная версия модели 7720, с 15-дюймовым FHD-дисплеем, CPU Intel i5-7300 и до 32 ГБ ОЗУ.

Диагональ дисплея Dell Precision 5520 также 15 дюймов; базовая комплектация предлагает CPU Intel Core i5-7440HQ и HDD 500 ТБ. В топовой комплектации Dell Precision 5520 поставляется с GPU Quadro M1200.

Самый бюджетный вариант — ноутбук Dell Precision 3520; за £1228 пользователь получает CPU Core i5-7440HQ и графику Nvidia Quadro M620 с 2 ГБ памяти.

«Мы рады объявить о появлении пяти новых компьютеров Dell Precision, поставляемых с предустановленной Ubuntu», последовал комментарий от Canonical.

ПО НОВЫМ ТЕХНОЛОГИЯМ

Память, ты же можешь!..

Компания Micron Technology удвоила емкость модулей NVDIMM.

В ходе состоявшейся 13–16 ноября в Денвере выставки-конференции SuperComputing 2017 (SC17) компания Micron Technology представила новые модули памяти NVDIMM-N, емкость которых — 32 ГБ — в два раза превышает существующие решения NVDIMM.

Модуль NVDIMM-N объединяет стандартную память DDR4 RDIMM и флэш-накопитель, что повышает производительность и позволяет восстанавливать данные в случае непредвиденной перезагрузки системы (например, из-за сбоя питания). Новый модуль, получивший обозначение MTA36ASS4G72XF1Z — 32 ГБ, оснащен 32 ГБ (4 GIG×72) памяти DDR4 RDIMM и 64 ГБ SLC Flash, его пропускная способность 23,47 Гбит/с, а скорость — 2933 МТ/с (мегатранзакций в секунду).

По форм-фактору модули NVDIMM-N идентичны DDR4 и в специальных слотах

не нуждаются. Micron рекомендует применять память NVDIMM-N для ведения журнала транзакций или хранения данных о файловой системе; в этих случаях на двухпроцессорном сервере достаточно будет иметь 64 ГБ NVDIMM-N.

По словам представителя Micron, в идеальном случае использование NVDIMM-N увеличит производительность платформы до 400%.

Сотрудничество Micron Technology с такими компаниями, как VMware и Dell, нацеливается на дальнейшее повышение производительности NVDIMM для виртуализированных приложений.

Подобная виртуальная энергонезависимая память позволяет клиентам запускать в виртуализированной среде сразу несколько операционных систем при одновременном снижении общего сетевого трафика. **LXF**

Новости короткой строкой

➤ Ускоритель [booster] поднял с 80 на 29 место в Top-500 созданный российской компанией «Т-платформы» для немецкого суперкомпьютерного центра в Юлихе суперкомпьютер JURECA. Источник: www.cnews.ru

➤ «Лаборатория Касперского» открыла в Московском техническом университете связи и информатики (МТУСИ) научно-образовательный центр по информационной безопасности. Источник: www.kaspersky.ru

➤ После выхода ядра 4.15 (февраль 2018 г.), RISC-V Linux ABI переведут в разряд стабильных интерфейсов с обратной совместимостью, готовых для повсеместного использования. Источник: www.opennet.ru

➤ В релизе свободной гоночной игры Supertuxkart 0.9.3 для Linux, Android, Windows и macOS оптимизирован mesh-формат для пространственного и аппаратного скиннинга. Источник: blog.supertuxkart.net

➤ Western Digital представила интеллектуальные встроенные флэш-накопители (EFDs) iNAND 8521 и iNAND 7550 для работы с контентом 5G, AR, AI и панорамного потокового видео. Источник: www.zdnet.com

➤ Флагманский чипсет Qualcomm Snapdragon 845 получил архитектуру ядер CPU Kryo 385, GPU Adreno 630, мультимедиа-процессор Spectra 280 ISP и изолированную подсистему безопасности Secure Processing Unit. Источник: www.theinquirer.net

➤ В Crossover 17 уровень совместимости поднят с Windows XP до Windows 7, обеспечена совместимость с MS Office 2016, улучшена поддержка MS Office 2010 и 2013. Источник: www.codeweavers.com

➤ В Android 8.1 добавлены API Neural Networks и WallpaperColors, класс SharedMemory и функция setAutofillOptions(), улучшены средства автозаполнения полей. Источник: android-developers.googleblog.com

➤ Ubuntu с Unity 7 возродится в одобренной Canonical сборке от энтузиастов-профессионалов, среди которых — бывшие разработчики из Canonical и лидер Ubuntu Mate Мартин Вимпресс [Martin Wimpress]. Источник: ubuntu-news.ru

СКАЖИ УЧЕБЕ

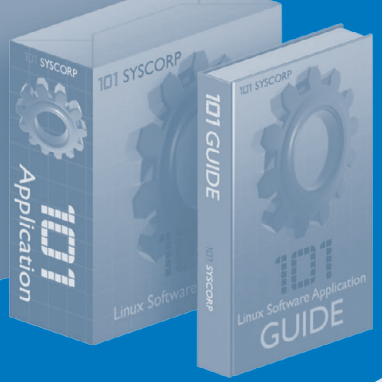
ДА!



**СЕРВЕРНЫЕ РЕШЕНИЯ
LINUX
SOLARIS
ХРАНЕНИЕ ДАННЫХ
JAVA
ANDROID
БЕЗОПАСНОСТЬ
ЗАЩИТА ДАННЫХ**

Санкт-Петербург,
ул. Яблочкова, 12, литер Ц
(812) 611 1575
unixedu.ru

UnixEducationCenter



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

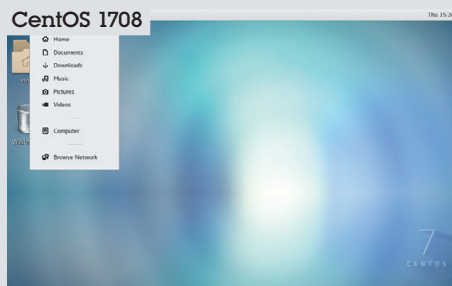
Обновление Cintu

Базис — 17.10, надстройка — Cinnamon 3.6. В инфраструктуре Ubuntu издавна представлены официальные дериваты «головной» системы с множеством рабочих окружений: от старинного KDE до недавно возникшего Budgie. Нет в этом списке только одного — Cinnamon. И его пакеты в официальной репозитории появились не так давно, не позволяя собрать самодостаточную систему, да и отстают от Mint'овского апстрима на полгода для текущего релиза Ubuntu, и на несколько версий — для релизов предыдущих. Причины этого явления автор обсуждать не будет (хотя и имеет свои соображения). Ибо предпочел исправить эту недоработку, представив систему, названную им Cintu (см. **LXF216**). Каковая вот уже более полутора лет собирается на базе Ubuntu 16.04 LTS со средой Cinnamon из PPA-репозитория Цветков Цветкова. «Жить можно бы бесечно при эдаком» наборе... по крайней мере, до 2021 г. Однако — скучновато, да и очередной LTS-релиз, 18.04, уже на горизонте. А поскольку (см. колонку **LXF229/230**) технические проблемы по сборке Cintu на базе «промежуточных» релизов оказались (пока?) решаемыми, эта система была представлена в виде микста из базиса Ubuntu 17.10 и текущей версии Cinnamon 3.6, пакеты для которой взяты из PPA-репозитория Гвендале Биана (Gwendal Le Bihan). Ссылка для скачивания образа — <http://alv.me/cintu-17-10-i-cinnamon-3-6/>.

Правда, оказалось, что у Cinnamon есть альтернативный источник пакетов, о чем — в следующий раз. alv@posix.ru

Сегодня мы рассматриваем:

- CentOS 1708** 12
Этот отпрыск коммерческого дистрибутива Red Hat заслужил репутацию стабильного и надежного, и его охотно выбирают администраторы серверов.
- Knoppix 8.1** 13
Дистрибутив, некогда славный распознаванием оборудования, отноше не утратил актуальности (правда, новую версию приходится всякий раз загружать с нуля). Свежая версия ориентирована на создание 3D-контента.
- Koora 26 Gnome** 14
Вроде та же Fedora, только с проприетарными кодеками и меньшим набором приложений — чтобы не пугать новичков; однако амбиции проекта растут, а его «домашние» инструменты хорошеют.
- КаОS 2017.09** 15
KDE-дистрибутив, построенный на чистом, беспримесном Qt, обзавелся интересными новыми компонентами. А если вы любитель рискованных предприятий, то для вас предусмотрена новая файловая система Btrfs.
- Intel Core i5 8400** 16
Голубой гигант вернулся со впечатляющей новой архитектурой процессора — Coffee Lake, превосходной для игровых целей.
- Intel Core i9 7960X** 17
16 ядер — это нечто, но расплачиваться придется повышенным энергопотреблением. Кстати, а нужна ли вам такая производительность?
- Google Pixel 2** 18
Столь классной камеры на смартфоне доселе не бывало, но во всем прочем его характеристики оставляют желать лучшего.
- Tacoma** 19
Детективная история на внезапно опустевшей космической станции: вам необходимо восстановить картину событий, приведших к гибели экипажа, на основании сохранившихся видеозаписей.



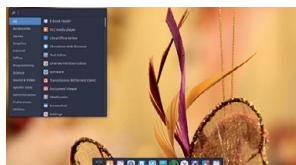
Для настольной ОС этот выбор нетипичен, но графический интерфейс всё равно вполне достойный.



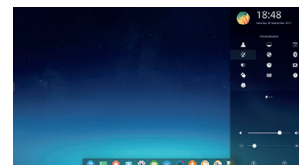
Хотя персонажи игры показаны безликими цифровыми призраками, каждый имеет свой характер.

Сравнение: Рабочие столы с. 22

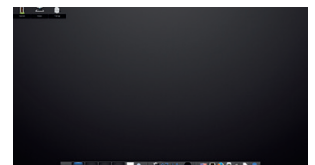
Budgie



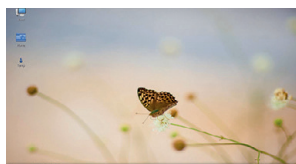
Deepin



Enlightenment



LXQt



Pantheon



Потребность в быстром рабочем столе-легковесе, приятном с виду и в работе, породила новое поколение. Стоит ли засиживаться на установленном по умолчанию? Вдруг ваша мечта совсем рядом?

CentOS 1708

Джонни Бидвелл сам признает, что стабильность и надежность — не его сильные стороны. Сумеет ли CentOS его вдохновить? Понадеемся...

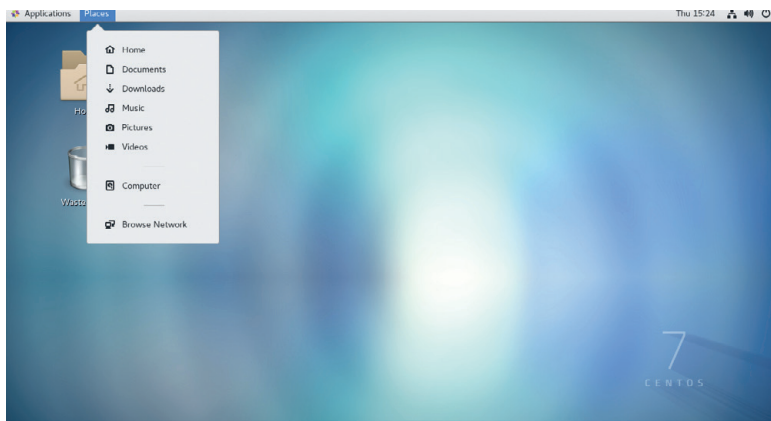
Вкратце

» Один из самых надежных и популярных серверных дистрибутивов. Основан на RHEL 7.4 и напигован всеми присущими тому корпоративными функциями, но без ценника. См. также: RHEL, clearOS, SUSE Linux Enterprise Server, Debian, Container Linux, Slackware.

После выпуска версии 7 (июль 2014 г.) мы не рассматривали CentOS, но только что вылетел новый релиз, и сейчас, пожалуй, пришло время компенсировать. CentOS основан на Red Hat Enterprise Linux (RHEL) с коммерческой лицензией. Исходный код RHEL по-прежнему открыт и доступен, но обновления делаются только по подписке, а символика Red Hat не бесплатна. Таким образом, удаление товарных знаков с исходников RHEL, их пересборка и разумное тестирование дадут вам ОС корпоративного уровня, без ограничений на распространение — короче, CentOS, и вы можете узнать больше о проекте из нашего выпуска LXF221.

Существует ряд ISO, из которых можно выбирать в зависимости от ваших целей. Есть минимальные опции и опции установки через сеть, для тех, кто хочет избежать лишних пакетов или не спалить полосу пропускания. Также доступны живые среды KDE и Gnome. Рекомендуемые действия — использовать (слегка неоднозначно названный) образ “DVD”, включающий все пакеты, устанавливаемые «из коробки». Он весит 4,5 Гб, и его-то мы и взяли. Более увесистый образ Everything содержит все пакеты, доступные для CentOS.

Установщик (Anaconda, как в Fedora и RHEL) предлагает выбор базовых сред, преднастроенных на популярные роли: Compute Node [Вычислительный узел], Virtualization Host [Хост виртуализации], а также Development [Разработка] и Creative Workstation [Рабочая станция для творчества]. Каждую из сред можно с того же



» Мягкие пастельные тона, значки на рабочем столе и меню сверху могут погрузить пользователей в мечтательные воспоминания о Gnome 2. Встряхнитесь, люди!

экрана настроить дополнительно, с помощью надстроек: например, добавить группу Development Tools [Инструменты разработчика] в среду Minimal, предположительно создав минимальную среду разработки.

Перебор выбора?

Пользователя настольного ПК такое многообразие может ошеломить, но CentOS — явно нетипичный выбор для настольной ОС. Тем не менее, мы установили сервер с графическим интерфейсом вместе с набором надстроек. Порядка 1400 пакетов, одна ошибка *QEti* и последующее принятие лицензионного соглашения длиной в одно предложение — и наша установка готова к работе. Мы обнаружили, что загрузили ядро 3.10, ветвь с долгосрочной поддержкой, на которую Red Hat портирует функции и исправления, и по своим возможностям она значительно отличается от стандартной версии 3.10 от 2013 г. Например, драйвер NVMe был переустановлен на ядро 4.10, а драйвер AMDGPU теперь поддерживает чипсеты до Арктических островов.

Наш вариант установки дал нам рабочий стол Gnome 3.22. Тем не менее, большинство установок CentOS предназначены для серверов, поэтому с рабочими столами не возитесь. Бывают ситуации, когда графический интерфейс полезен, и наличие всего одного такого — не позор, но в репозиториях CentOS не так много ПО для настольных компьютеров, а то, что имеется, будет намного старше, чем в дистрибутивах, более ориентированных на настольные ПК. Правда, репозитории можно

добавить; в частности, пакет Extra Packages для Enterprise Linux (EPEL) обеспечивает надежный перенос пакетов из Fedora для использования с RHEL и их производных.

Одной из основных причин, по которым люди выбирают CentOS, является его долговечность. Текущая версия, как и ее родич Red Hat, будет получать полные обновления до 2020 г. и поддерживаться посредством обновлений технического обслуживания до 2024 г., спустя целое десятилетие после появления. Люди выбирают CentOS не просто из-за ценника. Они делают это из-за надежной репутации. Многим пользователям хватает квалификации или уверенности для самостоятельной поддержки своих собственных установок, не только для проектов личных серверов, но и для бизнеса, малого и большого. LXF



Свойства навскидку

INSTALLATION SOURCE	Local media
SYSTEM	
INSTALLATION DESTINATION	Automatic partitioning selected
NETWORK & HOST NAME	Not connected



Файлы Kickstart

Установки могут быть автоматизированы и настроены с помощью текстовых файлов. Идеальны при массовых развертываниях.

Репозиторий EPEL

Популярные пакеты, портированные из Fedora, доступны здесь, и можно установить игры, рабочие столы и многое другое.

LINUX FORMAT Вердикт

CentOS 1708

Разработчик: Проект CentOS
Сайт: www.centos.org
Лицензия: Различные свободные

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

» Прекрасный пример того, что может создать открытый исходник и выделенное сообщество. Надежен и готов служить.

Рейтинг 8/10

Кнопrix 8.1

Спустя годы после первого выпуска дистрибутив по-прежнему свеж и надежен. Прими аплодисменты, Кнопrix, ты впечатлил **Шашанка Шарму**.

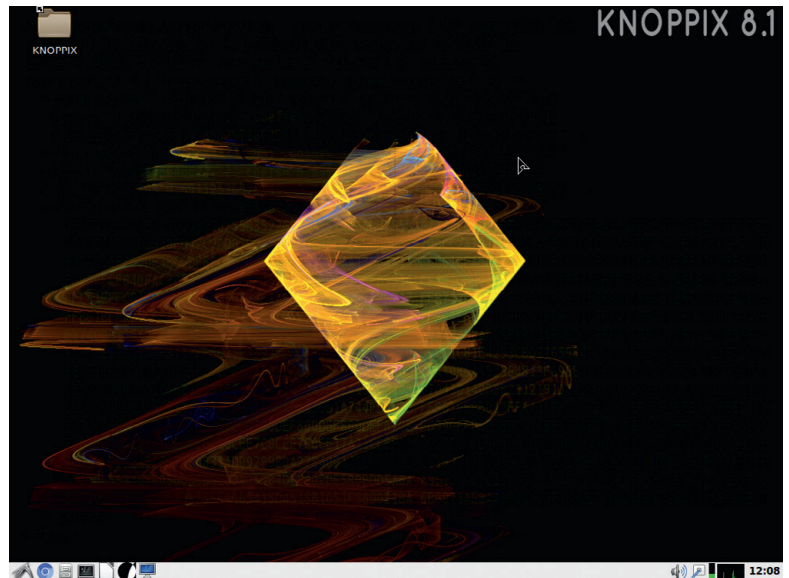
Вкратце

» Live Linux породил целое поколение гибридных дистрибутивов. Хотя этот проект по-прежнему поддерживает 32- и 64-разрядные системы, Live-CD больше не создается. Вместо этого пользователи должны загружать 4-ГБ ISO, заполненный лучшим ПО с открытым исходным кодом. Дистрибутив одинаково отзывчив на новых и более старых машинах, ладит даже с 2 ГБ ОЗУ. См. также: Debian.

Почти 18 лет спустя своего дебюта Кнопrix продолжает внедрять инновации. Правда, выход этой версии немного странен по нескольким причинам. Во-первых, в отличие от большинства других дистрибутивов, Кнопrix поставляется исключительно Live-DVD образы. Предложение на 4 ГБ, полное и практичных, и эзотерических приложений, вероятно быстрое. Во-вторых, дистрибутив является одним из тех, кто всё еще предлагает *Compiz*, ни много ни мало, включенный по умолчанию. Линуксоиды-ветераны помнят *Compiz* как диспетчер композитинга окон, вносящий ряд 3D-эффектов на рабочий стол Linux, но в последние годы он ушел в тень. Дистрибутив можно установить на диск USB или жесткий диск, но предназначен он для работы с USB в качестве живого носителя. Более того, можно скопировать ISO на USB-устройство, используя команды `cp` или `dd`. В обоих случаях вы получите самозагружаемое устройство.

Изобилие приложений

Последняя версия Кнопrix представляет собой сочетание приложений и технологий с открытым исходным кодом. В меню приложения сортируются по категориям, таким как Интернет, Игры и Звук и Видео. Особенно нам понравилась коллекция игр, головоломок и простых инструментов в помощь обучению детей программированию. Также включены *VLC* и другое ПО для воспроизведения мультимедиа-файлов, вместе с требуемыми кодеками. ПО можно



» Кнопrix предлагает ПО, выведенное из всех трех ветвей Debian — Stable, Testing и Unstable — и включает драйверы для некоторых новых видеокарт.

добавить (в т.ч. *Audacity*), используя установщик компонентов из меню Кнопrix.

Хотя рабочий стол по умолчанию — *LXDE*, DVD также включает KDE и Gnome. Загружаются эти среды командами `knoppix desktop=gnome` или `knoppix desktop=kde`.

Функция *ADRIANE* (Audio Desktop Reference Implementation and Networking Environment) [милый штрих: это имя супруги разработчика, — прим. ред.] позволяет запускать Кнопrix, не подключая машину к устройству визуального отображения, такому как монитор.

Последняя версия ориентирована на 3D и предлагает ряд приложений для создания контента, а именно, *Blender*, *FreeCAD* и *OpenSCAD*. Последний предназначен для пользователей, заинтересованных в 3D-дизайне, но не обладающих необходимыми навыками рисования. Также предлагаются инструменты для 3D-печати: *Slic3r* и другие.

Увы, с браузером *Firefox*, включенным в дистрибутив, не работает звук. Это давно документированная проблема *Firefox*, не имеющая ничего общего с упаковкой для Кнопrix. В качестве обходного решения в дистрибутиве прилагается *Chromium*; и эта самопровозглашенная ошибка — единственная, которую мы нарыли в Кнопrix за время его использования.

Этот дистрибутив не предназначен для замены вашего текущего дистрибутива

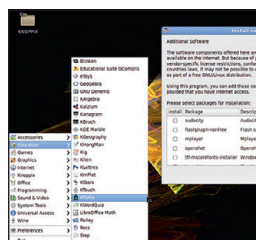
Linux. Его также нельзя установить на диск и использовать повседневно. Частично это объясняется тем, что дистрибутив, в отличие от своих коллег, не является обновляемым. Вы должны дождаться, когда последующие выпуски включают более новую версию ПО, и загрузить ISO заново, чтобы получить доступ к последним предложениям. Хотя Кнопrix может играть роль аварийного дистрибутива для вашей системы, он разработан скорее как средство продемонстрировать доблесть открытого исходного кода и приобщить пользователей к лучшему доступному ПО. **LXF**

Свойства навскидку



Adriane

Разработанная для слабовидящих пользователей, разговорная система меню также может использоваться детьми.



Коллекция ПО

4-ГБ ISO обуславливает широкий простор для обширного спектра программ с открытым исходным кодом.

LINUX FORMAT Вердикт

Knoppix 8.1

Разработчик: Клаус Кноппер [Klaus Knopper]
Сайт: www.knopper.net/knoppix
Лицензия: Разные

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

» Идеальный плацдарм для несведущих в Linux; впечатляет коллекция программ с открытым исходным кодом.

Рейтинг 9/10

Korora 26 Gnome

Попытка работать в Korora — один из способов знакомства с персонажами «В поисках Немо». Но **Шашанк Шарма** любит его за простоту и удобство.

Вкратце

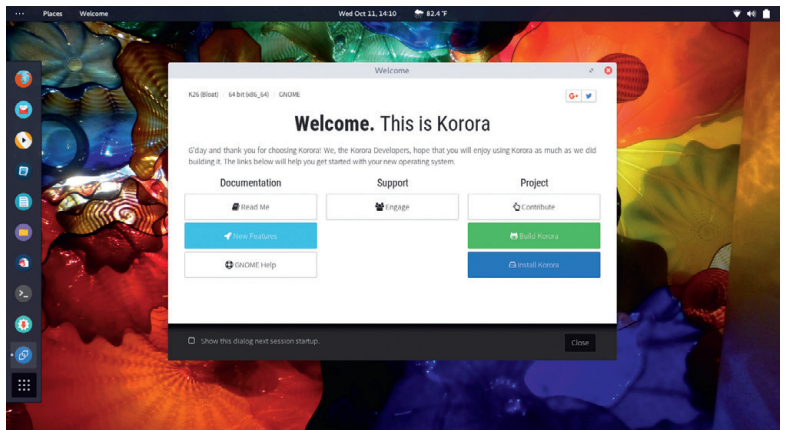
» Дружелюбный к новичкам дистрибутив на базе Fedora Linux сочетает в себе лучшие качества, а именно стабильность и современные пакеты, с проприетарными дополнениями, такими как кодеки и приложения типа VLC. Это делает его идеальным для новых пользователей, не имеющих опыта работы с дистрибутивами Linux. См. также: Fedora, Linux Mint.

Когога старается выпустить новый релиз в пределах нескольких недель после обновлений Fedora, в версиях KDE, Mate, Cinnamon, Xfce и Gnome. Это время уходит на ряд изменений, чтобы дистрибутив с ходу был удобным для более широкого круга пользователей. Вот почему Koroga поставляется с несколькими проприетарными кодеками и различными приложениями, не входящими в Fedora по умолчанию во избежание проблем с лицензированием. Еще одно серьезное различие между Koroga и Fedora — первый поставляется с меньшим набором приложений. Фактически, вы найдете только по одному приложению на каждую задачу. Правда, этой коллекции хватит для большинства пользователей; и вы всегда можете подсилить дистрибутив, обратившись к репозиториям ПО, которые содержат тысячи дополнительных пакетов.

Но не дайте наличию сторонних проприетарных предложений посеять в вас мысль, что код Когога не открытый. Исходный код доступен для загрузки через репозиторий Git. Как и родитель, Koroga использует для выпуска своих версий скрипты Kickstarter. Можно использовать инструмент *kr* для модификации кода Когога, чтобы создать собственный ремикс-дистрибутив.

Грядущие изменения

Из-за шестимесячного цикла выпуска следующие релизы дистрибутива могут выглядеть как предыдущие итерации, причем мало что отличает друг от друга пару



» Благодаря решениям вроде включения в дистрибутив Gnome Tweak Tool, Koroga отлично подходит новым пользователям.

выпусков, кроме номеров версий некоего включенного ПО. Но проект решил прислушаться к своему сообществу и внести необходимые изменения.

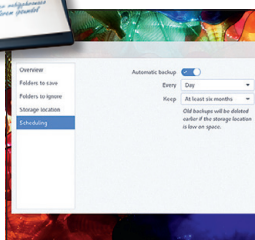
В Koroga 26 пользователям намного проще защитить свои установки от порчи, используя *Canvas*, еще один домашний инструмент для помощи в управлении и настройке Koroga. С помощью *Canvas* пользователи могут создать шаблон своей установки и восстановить систему, если что-то сломается при настройке конфигурации или установке нового ПО. В следующем выпуске проект надеется сосредоточить свои усилия на рабочих столах KDE, Cinnamon и Mate, а также улучшить установщик *Anaconda*. В перспективе дистрибутив может даже отказаться от *Anaconda*, введя альтернативу. Аналогично, хотя стандартным менеджером приложений в Koroga 26 является ПО *Gnome Software*, проект хочет оптимизировать свои предложения по различным редакциям и исследует решение, не зависящее от рабочего стола, на основе *Canvas*; но это в очень дальней перспективе.

Основное внимание мы уделили версии с Gnome, но также пробовали другие редакции Koroga, чтобы увидеть, нет ли такой, которая затмевает остальные. Но, хотя Gnome считается громоздким и медленным, особенно на фоне альтернатив типа Xfce, резких различий в производительности мы не обнаружили. Настройка по умолчанию не позволяет размещать значки на рабочем столе Gnome, но можно изменить это и ряд других параметров конфигурации через доступный по умолчанию

инструмент Gnome Tweak. Кроме того, дистрибутив в этом последнем выпуске добился прогресса по сравнению с исходной документацией. С ним теперь поставляется подробное руководство по передовым методам и стратегиям резервного копирования; но поле для улучшения пока еще есть.

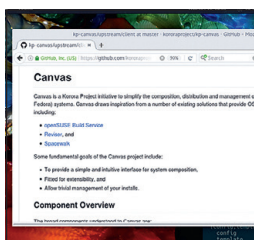
В прошлом Koroga всегда воспринимался как готовый к употреблению Fedora, идеальный для новых пользователей Linux. Однако отсутствие у него оригинальных функций и конкретных целей на будущее не давало матерым линуксоидам повода отказываться от своих установок в его пользу. План развития Koroga 27, а также будущие выпуски говорят о более агрессивной позиции, свидетельствующей о желании проекта подняться выше в рейтинге дистрибутивов. **LXF**

Свойства новскидку



Удобная утилита

Каждый выпуск свежей версии Koroga по умолчанию поставляется с инструментом резервного копирования.



На «Холсте»

Инструмент Canvas [англ. холст] пригоден для создания шаблонов и восстановления системы по этим шаблонам.

LINUX FORMAT Вердикт

Korora 26 Gnome

Разработчик: Проект Когога
Сайт: <https://korogaproject.org>
Лицензия: Разные

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

» Удобный для пользователя дистрибутив на RPM, который сразу работает, сделал гигантский шаг вперед с Canvas.

Рейтинг 9/10

KaOS 2017.09

Наша записная «Обезьянка Хаоса», **Джонни Бидвелл**, затребовал чистый, без примесей рабочий стол *Qt5*, и KaOS исполнил его пожелания.

Вкратце

» **Независимый, возобновляемый дистрибутив для фанатов современного оборудования и KDE/Qt. В официальных репозиториях нет 32-битных пакетов; ищите, где хотите. Но если вам нужен стильный рабочий стол и новейшие программы, добро пожаловать. См. также: Chakra Linux, KDE neon, Kubuntu, NetRunner Rolling.**

В зависимости от ваших предпочтений, первое, что вы заметите в KaOS, будет или а) вертикальное меню запуска приложений, правда, расположенное справа [Ред.: — Пошлость какая!] или б) удобный мастер первичной настройки *Kapitan*. *Kapitan*, ответившийся от турецкого дистрибутива Pisi Linux, поможет настроить внешний вид рабочего стола и выбрать меню на свое усмотрение (классика снова в моде!), и укажет на необходимую документацию. Это будет полезно новичкам в KDE, поскольку попытки отыскать требуемую настройку наверняка спровоцируют их на сравнения с кошмарным борхесовским лабиринтом.

В KaOS доступна стильная тема *Midna*, частично примененная по умолчанию (в остальном используется *Breeze*, тема по умолчанию в *Plasma*). Такая конфигурация, равно как и сама по себе *Midna*, выглядит неплохо. Фанатов темных тем (вас, несчастные) порадует, что в варианте *Midna Dark* читабелен текст во всплывающих подсказках, контекстных меню и в других местах, которые забыли затушевать. *Plasma*-виджеты стикеров и калькулятора включены по умолчанию и пригодятся тем, кому приходится часто что-то записывать и подсчитывать.

По нашим подсчетам, это наш четвертый обзор KaOS (см. **LXF188**, **197**, **212**), но команда не намерена почить на лаврах, так что в системе много новшеств. На самом деле, в данном релизе фигурирует



» Хорошо, что *Yakuake* был уже установлен, и нам очень понравилась стильная подсказка терминала. Стилей можно добавить, установив *powerline*.

целый ряд новых компонентов KDE — Frameworks 5.38, Plasma 5.0 и Applications 17.08. Всё это — поверх *Qt 5.9.1*. Ко времени, когда вы будете это читать, номера версий изменятся, что и свойственно возобновляемым релизам. *Qt4* удален из всех репозиториях, так что у любителей поплакаться об избыточности инструментария повода не будет. Только *Qt5*, ребята!

Фишки файловых систем

В KaOS, как и в RHEL 7.0, теперь по умолчанию используется файловая система XFS; более консервативным по душе придется Ext4, а более рискованным — Btrfs. Что интересно, если вы выберете вариант с UEFI, KaOS установит не *Grub*, а *systemd-boot*. Если ядро системы покажется вам недостаточно новым, то всего пара нажатий клавиш/щелчков отделяют вас от *linux-next*, а вместе с ним и от ошибок, которые, как мы надеемся, исправят к тому моменту, когда система доберется до версии 4.14.

Проприетарный драйвер *Nvidia* доступен в основном репозитории и не слишком отстал (384.67 против современного 384.90) от доступного на сайте *Nvidia*. В драйвере *Nouveau* применена экспериментальная заплатка, исправляющая систематическую ошибку при работе с *QtWebEngine*. Так что теперь можно беспрепятственно пользоваться браузером по умолчанию *Qupzilla* (который вскоре станет *Falkon*). Использованию *Plasma* на движке *Wayland* особого внимания не уделялось, но прогресс в этом плане

налицо, и в менеджере входа доступен экспериментальный режим.

Многие считают KDE неоп официальной выставкой новейших элементов *Plasma*, и вообще-то это справедливо. Но неоп (по словам самих же разработчиков) — «не вполне» дистрибутив, и ПО, не относящееся к *Plasma*, черпает из репозитория *Ubuntu 16.04*, так что оно не юное и может даже сбойть. Чтобы подчеркнуть свою независимость, KaOS завел свои репозитории, и все пакеты перед отправкой тщательно тестируются. Как в настоящем дистрибутиве, правда? В системе предустановлен ряд приложений *Qt*, в том числе *Krita*, *k3b*, *Gwenview* и аудиоплеер *Babe*. Хотя многие, конечно, предпочтут *LibreOffice* и *Inkscape* вместо *Calligra* и *Karbon*. **LXF**

Свойства навскидку



OctoPi

Теперь это не просто интерфейс для *Raspian*, а средство просмотра новостей KaOS и способ загрузить свой вариант системы на *GitHub*.



Image Writer

В KaOS теперь есть свой собственный инструмент для записи ISO. Ничего сверхъестественного, но это гораздо дружелюбнее, чем *dd*.

LINUX FORMAT Вердикт

KaOS 2017.09

Разработчик: Команда KaOS

Сайт: www.kaosx.us

Лицензия: Разные

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

» Прекрасный выбор для пуристов *Qt*, но кого-то намеренно небольшие репозитории скорее отпугнут.

Рейтинг 8/10

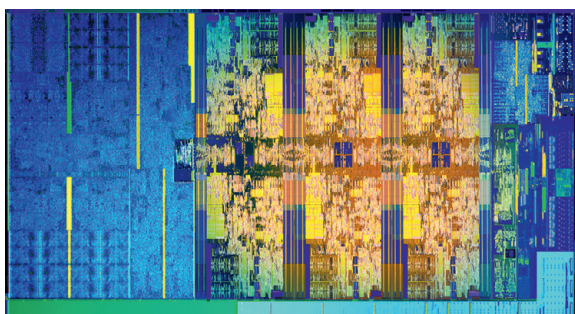
Intel Core i5-8400

После нескольких месяцев бесконечной похвалы AMD Джарред Уолтон наконец-то побалуует Intel и команду голубого гиганта.

Спецификация

- » Разъем 1151
- » Тип 64 бита
- » Технологический процесс 14 нм
- » Ядра Шесть
- » Потоки Шесть
- » Тактовая частота 2,8 ГГц
- » Турбо 4,0 ГГц
- » Кэш 9 МБ
- » Память DDR4-2667, два канала, 64 ГБ макс.
- » Тепловыделение 65 Вт
- » Графический процессор Intel UHD Graphics 630
- » Тактовая частота GPU 350 МГц (1,05 ГГц макс.)
- » Графические блоки 23
- » Виртуализация VT-x, VT-d, EPT

» Кристалл процессора Core i5-8400 во всей красе.



Coffee Lake — самая большая модернизация в поколениях основной линейки процессоров Intel с момента разделения на основные и для энтузиастов, которое в 2010 г. произошло в первом поколении Core i7. И Core i7, и Core i5 получают на 50% больше ядер и потоков, без снижения тактовой частоты, а Core i3 в основном занимает место предыдущего поколения Core i5 в части 4С4Т.

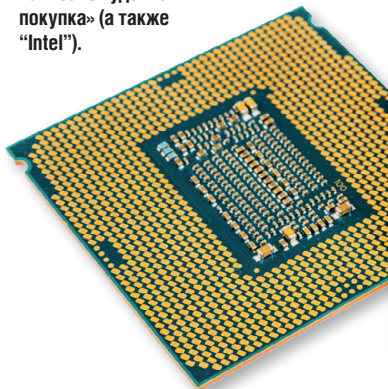
Прежде чем мчаться покупать Core i5-8400, следует вспомнить, что для процессоров Coffee Lake требуется новая материнская плата и набор микросхем серии 300. Прямо сейчас это означает последний Z370. Это связано с измененными требованиями к питанию Coffee Lake. Что касается набора микросхем Z370, то с точки зрения возможностей и функциональности он, похоже, идентичен Z270 — при некотором обновлении микрокода, который распознает только процессоры Coffee Lake.

Возьмите существующие процессоры Core i5, добавьте 50% ядер, и что это вам дает? Играть, уже достигшим максимальной производительности на i5-7600K, ждать особо нечего, но растет количество игр, которые при большем количестве ядер и/или потоков покажут улучшение от мало до умеренного.

Вообще говоря, Hyper-Threading на ядре процессора Intel дает 20–30-процентное увеличение потенциальной пропускной способности, поэтому наличие шести полных ядер, работающих на шести потоках, на самом деле будет быстрее, чем четыре ядра и восемь потоков — по крайней мере на одной тактовой частоте. В целом по среднему показателю игровой производительности «бюджетный» Coffee Lake i5-8400 побивает бывшего чемпиона i7-7700K.

Практически, большинство игроков не ощутили бы никакой разницы между

» На нем прямо-таки написано «удачная покупка» (а также «Intel!»).



любимыми процессорами Core i5 и Core i7 для игр, даже с GTX 1080 Ti. С более скромным графическим процессором и/или более высоким разрешением шестиядерный Ryzen 5 и 8-ядерный Ryzen 7 также хороши в сравнении с Core i5/i7. Они редко, если вообще когда-либо, быстрее в играх, но большую часть времени разница не будет ощущаться так сильно, как предполагалось.

Замер производительности

Как i5 Coffee Lake справляется с интенсивной нагрузкой? На сей раз новичок i5 чуть отстает от i7-7700K — в основном из-за тех тестов, которые вводят в игру более высокую тактовую частоту i7. i5-8400 отстает в *Cinebench 1T* на 16% и на 4% в тесте *Cinebench mT*. В видеокодировании — ничья, с небольшим преимуществом i5-8400 в более требовательном втором проходе x264. *VeraCrypt* отдает предпочтение i5.

Рассматривая другие сравнения, i5-8400 также отстает от процессоров Ryzen 7 в большинстве тестов, а также немного отстает от Ryzen 5 1600X. Ryzen 5 1600 — это точка перегиба, где i5-8400 возвращает себе общее лидерство, и когда мы дойдем до группы 4С8Т Ryzen 5, i5-8400 будет впереди в каждом тесте.

В плане ценообразования прямым конкурентом является AMD Ryzen 5 1600 (£190 против £200 для Core i5-8400), хотя у Ryzen 5 1600X цена немного выше — £220. Стоит отметить, что все эти процессоры включают кулер, а цены на материнские платы в целом предпочтительней у плат AMD B350: платы X370 AM4 в конечном итоге очень близки по цене к платам Z370.

До Coffee Lake процессоры Intel Core i5 всегда были компромиссом; с производительностью многопоточности приходилось проститься. Это и до сих пор верно, но, наконец, выйдя за рамки четырех ядер, Intel восполняет многие недостатки i5. По сравнению с предыдущим поколением i7-7700K, i5-8400 предлагает аналогичную производительность. Немного быстрее в игровых тестах, немного медленнее в общей производительности для приложений; но, что самое приятное, цена примерно вдвое меньше.

Сейчас на рынке процессоров за £200 есть три основных кандидата: Core i5-8400, Ryzen 5 1600 и Ryzen 5 1600X. Для игровых целей — это Core i5-8400, обеспечивающий производительность на 25–35% лучше, чем у Ryzen 5. Core i5-8400 — в целом явно лучший процессор на рынке обычных процессоров. **LXF**

LINUX FORMAT Вердикт

Intel Core i5-8400

Разработчик: Intel
Сайт: www.intel.com
Цена: £200

Функциональность	9/10
Производительность	9/10
Удобство работе	8/10
Оправданность цены	9/10

» Даже с заблокированным умножителем i5-8400 обеспечивает потрясающую производительность — для цены £200.

Рейтинг **9/10**

Intel Core i9 7960X

Теперь, когда Intel пошла вразнос со своими 16 и 18-ядерными Skylake-X, Джарред Уолтон счастлив, как рыба в воде.

Спецификация

- » Разъем 2066
- » Тип 64 бита
- » Тактовая частота 2,8 ГГц
- » Турбо 4,2 ГГц
- » Ядра 16
- » Потоки 32
- » Кэш 22 МБ
- » Память DDR4-2666, 128 ГБ макс., No ECC, 4 канала
- » PCIe 44
- » Тепловыделение 165 Вт
- » Виртуализация VT-x, VT-d

Core i9-7960X — разумная альтернатива 18-ядерному монстру i9-7980XE. Не будем заострять, но помимо Core i9-7900X становится трудно оправдать покупку каких-либо продуктов Intel Core i9, особенно для игровых задач.

Вам всё же нужна веская причина выйти за пределы базового уровня i7-7800X или Ryzen 7 1800X, по крайней мере, с точки зрения производительности процессора. Несомненно, дополнительные ядра помогут ускорить кодирование видео, а у 7960X есть четыре полных (более или менее) чипа i7-7700 внутри, но если вы не ждете от процессора участия в гонках, то лучше вложить ваши деньги во что-то другое.

Энергопотребление Core i9 очень велико. Полное энергопотребление системы складывается из множества компонентов, но у нас есть оборудование только для измерения общей нагрузки. При «штатном» запуске 7960X потребление составляет 369 Вт в *Cinebench* — и более при других нагрузках, особенно использующих систему команд AVX.

Вероятно, из-за расположения или, возможно, прошивки материнской платы, получается, что 16-ядерные процессоры фактически потребляют больше мощности, чем 18-ядерные. Это также распространяется на разгон, где для 7960X максимум — 587 Вт, против 576 Вт для 7980XE. Для материнской платы типа Asus Prime X299-A, с четырехконтактным разъемом ATX12V и восьмиконтактным EPS12V, это немалый ток, протекающий через стабилизатор.

И в штатном режиме, и при разгоне разрыв между i9-7960X и i9-7980XE очень мал.

В лучшем случае 7980XE показывает производительность до 10% выше (когда оба чипа разогнаны и потребляют около 500 Вт); но в среднем разница в производительности составляет всего 3%. С точки зрения стоимости это делает 7960X лучшим выбором.

Если вам интересно сравнение AMD 16-core Threadripper 1950X с 16-ядерными Intel, то в большинстве тестов выигрывает Intel. Threadripper вровень с ним в *VeraCrypt*, имеет небольшое преимущество по скорости распаковки ZIP, но в целом 7960X оказывается примерно на 13% быстрее. Победа отнюдь не всухую.

Не игрушки

С производительностью в играх — та же история, которую мы рассказывали еще о i9-7900X: чтобы заниматься играми, такая производительность процессора по сути не требуется. Даже с самой быстрой графической картой в виде GTX 1080 Ti FE, работающей с ультра-настройками 1080p, чтобы подчеркнуть различия в производительности процессора, практически нет различия между различными процессорами Core i7 и i9.

Из-за сочетания требований цены, охлаждения и мощности Skylake-X намного сложнее рекомендовать как платформу, по сравнению с предыдущими поколениями Haswell-E и Broadwell-E. Чипы Skylake-X почти всегда быстрее, и в обычном режиме, и при разгоне, но Broadwell-E более энергоэффективен. Если у вас уже есть шести-, восьми- или 10-ядерный процессор HSW-E или BDW-E, нет необходимости обновлять его.

Мы всерьез задаемся вопросом, скоро ли 16-ядерные процессоры станут «мейнстримом». Первые четырехядерные процессоры за £1000, такие как Core 2 QX6700, Core i7-920 затмил всего через несколько лет, а к 2010 г. четырехядерные процессоры были обыденностью. После FX-8150 восьмиядерные чипы в различных видах встречаются повсюду. Поскольку Coffee Lake, наконец, ввела в основную платформу Intel шесть ядер, этот барьер



» За этим непритязательным логотипом — сверхмощный процессор почти за £1500.

может, наконец, пасть, но потребуются десять или более лет, прежде чем 16-ядерные решения хоть как-то распространятся.

Этот 16-ядерный i9-7960X получает несколько дополнительных очков сверх своего старшего брата 7980XE за то, что он не настолько громкий и неприятный, но теряет очки за такое же высокое потребление энергии и отсутствие престижа, проистекающее из статуса второго по скорости процессора: «почти у всех это есть». Итак, закончилось ничьей. Так или иначе, вы можете провести высококачественную кодировку видео 1080p60 в режиме реального времени. Или запустить кодирование 4K и пойти на обеденный перерыв. **LXF**



» Если сомневаетесь, печатайте логотип на коробке крупными буквами.

LINUX FORMAT Вердикт

Intel Core i9 7960X

Разработчик: Intel
Сайт: www.intel.com
Цена: £1499

Функциональность	9/10
Производительность	9/10
Удобство в работе	7/10
Справданность цены	5/10

» i9-7960X чрезвычайно мощный, но высокая цена и энергопотребление — горькая пилюля.

Рейтинг **6/10**

Google Pixel 2

Мэтт Свайдер обомлел от самой лучшей из когда-либо им виденных камеры смартфона, но потом понял, что остальное — из 2016 года...

Спецификация

- » ОС Android 8.0
- » Экран 5", 1080×1920, Amoled, Gorilla Glass 5
- » CPU Snapdragon 835
- » GPU Adreno 540
- » Память 4 ГБ
- » SSD 64 ГБ, SD-слот отсутствует
- » Камеры
Тыльная: 12,2 МР, видео 2,160р@30fps.
Фронтальная: 8 МР, видео 1080р
- » Сенсоры (А-) GPS, отпечаток пальца, акселерометр, гироскоп, датчик приближения, компас
- » Габариты 145×70×7,8 мм
- » Вес 143 г

Рixel 2, с его пятидюймовым экраном, не похож на смартфон будущего. Благодаря размеру с ладонь и малому весу он легко умещается в одной руке. Несколько лет назад он считался бы «лопатой», но сегодня, рядом с Note 8, это телефон нормального размера.

Его степень водонепроницаемой защиты IP67 (а значит, он может находиться под водой на глубине до одного метра в течение часа), и на задней крышке сохранен дизайн из стекла и металла, хотя и с меньшим количеством стекла к вершине по сравнению с его предшественниками. Отпечаток пальца был перенесен на текстурную алюминиевую часть задней крышки, а камера осталась со стороны стекла и теперь обведена защитным кольцом.

На этом телефоне вы не найдете гнезда для наушников и даже вкладышей USB-C внутри коробки. Он поставляется только с простым адаптером 3,5 мм для USB-C. Звук по-прежнему великолепен, когда он соединен с наушниками, и нам даже понравилось слушать музыку через двойные фронтальные стереодинамики.

Стискиванием сторон телефона запускается Google Assistant, который, по нашему опыту, удобен. И это значительно лучше, чем добавление специальной кнопки AI, но Google не позволит вам настроить это. То, что отвлекает — это большие каемки. Это особенно заметно в 2017 г., при наличии стольких телефонов Android с экраном во всю ширь, в том числе и более крупного Google Pixel 2 XL. Понравился нам его постоянно включенный дисплей, отображающий значки даты, времени и уведомления, а также функция пробуждения экрана двойным касанием.

На Google Pixel 2 много людей впервые испытают Android Oreo. Несколько замечательных новых функций: мазок по заднему датчику отпечатков пальцев теперь вызывает уведомление. Картинка в картинке — повсеместно, а маленький плавающий блок работает не только с видеороликами YouTube, которые продолжают воспроизводиться, когда вы гуляете по всему остальному телефону.



» Пресные цветовые схемы с откровенно унылыми названиями тоже не радуют.

Устаревший чипсет

Pixel 2 не сумел поставить новый чипсет Snapdragon и работает на модели восьмимесячной давности, используемой в Galaxy S8 и LG G6. Он также подешевил с 4 ГБ памяти — само по себе неплохо, но большинство флагманов работают уже на 6 ГБ. Такая же история с хранилищем на всего 64 ГБ у базовой модели, причем нет слота для SD-карт. К его чести то, что результат 6260 в тестах *Geekbench* ставит Pixel 2 по скорости в один ряд со всеми существующими телефонами Android. Пережидает его только Apple iPhone 8+, но это не в счет.

Основная камера Google на 12,2 МР с диафрагмой f/1,8 и 8-МР селфи-камерой с диафрагмой f/2,4 выигрывает конкуренцию практически всесторонне. Как у фронтальной, так и у тыльной камеры есть режим размывания фона портрета, 240 кадров в секунду замедленного видео и настройки, которые проникают глубже, чем у любой камеры Google до сегодняшнего дня.

Однако ПО камеры, которое установлено по умолчанию, не идеально. И на Samsung, и на LG легко сделать замедленную съемку с помощью удобного жеста рукой, тогда как Google заставляет стучать по кнопкам. В остальном, качество фотографии является лучшим в своем классе, даже без второй

телефотокамеры, а качество видео устойчиво благодаря электронной стабилизации изображения.

Срок службы батареи разве что приемлемый. При среднем дисплее 1080р и энергосберегающих трюках под капотом Oreo это огорчает. Запуск зацикленного HD-видео с полной яркостью при полном заряде привел к тому, что осталось 86%; это вполне себе средне, так что резюме нового флагманского продукта Google разочаровывающее. **IXF**

LINUX FORMAT Вердикт

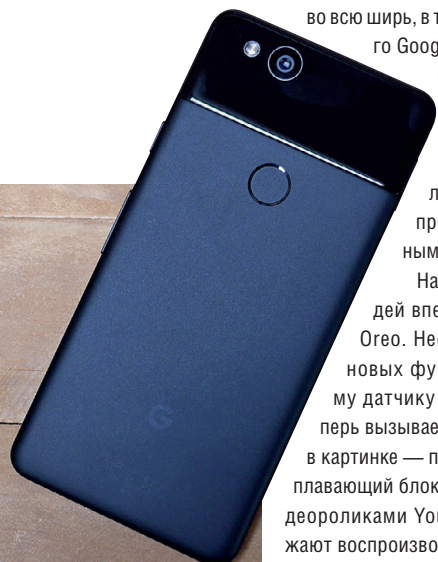
Google Pixel 2

Разработчик: Google
Сайт: <http://bit.ly/pixel-2>
Цена: \$ 629

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

» Лучшие в своем классе фотографии, двойные фронтальные динамики и водонепроницаемость не компенсируют малую производительность.

Рейтинг 7/10



Tacoma

Брошенный на расследование после ужасного и, без сомнения, щекотливого инцидента, **Энди Келли** реконструирует события, произошедшие в космическом подразделении *Linux Format*.

Спецификации

МИНИМАЛЬНАЯ

- » ОС 64-битная
- » Процессор 1,6 ГГц Intel Core i5 level
- » ОП 4 ГБ
- » GPU 1 ГБ VRAM
- » HDD 11 ГБ

РЕКОМЕНДУЕТСЯ

- » ОС Ubuntu 16.04 64-битная
- » Процессор 2,9 ГГц Intel Core i7 level
- » ОЗУ 8 ГБ
- » GPU Выделенная карта, 2 ГБ VRAM
- » HDD 11 ГБ

Пустая и заброшенная межпланетная станция Тасома дрейфует в космосе где-то между Землей и Луной. Некий инцидент заставил экипаж эвакуироваться, и вас, Эми Феррьер [Amy Ferrier], отправили для расследования и восстановления работы вездесущего ИИ, Одина [Odin]. И хотя тех шестерых человек, которые управляли станцией, здесь давно нет, их присутствие ощущается повсюду.

В распоряжении Venturis, корпорации, владеющей Тасома, есть 3D-записи каждой секунды года жизни, проведенной экипажем на борту, и вам выдан доступ к ним. Эти призрачные воспоминания помогут вам узнать о тех, кто жил и работал там до аварии. Их загадка захватывает и не отпускает вас с той минуты, как ваши магнитные ботинки коснулись станции. Когда вы первый раз в невесомости вливаете внутрь Тасома, двери отсеков поднимаются перед вами одна за другой, открывая длинный вращающийся коридор. Это и наглядное знакомство с обстановкой станции, и введение в особый мир данной игры.

Жуткую тишину нарушает только робот-уборщик, который иногда, гудя, пролетает мимо вашей головы. Вас окружают разноцветные двери, ведущие к разным частям станции, но пока вы не выполните какую-либо задачу, большинство из них недоступны. Игра направляет вас из одного места в другое, разворачивая перед вами свой линейный сюжет... такой линейный, что в нем бывает тесновато.

Первое крыло станции, куда вы можете попасть — это Персонал. Опустившись на пол под тягой искусственной гравитации, вы идете в пищеблок, где начинают



» Корпорация располагает записями о каждой секунде года жизни экипажа из шести человек на борту, и к ним вам предоставлен неограниченный доступ.

оживать очертания записей дополненной реальности [AR]. На игровой панели появляется временная шкала, позволяющая продираться через воспоминания, приостановить, перемотать или ускорить воспроизведение. Экипаж, показанный в виде цифровых силуэтов, готовится к празднику. Хотя с виду это безликие призрачные цифровые силуэты, их характеры очень выпуклы. В этом заслуга выразительной анимации игры и превосходной озвучки: вместе они создают персонажи, полные жизни.

Внимание к деталям

Другим великим талантом Fullbright является создание миров. Деталю в *Tacoma* уделяется необычайное внимание, что мотивирует вас тщательнее осматривать станцию. Чувствуешь, что каждый объект здесь не случаен и дорисовывает картину повседневной жизни экипажа. Как и в *Gone Home*, взаимодействие с окружающей средой ощущается очень реально. Можно выдвигать ящики и смотреть, что там; поднимать предметы и рассматривать их; отодвигать вещи в сторону, чтобы заглянуть под них.

Первый час проходит спокойно, давая вам шансы познакомиться с экипажем. Но затем внезапное происшествие превращает ситуацию на *Tacoma* в аварийную, и остаток игры посвящен тому, как обитатели станции пытаются с ней справиться. Атмосфера становится напряженной, опасной и нередко трагической. Но переход от наблюдения за тем, как люди делают торты и играют в бильярд, до борьбы

за свою жизнь, довольно резкий. История становится содержательнее, когда вы погружаетесь глубже в станцию. Появляется серьезный круг вопросов, касающийся взаимоотношений прав разумного ИИ и корпоративных и человеческих интересов. Это мрачная, с подтекстом, научная фантастика посвящена опасностям и этике передовых технологий и колонизации космоса.

Ставки растут, и вы начинаете искренне привязываться к этим шести уникальным людям. В *Tacoma* много еле уловимого, ожидающего своего открытия. В плане восстановления картины событий игра очень хороша и на Linux идет безупречно, но рекомендуется приличный графический процессор даже для умеренной скорости и скромного разрешения 720 пикселей. **LXF**



» Нас буквально поразило правдоподобие диалогов. Они очень естественны, не натянуты и не чрезмерно описательны. Мы хотим продолжения!

LINUX FORMAT Вердикт

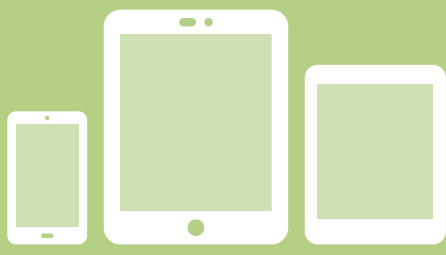
Tacoma

Разработчик: Fullbright
Сайт: www.tacoma.game
Цена: £ 14,99

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

» Умный, содержательный, но короткий квест в стиле научпоп, с невероятно продуманными, реалистичными персонажами.

Рейтинг 8/10



Мобильные НОВОСТИ

СЕТИ 5G

Рапортуют о достижениях

Intel представила портфолио коммерческих модемов стандарта «нового радио» для сетей 5-го поколения.

Intel анонсировала первое семейство коммерческих многорежимных модемов стандарта «нового радио» (NR) для сетей 5-го поколения (5G NR) Intel XMMTM 8000 series, запуск новейшего модема для сетей LTE — Intel XMMTM 7660, объявила об успешном выполнении задач, полностью осуществленных в сети 5-го поколения с использованием модема Intel 5G Modem, и продемонстрировала гигабитные скорости передачи данных на модеме Intel XMMTM 7560, представленном еще в июне на Всемирном мобильном конгрессе 2017.

Среди новинок Intel в портфолио для беспроводных сетей:

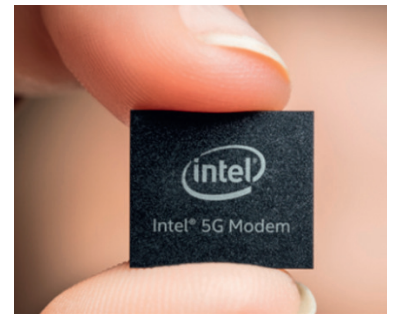
» модемы Intel XMM 8000 series — коммерческие многорежимные модемы для сетей пятого поколения, работающие как

в глобальном суб-6 ГГц диапазоне, так и на мм-волнах, которые обеспечат подключение к 5G сетям с самых различных устройств;

» Intel XMMTM 8060 — первый коммерческий модем Intel для сетей 5G, поддерживающий работу в мульти-режиме для полноценного 5G new radio как в варианте non-standalone NR, так и в standalone NR, а также в разнообразных 2G-, 3G- (включая CDMA) и 4G-сетях. Intel XMM 8060 появится в коммерческих устройствах заказчиков в середине 2019 г., что позволит ускорить внедрение устройств 5G-ready до предполагаемого повсеместного развертывания сетей 5G в 2020 г.;

» Intel XMM 7660 — новейший LTE модем, поддерживающий спецификации Cat-19

» Значительные достижения Intel в реализации планов по выпуску беспроводных продуктов ведут к ускоренному развитию сетей 5-го поколения.



и работающий со скоростями до 1,6 Гбит/с; обеспечивает расширенную поддержку технологии MIMO и агрегацию несущих частот и работает в широком спектре диапазонов; в коммерческих устройствах появится в 2019 г.

КОНКУРЕНЦИЯ

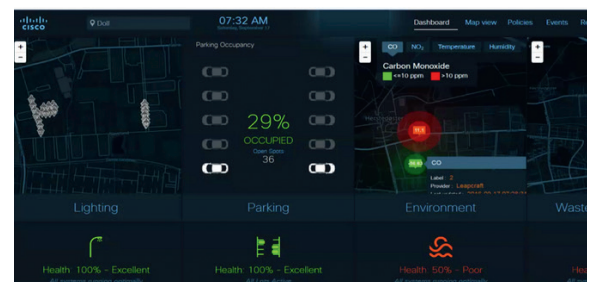
Все пути ведут в... «умный» город

Cisco и Huawei сражаются за «мозг» города будущего.

На Smart City Expo World Congress в Барселоне компания Cisco представила дополнительную функциональность для своей подключенной цифровой платформы, недавно получившей новое имя Cisco Kinetic for Cities, интегрированной с платформой обработки IoT-данных Cisco Kinetic. Новые функции предназначены для повышения общественной безопасности; все они — от новых опций автоматизации с полной поддержкой политик до модернизированного центра управления с интеграцией видео — прежде всего направлены на обеспечение безопасности жителей. Теперь для ускорения вызова аварийно-спасательных служб средствами оперативного оповещения о чрезвычайных ситуациях используются добавочные

источники IoT-данных и решение Cisco Spark Collaboration. Обращение Cisco к решениям для smart-городов неудивительно: компания десятилетиями специализируется на различных вариантах сетевых протоколов, а IoT и «умный» город, по сути, являются большой дисперсной сетью.

Теперь и компания Huawei, основной соперник Cisco на развивающихся рынках и в Китае, выпустила свой Intelligent Operations Center (IOC). Huawei IOC дополняет предложения компании для проводных и беспроводных широкополосных сетей, платформу IoT и интеллектуальную ОС компании LiteOS, которые образуют «центральную нервную систему, передающую собранную с датчиков и камер информацию в мозг — IOC».



» Так выглядит модернизированный центр управления платформы Cisco Kinetic.

Подходы Cisco и Huawei к управлению «умным» городом весьма схожи; с интересом будем следить за поединком двух компаний, предлагающим свой вариант «мозгового центра» для smart-сити.

ПЕРСПЕКТИВЫ

Huawei демонстрирует интеллект

Смартфон превращается в... «интеллектуальный телефон»!

Внедрение в смартфоны искусственного интеллекта «кардинально изменит» телефоны, поскольку вызовет полную смену технологий, уверен глава Consumer Software Engineering и директор Intelligence Engineering Феликс Чжан [Felix Zhang]. Появление ИИ он сравнил с изобретением парового двигателя — с точки зрения способности новых технологий коренным образом изменять жизнь людей. «Если рассматривать экосистему в целом, ИИ вызовет принципиальную трансформацию смартфона в интеллектуальный телефон», заявил Чжан в ходе Huawei Asia-Pacific Innovation Day 2017 [Дня инноваций Азии и Тихоокеанского региона], состоявшегося 9 и 10 ноября в Куала-Лумпуре.

Huawei характеризует представленный в сентябре чипсет с ИИ Kirin 970 как «будущее смартфонов»; концепцию мобиль-

ного ИИ компания видит в комбинации устройства с ИИ и облачного ИИ. «Huawei стремится развить устройства smart в устройства intelligent, создавая комплексные инструменты, поддерживающие разработку чипов, устройств и облаков», говорит CEO Consumer Business Group Ричард Ю [Richard Yu].

Новые технологии коренным образом изменят жизнь людей.

Анонсированные в октябре новые флагманские смартфоны Mate 10 и Mate 10 Pro поставляются с процессором Kirin, имеющим специализированный блок нейронной обработки, способный обрабатывать 2 тыс.

изображений в минуту с помощью технологий распознавания изображений.

Чжан также назвал 4 основные проблемы, с которыми пришлось столкнуться Huawei при внедрении ИИ на устройстве: восприятие, осознание, безопасность и энергопотребление. ОС телефона EMUI 8.0 осуществляет «исследование ИИ» по всей экосистеме, выполняя интеллектуальный прогноз поведения пользователя, интеллектуальное распределение ресурсов и интеллектуальное восприятие контекста с использованием машинного обучения. Системы масштабирования и распознавания объектов ИИ гарантируют, что камера будет знать, на чем следует сфокусироваться при съемке, а функция ускоренного перевода выполняет перевод с более чем 50 языков онлайн и оффлайн в 3 раза быстрее предыдущей технологии перевода.

ДОМАШНИЙ WI-FI

Расширяя «узкие места»

Mesh-сети — будущее «умных» домов.

Первый вице-президент Qualcomm Рауль Патель [Rahul Patel] полагает, что mesh-сети имеют решающее значение для будущего домашнего Wi-Fi; домашнюю широкополосную связь ждут серьезные изменения, ключевым элементом которых станут именно mesh-сети. Согласно прогнозу Gartner, до конца 2017 г. в мире будет насчитываться 8,4 млрд. подключенных устройств — на 31% больше, чем в 2016 г. Отчет GMSA “The Impact of the Internet of Things: The Connected Home” предполагает, что в каждом «подключенном» доме к 2020 г. будут использоваться в среднем 50 «умных вещей»; возникает необходимость в инфраструктуре, способной поддерживать различные потоки данных от нескольких устройств. Входящем году Qualcomm провела опрос среди пользователей домашних Wi-Fi-роутеров; 50% из 1,5 тыс. респондентов из UK, Франции и Германии заявили, что используют устройство в трех комнатах одновременно.

Современные домашние сети уже с трудом справляются с возникающими нагрузками, а переход на более качественное оборудование может вызвать трудности у среднего потребителя; можно заплатить за высокую скорость и надежность, но из-за дешевого оборудования и устаревшей инфраструктуры домашняя сеть уже не отвечает современным стандартам. Ухудшает ситуацию царящая в мире подключенных и IoT-устройств фрагментация.

Переход к Mesh-сетям обеспечит высокий уровень подключения и позволит автоматически исправлять возникающие проблемы, централизовать управление данными и устройствами, анализировать собранные данные.

Mesh-решения Qualcomm, в том числе Wi-Fi SON, уже используются Eero, Google Wi-Fi, TP-Link, Luma и Netgear. Qualcomm создала поддерживающую спецификацию IEEE 802.11ax 12-поточную платформу, основанную на четырехъядерном CMOS-



► Рауль Патель: интернет-провайдерам необходимо отказаться от консервативных подходов и оперативнее реагировать на возникающие вызовы, пока конкурентное преимущество не перешло к облачным провайдерам.

процессоре 64-разрядной архитектуры технологии 14 нм finFET. Выявленные недостатки ранних 8-поточных продуктов устранены; 12-поточные решения могут питаться от обычных кабелей Ethernet. LXF

Сравнение

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

Рабочие столы

Большинство людей придерживаются в своем дистрибутиве рабочего стола по умолчанию. **Шашанк Шарма** пробует более подходящие альтернативы...



Про наш тест...

В нашем Сравнении мы рассматриваем проекты рабочего стола, которые пока не столь популярны. Нам нужна была нейтральная база для тестирования, и мы начали с Fedora. Кроме рабочего стола Deepin, мы смогли установить в Fedora все проекты из нашего списка. Поскольку мы запустили DDE на Ubuntu, мы решили, что лучше всего будет установить и остальные рабочие столы, а также сравнить их на обоих дистрибутивах. До последнего релиза Ubuntu всю свою мощь вкладывал в Unity, но сейчас снова вернулся к Gnome как к рабочему столу по умолчанию, а также предлагает редакцию с *Budgie*. Там, где это возможно, мы также пробовали эти PC в их родной среде.

Мы ищем PC, стабильный и простой в работе. Бонусные баллы начисляются, если проект не очень прожорлив в плане ресурсов и дает возможность тонкой индивидуальной настройки рабочего стола.

Рабочий стол (PC) — важная точка контакта между пользователем и его дистрибутивом Linux. PC — это набор всех графических элементов, которые вы видите на рабочем столе, как то: окон, панелей инструментов и значков — а также Менеджер окон, отвечающий за внешний вид окон в GUI.

Все популярные настольные дистрибутивы предлагают одну из наиболее надежных сред рабочего стола, например, Gnome, KDE или Mate. Рабочий стол — настолько неотъемлемая и важная часть любого дистрибутива Linux, что некоторые

дистрибутивы часто определяются своим выбором рабочего стола (PC). Fedora, например, считается дистрибутивом Gnome, несмотря на редакции с другими PC, например, KDE. Но при всей важности роли PC в нашем повседневном использовании компьютеров, многие пользователи редко выходят за рамки PC по умолчанию своего дистрибутива.

И это не очень правильно, поскольку многие PC чудесны в работе и невероятно надежны, а состоявшиеся проекты, такие как Gnome и KDE, считаются громоздкими и медленными. Потребность в быстром рабочем столе-легковесе, приглядном и приятном в работе, породила новое поколение — в частности, те PC, которые составили наш выбор.

Потребность в быстром рабочем столе-легковесе, приятном в работе, породила новое поколение.

Наша подборка

- » Budgie
- » Deepin
- » Enlightenment
- » LXQt
- » Pantheon

Установка и наличие поддержки

Легко ли обзавестись новым рабочим столом?

Несмотря на изобилие внутренних компонентов, таких как обои, значки и темы, среда рабочего стола — всё равно не более чем программа. А значит, вы можете применять для ее установки инструмент управления программами своего дистрибутива, совершенно так же, как и для любой другой программы — например, медиа-плеера либо web-браузера.

Однако установка другой доступной среды рабочего стола — только первый шаг. Далее надо будет установить менеджер отображения, который позволит выбирать, какой PC загружать.

К сожалению, некоторые менеджеры отображения — например, *GDM*, менеджер отображения в *Gnome* — не предоставляют возможности такого выбора. *LightDM* — популярная альтернатива *GDM*, и он доступен в программных репозиториях практически всех популярных дистрибутивов Linux. Если вы работаете в *Ubuntu*, как и мы, вам не придется устанавливать *LightDM*, поскольку в этом дистрибутиве *LightDM* — менеджер отображения по умолчанию.

Поскольку в нашем Сравнении рассматриваются альтернативные PC, они, неизбежно, не все будут официально поддерживаться популярными дистрибутивами. Фактически, за исключением *LXQt* и *Enlightenment*, вам затруднительно будет найти остальные три в программных репозиториях своего дистрибутива. Но не отчаивайтесь!

Budgie является внутренней разработкой для дистрибутива *Solus OS* и предложением по умолчанию для этого проекта. Начиная с последнего релиза 17.04, есть *Budgie*-редакция *Ubuntu*, и *Manjaro Linux* тоже поддерживает *Budgie*. Если у вас *Fedora*, вы можете установить *Budgie* с помощью созданной сообществом сборки *Corp*, но будьте готовы к некоторым проблемам, например, невозможности загрузить обои.

Elementary OS, дистрибутив на базе *Ubuntu*, предназначенный специально для пользователей *Windows* и *MacOS*, предлагает собственный рабочий стол *Pantheon*. К сожалению, в настоящий момент *Pantheon* официально не поддерживается больше никаким дистрибутивом. Однако вы можете



» С релиза 17.04 *Ubuntu* предлагает редакцию *Budgie*, и он доступен также для *Manjaro Linux*.

установить его на *Fedora* с помощью *Corp* или в *Ubuntu* через PPA-репозиторий.

Deepin Desktop Environment, или *DDE* — один из самых молодых рабочих столов, это среда по умолчанию в *Deepin Linux*. Как и *Pantheon*, *DDE* пока что не поддерживается официально другими дистрибутивами, но можно использовать репозиторий PPA для установки проекта в *Ubuntu*. Хотя сборка *Corp* доступна, на *Fedora* во время наших тестов установить *DDE* мы не смогли.

В традиционном понимании менеджер окон — всего лишь компонент рабочего стола, однако он предлагает достаточно функций для самостоятельного применения в качестве PC. В сочетании с *Enlightenment Foundation Libraries (EFL)*, набором графических библиотек, *Enlightenment* считается рабочим столом, и поэтому достоин занять место в нашем списке.

Вердикт

LXQt
★★★★★
Enlightenment
★★★★★
Budgie
★★★★★
Deepin
★★★★★
Pantheon
★★★★★

» Два старейших проекта проекта доступны также почти для всех дистрибутивов Linux.

Поддержка приложений

Будут ли ваши любимые приложения работать с этими PC?

За исключением *LXQt* и *Enlightenment*, все наши PC спокойно запускают приложения, разработанные для других рабочих столов. Кроме того, известно, что среды рабочего стола также предлагают собственные приложения

и утилиты. Некоторые проекты из нашего списка предлагают достойный выбор приложений. Хорошим примером будет *Deepin Desktop Environment*: он предлагает собственный просмотрщик изображений, программу записи аудио и видео,

приложение терминала и менеджер файлов. *Enlightenment* тоже предлагает менеджер файлов и терминал, но это и всё. Однако есть несколько основанных на *EFL* приложений, таких как видеоплеер *Rage*, торрент-клиент *Eporur* и браузер *Elbow*. Превращение их и многих других в составную часть *Enlightenment* — всего лишь вопрос времени.

LXQt появился в результате слияния *LXDE-qt* и *Razor-qt*. И как таковой, он прекрасно справляется с рендерингом приложений *Qt*. *Budgie* тоже создан на *Qt* и отлично работает с этими приложениями. Но вы можете использовать все свои приложения в любом из этих PC.

Фактически, все проекты из нашего списка довольно неплохо справляются с использованием пакета приложений, установленного на вашем дистрибутиве. Во всяком случае, эти PC понимают искусство хорошей интеграции, и у нас не случилось проблем с запуском разнообразных приложений.



» *Deepin Desktop Environment* предлагает хороший выбор готовых к использованию приложений.

Вердикт

Budgie
★★★★★
Deepin
★★★★★
Enlightenment
★★★★★
LXQt
★★★★★
Pantheon
★★★★★

» И *Pantheon*, и *Budgie* предлагают широкий выбор приложений по умолчанию.

Взаимодействие с пользователем

С каким рабочим столом приятнее всего работать?

Хотя такие инструменты, как системный монитор *Conky*, заботятся о тех, кто жаждет добавить на свой рабочий стол всякие примочки, большинство рабочих столов Linux традиционно имели незамысловатый внешний вид. Но несколько лет назад это изменилось. Linux теперь не только поддерживает сенсорные устройства, но разработчики активно трудятся

над обеспечением своих пользователей максимально привлекательным внешним видом.

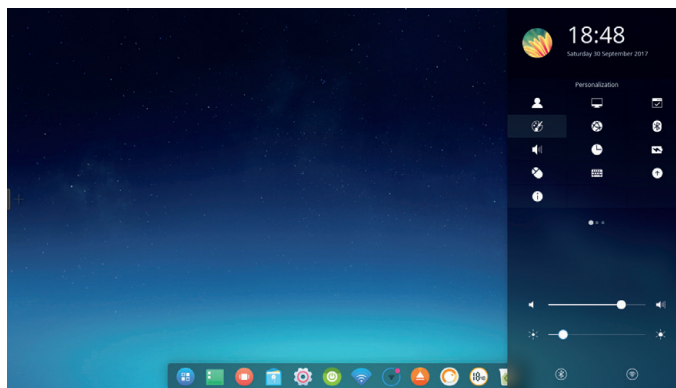
Когда разработчики таких именитых проектов, как Unity и Gnome, приняли сознательное решение изменить свой стиль работы, чтобы он больше соответствовал сенсорным устройствам, это разозлило многих их постоянных пользователей. Это, а также постоянно растущие требования

к системным ресурсам для поддержки всяческих красотостей подсказали некоторым пользователям идею поискать альтернативу.

Все проекты в нашем списке высокофункциональные и современные, а также высоко ценятся за свою способность к настройке. Они умудряются делать всё, что полагается, и при этом не пожирать системные ресурсы.

Budgie ★★★★★

Budgie написан с нуля для дистрибутива Solus OS, и прошел долгий путь. Окно приглашения сейчас имеется во многих PC, но окно приглашения *Budgie* элегантно и простое по дизайну. Нажмите на кнопку Recommendations [Рекомендации], чтобы просмотреть список рекомендуемых программ. Помните, что этот список зависит от используемой ОС и будет содержать разные инструменты в зависимости от того, работаете вы в Ubuntu или Solus. В противоположность традиционной метафоре рабочего стола, *Budgie* не поддерживает по умолчанию создание значков на рабочем столе. Вы должны вручную активировать эту опцию на боковой панели Raven. Хотя она не предлагает столько виджетов, сколько в устоявшихся PC вроде Gnome, проект постоянно совершенствуется, и разработчики постоянно предлагают разумное и интересное взаимодействие, уделяя большое внимание улучшению Raven.



Deepin ★★★★★

Изначально разработанный для Deepin Linux, это надежный рабочий стол. *DDE* хорошо работает в Deepin Linux и Ubuntu, но нам не удалось установить его в Fedora, несмотря на то, что мы пробовали разные сборки Comp. Жаль, что этот проект пока что не используется другими дистрибутивами Linux, потому что он прекрасен. Кроме того, в нем множество встроенных приложений, которые дополняют его дизайн и внешний вид. В отличие от Raven в *Budgie*, который иногда кажется несколько загроможденным, Центр Управления [Control Center] просторный и хорошо продуманный. Вы можете использовать его, чтобы настроить мышь и трекпад, горячие клавиши, изменить приложения по умолчанию и т.д.; настраиваемые горячие углы позволяют получить доступ к меню приложений и панели управления. Хотя *DDE* загружается немного медленнее, чем другие проекты в нашем списке, он невероятно отзывчив.

Пользовательская документация

Ни один из этих рабочих столов не предлагает особой поддержки.

Arch Linux Wiki (<https://wiki.archlinux.org>) — один из лучших информационных ресурсов почти для всех проектов Linux, и рабочие столы здесь не являются исключением. Фактически, Arch Wiki содержит больше практической информации о PC из нашего списка, чем их собственные сайты!

Как это ни удивительно, но у *Pantheon*, внутренней разработки дистрибутива Elementary OS, даже сайта не имеется. Он только вскользь упомянут на сайте и в блоге родительского проекта. Официальная

документация предлагает краткое введение в такие функции, как горячие углы и рабочие области, но на этом, собственно, всё и кончается.

Помимо страницы About, которая служит введением, *Enlightenment* не может похвастаться особым объемом документации. Конечно, можно многое узнать на сайте проекта, если вы — разработчик, который жаждет поучаствовать в проекте или EFL; но если вы — пользователь, просто пытающийся разобраться в опциях настройки *Enlightenment*, то Интернет

окажется для вас куда лучшим источником информации.

Wiki *LXQt* предлагает базовую информацию по таким вопросам, как установка и темы, но этим всё и ограничивается. *Budgie* сейчас обзавелся собственным сайтом, но пока там не размещено никакого контента, кроме блога с сообщениями о новых релизах.

Хотя *Deepin Desktop Environment* упоминается в wiki Deepin, там обсуждаются только основы, например, создание файлов и папок.

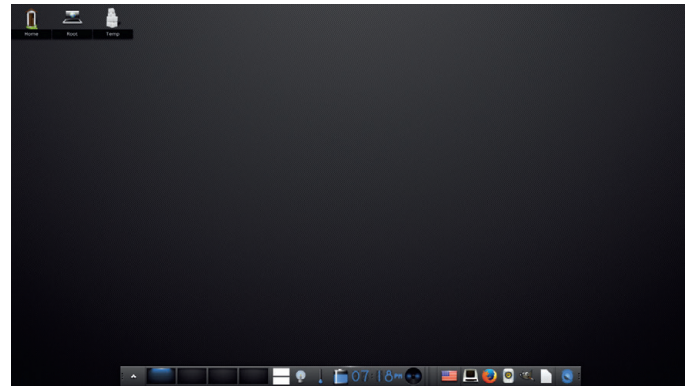
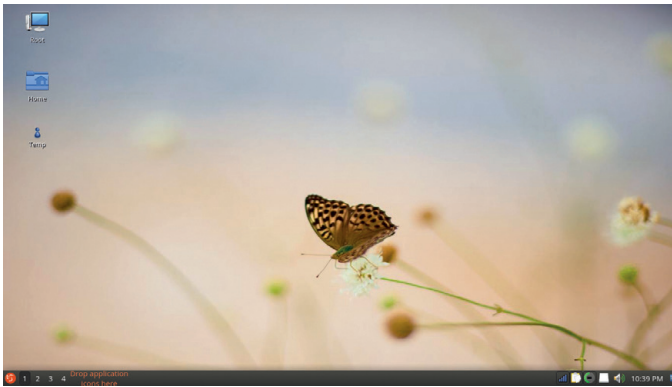
Вердикт

- Deepin ★★★★★
- Enlightenment ★★★★★
- LXQt ★★★★★
- Budgie ★★★★★
- Pantheon ★★★★★

» Для желающих узнать, как настроить свой PC, официальную помощь весьма слабая.

Enlightenment ★★★★★

Проекту *Enlightenment* уже исполнился 21 год, впервые он появился в 1996 г. Его долголительство — свидетельство популярности у пользователей. Проект идеален для маломощных компьютеров, поскольку ему не требуется большой объем ОЗУ, да и просто предлагает привлекательное взаимодействие с конечным пользователем. *Enlightenment* ничуть не проигрывает по сравнению с остальными, если вы уделите время, чтобы поближе познакомиться с его хитроумным дизайном. При первой загрузке вам зададут целую серию вопросов, чтобы вы могли настроить среду на свой вкус. Это, например, касается языка по умолчанию и размера верхней панели инструментов окон. Вы также можете настроить панель оболочки внизу рабочего стола — но опять же, всё это потребует от вас терпения и желания пробовать разные опции и узнавать, что делает каждая из них.

**LXQt** ★★★★★

LXQt доступен во многих дистрибутивах Linux, однако его производительность можно в лучшем случае описать, как капризную. Его взбалмошное поведение не позволяет с чистой совестью порекомендовать этот проект, особенно начинающим пользователям, которых его странности сильно огорчат. Минималистский, хотя и вполне функциональный рабочий стол предлагает несколько опций настройки. На панели внизу нет никаких приложений, что составляет резкий контраст с панелями других ОС. Вместо этого *LXQt* предлагает место-заполнитель в левой части панели, куда вы сможете перетащить свои любимые приложения. В меню содержится список приложений *Categories* [Категории]. Все настраиваемые элементы *LXQt* вы обнаружите в отдельном списке, расположенном внутри меню *Preferences > LXQt settings* [Предпочтения > Настройки LXQt].

Pantheon ★★★★★

Pantheon — еще один минималистский, но и стильный рабочий стол. Его интуитивный дизайн способствует приятному взаимодействию с пользователем. Хотя он не может похвастаться готовыми опциями настройки, это легко исправить, установив инструмент *Elementary Tweaks*. Рабочий стол предлагает панель вверху с программой запуска приложений в левом углу. В отличие от большинства других ОС, *Pantheon* не позволяет настраивать верхнюю панель. По умолчанию программа запуска не предлагает списка приложений по категориям, и просто выводит все приложения в списке, не особо отличаясь от Gnome. Желая, чтобы приложения группировались по типам, нажмите на кнопку *View applications by categories* [Просмотреть приложения по категориям]. Чтобы не прокручивать меню в поисках нужного приложения, просто напечатайте его название, и программа запуска автоматически отфильтрует результаты.



Производительность

Скорость захватывает, а ее отсутствие огорчает.

Внешняя привлекательность — важный критерий при выборе рабочего стола. В этой области все проекты в нашем списке показали себя прекрасно, предлагая анимацию и прочие эффекты вам на радость — но все эти предложения осуществляются за счет системных ресурсов. Громоздкие ОС вроде Gnome, KDE и Unity требуют ускорения графики и огромного объема ОЗУ; зато проекты в нашем списке не столь требовательны.

Enlightenment и *LXQt* совершенно спокойно функционируют на компьютерах,

находящихся на низшем конце спектра. Оба ОС загружаются быстро — но если *Enlightenment* работает, как обещано, то *LXQt* на Ubuntu и Fedora частенько сообщает об ошибках и склонен к периодическим зависаниям.

Текущий релиз *Budgie* тоже не особенно прожорлив в плане ресурсов, но в будущих релизах это должно измениться. Его разработчики уже сообщили о своих планах расширить набор функций *Budgie*, который может оказаться чрезмерным для компьютеров с 2 ГБ ОЗУ.

Pantheon демонстрирует наибольшую скорость в своем родном дистрибутиве. А вот на Fedora или Ubuntu начинает немного тормозить.

Deepin Desktop Environment, безусловно, быстр и отзывчив как таковой, но он самый медленный из нашего списка. Для оптимальной производительности ему требуется двудерный компьютер с более чем 2 ГБ ОЗУ.

LXQt загружается быстро, но малонадежное поведение отбрасывает его на последнее место в нашем тесте.

Вердикт

Enlightenment
★★★★★
Budgie
★★★★★
Deepin
★★★★★
Pantheon
★★★★★
LXQt
★★★☆☆

» Большинство этих ОС надежны и резко отключаются даже на самых дешевых ПК.

Расширения

Добавочные функции... кто-нибудь, ну хоть кто-нибудь?

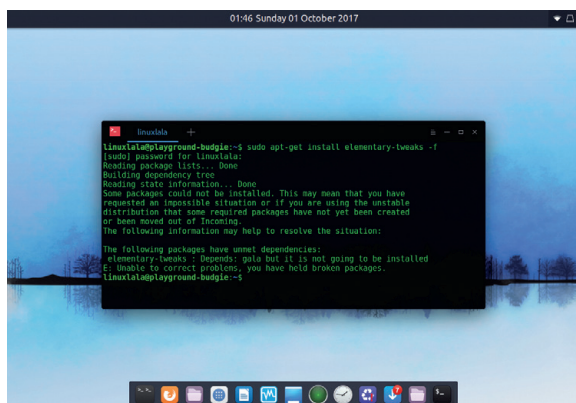
Budgie, разработанный проектом Solus OS, не поддерживает расширения. То же можно сказать про LXQt и DDE, разработанном для проекта Deepin Linux. Однако недавние разработки и объявления из лагеря Budgie могут привести к заключению, что проект стремится выйти за свои минималистские рамки и легковесные характеристики. Вероятно, что проект станет использовать расширения

в будущих релизах по мере того, как обретет больше пользователей и станет доступным для других дистрибутивов.

Подобно Budgie и DDE, Pantheon был разработан для конкретного дистрибутива Linux, в данном случае — для Elementary OS. Однако, в отличие от Budgie и DDE, Pantheon легко расширяется с помощью инструмента Elementary Tweaks, доступно в репозитории PPA. Движимые любовью

к простоте и элегантности Elementary OS, некоторые пользователи из сообщества разработали сторонние декоративные элементы, например, темы и значки, чтобы дополнить свою установку Elementary OS. После установки можно использовать инструмент Elementary Tweaks для изменения внешнего вида рабочего стола, настройки шрифтов, управления анимацией и т.д. В репозитории PPA размещается также несколько созданных пользователями тем. Инструмент Elementary Tweaks даже позволяет определять количество строк и столбцов для отображения в программе запуска приложений, и можно также настроить размер значков, темы, положение панели дока на экране. Но как бы чудесно всё это ни было, Elementary Tweaks не устанавливается на дистрибутивы без прямой поддержки Pantheon.

Немало сторонних расширений для Enlightenment доступно на сайте проекта opendesktop.org. Помимо темы фонов панели, имеется также анимированный фон, который реагирует на наведение мыши.



» К сожалению, Elementary Tweaks недоступен для Fedora и других рабочих столов, и его даже нельзя установить на Ubuntu 17.04.

Вердикт

- Enlightenment ★★★★★
- Pantheon ★★★★★
- Budgie ★★★★★
- Deepin ★★★★★
- LXQt ★★★★★

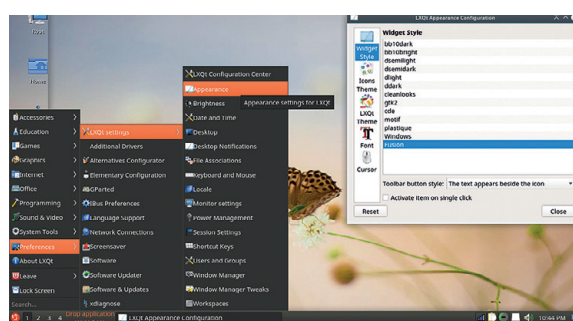
» Пусть результаты этого теста не влияют на ваше мнение о любом из этих проектов.

Настраиваемость

Когда вы в настроении что-то поднастроить.

Все проекты в нашем списке поддерживают основные опции настройки и позволяют сменить обои, настроить темы окон, определить шрифты, значки и курсоры.

Чтобы настроить все вышеперечисленные элементы в Pantheon, используйте инструмент Customize Look and Feel. Точно так же можно использовать в LXQt инструмент Appearance Configuration Tool в LXQt для смены курсора, тем окон и т.д. Желая настроить Panel, щелкните правой кнопкой и нажмите Configure Panel. Отсюда можно настроить положение панели и ее размер, и даже задать цвет фона. Настраиваемыми являются даже некоторые виджеты на панели. Нажмите на вкладку Widgets на левой боковой панели. Затем выберите виджет и нажмите на кнопку Configure, чтобы поиграть с различными имеющимися опциями. Можно также настроить Панель так, чтобы она постоянно отображалась или автоматически скрывалась. LXQt использует в качестве менеджера окон Openbox, который предлагает собственный инструмент Openbox Configuration. Вы можете



» Все среды рабочего стола в нашем списке поддерживают базовую настройку.

использовать этот инструмент, если вам хочется изменить на своем рабочем столе LXQt тему окон.

Доступ к опциям настройки Deepin можно получить из Центра Управления [Control Center]. Несмотря на свое название, Центр Управления в Deepin Desktop Environment не является окном, как в других дистрибутивах Linux. Это боковая панель, откуда вы можете управлять практически любым аспектом Deepin Environment. Переместите мышь в правый нижний угол экрана и щелкните. Активируется горячий угол и откроется Центр Управления. Теперь

нажмите на кнопку Personalisation для доступа к настраиваемым опциям.

Как и Deepin DE, Budgie предлагает опции настройки из панели. Вы можете получить доступ к этим настройкам из боковой панели Raven. Нажмите на квадратную кнопку в правой верхней части панели вверху рабочего стола, чтобы получить доступ к боковой панели Raven. Budgie также позволяет создавать дополнительные панели, вроде той, которая внизу рабочего стола, прямо из Raven. Затем можно добавить на эту новую панель виджеты и настроить их.

Вердикт

- Budgie ★★★★★
- Deepin ★★★★★
- Enlightenment ★★★★★
- LXQt ★★★★★
- Pantheon ★★★★★

» Пускай Enlightenment и не кон, но предлагает достойный и приятный внешний вид.

Они не просто попутчики

Вердикт

Для нашего *Сравнения* мы протестировали пару старых, независимых от дистрибутива рабочих столов и три новых, каждый из которых разработан под определенный дистрибутив. И даже нас удивил конечный результат. По логике, независимые от дистрибутива PC типа *LXQt* и *Enlightenment* должны были превзойти новые разновидности, но этого не случилось. Частично причиной тому внутренний код и унаследованные от него сложности.

Enlightenment существует более двух десятков лет; хотя он постоянно развивался, его надо практически полностью переписать, если он намеревается конкурировать с новичками.

Несмотря на мощное наследие, *LXQt* так и не удалось найти свою нишу. Если только этот проект не будет использоваться каким-то крупным дистрибутивом или не получит поддержку преданной группы разработчиков, вряд ли он протянет долго. Хотя он предлагается в программных репозиториях многих дистрибутивов,

реализован он слабовато и выдает ошибки на Fedora и Ubuntu.

Нам было в общем непросто не присудить *Enlightenment* призового места. Это в основном независимый проект, поддерживаемый только группой преданных разработчиков и сильным сообществом пользователей. В самом по себе *Enlightenment* нет особых проблем, но он проигрывает простоте и философии дизайна других PC.

Мы уже упоминали, что *Pantheon* лучше всего работает в своем родном дистрибутиве Elementary OS. Тем не менее, небольшая потеря производительности — не повод не попробовать его на других дистрибутивах. *Budgie* обгоняет *Pantheon* благодаря своим опциям настройки и инструменту боковой панели Raven.



Несмотря на отсутствие официальной поддержки со стороны основных дистрибутивов, победителем стал *DDE* благодаря своей боковой панели Центра управления [Control Center]. Все занявшие призовые места лучше всего соответствуют своим родительским дистрибутивам благодаря внутренним приложениям, дополняющим их. Хотя приложения Solus не предлагаются в качестве части *Budgie* на других дистрибутивах, *Deepin DE* предлагает ряд таких приложений, и это еще одна причина, по которой он выигрывает у *Budgie* и *Pantheon*.

» Можно смело сказать, что *DDE* и *Budgie* являются причиной роста популярности *Deepin Linux* и *Solus OS*.

По логике, независимые от дистрибутива рабочие столы должны были превзойти новые разновидности.

I Deepin DE ★★★★★

Версия: 15.4 Сайт: www.deepin.org Лицензия: GPLv3
» Очень многообещающий PC, особенно при своих приложениях.

IV Enlightenment ★★★★★

Версия: 0.21.9 Сайт: www.enlightenment.org Лицензия: GPL и др.
» Интересная среда, если вы найдете время изучить ее причуды.

II Budgie ★★★★★

Версия: 10.3.1 Сайт: www.budgie-desktop.org Лицензия: GPL
» Лучше всего работает на Solus.

V LXQt ★★★★★

Версия: 0.11 Сайт: www.lxqt.org Лицензия: LGPL и др.
» Идеален для маломощных компьютеров, но не лишен ошибок.

III Pantheon ★★★★★

Версия: 0.1.1 Сайт: www.elementary.io Лицензия: GPL и др.
» Легковес, но ему нужно больше опций настройки по умолчанию.

Обратная связь

Вы согласны с нашей оценкой? Мы не пропустили ваш любимый PC? Сообщите нам о своем мнении на lxformat@futurenet.com.

Рассмотрите также...

Если вам не жаль ОЗУ и ресурсов, то ничто не сравнится с такими монстрами среди рабочих столов, как Gnome и KDE. Те, кто ищет нечто легковесное, но при этом популярное, могут выбрать *Xfce* и *LXDE*. Они также официально поддерживаются популярными дистрибутивами, такими как Fedora и Ubuntu, со специальными версиями

для каждого. Иногда целого рабочего стола многовато, и многие пользователи обходятся менеджером окон. Если вы готовы рискнуть, попробуйте невероятно богатые функциями менеджеры окон *Blackbox*, *Openbox* или *IceWM*. Они не нагружают систему и предлагают отличные опции настройки, несмотря на свой минимализм. Дистрибутивы вроде

Puppy Linux предлагают только менеджер окон, давая в результате очень быструю систему, которая хорошо работает даже на старых компьютерах. Хотя Ubuntu уже отказался от Unity, его разработчики настроены продолжать работу над проектом, так что это может быть вполне достойная альтернатива, если вы готовы выделить ресурсы. **LXF**

UBUNTU 17.10: ПЕРЕНАСТРОИМ!

Последний релиз Ubuntu возвращается не только к началу алфавита, но и к рабочему столу Gnome, где и зародилась мечта...



Это особенное время наступило. Мы раскрасили обложку в оранжевый цвет Ubuntu, заполнили ее призывами к действию и одобренным маркетологами количеством восклицательных знаков. Мы делаем резервные копии наших данных, мы прощаемся с нашей версией 17.04, и рассказываем вам о новеньком Ubuntu.

Однако если взглянуть на Ubuntu за последние, скажем, пять лет, мало что изменилось. И это, надо сказать, Хорошо: это демонстрирует зрелость, стабильность и выбор в пользу постепенных изменений вместо полного разрушения и начала с нуля. Всё это хорошо, но усложняет жизнь нашим бедным авторам [Ред.: — Ох, помолчите, а как

насчет бедного редактора, который всех вас организует?], поскольку им приходится прославлять перемещение кнопки в окне направо, и прочие столь же малоэффективные вещи, как невероятно захватывающие штуки.

мы увидим, вероятно, самое радикальное изменение рабочего стола Ubuntu с момента появления Unity шесть лет назад. Возможно, оно даже сможет оправдать все эти превосходные степени и восклицательные знаки.

Вероятно, самое радикальное изменение рабочего стола Ubuntu с момента появления Unity...

Однако в апреле 2017 г. главный босс Canonical, Марк Шаттлворт [Mark Shuttleworth] шокировал мир Linux сообщением о том, что Canonical прекращает разработку Ubuntu Touch (мобильной Ubuntu OS) и что настольный Ubuntu откажется от собственной среды Unity. Итак, в этом релизе

Но не бойтесь: Canonical и команда, создавшая официальную версию Ubuntu Gnome, сделали всё возможное, чтобы Gnome стал удобнее для беглецов с Unity. И мы готовы рассказать вам, как обустроиться там и сделать Gnome, да и остальную часть Ubuntu домом родным.

Названное изменение, а именно, отказ от Unity и возврат к Gnome, может быть как хорошей, так и плохой новостью, в зависимости от ваших предпочтений рабочего стола.

Знакомьтесь: Aardvark

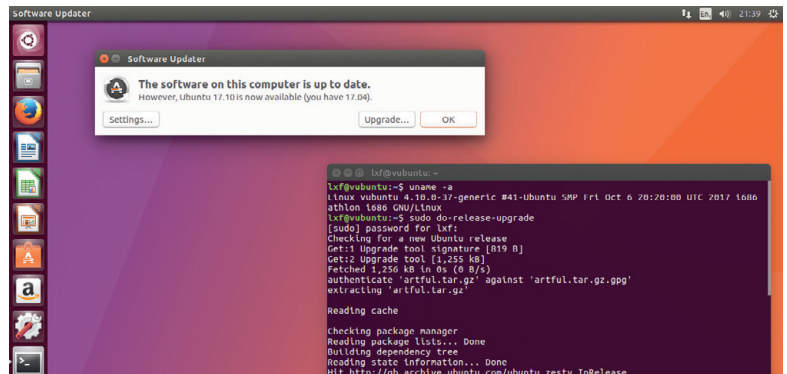
Свежий Ubuntu уже здесь, и он несет масштабные изменения на рабочем столе. Но не бойтесь — наше славное руководство обо всем расскажет...

Ubuntu 17.10, под названием Artful Aardvark [Творческий Трубказуб], уже вышел на волю, и помимо мощного нового рабочего стола Gnome 3.26 предлагает также множество новых функций и технологию **Wayland**.

Он работает на ядре 4.13, с поддержкой чипов Intel Cannonlake и Coffee Lake (см. стр. 16), основным обновлением механизма безопасности AppArmor и улучшением протокола Thunderbolt. Поддерживается также аппаратное ускорение декодирования видео с помощью VA-API, результаты чего можно увидеть, установив медиа-плеер *mpv* или прямо сразу (отметив окно Install third-party software [Установить стороннее оборудование] во время установки) с помощью *Totem*. С драйверами с открытым кодом Gnome по умолчанию использует композитор следующего поколения *Wayland*, но если у вас появятся какие-либо «странности», из окна приглашения доступна сессия *X.org*. Солидно улучшена также поддержка масштабирования HiDPI и настройки с несколькими мониторами. Обновлены приложения, улучшена поддержка печати, появилась аккуратная новая панель управления. Ах да, и кнопки управления окнами снова вернулись на правую сторону; но это легко изменить.

Безопасная работа

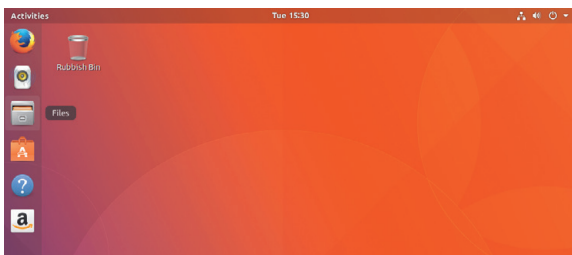
Если вы готовы рискнуть, можете выбрать чистую установку через испытанную временем программу *Ubiquity*. Или прочитать во врезке внизу, как обновиться из 17.04. Никогда не помешает сделать резервную копию данных, и всегда неплохо делать это перед обновлением ОС. Для тех, кто работает на 16.04, всё довольно непросто, но сперва подумайте: а надо ли вам обновляться? Релизы LTS



созданы для стабильности и будут поддерживаться до 2021 г., тогда как поддержка трех промежуточных релизов — только девять месяцев. Это скорее релизы разработки для тестирования новых функций и подготовки их к следующему LTS-релизу, 18.04, под названием Bionic Beaver (до него осталось всего полгода). Из-за различной природы этих каналов прямое обновление с 16.04 до 17.10 не поддерживается. Ядро 4.8 в LTS релизе весьма уложенное в плане исправления ошибок и обновлений безопасности, и в него портируются даже новые функции там, где это возможно и разумно. Если вам нужно новое ядро/видеодрайверы для лучшей поддержки оборудования, то лучшим выходом может оказаться установка пакета Hardware Enablement (HWE). Сейчас он дает вам ядро 4.10, но релиз 16.04.4, который должен выйти в феврале 2018 г., перейдет на 4.13.

Но даже без поддержки можно переключиться на версию разработки, и это можно сделать, запустив в окне команду с `-d` на конце, что обозначит желание перейти на эту версию. Однако это лишь половина пути, и обновит только до версии 17.04, а хотелось бы — до 17.10. То же самое можно сделать с помощью инструмента *Updater*, изменив настройку уведомлений во вкладке Updates с 'For LTS versions only [Только для версий LTS]' на 'For any new version [Для любой новой версии]'. Основное изменение в этом релизе — отсутствие 32-битного установщика, но пользователи старых ПК всё равно могут обновиться или выполнить чистую установку более подходящей версии, скажем, Lubuntu 17.10 или Ubuntu Mate.

➤ Если вы работаете на 17.04, Software Updater предложит вам сделать обновление (даже на 32-бит), или можно использовать командную строку.



➤ Фон рабочего стола по умолчанию показывает трубказуба, а заодно напоминает макушку пирата в бандане.

Обновление с 17.04

Обновление с 17.04 — процесс простой. Если у вас достаточно терпения, вам, вероятно, будет предложено обновление вскоре после официального релиза. Если этого не произойдет, откройте инструмент *Software & Updates*, перейдите на вкладку Updates и проверьте, чтобы выпадающее окно "Notify me of a new Ubuntu Version [Уведомлять меня о новых версиях Ubuntu]" настроено на "For any new version [Для любой новой версии]". В порядке альтернативы можете подпихнуть программу обновления, открыв терминал и введя следующее заклинание:

```
$ sudo do-release-upgrade
```

Обновление с 17.04 (или 16.04, если ваш компьютер не был включен в этот процесс) означает, что вы сохраните рабочий стол Unity и получите установленный Gnome. Наличие двух рабочих столов — это не страшно, и вы сможете выбирать на экране приглашения, какой из двух больше соответствует вашему настроению. Однако некоторые захотят полностью отказаться от Unity — или из мести, или просто ради ощущения свободы, создаваемого очищением системы. Какой бы ни была

причина, не запускайте эту команду в системе, где Unity — единственный рабочий стол, если только вы не хотите на самом деле избавиться от сервера отображения и графических приложений. Чтобы избавиться от Unity, надо всего лишь ввести

```
$ sudo apt purge unity unity-session
```

И тогда позади останутся многие ненужные зависимости. Их — вместе со всеми остальными ненужными пакетами — можно отправить в Валгаллу с помощью

```
$ sudo apt autoremove
```

Оглянемся на Unity

Прежде чем нырнуть в новый рабочий стол Gnome, на миг остановимся и вспомним прекрасный и отважный Unity и его яркую семилетнюю историю.

Когда в 2011 г. появился Unity, мир рабочих столов Linux был совершенно иным. Сейчас имеется как минимум шесть основных сред рабочего стола (и по своему Ubuntu на каждый из них). А тогда были Gnome, KDE или Xfce. В порядке альтернативы можно было употребить файл .xinitrc для запуска минималистичного менеджера окон плюс ряд вспомогательных программ для выполнения освященных этикетом сервисов рабочего стола — и, вероятно, приходилось склепать пару скриптов, чтобы они сработались. Заметим, что подобный хитроумный подход был не для слабонервных.

Есть мнение, что Linux Mint в большой мере обязан успехом массовому уходу от Unity.

Gnome тогда запустил третью итерацию своего рабочего стола, вызвавшую наполовину озадаченность, наполовину отвращение. Это был радикальный отход от нормы, отказ от традиционных меню и доков, и критики заявили, что он больше похож на мобильный интерфейс, чем на интерфейс рабочего стола. У KDE 4 были

свои сторонники, и поскольку он существует с 2008 г., то имел достаточно времени на устранение неполадок. Но он всё равно считался слишком ресурсозатратным (и считается таковым, хоть и менее заслуженно, и по сей день) и вряд ли мог привлечь много новых пользователей. За это время у Xfce появилось немало последователей, но он всегда был мелкой рыбешкой. Gnome 3 не был готов для Ubuntu 11.10, а ответвление Gnome 2 сочли неуместным (хотя именно это сделал Mate несколько лет спустя), и поэтому Unity, изначально разработанный как GUI для нетбуков, был адаптирован для настольных ПК.

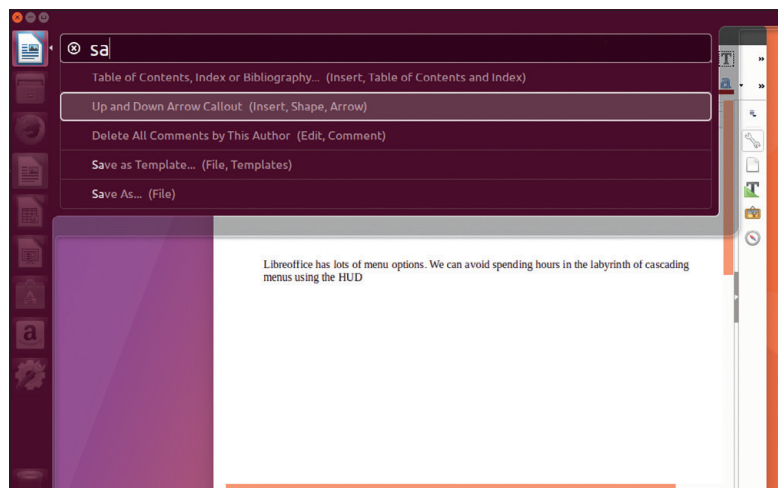
Обаяние унификации

Unity был инновационным рабочим столом, с его мощным HUD для прямого доступа к пунктам меню приложений и смелым использованием глобального меню. Конечно, у него были свои фанаты, но были и критики. Есть обоснованное мнение, что Linux Mint в большой мере обязан своим успехом массовому уходу от Unity, из-за его непривычного левостороннего размещения программы запуска, кнопок управления окон и т.д. Глобальное меню тоже вызвало недоумение пользователей, и в конечном итоге меню снова вернулось в окна приложений, хотя и несколько непривычно для их строк заглавия.

Помимо этой критики, были и другие проблемы. Установка Unity на другие дистрибутивы была крайне сложной. Частично это объяснялось необходимостью латать его под GTK, Qt и прочее, что, в свою очередь, требовало латать приложения, вызывая каскад пересборок для производителей пакетов и сильную головную боль у разработчиков, вынужденных заниматься рабочим столом, который, вполне вероятно, был им абсолютно безразличен.

Глава рабочего стола Ubuntu, Уилл Кук [Will Cooke], слова которого вы еще услышите в этой статье (спасибо, Уилл), объясняет: «Разработка Unity была полностью с открытым кодом и доступна любому на Launchpad. Из-за солидного объема доли Canonical в разработке и участия большого числа сотрудников Canonical многие считали его проектом „Исключительно Ubuntu“, хотя этого вообще не планировалось. Мы видим, как вокруг Unity 7 на Ubuntu формируется сообщество заинтересованных людей, чья цель — обеспечить его работу в будущем».

Мы еще услышим об этом сообществе, и если вы безутешны по поводу утраты Unity, загляните во врезку внизу. Но мы (и, по видимому, Canonical тоже) очень бы хотели, чтобы сначала вы предоставили шанс Gnome.



➤ HUD был одной из самых инновационных функций Unity, дающей мгновенный доступ к глубоко скрытым опциям меню. Но использовали ее лишь немногие.

Установка Unity

Unity 7 можно, условно говоря, воскресить из мертвых на чистой установке 17.10 (или если вы удалите его и в ужасе осознаете, что это была любовь). Есть несколько отладок, и они будут делаться и дальше, но жаждающие новых функций будут разочарованы. Технически этот проект уже несколько лет был законсервирован (до прошлого апреля, когда его слили), а разработка сосредоточилась

на Unity 8. Однако Unity 7 будет поддерживаться на протяжении всей жизни 16.04 (до 2021 г.) и в том или ином виде будет в репозитории Universe для 18.04 LTS.

Всё, что нужно —
`$ sudo apt install unity`

Держаться за нечто привычное хорошо и мило, но сейчас самое время начать привыкать к Gnome.

Даже если Gnome вам не по душе, в наши дни есть выбор из множества других рабочих столов из числа незамороженных проектов.

Если вы предпочитаете направление развития Unity 8, приглядывайте за проектом [yunit \(https://unity.io\)](https://unity.io), который разветвил свою кодовую базу и развивается неплохо. Больше о нем — ближе к концу нашей статьи.

Представляем Gnome

Настольный Ubuntu теперь с Gnome, современным и совсем другим. Его списали было со счетов, но мы рекомендуем как минимум дать ему шанс.

Уиллу Куку есть что сказать для успокоения тех, кто боится этого нового рабочего стола. «Еще в самом начале цикла разработки 17.10 мы намечали сделать переход на Gnome Shell как можно более гладким для всех наших пользователей... Мы подумали о том, что произошло бы, если бы кто-то, не слышавший последних новостей Ubuntu, получил обновление до 17.10 и обнаружил, что Gnome Shell теперь заменил Unity; и мы решили, что такое радикальное изменение рабочего стола многим будет непросто понять.

Поэтому в июле мы отправились на GUADEC [ежегодная европейская конференция пользователей и разработчиков среды рабочего стола Gnome, — прим. пер.], чтобы пообщаться с разработчиками Gnome, [и] уехали оттуда с планом загрузить расширение Dash to Dock (со знакомой программой запуска слева), и позднее расширили этот план, включив в него KStatusNotifier, чтобы добавить поддержку индикаторов приложений».

Gnome 3 тоже изменился с момента своего появления семь лет назад, и если вы уже какое-то время с ним не работали, стоит на него взглянуть. Первое, что обычно смущает пользователей

Gnome — это отсутствие меню приложений и дока (панели, где традиционно отображаются открытые окна). Как только у вас будет установлена и заработает 17.10, вы увидите, что Canonical обеспечила для этого замену. Она обеспечивается через Ubuntu Dock (ответвление расширения Dash to Dock для Gnome), которое гордо несет стражу слева.

Многоточие внизу открывает меню приложений, как в Unity. В Canonical также добавили поддержку индикаторов Application [Приложений], чтобы приложения могли обеспечивать сервисы из отдельных меню вверху справа. По словам Уилла: «Внеся эти два дополнения в стандартный Gnome Shell, мы надеемся, что долгосрочные пользователи Unity смогут быстро и легко перейти и начать работу с Gnome».

В Gnome приложения находятся и запускаются из вида Activities, который можно вызвать разными способами: щелкнув по надписи Activities наверху слева, переместив курсор в верхний левый угол или (проще всего) нажав на клавишу Super (Windows). Мы создали очень удобную схему с пояснениями всех прелестей Gnome, чтобы она была у вас под рукой (см. внизу).

Ориентируемся на рабочем столе Gnome

Dock

Запускайте или переключайте приложения здесь. Работаящие приложения помечены точкой.

Верхняя панель

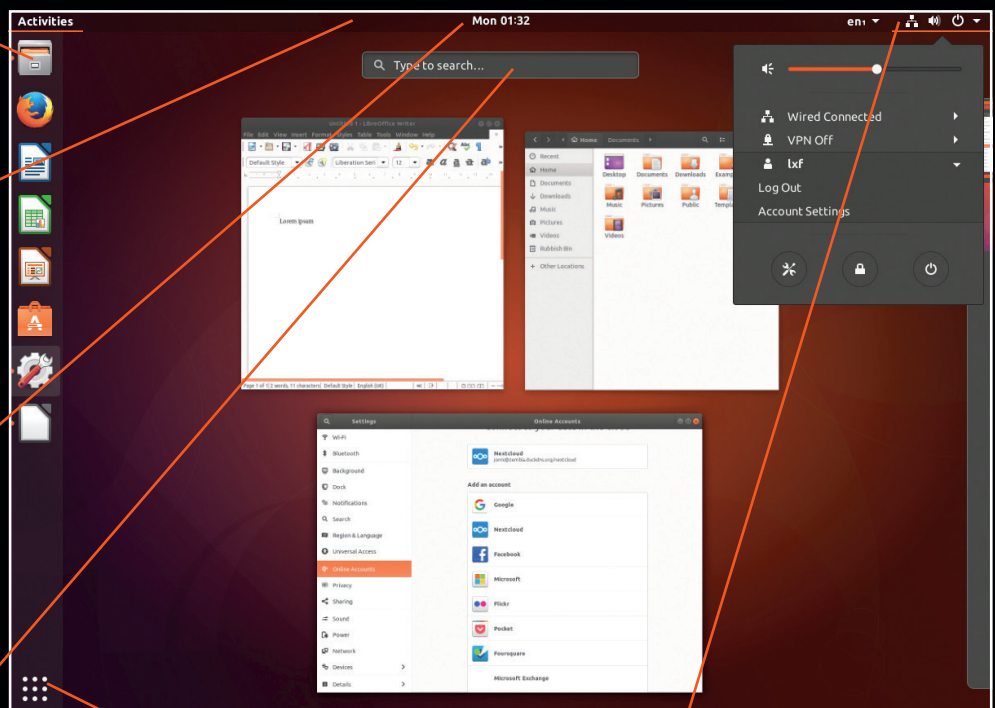
Глобального меню больше нет, но некоторые приложения и расширения используют эту область для меню. Загляните в Dash to Panel, если хотите использовать эту область.

Область Календаря и Уведомлений

Здесь появляются устройства USB и извещения программ; отсюда управляется большинство музыкальных плееров, а также назначенные мероприятия.

Окно поиска

При вводе имени приложения (внутри окна поиска щелкать при этом не надо) вы увидите несколько подсказок.



Меню приложений

Щелкните, чтобы ознакомиться со списком всех установленных приложений и восхититься эффектом размещения. Щелчок правой кнопкой по приложению добавит его в Dock.

Меню состояния

Сетевые настройки (включая VPN), громкость, индикаторы приложений находятся здесь. За меню скрыты динамические рабочие области (виртуальные рабочие столы).

Гноме в подробностях

Сделайте работу с Гноме лучше, изучив некоторые комбинации клавиш...

Гноме сильно изменился с тех пор, когда у него были характерные верхняя и нижняя панель в Ubuntu 10.10 и более ранних версиях. Название Гноме 3 не отражает всю его революционность. На него не похож ни один рабочий стол — по крайней мере, на его стандартную форму, в чем вы сами убедитесь, изучая указания во врезке на стр. 31.

Благодаря настройкам Ubuntu он более доступен для тех, кто переходит с Unity, но мы уже слышим критику. «Мне не нравится то, что я вижу», пишет один читатель, чьим взглядам вторят в Интернете. Проблема Гноме 3 в том, что он довольно минимален по дизайну, и ему не хватает немало компонентов традиционных рабочих процессов. В 2011 г. такая же критика звучала по поводу Unity; и хотя часть пользователей сбежала на Cinnamon, рабочий стол Mint, многие остались с Unity, выяснили, как взнудать его функции — и научились его любить.

Клавиши счастья

Отдельные среды рабочего стола укоренили в нас идею, что всё следует делать мышью, и в Гноме это вполне возможно — хотя и не очень эффективно. Работа с Гноме намного упростится,

если выучить несколько горячих клавиш. Не волнуйтесь, речь не о многоуровневой комбинаторике клавиш в стиле Emacs: эти несколько комбинаций довольно интуитивны. Все они включают клавишу Super (на большинстве клавиатур это клавиша Windows, однако по очевидным причинам мы не хотим так ее называть).

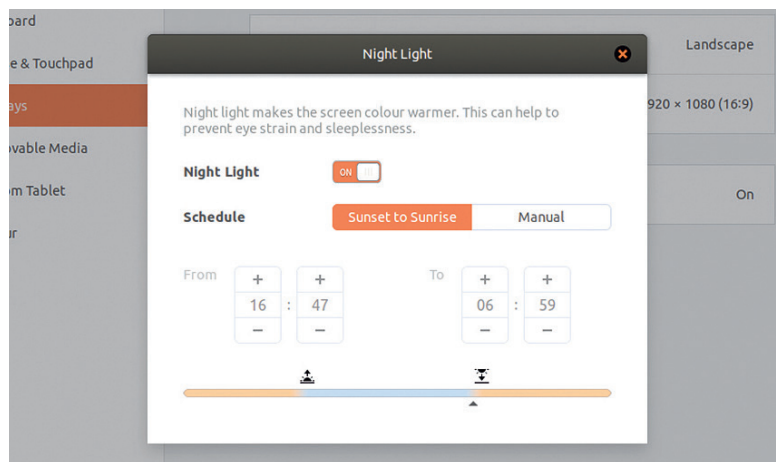
- » **Super** Выводит вид Activities
- » **Super A** Меню Applications [Приложения]
- » **Super Стрелки вверх/вниз** Разворачивает/восстанавливает активное окно
- » **Super Влево/Вправо** Прикрепляет активное окно к левой/правой половине экрана
- » **Super PgUp/PgDn** Перемещает рабочую область вверх/вниз

Помимо этого, обдумайте совет Уилла Кука: «Мой главный совет — поиграйте с ним. Через пару дней вы увидите, что всё потихоньку встает на свои места, и станет понятно, как тщательно всё продумывалось, чтобы всё просто работало». Gnome Shell приятно использовать, там множество новых функций для пользователей Unity 7, огромный каталог расширений и отзывчивое сообщество. Если у вас возникнут проблемы, загляните на форумы Ubuntu Forums (<https://ubuntuforums.org>) или Ask Ubuntu (<https://askubuntu.com>), и не забудьте также зарегистрировать обнаруженные ошибки в Launchpad.

Canonical немало сил вложила в Snap, универсальный формат пакетов. Однако он не скоро лишит нас наших Deb. Представитель Ubuntu Desktop Уилл поясняет: «...в 18.04 мы рассчитываем поставлять в виде Snap много стандартных настольных приложений. Предстоит еще немало трудов, но разработка ведется активно. У нас есть „платформенные“ Snap для Гноме, а у разработчиков KDE — для KDE, а значит, Snap, использующие эти наборы инструментов и библиотеки, могут быть намного меньше, все общие компоненты будут устанавливаться один раз для каждого компьютера».

Все имеющиеся в *Ubuntu Software* приложения распространяются в виде Snaps, а значит, всего одно нажатие отделяет вас от свежих версий, например, хипстерских текстовых редакторов *Atom* или *Brackets* или клиента *Corebird Twitter*. Snaps можно также установить из *Firefox* с сайта <https://snapcraft.io>, или из командной строки по `snap install`.

Запуск Ubuntu с видеодрайверами с открытым кодом включает *Wayland* по умолчанию. У нас это работало прекрасно, но *Wayland* —



» Night Light уменьшает усталость глаз в ночные часы. Можно также изменить температуру цвета через *dconf*.

Гноме каким он задумывался

Canonical сознательно сделала свои правки Гноме как можно более ненавязчивыми — в разительном контрасте с тем, как настольные компоненты вроде приложений *GTK* и Гноме густо залатывались ради соответствия Unity. Несомненно, проделана солидная работа по воссозданию чего-то, похожего по внешнему виду и ощущению на Unity — но всё это делалось в гармонии с лежащими в основе технологиями, а не в пику им. Два самых основных изменения, *Ubuntu Dock* и темы *Ubuntu*, реализованы в качестве расширения Гноме и *Shell Theme* соответственно.

Однако есть другие dock и другие темы; и на самом деле в стандартной версии Гноме вообще нет dock.

Отважные духом, способные жить без постоянно работающей программы запуска, или те, кто уже привык к Гноме, могут установить базовую сессию Гноме с помощью

```
$ sudo apt install gnome-session
```

Некоторые считают сочетание верхней панели Гноме и больших строк заголовка приложений напрасной тратой вертикального пространства. Если вы из таких, используйте расширение *Pixel Saver*.

Оно убивает строки заголовка развернутых приложений, размещая соответствующую информацию в заголовке окна.

Как было сказано выше, *Ubuntu Dock* — это ответвление популярного расширения *Dash to Dock*. На него стоит взглянуть, потому что оно более настраиваемое, но вы можете обнаружить, что жизнь без dock вполне возможна.

Dash to Panel также заслуживает вашего внимания: здесь используется «напрасно потраченное» пространство в верхней панели для отображения сжатого dock.

новая технология, и есть сообщения, что не всё вело себя так, как ожидалось. На случай нештатного поведения рабочего стола из окна приглашения доступна сессия X.org. Пока есть проблемы с программами записи с экрана, программами удаленного рабочего стола и некоторыми играми (см. https://fedoraproject.org/wiki/How_to_debug_Wayland_problems#Known_issues.2C_frequent_complaints.2C_fundamental_changes).

Есть и фундаментальные различия между Wayland и X.org. Одно из самых значимых — невозможность запускать приложения GUI с помощью `(gk)sudo`, что некоторые любят делать для менеджеров файлов или текстовых редакторов. Попытки сделать это приводят к появлению таких ошибок:

```
Unable to init server: Could not connect: Connection refused
[невозможно инициализировать сервер: Невозможно
подключиться: В подключении отказано]
```

Управление Wayland

Wayland имеет собственный, более надежный механизм запроса доступа root (протокол `admin://`), и именно приложения должны инкорпорировать поддержку. Обходной путь — предварив командой `sudo`, запустить `xhost si:user:root`, где `user` — ваше имя пользователя (или добавить его к `~/.bashrc`), хотя есть серьезные причины не запускать приложения внутри сессии GUI от имени root.

Gnome управляет расширениями через сайт <https://extensions.gnome.org>. В других дистрибутивах это работает через плагин браузера, который общается с собственной утилитой соединения дистрибутива. К сожалению, этот плагин браузера больше не поддерживается Firefox, но доступен в Chromium (`sudo apt install chromium-browser chrome-gnome-shell`, затем посетите приведенный выше URL, чтобы получить плагин браузера).

Ubuntu управляет расширениями оболочки через инструмент Software, в Addons > Shell Extensions > Extension Settings [Дополнения > Расширения оболочки > Настройки расширений]. Изначально вы найдете здесь только два расширения: Ubuntu Dock и ApplIndicators. Их нельзя отключить в стандартной Ubuntu Session, поскольку они являются неотъемлемой частью работы с Ubuntu.

В таком случае можно спросить: зачем тогда нужны настройки? И ответ будет двояким: если установить Vanilla Gnome Session (см. внизу), то эти два расширения будут деактивированы, а если выполнить `sudo apt install gnome-shellextensions`, то станет доступным управляемый Ubuntu выбор расширений.

При установке этого пакета в виде Activities установится еще и *Tweaks*, известный ранее как *Gnome Tweak Tool*. Для многих *Tweaks* — основной способ внести здравую нотку в Gnome. Он дает простой доступ ко всему, от изменения контуров шрифта [hinting] до значков на рабочем столе и былой комбинации клавиш `Ctrl+Alt+Backspace`. *Tweaks* также может управлять темами оболочки, но только если сначала установить расширение User Themes, самое популярное из расширений Gnome. Сторонние темы можно позднее посмотреть на www.gnome-look.org и затем скачать прямо из авторских репозиториях. Темы надо распаковывать в директорию `~/.themes`, откуда их будет брать *Tweaks*.

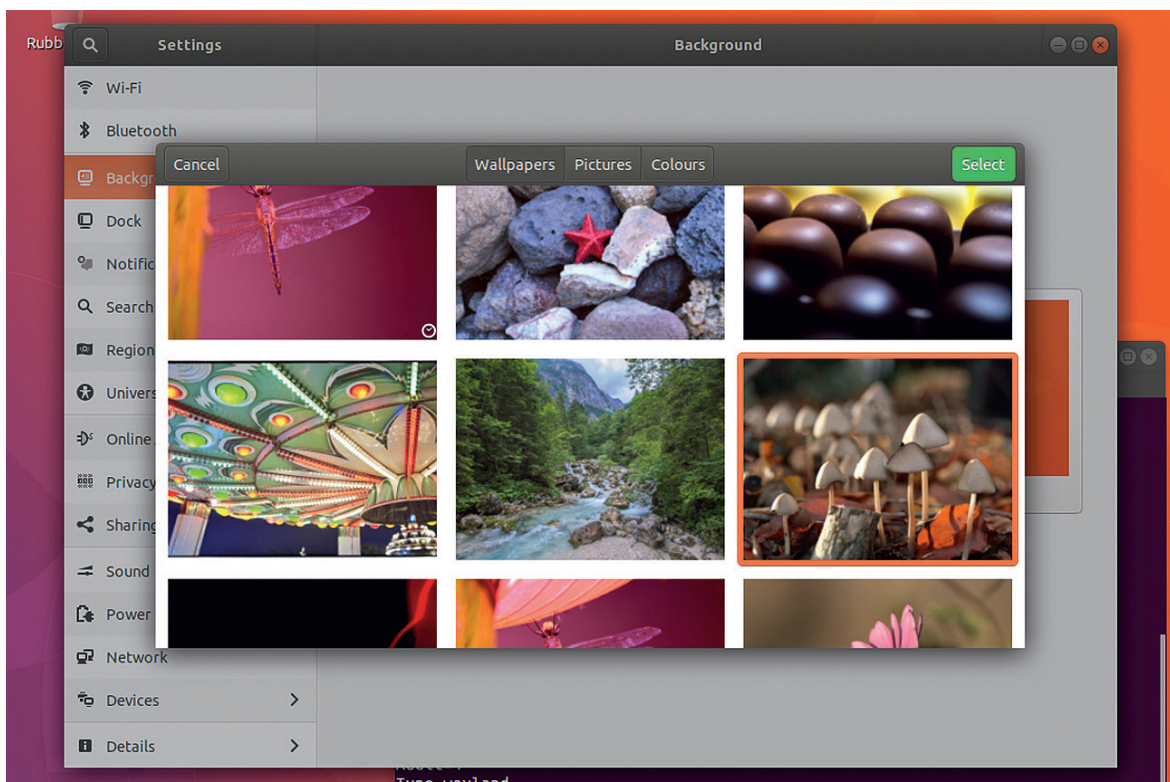
Tweaks дает доступ ко всему, от шрифта до комбинации клавиш Ctrl + Alt + Backspace.

Мы (надеюсь) показали вам, как перестать волноваться и научиться любить Gnome, но если не вышло — есть множество других вариантов Ubuntu со всеми основными рабочими столами. Вы даже можете попробовать эти рабочие столы (восемь из них, если быть точными) безо всякого риска с DVD этого месяца. И их очень просто добавить в вашу систему без установки новой версии (но узнайте на стр. 34, как это сделать). Чтобы установить сессию KDE Plasma (и специальных утилит KDE в Kubuntu), выполните

```
$ sudo apt install kubuntu-desktop
```

Замените последний аргумент на `xubuntu-desktop`, `budgie-desktop`, `cinnamon-desktop-environment`, `mate-desktop-environment`, `lubuntu-desktop` или `lubuntu-qtdesktop` в зависимости от ваших симпатий относительно рабочего стола.

> Имеется несколько вдохновляющих фонов, но пиратско-трубка тема нам нравится больше.



Разновидности Ubuntu

Если вы решите, что Gnome не для вас, есть целый ассортимент замечательных разновидностей Ubuntu на выбор...

Предыдущие поколения рабочих столов KDE (заслуженно или нет) имели репутацию ресурсоемких и чрезмерно настраиваемых. Но KDE Plasma 5 в том виде, в каком он присутствует в Kubuntu, является одним из самых шикарных рабочих столов. Он явно не подходит для старого «железа», и вам понадобится видеооборудование этого десятилетия, чтобы в полной мере оценить его затухания, тени и прочие эффекты (все они подвергаются рендерингу через QtQuick и поэтому не нагружают CPU).

Во многом KDE Plasma (особенно при настройке в Ubuntu) придерживается традиционной метафоры рабочего стола с меню внизу слева, программой вызова приложений внизу и системным лотком внизу справа. Но к этим традиционным идеям добавлены современные удобства. Нажмите на меню (или используйте клавишу Super) и начните печатать, и появятся соответствующие приложения. Нажмите Alt+F2, и появится диалоговое окно KRunner. Приложения и недавние документы запускаются так же, как обычные команды или tap-страницы (поставьте перед командой #, например, #uname) и многое другое (см. <https://userbase.kde.org/Plasma/Krunner/en>).

И несмотря на все эти современные навороты, Plasma 5, по всеобщему признанию, меньше нагружает ресурсы, чем Gnome. При этом на системах, где более 4 ГБ памяти, нет разницы ни туда, ни сюда. Как и Gnome, Plasma предлагает интеграцию с учетными записями онлайн, такими как в Google, Exchange, Jabber, Nextcloud, Telegram и т. д.

В Plasma недавно добавились поддержка папок на рабочем столе (и снова отсылка к традиционализму), и введена новая функция папок: пружинная загрузка [spring loading]. Это означает, что когда вы перетаскиваете файл на папку, папка открывается, благодаря чему перемещать файлы в поддиректории намного удобнее. А если вам интересны самые последние технологии Plasma, стоит обратить внимание

на KDE Neon. В нем используется база Ubuntu 16.04 и самая последняя, постоянно обновляемая начинка KDE.

Здесь всё меняется довольно быстро, и вполне можно понять нежелание Canonical поддерживать рискованные технологии даже в своих релизах разработки; поэтому они придерживаются Plasma 5.10. Kubuntu имеет PPA обратных портов, так что если вы хотите попробовать Plasma 5.11 (выпущенный в прошлом сентябре и использующий Vaults для шифрованного хранения), загляните на <https://launchpad.net/~kubuntuppa/+archive/ubuntu/backports>.

Ubuntu Mate

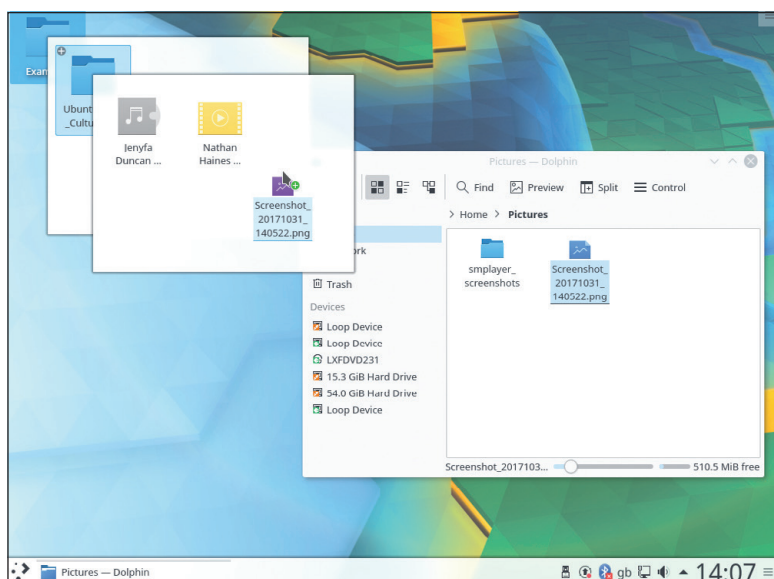
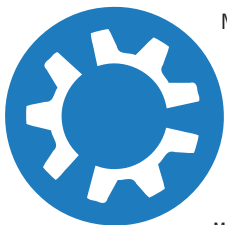
Если вы тоскуете по счастливым дням Gnome 2, тогда Mate (произносится Ма-тэ) как раз для вас. Рабочий стол является ответвлением Gnome 2 и предъявляет весьма скромные системные требования. Но это не просто атавизм рабочего стола: вы найдете здесь современный менеджер файлов (*Caja*), поддержку индикаторов Optimus (если у вас хорошее оборудование Nvidia) и пакет Snap, установленный по умолчанию (он первым из дистрибутивов делает это). Этот Snap — *pulsemixer*, основанный на консоли микшер, которого нет в стандартных репозиториях Debian или Ubuntu. Есть также *Redshift* для уменьшения нагрузки на глаза в ночные часы (в Gnome эта функция встроена в виде *Night Light*).

Вероятно, самое потрясающее в том, что с Mate 1.18 (новинка Ubuntu 17.10) идет подборка новых раскладок панелей. Они доступны через Mate Tweak (в разделе Panel), и некоторые из наших фаворитов представлены ниже:

- » **Traditional** [Традиционная] Раскладка по умолчанию, вдохновленная Gnome 2 из времен нашей беззаботной юности
- » **Mutiny** День Unity, с боковой панелью, встроенным в приложения HUD и Global Menu (которые также присутствуют в других раскладках)
- » **Cupertino** Раскладка, вдохновленная macOS, которая использует док Plank, чтобы вы могли наслаждаться работой мыши
- » **Redmond** Меню, ярлыки, пейджер приложений и системный лоток — всё расположено внизу. Проклятие, наложенное Windows 95.
- » **Netbook** Наш научный редактор и некоторые другие по-прежнему используют эти небольшие устройства. Данная раскладка размещает значки в верхней панели (в стиле Dash to Panel), максимально экономя пространство по вертикали.

Мы бы включили на DVD 32-битный Ubuntu Mate, но ISO немного велик, и вместо него мы выбрали Lubuntu (со слегка устаревшим набором инструментов *GTK2* и без встроенной поддержки мультимедиа). Но не тревожьтесь: Ubuntu Mate спокойно будет работать на компьютере с 1 ГБ ОЗУ (а возможно, и меньше, но вряд ли вам от этого будет много прока при просмотре сайтов с избытком JavaScript).

Доступна также версия Raspberry Pi 2 или 3 — для тех, кому требуется немного больше «нормального рабочего стола» на своем компьютере-крошке. Лозунг Mate «За ретроспективное будущее» никогда не звучал правдивее — это передовая отсылка к прошлому.



» Plasma 5 предлагает эти безумные авангардные фоны и папки с пружинной загрузкой. В пору нашей молодости папки были непроницаемы и бессердечны.

Lubuntu

Это наименее требовательный из всех дистрибутивов рабочего стола Ubuntu, и вы найдете его на DVD (в 32-битной форме). Lubuntu использует рабочий стол LXDE, работающий на легковесном менеджере окон *Openbox*. LXDE — основа рабочего стола Raspbian и поэтому покажется знакомым пользователям Raspberry Pi.

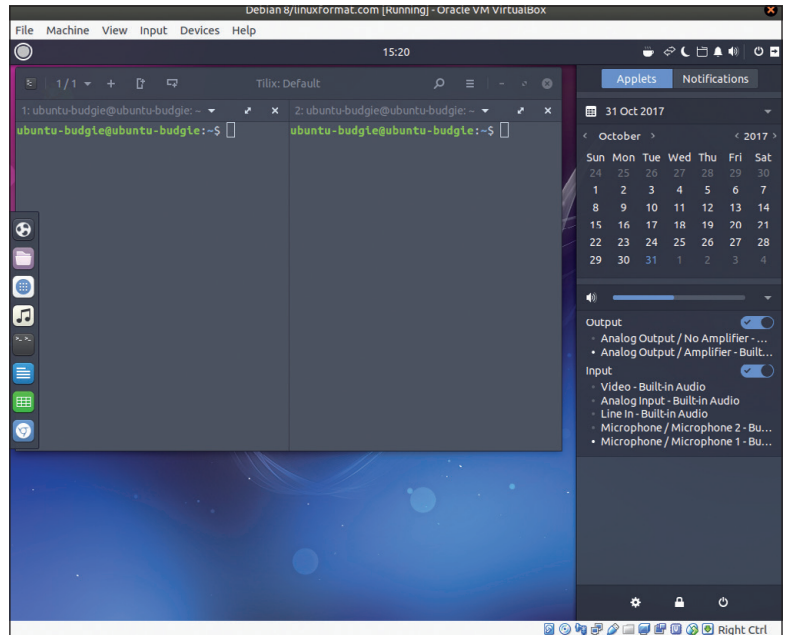


Будущее Lubuntu неведомо — не в том смысле, что он может исчезнуть, а в том смысле, что вскоре он может весьма сильно измениться внешне. LXDE в настоящее время использует GTK2, уже старый, обремененный ошибками и без шансов на поддержку Wayland; и самого его поддерживать сложно. Зато он очень легковесный, и многие очень милые приложения всё еще с ним ладят. На некой стадии команда LXDE была полна решимости перейти с GTK2 на Qt5, создав проект LXQt совместно с Razor-Qt. Но, судя по всему, LXDE и LXQt продолжат существовать отдельно, так что вы и дальше сможете наслаждаться PCManFM, Gnome Mplayer и почтовым клиентом Sylpheed.

Собственное правило номер один Lubuntu гласит, что дистрибутив должен работать на любом компьютере не старше 10 лет. Можно загрузить рабочий стол с 256 МБ ОЗУ, но вам понадобится вдвое больше, чтобы запускать LibreOffice, и еще в два раза больше для продвинутой работы в Сети. Желающие попробовать LXQt могут скачать дневной образ Lubuntu Next с <http://cdimage.ubuntu.com/lubuntu-next/daily-live/pending>. Нас восхищает вот что: собственно Qt5 (без библиотек KDE, которые с ним ассоциируют) весьма легковесен, но при этом придает всему блестящий вид новизны. LXQt разрешает расширенное использование тем и настраиваемость, и сейчас (в версии 0.12.0) улучшил поддержку дисплеев HiDPI.

Ubuntu Budgie

Это самая новая разновидность Ubuntu (возникшая в 17.04), и она использует рабочий стол Budgie из популярного дистрибутива Solus (бывшего Evolve OS). Одной из ключевых функций Budgie является его мощная боковая панель Raven [англ. Ворон], которую



► Ворон молвил: «Знай, Ленора, всё настроить будет можно — от звучанья до известий — никуда не выходя».

можно настроить как душе угодно. Добавлять программы и проводить тонкую настройку можно прямо на экране приветствия, так что вы можете с ходу настроить рабочий стол на свой вкус.

Budgie использует прекрасный эмулятор терминала Tiling. На данный момент Budgie собран на технологиях Gnome, и помимо Tiling и почтового клиента Geary предлагает те же программы, что и стандартный релиз Ubuntu. Сюда входят Rhythmbox с новой внешностью, LibreOffice и все остальные программы Gnome 3.26. Однако следующий релиз рабочего стола, Budgie 11, переключится с GTK на инструментарий Qt, и к нему стоит приглядываться.



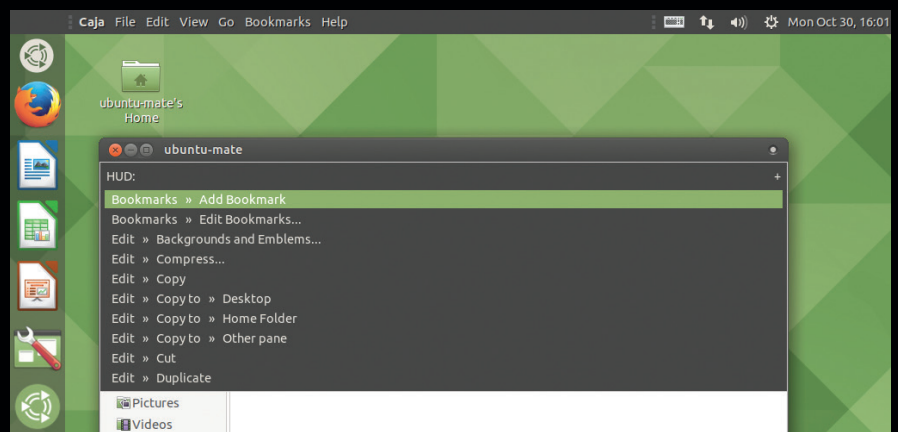
И кое-что другое...

Xubuntu популярен уже много лет. Он использует рабочий стол Xfce, тоже легковесный, хотя и не так, как LXDE. После долгого процесса исправления ошибок и тестирования Xfce близок к релизу 4.14. В нем завершится переход рабочего стола на GTK3. А до тех пор Xfce остается отличным рабочим столом, и в этом релизе предлагаются улучшения производительности медиа-плеера Parole. Для китайских пользователей предназначен Ubuntu Kylin, использующий UKUI, ответвление рабочего стола Mate. Этот релиз добавляет новую тему значков и улучшения производительности.

Ubuntu Studio — дистрибутив для любителей творить мультимедиа. Он по умолчанию включает JACK (что, вероятно, спасет вас от изрядной головной боли), цифровую рабочую станцию аудио (DAW) Ardour, Scribus (для настольной публикации), Darktable, Blender, Krita и многое другое. Используется рабочий стол Xfce — для ускорения отклика. Опытные пользователи захотят попробовать использовать ядра real-time или low-latency. Очень легко добавить в вашу установку другие рабочие столы и менеджеры окон, для которых

нет специального дистрибутива Ubuntu. Нас весьма впечатлил проект Sway — это работающая на Wayland замена i3, мозаичного менеджера окон, популярного среди разработчиков. Нам также

нравится Enlightenment, который можно попробовать на нашем DVD или установить из E21 PPA (инструкции см. на www.enlightenment.org/docs/distros/ubuntu-start).



► Новая раскладка Mutiny возвращает Unity HUD и Global Menu.

Восставший из пепла

В Canonical, возможно, больше не будут работать над Unity 8, мобильными устройствами или конвергенцией, но эти проекты отнюдь не мертвы.

Марк Шаттлворт в беседе с eWeek недавно сообщил о причинах, по которым его компания сменила курс. TL;DR Canonical — это бизнес, причем бизнес, особенно успешный в сфере облачных технологий и IoT.

Она выпускает популярный рабочий стол, который во многом работает с ними в тандеме и является жизненно важным для корпоративных клиентов. По сумме количества настольных ПК и серверов (на одной базе), это самый популярный дистрибутив Linux, и полностью от него отказаться было бы безумием. Однако было

могли увидеть Ubuntu на Unity 8 и Mir на каком-нибудь более привлекательном оборудовании, но время — деньги, и в этом-то всё и дело. Canonical хочет стать более привлекательной для внешних инвестиций, а для этого нужна рентабельность. И, без обременения Unity и Ubuntu Phone, она будет достигнута.

Мультиплатформенные мечты

Любили ли вы Unity или ненавидели, в него вложен колоссальный труд, и до последнего он выражал смелые новые идеи о будущем интерфейсов Linux. Сервер отображения следующего поколения, Mir, на котором должен был работать Unity 8, предназначался также для реализации мечты о «конвергенции», когда приложения могли бы трансформироваться и быть пригодными и для настольно-го, и для мобильного интерфейса.

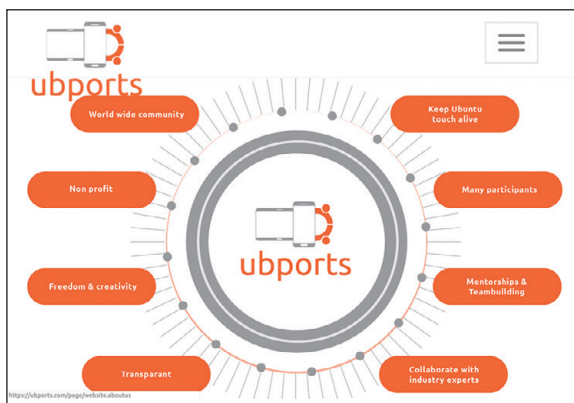
При этом Mir вполне жив и здоров. Уилл говорит: «Canonical считает Mir будущим решением для IoT, где требуется более надежный стек отображения. Mir версии 0.28 теперь доступен в Artful, с изначальной поддержкой клиентов Wayland, и мы продолжим разрабатывать его в следующем цикле разработки».

Wayland и Mir могут показаться странной парой, но от этого выиграют все, в частности, команда Ubuntu Mate, которая, возможно, предпочтет не тратить ресурсы на создание собственного композитора Wayland. Фанаты Unity, не вполне довольные настройками Canonical Gnome, захотят взглянуть на рабочий стол Artemis (<https://artemis-project.github.io>), цель которого — воссоздать Unity внутри Plasma.

Что касается будущего поддержки 32 бит, у Уилла есть слова утешения: «...уходят только ISO, 32-битные пакеты будут создаваться по-прежнему, и для 18.04 LTS Canonical решил предоставить 32-битный путь обновления с предыдущего релиза 16.04 LTS. Это означает, что обновление с предыдущего релиза будет работать, а для новых установок LTS может стать лучшим выбором, или даже вариантом, который хорошо работает на более старых компьютерах». Так что с вашим оборудованием эпохи неолита всё будет хорошо, только не ждите, что Gnome заработает на вашей видеокarte AGP [Ред.: — Я возмущен!] с 16 МБ видеопамяти. **LXF**

Canonical считает Mir решением для IoT, где требуется надежный стек отображения.

признано невыгодным тратить время разработчиков на амбициозную среду рабочего стола, которая на годы отставала от графика, и на мобильный проект, рассчитанный на пару бюджетных телефонов и планшет, работающий на устаревшей базе. Это немного жестоко — если бы дать ему еще немного времени, мы бы, без сомнения,



➤ Первые телефоны BQ и Meizu поддерживаются как унаследованные устройства, а на нескольких новых устройствах сейчас может запускаться Ubuntu Touch.

Ubuntu Phone и Yunit

В мире мобильных ОС господствуют Android и iOS, и грустно признавать, что многие проекты открытого кода не сработали (см. нашу статью в **LXF228**). И несмотря на то, что Ubuntu Touch (с его Scores и странными жестами) подходил далеко не всем, новость о том, что Canonical прекращает его разработку, заставила нас дружно лить слезы. Однако мы утешились тем, что его разработкой займется сообщество UBPorts. UBPorts выпустило два обновления Over The Air (OTA) и успешно портировало Ubuntu Touch на ряд устройств. До недавнего времени сборки UBPorts основывались на устаревшей базе 15.04 Ubuntu, но сейчас наблюдается определенный прогресс (в частности, для Fairphone 2, первого в мире этического, модульного смартфона с обновляемым

оборудованием — см. www.fairphone.com/en/2017/09/20/newcamera-modules-make-the-fairphone-2-upgradeable), поскольку оно переходит на 16.04.

От кодовой базы Unity 8 отпочковался проект Yunit (<https://yunit.io>), и пакеты доступны для Ubuntu 16.04 и всех трех каналов Debian. Ведется также работа над пакетами для 17.10. И UBPorts, и Yunit хотят перейти с Mir на Wayland, и недавняя работа по поддержке Wayland в Mir будет, вероятно, полезна, по крайней мере, в качестве механизма перехода. Yunit сосредоточен на рабочем столе, но поскольку в основе Unity 8 лежит идея конвергенции, будет вполне естественным пересечение с вариантами для мобильных устройств, и Yunit и UBPorts сообщили о предстоящей совместной работе.



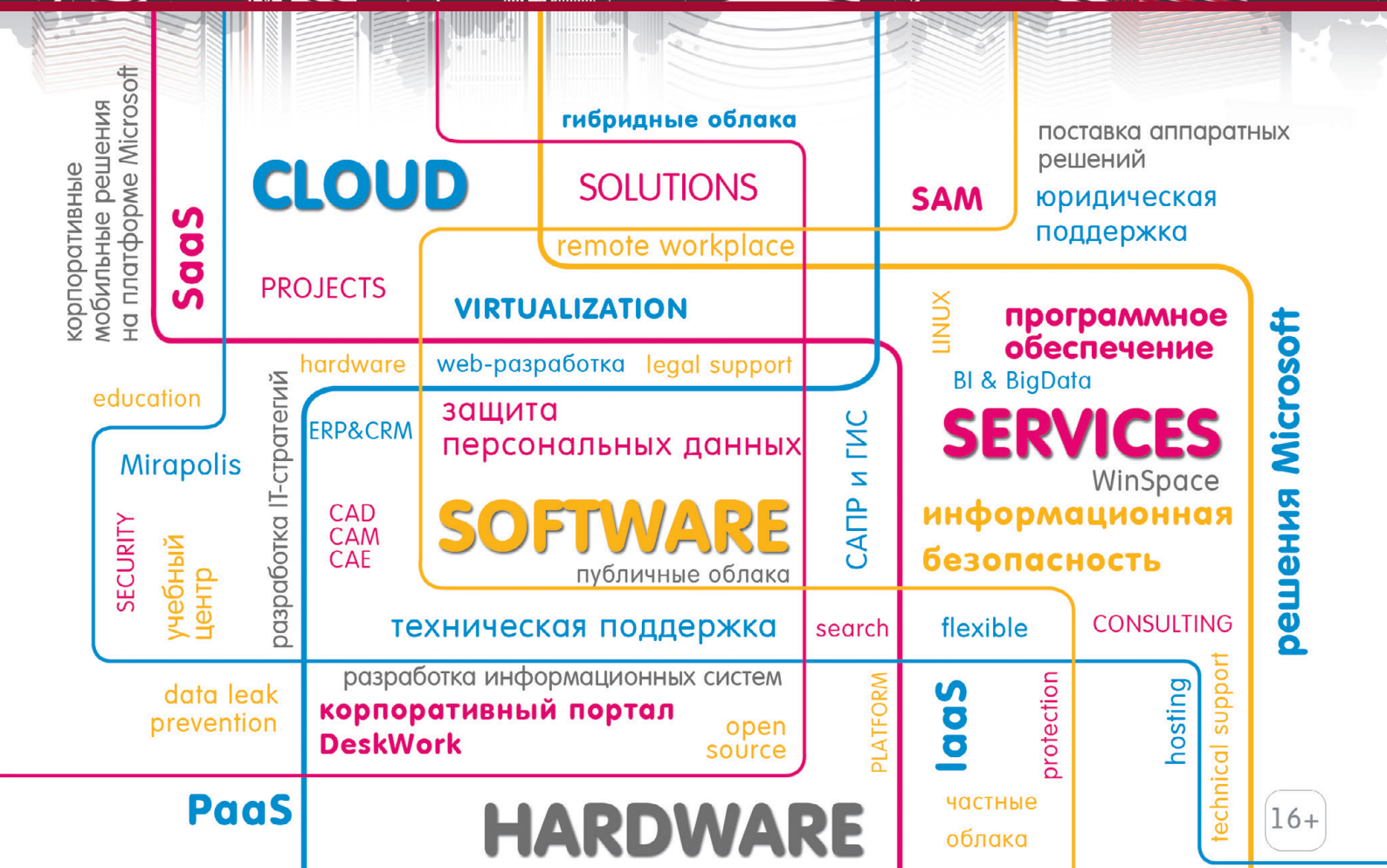
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



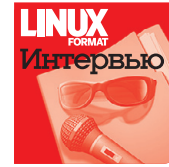
Мартин Вудвард

Тот самый

С целью явной провокации, Джонни Бидвелл отправился в штаб Microsoft в Великобритании и пообщался с гостеприимнейшим Мартином Вудвардом.

О ТОМ, КАК ВСЕ НАЧИНАЛОСЬ В MICROSOFT
 Нам пришлось задуматься,
 как продолжать производить
 открытый код.

FOSS



Читатели, вылы наголо!
 На самом деле, не надо. Все мы помним эти Хэллоуинские мемы, но им уже сто лет в обед. Кое-кто еще сохранил неприязненные пред-

ставления о Microsoft и его вражде с Linux, но времена настали другие. Настольный Linux никто не собирается удушать. Благодаря Сатье Наделле [Satya Nadella] мы поняли, что Microsoft полюбил Linux. Даже подарил нам Skype и SQL-сервер.

Как и во многих других коммерческих организациях, в Microsoft открытый код играет сейчас ключевую роль, а роль нынешнего главы проектов, Мартина Вудварда [Martin Woodward], в этом отнюдь не последняя. На конференции O'Reilly Software Architecture в октябре 2017 г. он рассказал, какие титанические усилия ему потребовалось приложить, чтобы перевести 65 000 инженеров на единую платформу, основанную не на чем-нибудь, а на нашем любимом Git. Мы знаем, что есть люди, неспособные изменить свои взгляды насчет Редмондской империи, но надеемся, что кому-нибудь это всё же поможет переосмыслить ситуацию...

Linux Format: Как вы стали частью клана Microsoft?

Мартин Вудвард: Я работал в небольшом стартапе, который выпускал плагины для Eclipse, и мы создали кое-что для Team Foundation Server от Microsoft, для контроля за исходным кодом и т.п. В конце концов в 2009 г. Microsoft нас купил, ну и я присоединился к компании.

Когда они нас покупали, было много разговоров в духе: «А что это за открытый код, который вы используете?» И нам пришлось задуматься, как продолжать производить открытый код, работая в Microsoft. Потом, немного освоившись в компании, мы стали задаваться вопросами: «А как помочь Microsoft использовать больше открытого кода? Как бы сделать так, чтобы компания сотрудничала с Open Source, а не боролась с ним?»

LXF: Когда вы только присоединились, вы, наверное, были в шоке от того, как ведется разработка? Понятно, что при переходе из небольшой команды в огромную организацию культурный шок всегда неизбежен, но что поразило именно вас?

МВ: Это была отлаженная на протяжении двух десятилетий система, при этом внутри себя очень разрозненная. Каждый всё делал по-своему. До того я работал в стартапе, где мы всё время обновляли код. А в Microsoft были эти девятимесячные циклы разработки, где на собственно создание кода отводилось всего шесть недель. Первая мысль у меня была: «А что же происходит в остальное время?» Пока я во всё это не вник, был немного шокирован. Первый же курс, который мне нужно было пройти — помимо стандартной переподготовки, ну знаете, основы предпринимательства и т.д. — был о том, как улучшить голограмму на обложке DVD, который затем упаковывается в коробку для поставки. «У нас что, поставка на дисках?» — помнится, меня это порядком удивило. »

Три главных подвижки в плане философии компании, на моей памяти, это переход от разных инженерных систем на One Microsoft; переход к активному использованию и производству открытого кода и внедрение постоянно действующих сервисов. Раньше установка была на создание готового ПО, которое оптимизируется, чтобы оно не ломалось. А с сервисами всегда оптимизируют, создавая возможность скорее исправить что-то, если вдруг оно обвалится. Это принципиально иной подход.

LXF: Мы заметили резкое изменение отношения Microsoft к открытому исходному коду, особенно когда компанию возглавил Сатья Наделла. На каком-то этапе, пожалуй, от них был самый крупный вклад в развитие ядра. Многие наши читатели скептически относятся к намерениям, стоящим за этим, но я думаю, что есть компании, которые сейчас гораздо враждебнее к открытому коду *кхм*... то, что осталось от SCO, например.

МВ: Я думаю, важно понимать, почему мы так переменялись. Мы по-прежнему бизнес, просто мы целенаправленно изменили свой подход к ведению бизнеса, чтобы сделать его более дружелюбным к открытому коду. Большую роль в этом сыграл Azure [облачный сервис Microsoft] — теперь мы зарабатываем деньги, продавая доступ к Linux, мы продаем огромное количество виртуальных машин Linux.

Но, что еще важнее, мы хотим, чтобы клиенты размещали свои разработки в нашем облаке. Таким образом, всё, что создается в Linux, очень хорошо работает в Azure, а значит, и под Huger-V, потому-то мы и сделали кучу исправлений ядра Linux. Поскольку мы продаем много Linux, он должен хорошо работать в нашем центре обработки данных. Если мы сможем повысить производительность хоть на 0,05%, да хоть на сколько, будет уже огромная экономия средств, а значит, это оправдывает содержание инженеров, улучшающих ядро.

Главное изменение в компании произошло, когда мы научились объяснять это менеджерам в терминах бизнеса: если заниматься открытым кодом, нам будет гораздо проще набирать людей. И это улучшает имидж, популярность Microsoft. Но всякие лозунги, косвенные выгоды — это не коммерческие факторы. Последние заключаются в том, что нам нужен более эффективный хостинг, чтобы наши программы были более востребованными, вот почему мы создаем так много проектов с открытым кодом.

У нас есть три причины создавать открытый код. Во-первых, некоторые проекты, такие как ядро Linux, лицензируются GPL, так что мы обязаны это делать. Мы должны не только сделать код доступным, но если кто-то отправляет нам откры-

тую версию, и это всё равно что создавать проект самим.

Еще одной причиной было само ядро Linux, которое мы улучшали для ускорения работы внутри Azure или Windows Server или где-либо еще. Так почему бы нам не поделиться этими изменениями со всеми — мы же по-прежнему в выигрыше?

Мы открыли код .NET под достаточно свободной лицензией, лицензией MIT. Причина в том, что мы хотим привлечь на эту платформу как можно больше проектов. Нам не нужно проблем с лицензиями Apache и GPL, мы хотели, чтобы она была доступна везде. В итоге, остановились на MIT с небольшой припиской в кодовой базе, где говорится: «Кстати, мы это делаем не ради патентованная — вот вам и патенты».

О «ПОДЧИЩЕННОМ» OPEN SOURCE Пришлось выпиливать брань из кода Линуса Торвальдса, а браниться он любит!

ку, скажем, с пятью книгами, мы должны отправить им CD с кодом. Этого требует лицензия, и если мы хотим, чтобы люди уважали наши лицензии на коммерческое ПО, мы должны уважать лицензии с открытым исходным кодом.

Но главная причина, почему мы это делаем даже для кода с лицензиями MIT и Apache, в том, что если вы используете свою внутреннюю копию какого-нибудь открытого проекта, то это становится ответвлением. И чем дальше вы развиваете ответвление отдельно, вне основного проекта, тем сложнее интегрировать изменения основного проекта в вашу кодовую базу. Со временем вы получите совершенно самостоятельную ветвь исходного открытого проекта. Вы больше не сможете получить от остальной части сообщества добавления

Так что есть и наш прямой вклад в деятельность сообщества, а есть и сотрудничество. Те же .NET или TypeScript, что-то в стиле языков, трудно убедить людей использовать самую новую версию чего-либо. А если это запустить на сервере сборки CI с открытым кодом, мы получаем прямую обратную связь, и разработка идет активнее. Что способствует в Microsoft изменению подхода к работе. Процесс разработки стал гораздо более гибким и интенсивным. Мы по сути этого не монетизировали — это было или связано с Windows, или на этом строились какие-то сервисы. Там было много интеллектуальной собственности, но от коммерческой модели мы не отказались.

LXF: Вы ведь также открыли компилятор C#, Roslyn, да? Я его помню, потому что он похож на НЛО.

МВ: Ага, код опубликовал парень по имени Андерс Хейлсберг [Anders Hejlsberg], и это хорошая иллюстрация того, как MS переходил к открытости. Я тогда отвечал за Codeplex, это наша неизменная хостинг-платформа для открытого кода. И вот он решил открыть код Roslyn, такое приходится делать нечасто. Так что мы какое-то время даже не решались нажать кнопку публикации, все думали: «Будет ли оно работать, получится ли?» Но решились, и оно заработало, и это было здорово, но на самом деле мы выкладывали его на открытой хостинг-платформе.

Сегодня мы бы не назвали это настоящим открытым кодом. Несмотря на то, что он находится под лицензией Open Source и исходники доступны, внутри все инженеры работали на TSS (внутренняя версия VSTS), то есть в другом месте. Пользователи получали версию и могли отправить запрос на включение изменений, а затем инженерам приходилось эти изменения накатывать, согласовывать внутри системы, что-то там с ними колдовать и потом выдавать обратно.



Когда мы перешли на *Roslyn*, с внесением кода произошел резкий сдвиг, и мы соответственно перевели его на GitHub, но теперь это было уже не «разработчики — отдельно, сообщество — отдельно», а вся команда перебралась на GitHub. Так что репозитории были общие. И если разработчики хотели внести изменения в *Roslyn*, запрос на включение для них был тот же, что и для вас.

Для инженеров, никаких секретов нет — свой код вы добавляете в тот же репозиторий. Это много что поменяло в процессе документирования, мы убедились, что лучше добавлять информацию непрерывно и из многих источников, а не собирать, чтобы затем выдать целиком. Это развивает дух сообщества внутри бизнеса.

Я уже упоминал о переходе от отдельных групп к единой команде инженеров. Теперь мы все используем одну и ту же инженерную систему, и все используем Git. Если я хочу внести изменения, скажем, в *Notepad*, чтобы, наконец, добавить поддержку перевода строки в Unix (я уже не раз приходил к этой мысли) — я могу отправить запрос на включение команде Windows с этим изменением. Я вижу весь их код, так что это внутренняя модель исходника. Таким образом, благодаря нашему взаимодействию с открытым кодом команды инженеров поняли и смысл, и принцип работы с запросами на включение. Сотрудничество с открытым кодом также сильно изменило наш способ обработки запросов.

Основную ветку мы стараемся держать чистой, поэтому, если подается большой запрос на включение, мы тщательно проверяем, чтобы там не было сквернословия. Мне, например, пришлось выпилить брань из кода Линуса Торвальдса, а браниться он любит! Сделав код, добавляемый в основную ветку, чистым и красивым, мы будем счастливы включить его в продукт Microsoft.

LXF: Как насчет синдрома «стеклянного дома», когда люди понимают, что их открытый код доступен для всеобщего обсуждения? Поминать, в 2004 г., когда в открытый доступ утекла некоторая часть кода Windows 2000, многих буквально потрясло то, насколько здорово этот код был написан.

МВ: Когда мы открывали исходники для многих наших проектов, было удивительно, как мало понадобилось доделывать и подчищать. Любому инженеру знакомы эти настроения, когда ты думаешь: «Хорошо, мы откроем исходники, но мне нужно шесть месяцев, чтобы кое-что там доделать, прежде чем поставить рядом свое имя». Обычно я отвечаю: «Чувак, это поставяется на восемь миллиардов устройств, а ты говоришь мне, что код не хороший?» Если хороший, выпускай. Это то, что инженерам нужно просто преодолеть, и потом уже всё в порядке.

Мы всегда проводили оценку кода внутри Microsoft. В настоящее время это делается по запросу на включение, но делается всегда. Это способствует тому, чтобы держать вещи в порядке, и для кода всегда были свои стандарты. Проверяете код на безопасность — и никаких «пасхалок» и прочего.



LXF: Ну была же классика *Word Pinball*, и симулятор полета, который всплывал, если набрать какое-то безумное число в определенной ячейке в электронной таблице *Excel*...

МВ: Да раньше в *Excel* была целая игра *Doom* (см. <http://eeggs.com/items/719.html>). Это было потрясающе. Самое близкое к «пасхалке», что находил я — это когда вы клонируете репозиторий Git из VSTS и получаете логотип в ASCII-арте. Пожалуй, это самое близкое, и я без труда с этим справился

LXF: В наши дни всё серьезно.

МВ: Да, но нам и платят за работу с ПО с открытым исходным кодом. Так что это работа мечты!

LXF: Расскажите о новой Windows Subsystem for Linux (WSL); это очень интересно. Правда, много, наверное, разочарует, что она не позволяет запускать полностью рабочий стол, чтобы избавить всех от страха получить незагружаемую систему после установки Linux. Но и полноценный доступ к командной строке без мороки с *Cygwin* — уже здорово. Разработчики порадуются.

МВ: Именно, это и делалось изначально для разработчиков. Парень, который руководит этой командой, тоже британец — если захотите пообщаться, вам не придется ехать за тридевять земель! Проект был сделан для инженеров. Мы в принципе хотим сделать Windows отличной ОС для разработчиков, чтобы запускались какие угодно приложения, и *Office*, и *Outlook*, и прочее. Но если надо запустить *apt-get* или воспользоваться *Vim* или *Emacs* (давайте не будем развивать эту тему), то всё это там тоже есть, и нормальный *Bash*, где вы вводите код.

И разнообразные дистрибутивы есть. Виртуализированы они просто потрясающе, это вовсе не как в *Cygwin*, они настоящие, всё отлично спроектировано. Если вы еще не видели, обязательно попробуйте. Вы получаете все преимущества групповой политики и AD, но также и свободный доступ к оболочке *Bash* и возможность делать что-то вручную. Если захочется, запускайте X и настольные

приложения через него, но большинству людей нужна просто командная строка. Почему Mac'и так популярны среди разработчиков? — а это почти готовый рабочий стол, в котором всё есть. Хотя в Ubuntu зашли далеко — компания просто изменила свой рабочий стол, так ведь?

LXF: Да, перешли с одного непопулярного на другой, тоже непопулярный, но по другим причинам. Но не волнуйтесь, в главной статье этого месяца как раз всё про *Gnome* и его дикие и безумные расширения *JavaScript*.

МВ: Ха! Я думаю, что большинство пользователей Linux любят быть в курсе того, что происходит, и с радостью замажут руки, настраивая такие штуки. Что хочешь — то и получишь. А это, по-моему, самое главное. Вот чего я не ожидал, так это что *TypeScript* так взлетит. Это тоже классика от Андерса. Тогда я думал: «Кому нужен *JavaScript* от Microsoft?», и вдруг за него взялась команда *Angular.js*, немного его доделала и получился типобезопасный вариант *JavaScript* для больших приложений. Всё это происходило у меня на глазах.

Опять же, это мы изначально создавали для себя, а потом открыли, поскольку почему бы и нет? Когда это открытый код, появляется много людей, желающих его улучшить, но также и много инструментов для этого, и *Angular* тоже это зацепило. Теперь вопрос уже не «Почему мы должны это открывать?», а «Почему бы нам это не открыть?»

LXF: Наряду с внедрением в MS (более) безумных идей об открытости, вы перевели 65 000 инженеров на одну платформу Git. Почему именно Git?

МВ: Это было сразу после моего присоединения, в начале 2010 г. Управление распределенными версиями не соответствовало принятой политике контроля версий; оно реализуется совсем иначе, чем в централизованных системах. Для Microsoft было три разных решения: «покупать, создавать или использовать открытый код». Мы могли купить такую платформу у какой-нибудь коммерческой »

компания, к чему и склонялись. Мы могли бы построить свою, что предпочли бы многие сотрудники MS, потому что позиция «это же не наше» была всё еще распространена.

И на тот момент вариантами с открытым кодом были Mercurial и Git. Вам любой сказал бы, что система управления версиями Mercurial гораздо удобнее, и в Windows она работает лучше. Тогда было принято думать именно так. А копнув вглубь, мы обнаружили, что Git применяется как сетевой протокол для контроля распределенных версий между провайдерами DVCS. Но когда мы заглянули в облако, то увидели, что протоколы Git используются там всё больше и больше.

О ХОРОШЕМ СТИВЕ БАЛЛМЕРЕ

Кое-какие открытые разработки начались именно под его началом.

Если присмотреться к сетевому протоколу Git, на самом деле это очень тонкая прокладка в каталоге `.git` в вашей файловой системе. И мы подумали: «Хм. Это интересно». Так что плюсы использования Mercurial по сравнению с Git сводились к тому, что он лучше работает в Windows. Но если перевести команду Windows на Git, это можно исправить. Тогда мы еще не знали, победит ли Git — в 2010 г. сказать было трудно; но, к счастью, наша ставка оказалась верна. Сейчас около 70% разработчиков используют Git, так что мы сделали правильный выбор.

LXF: Хорошо. Расскажите немного о новой платформе DevOps.

МВ: У нас есть одна платформа на базе Visual Studio Team Services (VSTS), это наша инженерная система, та самая, что мы продаем. Она основана на Git, и там есть отслеживание работы, сборки, релизы и всё такое — это основная инженерная система.

Инженерная система поверх Git — интересная задача сама по себе. Git действительно хорош

для открытых разработок, для чего и создавался. Но если у вас очень большая кодовая база, вы можете использовать двойное масштабирование. Например, кодовая база самого Git весит около 34 МБ. Linux — самый увесистый из проектов с открытым кодом — это 650 МБ, если вы берете самый полный вариант. Но кодовая база VSTS, над которой я работаю, — это порядка 3 Гб, и это только последняя версия, ведь создавалась она годами. Код Windows без истории — 250, может быть, ближе к 300 Гб, и это буквально только последний код. Таким образом, можно сказать, что Windows нарастает на один Linux каждые несколько недель, что, возможно, не самое лучшее решение — это

говорит скорее в пользу небольших модульных кодовых баз. Но всё дело в том, что у нас 4000 инженеров, работающих над кодовой базой, которой уже 20 лет. Даже просто свести их в одно хранилище Git было безумно трудно.

LXF: Как Git справился с такой огромной кодовой базой? Подозреваю, это был не тот случай, когда всё Просто Заработало.

МВ: Ну, мы внесли некоторые изменения на сервере, чтобы справиться с этой нагрузкой. Но мы также создали такую штуку, как Git Virtual Filesystem (GVFS), это расширение на Git с открытым исходным кодом.

LXF: С ума сойти. Особенно учитывая, что Линус писал Git как файловую систему.

МВ: Да, это файловая система с направленным ациклическим графом. И самое безумное то, что с GVFS Git перестает быть распределенным. Куча всего виртуализируется в каталоге `.git/`. Идея в том, что когда вы клонируете репозиторий Git, вместо того, чтобы сохранять всё локально, она передает его частями, по запросу. Тогда можно применять масштабирование к таким массивным репозиториям. То есть, чтобы клонировать его, вам нужно всего лишь скопировать кучку указателей.

Когда вы пытаетесь получить доступ к материалам, они динамически подгружаются, и на заднем плане происходит много хитрых операций. Но он также может получать данные и из локального кэша. Таким образом, вы можете использовать один сервер в качестве главного репозитория, с которым все синхронизируются, но когда вы делаете клон, он предусмотрительно подтягивает двоичные файлы, скажем, из вашего локального кэша в Японии. Были из-за этого некоторые странности: например, команда в Японии могла клонировать исходный код Windows быстрее, чем команда в Редмонде, потому что команде в Редмонде приходилось прошагать по территории, чтобы получить кэш, а японцам — просто добраться до сервера, который находится у них в шкафу.

LXF: Видимо, многое изменилось с 2009 г., когда вы присоединились к Microsoft. Настольный Linux тогда был тем еще раздражителем.

МВ: Я пытаюсь вспомнить, когда я впервые запустил Linux, где-то около 1998 или 1999 г. У меня был какой-то дистрибутив, нарезанный на ISO, и старый сервер в шкафу. Тогда надо было компилировать практически свое личное ядро, чтобы система могла работать.

LXF: Я попытался запустить Slackware в 1996 г. Мне это не удалось, и потом за Linux я не брался лет пять.

МВ: Вот-вот. Помнится, один приятель тогда показал мне удивительную вещь, называемую форматом файлов MP3. «Послушай, звук-то почти не дребезжит!» Это было весьма впечатляюще. Как сильно всё изменилось...

LXF: Думаю, патенты на MP3 уже истекли, поэтому вы можете кодировать и декодировать MP3 в Linux. Увы, теперь это почти никому не интересно, ведь есть более продвинутые бесплатные форматы. Но я всё еще храню 128-битные MP3-версии CD, которые нарезал во времена Pearl Jam.

МВ: Поразительно, как много изменилось с 2010 г. Github появился не раньше 2006 г., но расти начал только в 2011/2012 г. А теперь у нас есть эта огромная коллекция открытого кода и целое сообщество кодеров. Приятно видеть, как меняется мир. Всего за последние два года Microsoft здорово изменился.

LXF: Он претерпел много изменений, и я надеюсь, мы убедили некоторых наших читателей, что это больше не враг. Я помню, как получил книгу «Дорога в будущее [The Road Ahead]» Билла Гейтса на Рождество (спасибо тете). Это было в далеком 1996 г., он же просто пересоздал компанию, осознав значимость Интернета. Потом Стив Балмер [Steve Ballmer] понял, какая важная штука мобильный телефон, и была эта история с Nokia. Но это, а также выходы к швырянием стульев, вышло уже не так хорошо...

МВ: Стив действительно отлично организовал продажу. Кое-какие открытые разработки начались именно под его началом, верите вы мне или нет. Его иногда недооценивают. В смысле, есть люди, не желающие его простить за Linux, его фразу, что Linux — это рак (в 2016 г. он всё же признал, что полюбил Linux). Но многое в One Microsoft началось на его веку.

Когда же пришел Сатья, который сам инженер, мы как бы получили разрешение делать всё иначе. Не в логике «А вот мы в Microsoft делаем это только так», а так, как правильно. Я сейчас читаю его книгу, она называется *Hit Refresh*, она очень хороша. Удивительно хороша для книг такого типа. Он вдумчивый инженер. Раньше он был главой нашего подразделения Облачных и коммерческих технологий [Cloud and Enterprise], но он всегда был инженером. В Microsoft вообще сейчас рулят инженеры, был Баллмер, есть Терри Майерсон [Terry Myerson] — он всегда был таким, а сейчас отвечает за Windows. А еще есть Скотт Гатри [Scott Guthrie]... то есть в Microsoft вообще сейчас рулят инженеры, и это здорово! **LXF**



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!

СОБИРАЕМ ДИСТРИБУТИВ

**Нейт Дрейк
помогает вам
расправить крылья
и взлететь с собст-
венной версией Linux.**



Как бы мы ни любили Джастина Бибера, всё же спорно, что поп-звезда заслуживает отдельной версии Linux. Тот факт, что интернет-сообщество выдало нам “Biebian” Linux (<http://biebian.sourceforge.net>), свидетельствует как о его избрательности, так и о чувстве юмора.

Некоторым облегчением для не-Биберирующих послужит тот факт, что данная операционная система не была создана с нуля, но является вариацией супер-легковесного Purru Linux. Даже не рассчитанные на малолетних фанатов версии Linux, например, Ubuntu, основываются на более старых системах (в данном случае, Debian).

Это — преимущество жизни в мире, который подарил нам операционные системы, посвященные Джастину Биберу, Ханне Монтане и Сатане.

Свободное ядро Linux с открытым кодом позволяет существовать дистрибутивам в тысячах так называемых разновидностей [flavours].

Вас как конечного пользователя это может изрядно ошарашить, поэтому большинство новичков в Linux придерживаются хорошо известных дистрибутивов, таких как Ubuntu и Linux Mint.

Мы поможем вам выбрать дистрибутив Linux — и сделать его вашим личным.

Хотя мы от души одобряем ваше желание сделать жизнь проще, Linux не имеет таких ограничений, как Windows. Большинство дистрибутивов поддерживают ряд сред рабочего стола, разработанных ради блестящих визуальных эффектов или эффективности. Вы можете менять

аспекты вашей системы, от таких простых, как цвет значков и окон, до того, как размещаются окна на экране и как вы запускаете приложения.

В этом руководстве мы поможем вам выбрать базовый дистрибутив Linux с рабочим столом — и сделать его вашим личным. В первых нескольких разделах мы исследуем, как изменить некоторые аспекты графического интерфейса вашей текущей системы посредством установки менеджеров окон, панелей и индивидуальных тем. Если вам требуется большая управляемость своей систе-

мы, вы также можете обратиться к удобному онлайн-мастеру SUSE Studio — он позволяет создать устанавливаемый образ Linux с фоном рабочего стола, предустановленными программами и даже вашими собственными персональными файлами.

Рабочие столы

Выберите базовую систему, которую будете настраивать.

Одна из радостей, обусловленных открытостью кода Linux — возможность выбора из множества рабочих столов. Это исключительно дело персональных предпочтений, которое зависит от ваших потребностей и мощности вашего оборудования.

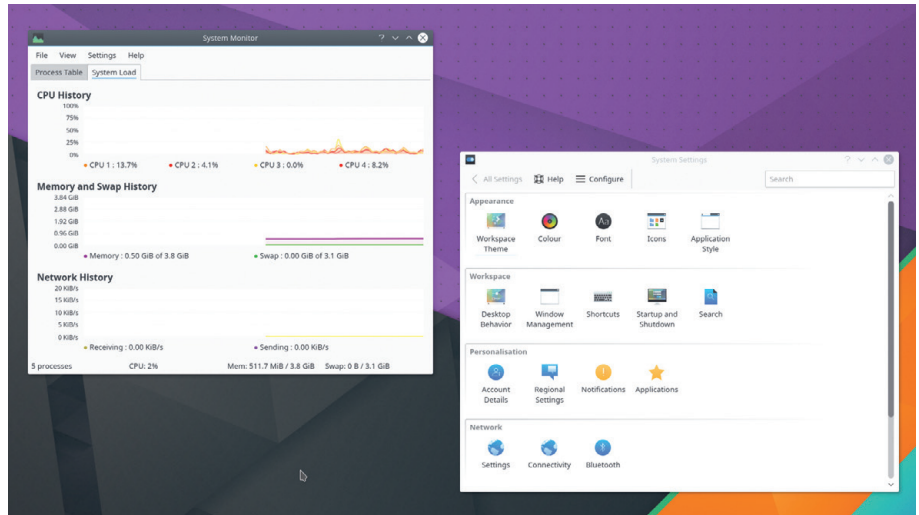
Хотя и можно установить на один компьютер несколько рабочих столов, проще выбрать дистрибутив Linux с предустановленным. Большинство операционных систем Linux также поставляется с набором приложений, оптимизированных для среды рабочего стола по умолчанию.

Чейbuntu?

До Ubuntu 17.04 (Zesty Zapus) версия по умолчанию вечно популярной операционной системы использовала рабочий стол Unity, который по-прежнему используется в LTS (Long Term Support) версиях Ubuntu. Однако Ubuntu 17.10 и последующие версии перешли на Gnome.

Gnome разработан во имя простоты использования и продуктивности. Его официальный интерфейс, Gnome Shell, предлагает категоризированное меню Applications, системное меню и часы. Поддерживается также режим быстрого обзора, позволяющий просматривать все активные в данный момент окна.

Некоторые приложения, например, текстовый редактор *AbiWord* или программа просмотра PDF *Evince*, оптимизированы для использования с Gnome, и это означает, что они будут корректно отображаться и работать с виджетами рабочего



► Системные настройки в KDE Plasma 5 поддерживают смену темы и управление эффектами рабочего стола.

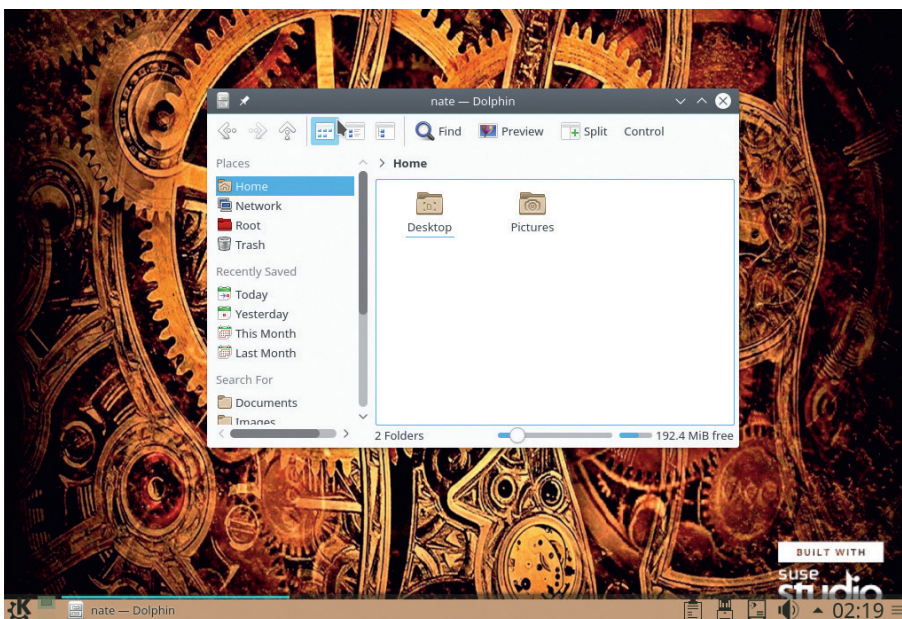
стола. Вы можете просмотреть их список на <https://wiki.gnome.org>, если хотите поэкспериментировать. Загляните на www.ubuntu.com/download/desktop, чтобы скачать Ubuntu 17.10.

Основной соперник Gnome — рабочий стол KDE Plasma, это среда выбора для таких операционных систем, как openSUSE и Kubuntu. По умолчанию KDE включает небольшую панель задач внизу экрана.

Имеется также огромное сообщество KDE, работающее в тандеме с разработчиками Plasma над созданием индивидуально настраиваемых

приложений, таких как универсальная программа просмотра документов *Okular* и *Kontakt*, пакет производительности в Ubuntu. KDE-версия самого Ubuntu называется Kubuntu и доступна на <https://kubuntu.org>.

Рабочий стол KDE Plasma также поддерживает ряд эффектов рабочего стола, например, Present Windows, который позволяет просматривать и переключать приложения, подобно функции быстрого просмотра Gnome. Более подробную информацию см. на https://userbase.kde.org/Desktop_Effects_Performance#Desktop_Effect_Performance.



► Личный Steampunk Linux автора, с индивидуальной темой; создан в SUSE Studio.

Легкий и скучный

Если вас заботит скорость, а не гламур, подумайте о выборе версии Linux с LXDE (Lightweight X11 Desktop Environment). У LXDE самое низкое потребление памяти из всех основных рабочих столов, и поэтому он идеален для более старых компьютеров. Lubuntu, версия Ubuntu с LXDE, преспокойно может работать на 512 МБ ОЗУ.

Быстрый интерфейс LXDE включает *LXLauncher*, который запускает приложения с невероятной скоростью. Менеджер файлов по умолчанию, *PC-ManFM*, столь же быстр, но довольно примитивен. LXDE также использует легковесный эмулятор терминала *LXTerminal* и стремительный *Openbox Window Manager*. Скачайте Lubuntu с <http://lubuntu.net>.

Специального упоминания заслуживает *Xfce*, тоже разрабатываемый как легковесный и быстрый, но с большей внешней привлекательностью, чем LXDE. Он поддерживает более интересные эффекты, например, тени окон и прекрасный менеджер файлов. Вариант Ubuntu с *Xfce* называется Xubuntu и доступен на <https://xubuntu.org>.

Окна и панели

Тонко настройте свою версию Linux, со своими менеджерами окон и файлов.

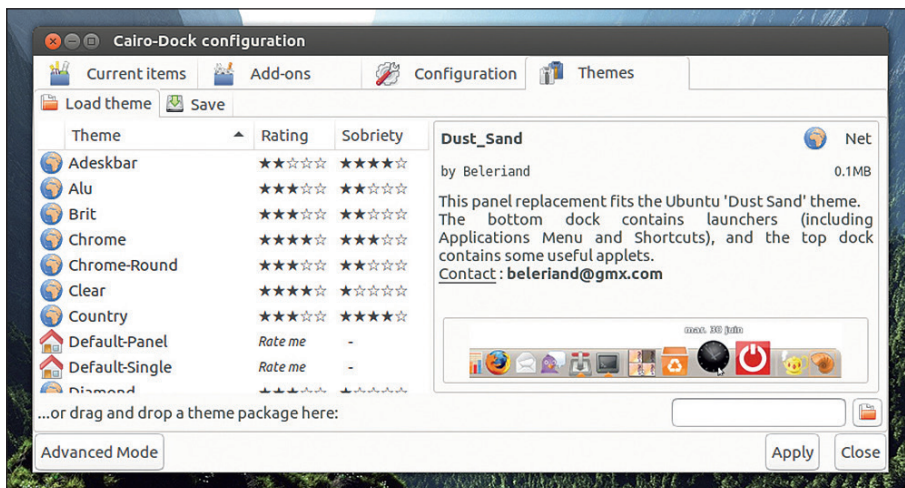
Какую бы версию Linux вы ни выбрали, почти наверняка в ней будет встроенный Менеджер окон. Они служат клиентами системе окон 'X' Windows и в целом соответствуют своему названию, поскольку определяют внешний вид и работу всех окон, используемых графическими приложениями.

Хотя вы можете полагать, что окна все на одно лицо, выбранный вами менеджер может сильно повлиять на вашу производительность. Например, стекловый менеджер окон позволяет размещать окна одно над другим, как на компьютере с Windows или Mac. А мозаичный менеджер окон сделает так, чтобы окна не перекрывали друг друга. Есть даже динамические менеджеры окон, способные переключаться между этими двумя.

Arch Linux wiki содержит исчерпывающий список разных менеджеров окон (https://wiki.archlinux.org/index.php/Window_manager). Однако не думайте, что из-за него вам придется использовать Arch.

Установка разных менеджеров окон тоже разная. Например, если вам захотелось установить легковесный менеджер окон *Openbox* в Ubuntu, то понадобится всего лишь открыть Терминал и ввести `sudo apt-get install openbox obconf`. Утилита *obconf* не является обязательной, но она помогает менять другие настройки системы, например, цвет и шрифты окон.

Панели используются для перечисления и запуска программ и местоположений на вашем компьютере. Например, рабочий стол Gnome по умолчанию имеет верхнюю панель с меню навигации на настройках Applications [Приложений], Places [Мест] и System [Системы], тогда как нижняя панель используется для переключения между



► Вот Ubuntu 16.04, где работают и Variety, и Cairo Dock. В Cairo есть встроенный инструмент настройки, сильно упрощающий выбор новых значков и тем установки.

активными окнами. Большинство панелей можно индивидуально настроить, добавив новые категории меню, а также программы запуска ваших любимых приложений.

Обычно ваши панели определяются выбранной вами средой рабочего стола, но можно установить и другие — например, *Cairo Dock* в стиле Mac (см. вверху) или быструю и легковесную *Docky*; обе доступны для установки через *Ubuntu Software*. Если вы используете Ubuntu с Unity или Gnome, можете найти больше программ запуска и доков на www.gnomelook.org/browse/cat/277.

Если вас устраивает ваш рабочий стол и вы хотите только изменить цвет или раскладку меню и панелей, на помощь вам придет Менеджер Тем

(как с ним работать, см. пошаговую инструкцию внизу стр. 47).

Острота разнообразия

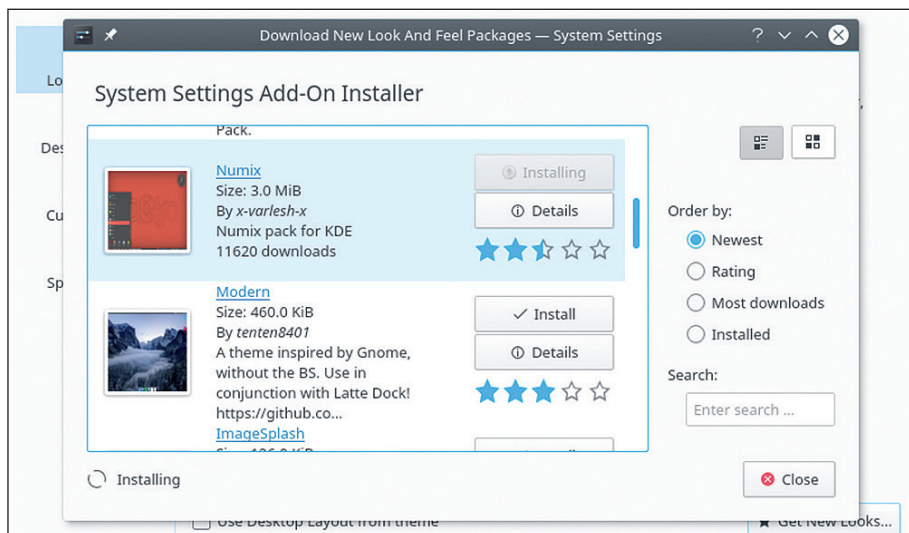
Обзаведясь окнами и панелями, уделите время выбору фона рабочего стола. Для большинства дистрибутивов на базе Ubuntu надо просто щелкнуть правой кнопкой в любом месте рабочего стола и выбрать *Change Desktop Background* [Изменить фон рабочего стола]. Вы можете выбрать из предустановленных фонов или нажать на кнопку +, чтобы добавить собственный.

Если вы считаете, что новизна этого несколько полекла, попробуйте с помощью *Ubuntu Software* установить *Variety*. Эту крошечную, но мощную программу можно настроить так, чтобы она проходила по разным фонам рабочего стола на вашем компьютере или онлайн. Изображения поступают из разных источников, например, видов Земли от НАСА, с Flickr и даже Reddit. Периодичность смены фона контролируется в пределах от нескольких секунд до 24 часов.

Будьте в теме

Большинство версий Linux поддерживают установку тем. Это файлы, способные изменять общий внешний вид и ощущение от вашей операционной системы: например, значки для приложений, цвет окон и доков, а иногда даже новые шрифты и курсоры.

Версии Linux, использующие KDE, позволяют скачивать и устанавливать новые темы и наборы значков прямо из системы. Нажмите на меню запуска и выберите *Settings > Workspace Theme* [Настройки > Темы рабочей области]. Если вам не понравится ни одна из предустановленных тем,



► В дистрибутивах с KDE можно просматривать и скачивать темы из System Settings.

нажмите на Get New Looks, чтобы просмотреть список совместимых тем в KDE Store. Выберите Install рядом с каждой из них, по завершении скачивания выберите тему и нажмите на Apply, чтобы ее использовать.

Процесс установки нового набора значков работает похоже: просто выберите Icons [Значки] вместо Workspace Themes, затем выберите или установите новые значки точно так же.

Если ваш дистрибутив Linux поставляется с Unity или Gnome, например, Ubuntu 16.04 LTS или Ubuntu 17.10 соответственно, можете скачать и установить новые темы с помощью инструмента *Gnome Tweak*. Следуйте шагам инструкции, приведенной внизу, чтобы получить подсказку. Вы можете отыскать наборы тем и значков онлайн. Два популярных сайта — <http://gnome-look.org> и <http://ubuntu-themes.org>.

Если английский не является вашим родным языком или если вы не придерживаетесь американского правописания и формата даты, вам, вероятно, придется изменить настройки вашего дистрибутива по умолчанию.

Придержитесь язык

В Ubuntu это делается из раздела Language Support [Поддержка Языка] в настройках Windows. Если во время настройки вы выбрали не тот язык, который используется любым из приложений системы (например, британский английский), Ubuntu спросит, не хотите ли вы скачать дополнительный язы-

Если у вас есть излюбленные приложения, теперь может быть самое время установить их через менеджер пакетов вашего дистрибутива. Если вы выбрали версию Linux, которую не использовали ранее, попробуйте использовать предустановленные приложения: они были оптимизированы для использования со средой рабочего стола.

Можете скачать и установить новые темы с помощью инструмента Gnome Tweak.

ковой пакет. Нажмите на Regional Formats [Региональные форматы] для выбора вашего региона, и вам сообщат о вашей валюте по умолчанию, времени и отображению чисел.

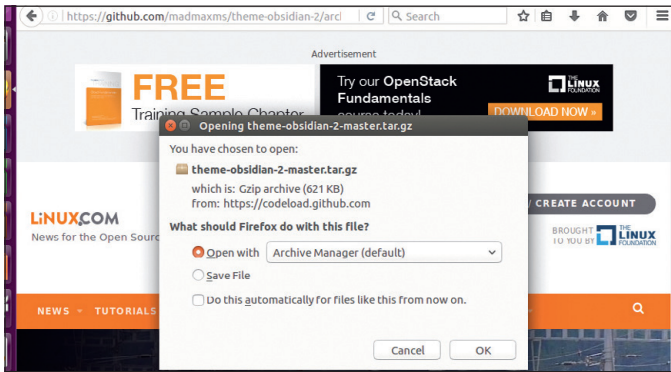
Если вы используете дистрибутив с KDE, перейдите в Settings > Language [Настройки > Язык]. Отсюда можно настроить выбранный язык. Можно также нажать на Formats [Форматы] на боковой панели, чтобы настроить свой регион.

Когда вас устроит внешний вид вашего индивидуального дистрибутива Linux, вы можете избежать прохождения процесса настройки заново с нуля, если будете регулярно делать резервные копии всего своего жесткого диска с помощью утилиты вроде *Clonezilla*, чтобы при необходимости его восстановить. Подсказку по этому процессу вы найдете на <http://www.techradar.com/how-to/how-to-clone-your-hard-drive-with-clonezilla>.

НОВЫЕ ТЕМЫ И ЗНОЧКИ

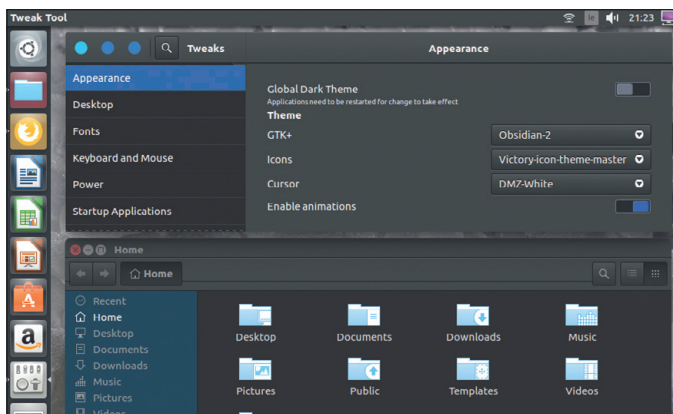
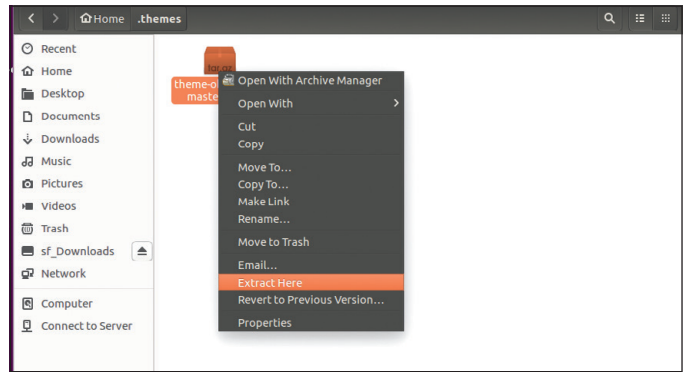
1 Найдите свою тему

Перейдите на <http://gnome-look.org>, чтобы найти выбранную вами тему. Загляните в категории GTK2 Themes и GTK3 Themes. Если вы используете Ubuntu со стандартным рабочим столом Unity, а не Gnome, некоторые из этих тем могут быть несовместимыми. Читайте раздел комментариев по каждой теме, чтобы выяснить, устанавливалась ли она ранее в Ubuntu. Чтобы установить набор значков, нажмите на раздел Icon Themes.



2 Сохраните тему

Откройте свой менеджер файлов и перейдите в свою домашнюю папку. Удерживайте Ctrl+N для отображения скрытых файлов, затем прокрутите вниз, чтобы найти папку с именем `.themes`. Если этой папки там нет, создайте ее, щелкнув правой кнопкой и выбрав New Folder [Новая Папка]. Скопируйте и вставьте скачанный файл в эту папку. Если он сжатый, нажмите на него правой кнопкой и выберите Extract Here [Распаковать здесь].



3 Загрузите свою тему

Откройте *Ubuntu Software* и введите слова 'Gnome Tweak' в строку поиска. Нажмите Install, чтобы начать скачивание приложения. Если вы используете популярную версию Unity, то значок для Tweak tool автоматически появится в программе запуска.

Откройте Tweak из программы запуска слева и выберите Appearance (с левой стороны). Нажмите на выпадающее меню рядом с GTK+ и выберите нужную тему.

Если эта тема не включает набор значков, можете их установить, повторив вышеописанные шаги, но вместо папки `.themes` сохраните файл в папку `.icons` в вашей домашней папке. Можно выбрать новый набор значков из раздела Appearance в Gnome Tweak tool, нажав на соответствующее выпадающее меню.

Волшебство дистрибутива

Создайте свою скачиваемую версию Linux с помощью онлайн-мастера SUSE.

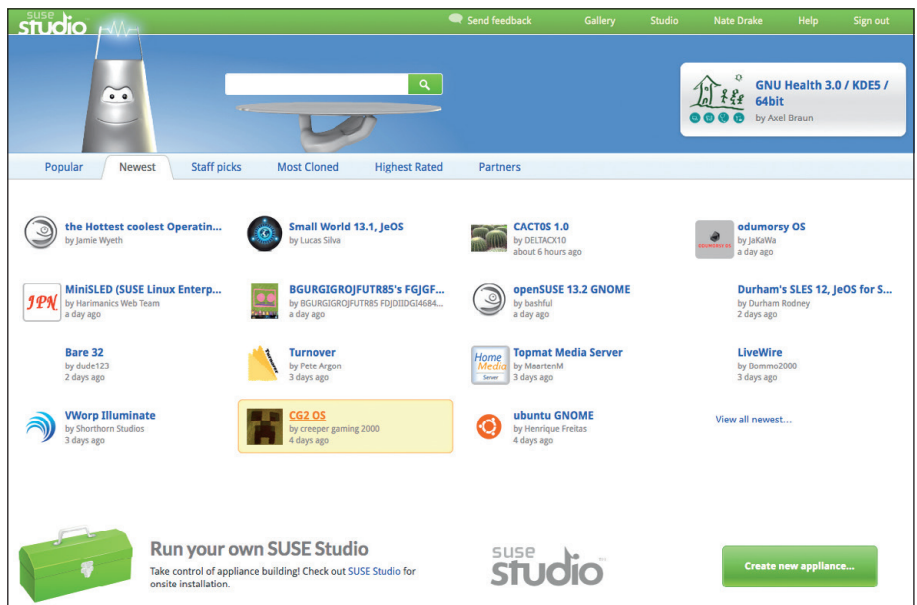
Как ни интересно методично прорабатывать разные функции Linux, было бы намного удобнее иметь программу, способную сделать за вас всю тяжелую работу.

За долгие годы появлялось немало инструментов типа *UCK (Ubuntu Customization Kit)*, с целью помочь вам создать свой собственный дистрибутив. Однако во время наших тестов ни один из них не работал с последними версиями Ubuntu.

Для тех, кто предпочитает дистрибутивы Linux на базе RPM, история будет несколько иная, поскольку добрые люди из SUSE разработали отличный онлайн-мастер для создания персонально вашей устанавливаемой версии Linux на основе openSUSE или SUSE Enterprise.

SUSE Studio, под которым именем он известен, проводит вас по процессу создания вашего собственного т. н. 'appliance [приспособления]', определяя, какие программы предустанавливать, выбирая фон рабочего стола и логотип и загружая ваши личные файлы. Если вам не хватает вдохновения, зайдите в галерею SUSE (<https://susestudio.com/browse>), где демонстрируются индивидуальные дистрибутивы SUSE, созданные другими пользователями.

Руководство на стр. 49 проведет вас по базовым шагам настройки вашего дистрибутива, вклю-



Если вам не хватает идей по индивидуальной настройке своего дистрибутива, поможет галерея SUSE Studio.

Если вы не вняли нашим зловещим предупреждениям и жаждете приступить к созданию собственной версии Linux в SUSE Studio, то вам

Пользователи Linux, которые скачивают ISO-образ своего дистрибутива, могут установить утилиту записи диска, например, *Brasero* или *K3b*, чтобы скопировать файлы на DVD. И, наконец, если на вашем целевом компьютере отсутствует DVD-привод, рассмотрите вариант скачивания *Unetbootin* с <http://unetbootin.sourceforge.net>: его можно использовать с USB-брелком, чтобы создать загрузаемый диск. **LXF**

Менеджер пакетов openSUSE отличается от используемого в дистрибутивах на базе Debian.

чая конфигурацию настроек запуска, таких как автоматический вход в систему и запуск программ и/или скриптов при загрузке.

Пробуйте до покупки

Если вы никогда не использовали RPM-дистрибутив Linux, такой как Red Hat, Fedora или openSUSE, то, возможно, захотите скачать стандартную версию, чтобы познакомиться с ней перед созданием вашего собственного индивидуального Linux.

Менеджер пакетов openSUSE отличается от используемого в дистрибутивах на базе Debian, таких как Ubuntu. Это означает, что любое руководство Linux, которое вы найдете для Ubuntu, онлайн или в *Linux Format*, в openSUSE может не работать.

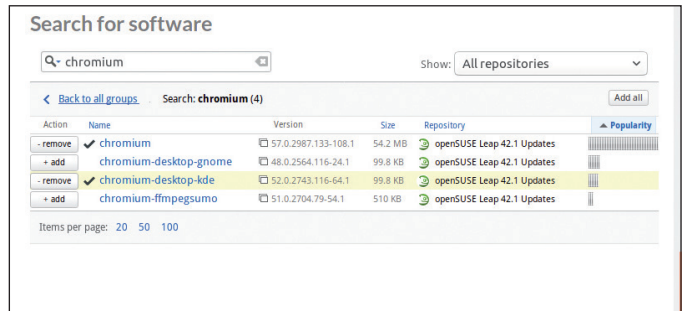
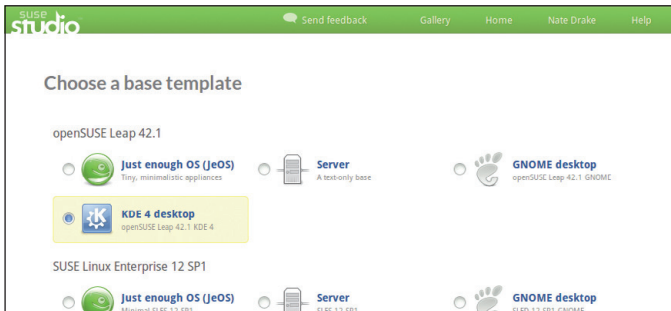
У самых популярных программ Linux, таких как *Mozilla Firefox*, есть RPM-версии, но обязательно проверьте их в разделе Software перед установкой. SUSE Studio поддерживает добавление репозитория и загрузку вручную файлов RPM, если там имеется какое-то конкретное необходимое для вас приложение.

понадобится перенести законченные файлы на носитель, например, на USB или на DVD, чтобы установить их на свой компьютер.

Когда настанет время собрать ваш личный дистрибутив Linux, советуем вам выбрать Live CD/DVD. Это позволит загрузить и протестировать вашу индивидуальную версию перед установкой. Если вы хотите сделать какие-то изменения, это намного проще сделать в SUSE Studio с помощью мастера. В порядке альтернативы можете использовать функцию Testdrive в SUSE Studio, чтобы поэкспериментировать со своим дистрибутивом в браузере.



Создайте свой дистрибутив

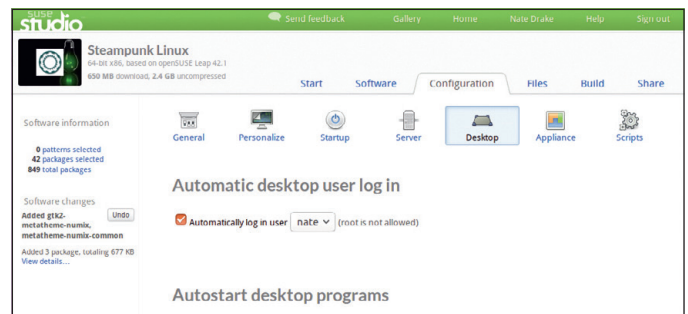
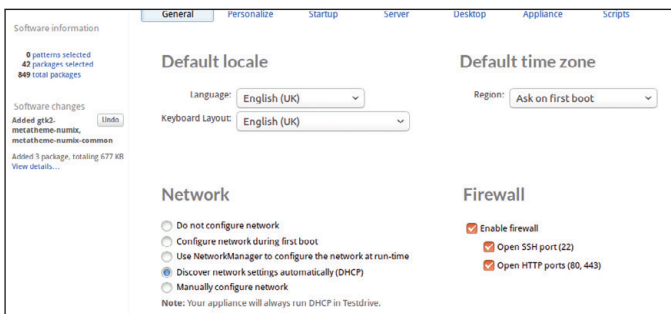


1 Выберите шаблон для установки

Зайдите на <https://susestudio.com>. Зарегистрируйтесь на этом сайте и нажмите на Create Appliance [Создать приспособление] вверху слева. Выберите базовый шаблон — либо openSUSE (рекомендуемая опция), либо SUSE Enterprise. Из рабочих столов вам предлагаются KDE Plasma или Gnome. Прокрутите до самого низа экрана, до опции Name your appliance [Задайте имя своему приспособлению], и нажмите Create. После этого SUSE Studio создаст ваше собственное «приспособление».

2 Управляйте своими программами

В Welcome нажмите на вкладку Software. Прокрутите до Search for software [Поиск программ]. Нажмите Add рядом с именем приложения, чтобы его включить. Не нашли какую-то программу? Прокрутите до Software Sources [Источники программ]. Нажмите Import new repository [Импортировать новый репозиторий] и выберите Add [Добавить]. Не можете найти свой любимый репозиторий? Нажмите Add a Repository via URL [Добавить репозиторий через URL]. Введите имя и URL репозитория в соответствующих окнах и нажмите Add Repository [Добавить репозиторий].

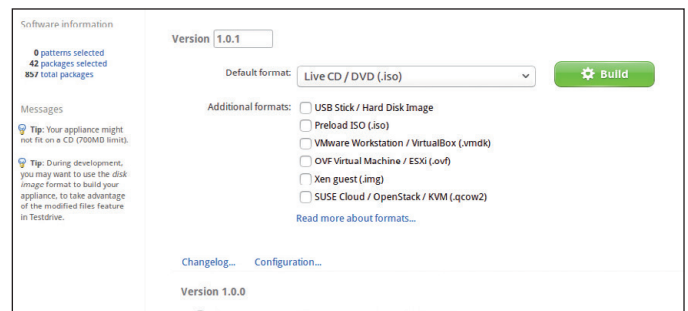
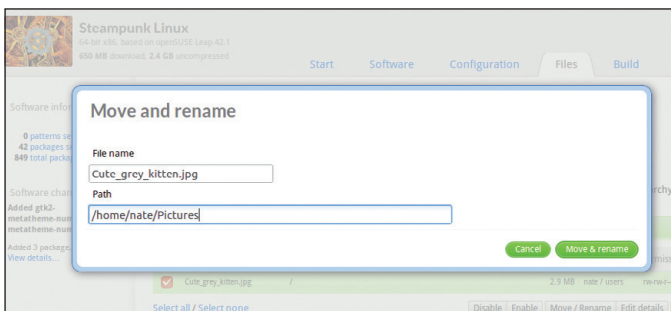


3 Настройка

Нажмите на вкладку Configuration. Установите Default locale [Язык по умолчанию] и Default Time Zone [Часовой пояс по умолчанию]. Настройки Network [Сеть] и Firewall [Брандмауэр] игнорируйте. Выбрав Users and Groups [Пользователи и группы], обновите имя пользователя и пароль. Нажмите Personalize [Личное] в Configuration [Настройке]. Нажмите Upload new logo [Загрузить новый логотип] и Upload Desktop background [Загрузить фон рабочего стола], обязательно выбрав обновленные. Прокрутите вниз, чтобы увидеть это на рабочем столе, экране приглашения и т. д.

4 Настройки запуска

Раздел Startup предусмотрен на случай, если вы не собираетесь загружаться в рабочий стол. Если раздел Server не является вашей целью, его тоже можно проигнорировать. Нажмите на Desktop, чтобы включить автоматический вход в систему и команды, которые будут запускаться при загрузке Linux. Если вы хотите при этом запускать скрипты, перейдите в раздел Scripts — там вы сможете написать код, который будет запускаться после установки или при каждой загрузке вашей операционной системы.



5 Управление файлами

Вкладка Files определяет автоматически включенные файлы. Используйте Upload File [Загрузить Файл], чтобы скопировать локальный файл, или Add from the Web (URL) [Добавить из web (URL)]. Подождите, пока SUSE Studio загрузит файлы, затем нажмите Edit details [Редактировать подробности]. Это позволит вам управлять разрешениями. Выберите свое имя пользователя в качестве владельца, затем назначьте уровень доступа для других. Нажмите Move/Rename [Переместить/Переименовать] для выбора места хранения ваших файлов — скажем, в папке Pictures.

6 Создайте образ установки

Нажмите на вкладку Build [Создать]. Установите версию в тестовом окне Version. Нажмите на меню Default format [Формат по умолчанию]; настройки по умолчанию (USB Stick/Hard Disk image [Флэшка/Образ жесткого диска]) являются самыми удобными, за ними идут (Live CD/DVD). Нажмите Configuration [Настройка], чтобы увидеть общий образ всех изменений. Если вас всё устраивает, нажмите Build [Создать], чтобы начать процесс. Не бойтесь нажимать Testdrive, чтобы всё проверить. И, наконец, выберите Download [Скачать], когда результаты вас удовлетворят.



Наш эксперт

Д-р Валентин Синицын перестал писать код для KDE, и ему полагается всего 0,16 студента Summer of Code в год. Так что у него масса времени для создания облаков на Linux и отправки статей в LXF.

Открытый код и просто открытость

Допустим, ваша организация решила открыть часть своего кода. Например, вы хотите, чтобы участники сообщества помогли вам разрабатывать и поддерживать этот код. Или ваши конкуренты открыли свой код, и пока вы не сделаете то же самое, клиенты не станут принимать вас всерьез. Какой бы ни была причина, вы создаете учетную запись на GitHub, отправляете код и нанимаете дизайнеров сделать хорошую главную страницу. И всё, да? Нет.

Что произойдет с учетной записью GitHub и с доменным именем, если ваша организация прикроет дело? Или ее купит некий противник открытого ПО? Думать, что ваша организация слишком велика, не стоит, как показали последние новости о Solaris.

Если вы любите кого-то (или что-то), отпустите это. Желая, чтобы ваш новорожденный открытый проект стал успешным, отдайте его выбранному вами фонду. Да, тогда вы не сможете контролировать всё происходящее с кодом. Вам придется передать учетную запись GitHub, доменное имя и всё прочее, требуемое для развития проекта. А может, и права на бренд. Кроме того, надо будет развивать свой проект в соответствии с принципами сообщества.

Всё это можно считать минусами. Но есть и плюсы. Вы привлечете больше участников извне, так как они будут уверены, что ваш проект сохранится. На некоторых из них вы сможете переложить финансовое бремя работы над проектом. Но главное — ваш проект будет жив до тех пор, пока он интересен обществу, независимо от интриг бизнеса.

Открытый код — не только открытость. Это еще и независимость. Пусть история с Solaris станет нам уроком.

По рецептам Д-ра Синицына

Эзотерическое системное администрирование из причудливых заворотов кишок серверной.

Закат Solaris

В 2005-м мы приветствовали Solaris в мире открытого ПО, сегодня пора сказать «Прощай».

С тех пор как Oracle купила Sun Microsystems еще в 2010 г., заметных новостей из лагеря Solaris не было. Разработка OpenSolaris вскоре прекратилась, и то же самое, казалось, угрожало коммерческой версии. В 2016-м Oracle перевела Solaris в «режим обслуживания», и теперь всё закончилось: в начале сентября 2017 г. «ведущих специалистов Solaris» (включая большую часть управляющего персонала) сократили. Что называется, «тихий EOL». Конечно, у нас по-прежнему есть Illumos, но это ответвление.

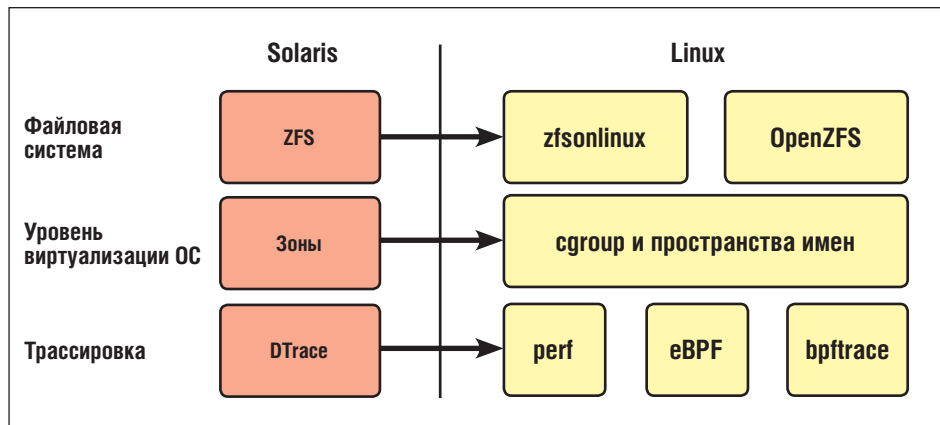
Некоторые могут возразить: это случилось годом раньше, когда разработчикам Solaris в Oracle предложили перейти на Linux или выйти вон. Но не будем обвинять Oracle. Любите вы Solaris или нет, но это собственность компании, и компания может делать с ним всё, что захочет. Разумеется, отсюда следует извлечь уроки (см. врезку слева). Тем не менее, давайте рассмотрим, что это означает для сообщества разработчиков.

Для предсказания, что организации рано или поздно начнут отказываться от систем на основе Solaris, не обязательно быть пророком. И если вы специалист по Solaris, то ваш набор знаний со временем

обесценится. Ничего хорошего тут, естественно, нет, но это повод изучить что-то новое. Брендан Грегг [Brendan Gregg], некогда перспективный специалист в Sun Microsystems, считает реальной альтернативой переход на Linux. Последние разработки в экосистеме Linux, такие как zfs on linux, контейнеры или eBPF, которые мы обсудим в этой статье, позволяют закрыть пробел между Solaris и Linux с точки зрения функций, и вы будете чувствовать себя в Linux как дома.

У Грегга огромный опыт в Solaris и Linux, и его статья о переходе с Solaris на Linux (www.brendan-gregg.com/blog/2017-09-05/solaris-to-linux-2017.html) наверняка покажется вам увлекательной. Грегг описывает основные технологии Solaris и то, как они были «повторно реализованы» в Linux, а также упоминает полезные ресурсы, например, Rosetta stone of Unix (<http://bhami.com/rosetta.html>). Взгляните на эту статью.

Впрочем, картинка лучше тысячи слов, и я попробую связать технологии Solaris и Linux на схеме (внизу). И если всё это кажется вам интересным, давайте углубимся в одну из технологий, показанных на схеме: eBPF.



На схеме показано соотношение технологий-аналогов в Solaris и Linux. Заметьте, что bpftrace здесь другая: эта утилита имитирует DTrace поверх eBPF.

eBPF: Не плачьте о DTrace

Хотите глубже проникнуть в действия своей системы? Узнайте, как в последних изменениях ядра улучшены средства анализа и устранения неполадок...

По части устранения неполадок Linux предлагает массу утилит. Исследованию процессов поможет *top* или более симпатичный *htop* (см. экранный снимок справа), для исследования виртуальной памяти — *vmstat*. Более глубокий анализ и просмотр текущих системных вызовов осуществляет *strace*; *netstat* или *ss* отображают информацию о сетевых подключениях, а для перехвата сетевых пакетов пригодится *tcpdump* или *tshark*.

Однако вам может потребоваться сделать нечто непосильное стандартным инструментам. Допустим, ваш сервис медленно отвечает на запросы. В чем проблема? Он слишком долго ожидает ввода/вывода на диске? Или проблема в сети, занятой пересылкой пакетов? С помощью eBPF вы узнаете причину наверняка (подробнее см. в учебнике из LXF229/230).

Классика BPF

По сути, в BPF нет ничего нового. Это фильтр пакетов Беркли, созданный еще в 1992 г. для BSD, и его изначальной задачей была фильтрация на уровне сокета. Он особенно полезен для сбора пакетов, и в *libpcap* реализован уже давно. По команде `tcpdump -i any tcp port 80` фильтр будет сравнивать с BPF коды операций, представляющие собой команды виртуальной машины BPF, похожие на команды Ассемблера. Их можно просмотреть с помощью параметра `-d`:

```
$ tcpdump -d -i any tcp port 80
(000) ldh [14]
(001) jeq #0x86dd jt 2 jf 8
...
```

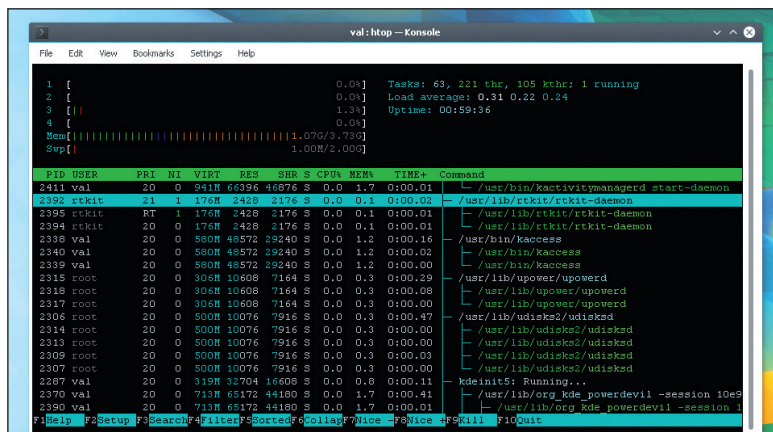
Этот простой фильтр охватывает 19 команд BPF.

Виртуальные машины сейчас не особо в моде, но BPF выделяется из толпы. Во-первых, программы BPF упрощают компиляцию JIT [точно в срок] на большинстве современных архитектур процессоров. Во-вторых, допустимы только программы BPF с гарантированным завершением. Т.е. в них запрещены циклы или переходы назад. Благодаря этому BPF достаточно безопасен, чтобы выполнять машинный код внутри ядра. Включается BPF JIT командой

```
$ echo 1 > /proc/sys/net/core/bpf_jit_enable
```

Если она не сработает, Linux вернется к внутреннему интерпретатору ядра. Существуют и реализации BPF в пространстве пользователя, например, uBPF от IOVisor (<https://github.com/iovisor/ubpf>).

eBPF кое-чем отличается от BPF. Идея состоит в том, чтобы расширить (“е”) сферу его применения за пределы фильтрации



» *htop* — это гибрид *Midnight Commander* и *top*. Пользуйтесь этой утилитой, если вам нравится приятный дизайн и вам не упомянуть все горячие клавиши *top*.

пакетов. Автор eBPF, Алексей Старовойтов [Alexei Starovoitov] из Facebook, описывает eBPF как «10% классического BPF + 70% x86 + 25% arm64 + 5% RISC». Большинство команд и регистров eBPF (а их там одиннадцать) точно соответствуют своим аппаратным аналогам. В eBPF также можно вызвать определенные функции ядра (называемые помощниками [helper]) и использовать так называемые «отображения [maps]» для обмена данными между программами eBPF и пространством пользователя. Последнее (но не по значению) — ядро не только гарантирует, что программа eBPF в конечном счете завершится, но и ограничивает в ней число команд (4096). Однако можно создавать «цепочки» из программ, вызывая их «с хвоста [tail call]». Первое условие является гарантией безопасности; последнее означает, что программы eBPF отработают быстро, и это позволяет применять их на рабочих серверах.

Привязан к ядру

При всей своей скорости и безопасности программы eBPF бесполезны, пока кто-то в системе их не вызвал. В ядре Linux немало мест, где это происходит. Обычный сценарий использования eBPF включает сети, трассировку или профилирование производительности. Это не означает, что eBPF не пригоден для других целей, например, аутентификации, авторизации и работы с учетными записями, но Алексей склонен называть это «безумием».

Начинается всё в сетях. Классический BPF выполнял фильтрацию пакетов, современный же связывается с подсистемами

»

Пример использования eBPF: Cilium

В современном мире всё больше приложений развертывается в контейнерах. Это удобно: контейнеры автономны, и незачем беспокоиться о различиях среды тестирования и рабочей среды. Контейнеры также предоставляют стандартные методы доставки. Вдобавок они легковесны и развертываются за несколько секунд. Всё это делает контейнеры перспективной целью для атаки. Ситуацию усугубляет то, что при взломе контейнера часто страдает и хост, на котором он расположен. К безопасности контейнеров следует подходить очень серьезно.

Cilium — современный механизм безопасности контейнеров. Его можно представить как *iptables*, которые работают на уровне отдельных HTTP-запросов. Благодаря этому, например, вы можете разрешить отправлять запросы GET к каталогу `/api/v1/foo` и одновременно запретить POST-доступ к каталогу `/api/v1/foo` для тех же самых участников. В основе Cilium лежит набор программ eBPF, привязываемых к уровню управления трафиком. Эти программы отвечают за применение политик, ответы ARP (таким образом снижая широковещательный

трафик), преобразование сетевых адресов (NAT) и балансировку нагрузки. Они также способны направлять трафик к прокси, который применяет политики на уровне приложения (например, запрещает доступ по протоколу POST, как в нашем примере). В последних версиях этот прокси-сервер даже может запускаться внутри ядра с шифрованием TLS.

Дополнительную информацию см. на сайте www.cilium.io или в GitHub: <https://github.com/cilium/cilium>. Код eBPF можно найти в каталоге “bpf”.

управления трафиком (TC) и eXpress Data Path (XDP). Вы можете улучшить работу с очередью трафика, хитроумнее формируя трафик, или реализовать интеллектуальные брандмауэры вроде Cilium (см. врезку внизу стр. 51).

XDP поступает в пакет сразу после того, как сетевая карта пометила его в буфер памяти, и задолго до того, как сетевой стек Linux замечает происходящее и выделяет соответствующие ресурсы ядра. Сейчас основной сценарий — это ослабление атак DDoS, но в будущем мы можем встретить механизмы обработки пакетов на основе XDP, способные конкурировать с DPDK по скорости, работая в ядре. Кроме того, некоторые сетевые карты аппаратно поддерживают разгрузку eBPF. Проще говоря, XDP открывает массу новых возможностей, но чтобы ими воспользоваться, нужно устройство с их поддержкой. Правда, в Linux также есть «универсальный XDP», который работает не слишком быстро, зато везде.

Теперь о трассировке. Благодаря трассировке вы подключаетесь к работающему ядру Linux или запущенной программе пользователя. При этом eBPF чуть больше тяготеет к трассировке на уровне ядра, так как в пространстве пользователя доступно множество других методов трассировки.

Существующая инфраструктура трассировки в ядре включает точки трассировки, которые представляют собой источники статических событий трассировки, разбросанных по коду, и события *kprobe*, которые являются динамическими (и низкос затратными), и их можно снимать с любой точки рабочих систем. Также есть события *xprobe*; они похожи на *kprobes*, но предназначены для пространства пользователя. Наконец, в *DTrace* есть *Userland Statically Defined Tracing* [Статически определяемая трассировка пространства пользователя], или USDT. Хотя *DTrace* так и не попала в Linux, события USDT в Linux использовать можно. Охватывается даже такое высокоуровневое ПО, как web-приложения: Python 3

и Node.js можно скомпилировать с поддержкой USDT. Не будем углубляться в детали, но если у вас есть некоторый опыт в Solaris, советуем взглянуть на USDT. Отличное место для старта — блог Брендана Грегга (www.brendangregg.com/blog/index.html). Идея здесь в том, что программу eBPF можно связать с любым из механизмов трассировки. Это бывает удобно для перехвата аргументов, переданных функции, или расчета времени, необходимого на возврат результата. При достаточном понимании внутренней структуры ядра вы можете измерить задержку ввода/вывода диска или подсчитать количество повторных пересылок сегментов TCP — с помощью других инструментов это сложно сделать сразу.

Это подводит нас к анализу производительности — теме обширной и сложной, и вы вправе сказать, что измерение задержки ввода/вывода диска уже связано с производительностью. Но область, где eBPF поистине сияет — это так называемое «профилирование вне процессора». Отслеживаются потоки, которые по какой-либо причине заблокированы и не выполняются. Количество аппаратных потоков на компьютере не может превышать количество ядер процессора, однако ограничений на количество заблокированных потоков нет. Другими словами, профилировщик вне процессора должен обрабатывать большой объем данных, и необходимость перебрасывать эти данные из ядра в пространство пользователя и обратно сразу же образует узкое место. Программа eBPF решает эту проблему, выполняя все сложные операции внутри ядра. Но чтобы получить преимущества eBPF, потребуется относительно новое ядро: версии не ниже 4.9.

CLang во спасение

Если вы возмущены, что для применения eBPF придется в 2018 г. писать программу на машинном коде, не бойтесь. Программирование на eBPF очень похоже на обычное программирование для ядра, за исключением того, что оно надежно и на него распространяются некоторые ограничения. Все это потому, что в LLVM теперь есть движок eBPF, и вы можете написать код на C и затем скомпилировать его в eBPF с помощью *CLang*, компилятора C/C++ для LLVM. Сегодня это стандарт для программирования eBPF, хотя существуют и другие языки высокого уровня, например, P4 (<http://p4.org>), которые позволяют компилировать программы для eBPF.

Эти программы на C немного отличаются от того, к чему вы привыкли. В них нет метода *main()* или глобальных переменных. Вспомогательные функции встроены. В коде встречаются атрибуты секций ELF, что тоже встретишь нечасто, даже в ядре. Точное представление кода зависит от двух вещей: где именно в ядре он будет использоваться и как будет загружаться. Вся инфраструктура eBPF доступна с помощью одного системного вызова *bpf(2)*. Это напоминает старые добрые времена *ioctl(2)*, но *bpf(2)* нельзя вызвать напрямую. Вместо этого *CLang* формирует объектный файл ELF (*.o*), который разбирает утилита *ip* и передает ядру.

➤ На рисунке показан *opensnoop*, запущенный в моей системе Arch Linux.

```
val@y50p ~$ sudo python /usr/share/bcc/tools/opensnoop
PID  COMM          FD ERR PATH
11586 console      16  0 /proc/11590/status
11586 console      17  0 /proc/11590/stat
11586 console      16  0 /proc/11590/cmdline
11586 console      16  0 /proc/11590/status
11586 console      16  0 /proc/11590/stat
11586 console      17  0 /proc/11590/stat
11586 console      16  0 /proc/11590/cmdline
11586 console      16  0 /proc/11590/status
11586 console      17  0 /proc/11590/stat
11586 console      16  0 /proc/11590/cmdline
2366  plasmashell   81  0 /usr/share/icons/breeze/apps/48/utilities-terminal.svg
11586 console      16  0 /proc/11590/status
11586 console      17  0 /proc/11590/stat
11586 console      16  0 /proc/11590/cmdline
2366  plasmashell   81  0 /usr/share/icons/breeze/apps/48/spectacle.svg
180   systemd-journal 19  0 /proc/2366/cgroup
180   systemd-journal 19  0 /proc/2366/cmd
180   systemd-journal 19  0 /proc/2366/cmdline
180   systemd-journal 19  0 /proc/2366/status
180   systemd-journal -1  2 /run/log/journal/607ff23ae23b475f1444e070000056e/system.journal
180   systemd-journal 19  0 /proc/2366/cgroup
180   systemd-journal 19  0 /proc/2366/cmd
180   systemd-journal 19  0 /proc/2366/cmdline
180   systemd-journal 19  0 /proc/2366/status
180   systemd-journal -1  2 /run/log/journal/607ff23ae23b475f1444e070000056e/system.journal
747   syslog-ng      66  0 /var/log/evereverything.log
11586 console      16  0 /proc/11590/status
11586 console      17  0 /proc/11590/stat
11586 console      16  0 /proc/11590/cmdline
11586 console      16  0 /proc/11590/status
11586 console      17  0 /proc/11590/stat
```

Пример использования: Linux VRF

Если вы знаете, что такое VRF, то, скорее всего, у вас есть некоторый опыт работы с коммерческими аппаратными сетевыми коммутаторами. Несколько лет назад это были Cisco, Juniper, Huawei и т. д., но сегодня «решения в белых коробках» с Linux также пытаются завоевать этот рынок.

Если вкратце, то VRF означает «Виртуальная функция маршрутизации [Virtual Routing Function]». Она позволяет иметь отдельные стеки уровня 3 (IP) в одном маршрутизаторе, а благодаря усилиям Cumulus теперь доступна в Linux. VRF — не та технология, которая потребует дома (хотя как знать?),

но для сетей уровня центра обработки данных она незаменима. Например, вы можете отдать «управляющие службы», такие как SSH, «под управление VRF», тем самым отделив их от трафика данных.

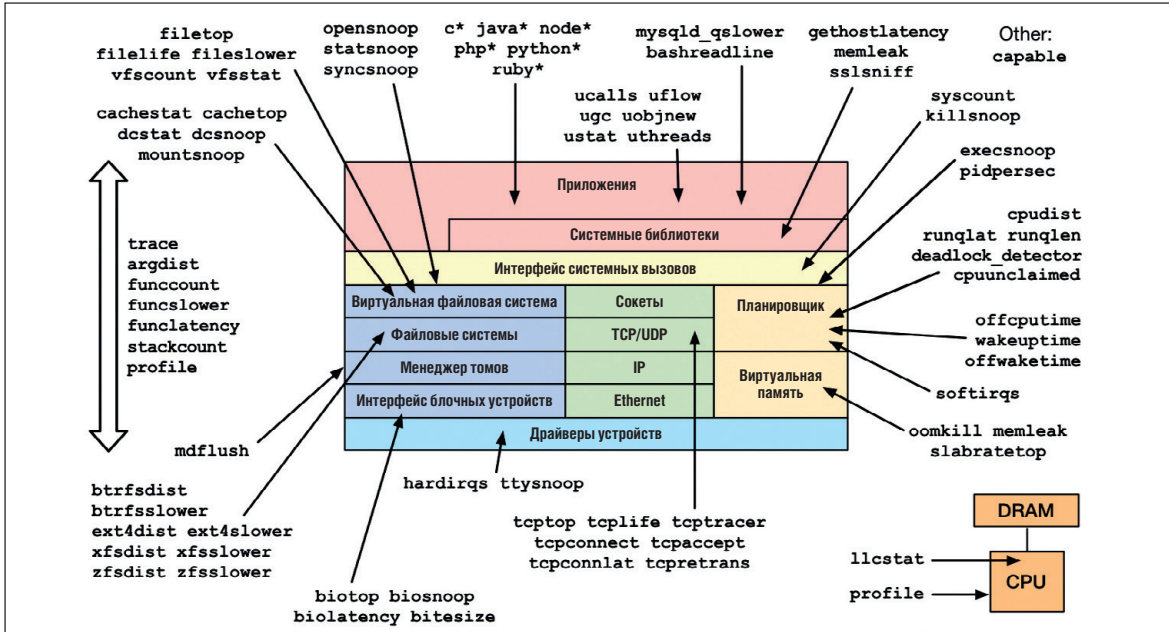
VRF может напомнить сетевое пространство имен, но на самом деле это совсем не так. Пространства имен изолируют весь сетевой стек, а VRF — только уровень IP. Поэтому при запуске команды *ip link show* вы по-прежнему видите все сетевые устройства, независимо от того, к какой VRF они подключены. Однако у VRF и пространства имен есть и кое-что общее: можно запускать некоторые

команды в контексте указанного пространства имен или указанной VRF. Для первого служит команда *ip netns exec*. Она выводит системные вызовы *setns* и *unshare* в требуемое пространство имен.

ip vrf exec похожа на нее, но она создает *cgroup* с подключенной программой eBPF. Эта программа связывает все вновь создаваемые сокеты с устройством VRF. Ее код состоит всего из шести инструкций и включен в команду *ip* прямо в виде кодов операций.

Подробности см. в следующем наборе исправлений: <https://lwn.net/Articles/708019>.

► Брендан Грегг проделал довольно много работы, чтобы определить подходящие инструменты трассировки для задания.



Поэтому нам нужны секции, и поэтому они зависят от загрузчика: различные инструменты используют различные протоколы.

eBPF делает программирование для ядра безопасным, однако чтобы получить практическую пользу, следует понимать внутреннюю структуру ядра. Это устроит не всех; но, к счастью, это не жесткое требование. Хотя для написания собственных программ BPF вам определенно нужны некоторые знания ядра, существует множество готовых программ, которые подходят почти для любых сценариев. Пожалуй, самая большая коллекция таких программ — BCC (<https://github.com/iovisor/bcc>).

Объединяем всё вместе

BCC, по самоописанию, это «инструменты для анализа ввода/вывода, сетей, мониторинга и др. в Linux на основе BPF». Если кратко, то это две вещи: набор инструментов, пригодных для решения проблем, и фреймворк Python, на котором разрабатываются новые инструменты на основе eBPF. Воспользуйтесь инструкциями в файле `INSTALL.md`, чтобы скачать пакеты для вашего дистрибутива, или скомпилируйте их сами из исходников.

Если кому интересно, BCC — это сокращение от “BPF Compiler Collection [Коллекция компилятора BPF]”. Наряду с тремя основными сферами применения, указанными выше, это средство умеет выполнять трассировку и чаще всего используется именно с этой целью. С программой поставляется хорошая документация: вы сможете изучить утилиты из состава коллекции и понять, как применять их в реальном мире и как создать собственные утилиты.

Пора взглянуть на код eBPF! Рассмотрим файл `opensnoop.py`. Эта программа показывает, какие файлы открыты в системе, в режиме реального времени. Если вы подозреваете, что кто-то забрался куда не надо, программа поможет это проверить.

Каждый инструмент BCC состоит из вложенной программы на C, которая делает всю работу, и загрузчика Python, ее запускающего. Для `opensnoop.py` программа на C выглядит примерно так:

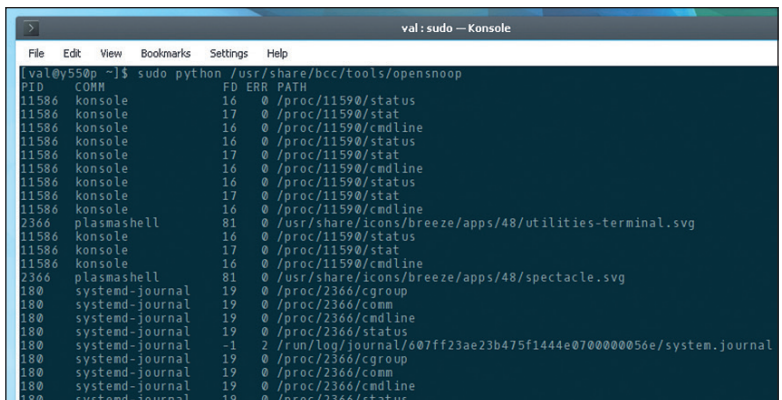
```
int trace_entry(struct pt_regs *ctx, const char __user
*filename)
{
    struct val_t val = {};
    u64 id = bpf_get_current_pid_tgid();
    if (bpf_get_current_comm(&val.comm, sizeof(val.comm)) == 0) {
        val.id = id;
        val.fname = filename;
        // store val in the map
    }
}
```

```
}
return 0;
};
```

`opensnoop.py` связывает эту функцию с `kprobe`. Такие программы eBPF получают в качестве первого аргумента указатель на аппаратные регистры оборудования. BCC немного разбавляет синтаксис, чтобы аргументы `kprobe`, которые на самом деле проходят через эти регистры, стали доступны сами по себе. В данном случае `trace_entry()` подключается к `kprobe` в начале функции `sys_open()`, которая представляет собой реализацию системного вызова `open(2)` в ядре. Это означает, что первый «декодируемый» аргумент представляет собой имя открываемого файла.

`bpf_get_current_pid_tgid()` и `bpf_get_current_comm()` — вспомогательные методы. Они контролируемо — и потому безопасно — делают различные функции ядра доступными для eBPF. В данном случае они используются для получения идентификатора процесса и имени программы (“`comm`”, в терминах ядра), которая пытается открыть файл. Программу в действии см. на скриншоте внизу.

Теперь вы знаете, что такое eBPF и как он может упростить жизнь сисадмина. Однако в BCC почти сотня инструментов, и найти нужный непросто. Если у вас возникли затруднения, на сайте Брендана Грегга (www.brendangregg.com/ebpf.html) есть «кулинарная книга», которую мы от души рекомендуем прочесть. Начните со схемы вверху и с раздела «Примеры однострочного кода». **LXF**



► Вы видите, как в `Konsole` опрашивается `/proc` для получения текущего заголовка окна и как “`systemd-journald`” заносит текущие записи в журнал, работая в фоновом режиме.



ГНУ/Линуксцентр

*Ваш поставщик свободного программного
и аппаратного обеспечения*

**Комплекты
легализации СПО**

**Дистрибутивы
GNU/Linux и СПО**
на DVD и загрузочных
флешках

**Дистрибутивы
GNU/Linux
и СПО**
с сертификатами
ФСТЭК, ФСБ
и Минобороны

**Межсетевые
экраны**
с сертификатами
ФСТЭК, ФСБ
и Минобороны



**Свободное
аппаратное
обеспечение**
Arduino, oLinuxino,
Cubieboard, Raspberry Pi,
Intel Edison, Digilent,
3D-принтеры
и робототехнические
конструкторы

**Аппаратное
обеспечение
с прошивками
на базе СПО**

**Обучающая
литература**

Атрибутика

Фирменный магазин и сервис-центр

Санкт-Петербург, пр. Медиков, 5, корп. 7

+7 812 309 06 86 | www.linuxcenter.ru



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Спираль в небеса

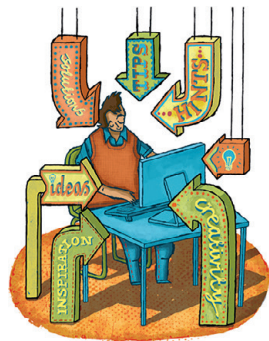
Глупые существа, ведомые спиральным инстинктом, всегда будут слепо рваться к небесам.

Лоренсом. Гуррен-Лагана, эпизод 18

До Гегеля люди сравнивали состояние общества с кругом или маятником — всё случается так, как уже бывало ранее. Это объяснимое состояние ума для аграрных обществ: урожай надо сажать, а затем убирать, и всё — инновации не нужны. Гегель же развернул круг в спираль, которая уткнулась в небеса. Конечно, можно найти сходство современного состояния дел с ситуациями, бывшими 80 лет назад, 100 лет назад и т. д. Но интереснее не сходство, а различие: то, что выводит общество из цикла и позволяет развиваться.

Вот уже восемь лет как FSF перед Рождеством публикует «Этические рекомендации по технически сложным подаркам [Ethical Tech Giving Guide]». И каждый раз хулиители устраивают гнусные пляски на форумах, со ссылками на этические рекомендации. Сходство очевидно: полный и неизменный цикл советов и взрывов негодования. А что же выведет нас в небеса, безотносительно к тому, что этика — это не закон, а лишь шаблоны поведения?

В этом году в списке, кроме рекламы рабочих станций с возможностью залить Libre Boot, есть пункты об этическом выборе коммерческих сервисов, предоставляющих электронные книги, музыку и видео без DRM-защиты. Сейчас это передний край защиты «электронных свобод». E.m.Baldin@inp.nsk.su



В этом месяце вы научитесь...

Привязывать ядра 56 Связывать команды 58

Как изловчиться распределить выполнение программ по разным ядрам процессора, показывает **Джон Найт**. Такой прием способен прилично повысить производительность.

Джон Найт нанизывает цепочки команд, передавая данные от одной утилиты к другой — эта мощная функциональность унаследована от UNIX. Корифеи былых времен не стареют...

Работать с данными 60 Ускорять сети 64

Электронные таблицы — полезнейший инструмент компьютерного арсенала **Бобби Мосса**, и он научит вас задействовать многочисленные возможности таковых. Гистограммы — штука эффектная.

Всё больше устройств обзаводятся совместимостью с GigabitEthernet 10G. **Тим Армстронг** разбирается, кому это выгодно и стоит ли сразу мчаться за новым коммутатором.

Настраивать VPN 66 Подключаться по HTTPS 70

По опыту **Ника Пирса**, общественные точки доступа Wi-Fi отнюдь не безопасны; но их можно заранее обезвредить, создав внутри них зашифрованный канал, и преспокойно пересылать свои личные данные.

Шон Конвэй гремит ключами шифрования и входит в доверие к пользователям, генерируя запросы на подпись сертификата. Причем всё — напрямую из командной строки. Перенимайте.

Двигать слайдер 74

Максиму Черепанову показалось недостаточно только включать/выключать индикаторы своего Послушного Дома: регулирование должно быть плавным!

АКАДЕМИЯ КОДИНГА



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

BASIC возвращается 78

Взгрустнув по дням молодости, **Нейт Дрейк** внезапно обнаружил, что язык BBC BASIC вполне работоспособен и сейчас — и тут же написал на нем вполне приличную игру.

Kotlin и еще раз Kotlin 82, 86

Михалис Цукалос подробно изучил всевозможные конструкции языка Kotlin, после чего приступил к созданию системных утилит и потоков, а также случайных чисел.

Taskset: Определите привязку к процессору

Джон Найт предлагает фанатам открытого ПО попробовать вкус наивысшего могущества, связывая программы с ядрами процессора по своему выбору.



Наш эксперт

Джон Найт
Когда Джон не играет в видеоигры на французском языке, его обычно можно заставить за избиением большого барабана до тестового состояния.

В новом мире многоядерных вычислений *taskset* позволяет пользователю связывать программу или «создать привязку» к определенному процессору или ядру процессора. Зачем? Иногда привязка процесса к одному ядру может повысить производительность благодаря более эффективному использованию кэша процессора. Кроме того, *taskset* позволяет распределять процессы по нескольким ядрам, предлагая полный контроль над распределением нагрузки в системе.

Эта тема перегружена жаргоном и жуткими шестнадцатеричными величинами, и вместо исчерпывающего отчета о работе *taskset* мы надеемся разбить *taskset* на ключевые и наиболее используемые компоненты. В конечном счете наша цель состоит в том, чтобы эта статья могла быть понятна любому со средней квалификацией.

Комплект ядер

Для начала запустим новую программу и привяжем ее к определенному ядру процессора:

```
$ taskset -c your-cpu-core-here your-command-here
```

Чтобы обеспечить максимально универсальный сценарий, предположим, что у вас двоядерная или больше система, и по какой-то причине вы хотите запустить *VideoLAN* на втором ядре. Команда будет выглядеть так:

```
$ taskset -c 1 vlc
```

Как уже упоминалось, *taskset* также может создать привязку к нескольким ядрам — просто вставьте запятую и припишите еще один номер ядра. В дополнение к последнему примеру: допустим, вы используете четырехъядерный или больше процессор, и хотите запустить *VideoLAN* на втором, третьем и четвертом ядрах. Команда будет выглядеть так:

```
$ taskset -c 1,2,3 vlc
```

Ряд	Пользователь	PR	NI	Вирт	RES	SHR	S	%CPU	%MEM	TIME	КОМАНДА
18642	nhoj	20	0	485848	46288	28332	S	12.5	0.6	0:01.32	chrome
14278	nhoj	20	0	296712	107996	50404	S	6.2	1.3	9:30.28	steam
18463	nhoj	20	0	1198644	204564	97328	S	6.2	2.5	0:09.82	chrome
13516	nhoj	20	0	630324	161068	91488	S	6.2	2.0	0:04.33	chrome
18606	nhoj	20	0	1180960	344124	148540	S	6.2	4.3	0:13.99	chrome
1	root	20	0	120200	4856	3112	S	0.0	0.1	0:02.17	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.41	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworke/10:0H
7	root	20	0	0	0	0	S	0.0	0.0	1:29.23	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.12	migration/0
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.15	watchdog/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.15	watchdog/1
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.08	migration/1
13	root	20	0	0	0	0	S	0.0	0.0	0:00.32	ksoftirqd/1
15	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworke/11:0H
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.15	watchdog/2

top покажет вам PID и имя программы. Здесь *taskset* проверяет привязку *Chrome*.

Просто, не так ли? Однако это относится только к еще не запущенным программам; а что делать с уже запущенными? Это несколько сложнее. Скажем, у вас есть *Firefox* на шестиядерной системе; вы недовольны тем, как он использует свои ресурсы, и хотите переместить его на последнее ядро.

Сначала вам нужно определить идентификатор процесса — или, сокращенно, PID. Обычно его можно найти в таблице вашего менеджера процессов, например, *KSysGuard*, если вы пользователь KDE. Однако большинство постоянных пользователей терминала предпочтут использовать старый добрый *top*. Для новичков — введите команду

```
$ top
```

Если у вас не запущена куча увесистых процессов, *Firefox* должен быть где-то ближе кверху, а его PID будет слева. На нашей машине *Firefox* сейчас имеет PID 25549, а вам надо подставить его PID в вашей машине в следующий пример:

```
$ taskset -cp 5 25549
```

Каковы результаты?

Совершенно неубедительные. Даже при масштабировании *DiRT Rally* на все ядра процессора и обратно на одно различия были незначительны [Ред.: — Плохой GPU?], особенно для счетчика FPS, прыгающего в зависимости от того, что находится в поле зрения игрока. Это не означает, что всё впустую: разные игры и разные программы

будут иметь разные результаты, а игры с большим объемом информации, такие как *XCOM* или *Football Manager*, могут дать интересные тесты.

В других экспериментах мы играли с *Netflix* в *Chrome*, у которого есть раздражающая привычка пропускать кадры при распределении по умолчанию по всем ядрам. Но когда мы установили

привязку к двум ядрам, воспроизведение стало вполне устойчивым. С другой стороны, услышав в видео, что PlayStation 2 предназначалась для двоядерного процессора, мы решили попробовать эмулятор *PCSX2* с двоядерной привязкой, надеясь приблизиться к оборудованию оригинала. Результат? Производительность упала почти на треть!

Инструкции для нетерпеливых

Запустите новую программу на одном выбранном вами ядре:

```
$ taskset -c 1 gparted
```

Параметр `-c` выбирает ядро (в данном случае — 1, что означает второе ядро), а `gparted` — выбранная команда.

Запустите новую программу на нескольких ядрах:

```
$ taskset -c 0,1,2,4 gparted
```

В приведенной ситуации *GParted* привязывается к первому, второму, третьему и пятому ядрам процессора.

Обратите внимание, что “0-2,4” эквивалентно синтаксису “0,1,2,4”.

Проверьте PID имеющегося процесса:

```
$ taskset -cp PID-number-here
```

Обратите внимание на недостающий номер процессора — он, по-видимому, предназначен для запуска следующей командой. Затем измените привязку имеющегося процесса:

```
$ taskset -cp 4,5 20000
```

Эта команда назначит пятое и шестое процессорные ядра уже запущенной программе, с идентификатором процесса 20000.

В этой команде, как вы, возможно, догадались, ключ `-cp` означает `c` для ядра процессора и `p` для PID; 5 — назначенное ядро процессора; а 25549 — вышеупомянутый PID. Теперь нагрузка уже запущенного *Firefox* перенесется на ядро 5. Привязку можно переназначать сколько угодно раз, а также на сколько угодно ядер.

Вы можете спросить, а как проверить первоначальную привязку? Ну, это легко. Просто в последней команде опустите номер ядра:

```
$ taskset -cp 25549
```

```
pid 25549's current affinity list: 0-5
```

Устаревшие приложения

Устаревшие 32-битные приложения являются отличным примером программ, способных извлечь выгоду из указанной привязки процессора — особенно старые приложения Windows под *WINE*. Плохо запрограммированные приложения переводились на 64-разрядную платформу не лучшим образом. Мир GNU, как правило, довольно хорошо разобрался в проблемах миграции с 32 на 64 разряда, но если у вас есть старая программа, которая работает с причудами, привязка к одному ядру процессора может решить проблему.

Что касается мира Windows и, более конкретно, игр, которые стоит попробовать под Linux с *WINE*, есть много примеров 32-разрядных приложений, плохо работающих при привязке по умолчанию. *The Saboteur* — особенно яркий пример плохо кодированной игры.

Выпущенная еще в 2009 г., эта небрежно портированная на PS3 игра поставит ваш новый компьютер на колени. Однако просто установите перед запуском привязку к одному ядру, и разница настолько значительна, что простительно вообразить существенное обновление оборудования.

Для желающих попробовать *The Saboteur* мы сначала установили его в удобный для терминала каталог (без пробелов и с коротким путем) и отредактировали команду запуска по значку *Exec* на рабочем столе следующим текстом, используя шестое ядро процессора:

```
Exec= taskset -c 5 wine /home/nhoj/The_Saboteur/Saboteur.exe
```

Отлично, теперь, со знанием основ, самое время для...

...чего-то более амбициозного

Steam удобен и превосходен для игр, но алчен до ресурсов! Неужели эти издержки будут портить любую навороченную современную игру? Часто говорят, что лишние ядра редко используются, поэтому, возможно, распределение ресурсов по нашему желанию было бы интересным экспериментом.

Тем, кто хочет повторить на своем ПК: наша машина шести-ядерная, но если у вас другая — что вполне возможно — то подставляйте свое количество ядер и распределите привязку так, как считаете нужным.

На нашей машине *Steam* распределялся по ядрам от 0 до 5, а игра, которую мы запускали — *DiRT Rally*. Чтобы убедиться, что *Steam* имеет всё необходимое, мы дадим ему первые два ядра:

0 и 1; а тестируемая игра получит оставшиеся ядра: 2–5. Если повезет, обе программы будут разделены, а исполнение игры будет более плавным.

Для этого сначала надо запустить *Steam* в терминале с выбранной привязкой. Затем запустите игру в *Steam*, временно оставьте ее, затем найдите ее PID и используйте PID, чтобы переназначить привязку игры.

Сперва запустите *Steam* на ядрах 0 и 1:

```
$ taskset -c 0,1 steam
```

Теперь, когда *Steam* работает, пришло время начать игру. Как уже упоминалось, мы используем *DiRT Rally*, но вы берите любую понравившуюся вам игру, если она объемистая и жадная до процессора. Вы начнете игру, а затем найдете ее PID, но возникает вопрос: как временно покинуть игру для ввода команд и вернуться в нее?

Можно, конечно, играть в окне, но веселого тут мало! Один из способов — рискнуть через `Alt+Tab` выйти из игры и надеяться, что она не упадет. Но если вы хотите быть крутым, более эффективный способ состоит в том, чтобы выйти в системный терминал, запустить команды оттуда, а затем в три пальца вернуться в X.

Итак, чтобы выйти в системный терминал, нажмите `Ctrl+Alt+F1`. Введите свое имя пользователя и пароль, и снова просто запустите *top* вот так:

```
$ top
```

Ваша большая прожорливая игра должна быть в верхней части списка — запишите ее PID слева (в данном случае это 12144) и выйдете из *top* по кнопке `q`. Теперь, чтобы установить *DiRT Rally* на использование ядер с 2 по 5, мы используем следующий синтаксис:

```
$ taskset -cp 2-5 12144
```

Однако с тем же успехом можно использовать синтаксис

```
$ taskset -cp 2,3,4,5 12144
```

Это может показаться излишним, но бывает удобно, если вы собираетесь повторить команду и захотите вырезать одно из чисел посередине — например, “2,3,5”.

Теперь, чтобы вернуться в игру, повторно войдите в X, нажав `Ctrl+Alt+F7`, и если это не сработает, то `Ctrl+Alt+F8` — это зависит от того, как настроен ваш дистрибутив.

И — да, для тех, кто вас видит, вы отныне круты. **LXF**

Скорая помощь

Схема нумерации процессоров может сначала ввести в заблуждение новичков Linux, но Linux использует систему нумерации Unix, где отсчет начинается с 0, а не 1. Таким образом, для четырехъядерной конфигурации ядра нумеруются 0 1 2 3, а не 1 2 3 4.



► *The Saboteur* — великолепная игра, которая швырнет современный ПК на колени. Однако установите привязку к одному ядру, и он заработает как новый.

Каналы: Свяжем команды

Джон Найт объясняет, что делает этот таинственный персонаж: `|`. Надеемся, вам будет весело, а заодно выполним задуманное...



Наш эксперт

Джон Найт
Когда он не играет в видеоигры на французском языке, его обычно можно застать за избиением большого барабана до тестового состояния.

Начинающие пользователи, возможно, недоумевают: что это за символ — `|`? А это канал. Хотя о нем и не часто упоминается, но это одна из самых крутых штук в Unix. Итак, давайте заново узнаем, что такое каналы.

Каналы в 1973 г. изобрел Дуглас Макилрой [Douglas McIlroy] из Bell Labs. Они подключают одну программу к другой, чтобы объединить функциональность. Не углубляясь в технические детали, основные утилиты систем Unix предназначены для использования в стандартной потоковой системе: программы имеют потоки STDIN (стандартный ввод) и STDOUT (стандартный вывод) и могут принимать выходной поток предыдущей программы, подключая их через канал к своему входному потоку.

Это дает возможность строить цепочки из утилит: каналы позволяют создавать совершенно новые устройства и функциональности, просто стыкуя существующие инструменты вместе. Кроме того, в эпоху, когда был дорог каждый килобайт — а то и каждый байт — такой конвейер позволял программистам забыть мороку с временными файлами и создавать программы с гораздо более эффективным потреблением памяти.

Безумная корова

Начнем с простого примера, а затем перейдем к чему-нибудь более полезному.

Для затравки возьмем старинную команду `fortune`, воспроизводящую случайную цитату или что-нибудь неопределенно-забавное. Попробуйте, если хотите:

```
$ fortune
```

Чтобы создать свой первый канал, вы можете подключить вывод `fortune` к чему-то другому — в этом случае давайте используем `cowsay`. Это еще одна забава, вшитая в терминал, которая выводит облачко с текстом для ASCII-рисунка коровы. `Cowsay` обычно полагается на того, кто после этой команды введет текст, но также может брать текст из канала. Итак, давайте передадим ему что-нибудь от `fortune`:

```
$ fortune | cowsay
```

Для большей своеобразности и для дальнейшей практики со структурой команд, попробуйте заменить `fortune` командой `ls`. Это даст корове список содержимого вашего текущего каталога (вероятно, вашей домашней папки):

```
$ ls | cowsay
```

`cowsay -f tux` “так полезней”

Лучшей помощью и в работе, и в устранении неполадок будет непосредственное применение, поэтому давайте сразу к делу — рассмотрим некоторые примеры в этих областях.

В частности, распространенным сценарием на новых системах является поиск имени команды, и каждый, кто несколько лет пользовался Linux, знает, что для этого придется прочесывать `/usr/bin` в поисках подсказок. Трудность этой задачи заключается в том, что папка `/usr/bin` чрезвычайно велика; особенно когда вам не повезло застрять только с системным терминалом и без графического интерфейса, когда нельзя прокручивать вывод! И как же нам прокручивать эту мешанину? Один из способов — вывести вывод в `less`, например:

```
$ ls /usr/bin | less
```

Для новых пользователей, которые не в курсе: команда `ls` отображает содержимое каталога, а `/usr/bin` — основной каталог системы для хранения двоичных файлов программ. Однако этот каталог нередко содержит сотни, если не тысячи файлов. Команда `less` обычно используется для чтения текстовых файлов в терминале, и ее главное функциональное преимущество состоит в возможности прокрутки вверх и вниз. Добавляя `less` к команде `ls` выше, вы переориентируете `ls` на нечто новое, разрешив пользователю прокручивать вверх и вниз во время просмотра файлов в терминале.

Имея в виду концепцию использования `less` вместе с каналами, давайте применим ее к другому полезному реальному сценарию. Полную картину того, насколько хорошо ваше оборудование работает, предоставляет команда `dmesg`. Однако, как и в предыдущем примере, вывод ее огромен, и в итоге вы видите только последнюю

Поманим морковкой...

Давайте введем в этот канал морковку! Мы не стали помещать это в основную статью, чтобы не слишком всё усложнять, а для тех читателей, которым всё понятно и они хотят узнать больше — читайте дальше.

До сих пор мы использовали каналы в основном для вывода информации на экран, но если вы хотите выводить данные прямо в текстовый файл, морковки вам помогут. Морковка работает,

просто добавляя знак `>`, за которым следует имя выходного файла.

Взяв команду аппаратной ошибки Брайана, ее можно изменить так, чтобы она выводила все ошибки в текстовый файл, который вы потом можете отправить почтой, например:

```
$ dmesg | egrep -i 'error|fault|failed' | awk '{sub(/.*[0-9]/, ""); print}' | sort | uniq -c | sort -n > system-errors.txt
```

Обратите внимание, что одна морковка сотрет всё содержимое файла, если тот ранее существовал, и запишет новое. Если вы хотите добавить информацию в существующий файл, а не перезаписать его, можете использовать «двойную морковку», т.е. просто два символа `>>`. Это особенно полезно для слияния большого количества текстовых файлов в один общий, и может даже использоваться для сшивания нескольких видеофайлов!

Первый скрипт spell

Примерно в конце 1973 — начале 1974 г. необходимость заставила Стива Джонсона взять старый Teletype Model 37 и основать на нем существующую программу *typo* — она хорошо находила общие типографские ошибки, но была сильно ограничена в функциональности. Где-то внутри Bell Labs появилась магнитная лента, содержащая миллион английских слов, которая была урезана до основного компонента файла, содержащего самые важные слова: это было в 2004 г., поэтому файл назван

w2004. С некоторыми продуманными экспериментами в английском синтаксисе, новой системой каналов в Unix и здравым воображением Стив создал первый настоящий прототип проверки орфографии. Окончательно канал выглядел следующим образом:

- 1 ввод текста => список слов
- 2 сортировка
- 3 отбрасывание дублей
- 4 удаление слов также в w2004
- 5 изменение слова

6 применение правил суффикса (теперь правил префикса)

7 снова изменение слова

8 удаление слова также и в словаре

...а на выходе был список слов, вероятно, неправильно написанных. С годами скрипт много раз менялся и улучшался, но это дедушка всех программ проверки орфографии, от которого они все и произошли. Если боги к Unix благосклонны, авось у кого-то найдется оригинал!

страницу. Но добавление прокрутки упрощает дело — снова переведем вывод в *less*:

```
$ dmesg | less
```

Пока мы объединяли через канал только два компонента. Но не думайте, что этим всё и ограничивается: можно использовать столько каналов, сколько хотите, подключив столько команд, сколько пожелаете. Не вдаваясь в подробности, немного поднимем ставки, добавив в эту цепочку еще одну команду и еще один канал.

Предположим, что вы что-то ищете с помощью *dmesg* и хотите изменить порядок его вывода. Как и в предыдущем примере, мы собираемся сохранить компоненты *dmesg* и *less*; однако, чтобы обратить вывод, введем в середину цепочки команду *sort*, например:

```
$ dmesg | sort -r | less
```

Теперь вы можете прочитать вывод *dmesg* в обратном хронологическом порядке. Команда *sort* — это еще одна базовая системная команда, а *-r* — это параметр, который велит *sort* упорядочить вывод в обратном порядке.

Затеем показуху

Пока что всё полезно — и довольно скучно. Мы опросили сообщество OSS, как сделать с помощью каналов нечто более амбициозное. Наш старый товарищ Жан-Батист из VideoLAN указал нам на следующее:

Слушать радио в *VLC*:

```
$ curl http://insert-streaming-url-here | vlc -
```

И для отладки модуля *VLC*:

```
$ vlc -vvv | grep <module-name>
```

В поисках чего-то действительно особого мы имели честь побеседовать с двумя из соавторов Unix: Стивом Джонсоном и Брайаном Керниганом. Стив Джонсон [Steve Johnson] был автором *uacc*, *spell* (подробнее об этом позже), *lint* и *Portable C Compiler*. Брайан Керниган [Brian Kernighan] — соавтор языков AWK и AMPL и буквально написал книгу *The C Programming Language*. И, вероятно, самое крутое из всего: именно Брайан придумал термин Unix!

Мы спросили Брайана, какие интересные штуки он любит делать с каналами, и он ответил: «Мои любимые примеры каналов, как правило, примерно такие»:

```
$ awk 'get something' | sort | uniq -c | sort -n
```

Брайан также снабдил нас чрезвычайно полезным скриптом:

```
$ dmesg | egrep -i 'error|fault|failed' | awk '{sub(/.*[0-9]/, ""); print}' | sort | uniq -c | sort -n
```

Этот шестикомпонентный канал принимает вывод *dmesg*, который мы рассматривали раньше; фильтрует его по конкретным выражениям: *error*, *fault* и *failed*; а затем сортирует его так, чтобы мы получили аккуратное резюме всех проблем с оборудованием, возникающих в системе.

Любому IT-специалисту удобно держать эту команду под рукой на случай устранения неполадок у компьютеров по соседству. Попробуйте загрузить liveCD и ввести ее в терминал. Без учета времени загрузки, процесс, подобный этому, превращает отслеживание аппаратной ошибки в десятисекундную задачу.

Брайан особенно хотел обратить внимание на скрипт Стива Джонсона, написанный еще в первые дни Unix. Своей программой *spell* Стив, по сути, создал первый прототип проверки орфографии, объединив несколько файлов и сделав сравнения, используя продуманную цепочку каналов. Подробности см. во врезке вверху.

Мы надеялись увековечить ее первоначальный код в архивной печатной форме, но в ходе нашей переписки Стив сказал нам, что покидая Bell Labs, он ничего с собой не взял, а программа *spell* с годами изменялась и развивалась. Дуглас Макилрой упомянул, что возможный источник может быть в пятом издании *Research*. Но в противном случае, если кто-то из читателей поможет нам отследить исходный скрипт — возможно, где-нибудь на старой дискете или ленте — мы с радостью сохранили бы его раз и навсегда в печатной форме.

С каналами разобрались, ребята

Ну вот мы и рассмотрели каналы, дорогие читатели. Мы надеемся, что каким-то образом вдохновили вас использовать эту мощную функциональность, лежащую в основе систем Unix. В любом случае, когда у вас в следующий раз засвербит желание покодировать, вы, вместо того чтобы попусту искать какой-то новый инструмент, сможете потратить несколько минут, соединяя существующие основные утилиты каналами для создания чего-то нового, способного пригодиться всем нам. **LXF**

```

\      ^__^
\      (oo)\_______
           ||----w |
           ||         ||
nhoj@nhoj-desktop ~ $ ls | cowsay

 20170917-chelsea-arsenal-1-eng-720p.mkv
 audacity-projects-mint-18 Desktop
 Documents Downloads dwhelper franbis
 GOG Games kdenlive lms Michel Thomas
 Complete V3 mnt movies mp3 mpegetc
 Music musostuff My Games Pictures
 pop.ppst Public Release.key
 rosetta-stone Sports Interactive src
 system-errors.txt temp Templates
 temp-torrents test.txt The_Saboteur
 torrents tv-shows Videos work

\      ^__^
\      (oo)\_______
           ||----w |
           ||         ||
nhoj@nhoj-desktop ~ $

```

Скорая помощь

Каналы могут использоваться для передачи всех видов данных. Коль скоро программа соответствует системе потоков Unix STDIN, STDOUT и STDERR, каналы могут передавать практически всё: текст, аудио, видео... что угодно!

► Нечто вроде *cowsay*, возможно, не очень полезно, но это хороший способ демонстрации для своих приятелей, чтобы те сразу запомнили концепцию каналов.

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

LibreOffice: Изучаем таблицы

Тех, кто хочет разобраться в сводных таблицах или построении гистограмм, Бобби Мосс мигом превратит в ниндзя обработки данных.



Наш эксперт

Бобби Мосс разрабатывает облачные микросервисы для глобальной ИТ-консалтинговой компании и ее клиентов. В свободное время он работает над проектами свободного программного обеспечения, возится со старым оборудованием и пишет всякие слова!



Следите ли вы за своими финансами, коллекцией игр или прогрессом любимой спортивной команды, электронные таблицы — это ужасно полезный инструмент для вашего компьютерного арсенала. С появлением *VisiCalc* и распространением *Microsoft Excel* эти мощные инструменты созрели для включения целого ряда функций, которые мы в последнее время обычно видим в приложениях для обработки текста, таких как мастера автоматического форматирования и инструменты для создания элегантных отчетов.

Родительским проектом полнофункционального офисного пакета в Linux был *OpenOffice.org*. Мы выбрали его ответвление, *LibreOffice*, на том основании, что в настоящее время он поставляется по умолчанию с большинством основных дистрибутивов Linux и, пожалуй, лучше поддерживается.

Теоретически вы можете использовать этот учебник для обоих пакетов, но интерфейс пользователя *LibreOffice* сильно изменился с 5-й версии, и формат файла по умолчанию у него теперь Extended ODF, поэтому пользователи *OpenOffice.org* могут найти много нового.

Здесь мы рассмотрим основные принципы применения электронных таблиц *LibreOffice*, затем перейдем к способам автоматизации ввода данных, созданию сводных таблиц и привлекательных графиков. Оно, может, и менее интересно, чем развернуть собственный сервис облачных хранилищ в AWS (это в следующем месяце), но, надеюсь, наши советы высвободят вам добавочное время для досуга.

Вне сетки

Первое, что вы увидите при открытии модуля *Calc LibreOffice* — это гигантская сетка с буквами вверху и цифрами сбоку. Пользователи

Excel и *Gnumeric* почувствуют себя как дома: всё, как в тех пакетах. Выбрав ячейку, вы увидите ее адрес, выделенный в верхнем левом углу экрана.

Вы должны иметь возможность вводить числа и текст непосредственно в любую ячейку. Желая обратиться к содержимому одной ячейки из другой ячейки, просто поставьте перед адресом ячейки знак равенства. Можно также выполнять арифметические операции, такие как $=(A1+B3)*2$, которые в данном случае складывают A1 и B3 и умножают результат на 2.

Чтобы выполнить более сложные вычисления, такие как получение результата на основе значения в ячейке, подсчет списка значений или сложение ряда чисел, можно использовать встроенную функцию. Полный список доступных функций и их назначение можно посмотреть, выбрав Вставка > Функция.

В ячейках могут быть данные определенного типа, такие как валюта, число или текст. Хотя *Calc*, как правило, достаточно умен, чтобы определить тип вводимых в таблицу данных, тип можно определить вручную, выбрав ячейку или диапазон ячеек и нажав $Ctrl+1$ или щелкнув правой кнопкой мыши на ячейке и выбрав Формат ячеек. В этом же меню можно установить цвет фона и изменить шрифт, как вам нравится.

Если конкретные адреса ячеек сложно запомнить (например, при выборе диапазона вроде 'A1:A5', состоящего из пяти ячеек, расположенных подряд в столбце A), то для них можно установить собственное имя. Просто выделите нужный диапазон, затем выберите Лист > Именованные диапазоны и выражения и Задать название и диапазон или выражение > Имя. С этого момента вы можете

Price Manifest					
Store Location	Item	Salesperson	Wholesale Price	Quantity	
London	Keyboard Type A	Smith	£16.03		
Manchester	17" LCD Monitor	Jones	£84.75		
London	Keyboard Type B	Bloggs	£32.53		
Bristol	Motherboard Type E	Smith	£92.38		
Manchester	Keyboard Type A	Jones	£17.25		
Swindon	17" LCD Monitor	Jones	£98.75		
Leeds	Motherboard Type D	Jones	£94.76		
Swindon	Keyboard Type A	Bloggs	£14.99		
Manchester	Keyboard Type B	Jones	£14.50		
Cardiff	Android Tablet A	Smith	£142.54		
London	Keyboard Type A	Smith	£15.75		
Glasgow	17" LCD Monitor	Jones	£103.45		
Glasgow	Android Tablet A	Bloggs	£175.99		
Swindon	Motherboard Type E	Smith	£95.26		
Bristol	Keyboard Type A	Jones	£13.50		
Manchester	Motherboard Type D	Bloggs	£76.99		
London	17" LCD Monitor	Jones	£98.00		

▶ Применяя форматирование ячейки, слияние и центрирование, границы таблицы и разные типы данных, можно создавать полезные (и годные для печати) таблицы.

Скорая помощь

Если в начале формулы вы поставите апостроф, она будет рассматриваться как обычный текст. Например, $'=1+2$ будет отображаться как $"=1+2"$, а не как результат — $"3"$.

во всех формулах и на каждом мастере и панели инструментов ссылаться на свой диапазон как '=мойдиапазон'.

Также можно настроить в вашей рабочей книге отдельные электронные таблицы. Вы увидите вкладки в нижней части окна с именами Лист1 и Лист2. Щелкнув по ним правой кнопкой мыши, можно изменить их имена и цвета фона. Листы также можно перепорядочить нажатием и перетаскиванием. Можно добавлять новые и удалять листы, а если вам надо сослаться на ячейку на другом листе, пользуйтесь формулой, например, '=Лист2.A1'.

Как и в текстовых процессорах, в Calc вы найдете традиционные сочетания клавиш, такие как Ctrl+X, Ctrl+C и Ctrl+V для вырезания, копирования и вставки. (Нажатие Enter удаляет анимированную линию вокруг скопированного диапазона.) Также можно форматировать текст ячейки жирным шрифтом, курсивом и подчеркиванием с помощью сочетаний клавиш Ctrl+B, Ctrl+I и Ctrl+U соответственно.

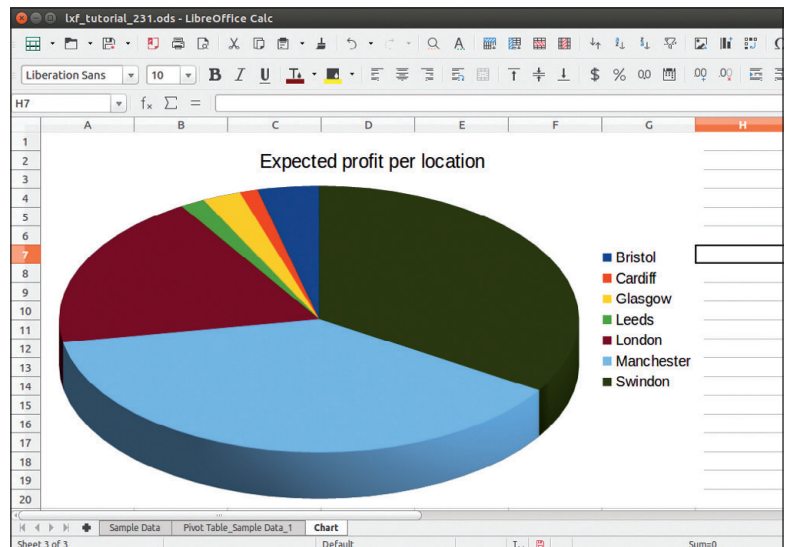
Обнаружив, что данные не помещаются в одной ячейке, вы можете дважды щелкнуть по правой вертикали, и столбец автоматически расширится в соответствии с содержимым. Если вы не хотите изменять размер столбца, поскольку ниже есть данные, формат которых вы хотите сохранить, можете выбрать две соседние ячейки, затем щелкнуть правой кнопкой мыши и выбрать Объединить ячейки.

В порядке последнего примечания к основным принципам заметим, что при нажатии и перетаскивании нижнего правого угла ячейки возникает интересное явление. Если это текстовое поле, содержимое будет дублироваться, а если это число, то Calc обычно начинает увеличивать значение на 1.

Работа с данными

Итак, основы мы рассмотрели; теперь имеет смысл подумать о представлении и обработке данных в электронных таблицах. Чтобы вам не вводить кропотливо собственные данные, мы включили два образца таблиц на диск этого месяца, чтобы вы могли с ними пробовать и экспериментировать. Они содержат условный список частей компьютера, купленных разными сотрудниками для воображаемого сетевого магазина.

Первым делом вам захочется форматировать таблицу данных, чтобы она выглядела приятнее. LibreOffice охотно сделает это за вас, если вы выберете весь диапазон ячеек и перейдете в **Формат > Стили автоформата**. А если вы предпочитаете делать это сами, можете задать границы по соответствующему значку на панели инструментов или использовать комбинацию клавиш Ctrl+1 и посмотреть на вкладке **Обрамление**. Цвет фона ячеек меняют примерно так же.



» LibreOffice Calc предлагает массу вариантов настройки, пригодных для создания диаграммы любого типа, будь то столбчатая диаграмма, круговая диаграмма, диаграмма рассеяния или гистограмма.

Вы заметите, что данные в электронной таблице примеров довольно беспорядочны. Разумеется, имеет смысл упорядочить и отфильтровать их... Упорядочить можно, выбрав **Данные > Сортировка**. Здесь можно выбрать, по каким столбцам вы хотите сортировать и в каком порядке. Если вы включили в свой диапазон заголовки столбцов, это должно сделать вашу жизнь немного легче, поскольку они будут отображаться вместо букв столбца!

Что касается фильтрации, можно использовать инструмент Автофильтр из того же меню **Данные**. Это даст вам возможность выбрать и отменить выбор определенных значений в столбце; строки, к которым они относятся, будут динамически скрыты и показаны соответственно. В случае с образцом документа вы могли бы, например, решить показать только товар в Глазго [Glasgow] или только записи, касающиеся материнских плат.

Введение в функции

Ранее мы рассматривали, как вставлять функции из списка, но есть некоторые ключевые функции, которые во время работы с образцом электронной таблицы вы найдете особенно полезными. Например, чтобы сложить все элементы в столбце, надо использовать функцию '=SUM()' с интересующим вас диапазоном внутри скобок. Например, '=SUM(H3:H23)' даст общее количество

Скорая помощь

Блокировка вашей книги очень проста. Просто выберите **Сервис > Защитить документ**, чтобы предотвратить внесение изменений лицами, не владеющими паролем. Для блокировки только отдельных ячеек в документе послужит **Сервис > Защитить лист**.

Открытые форматы файлов

По умолчанию Calc будет сохранять ваш файл как файл .ods, а не .xls или .xlsx (оба — форматы Excel). Это связано с тем, что по умолчанию используется стандарт ODF Document Foundation (Open Document Format). Он одобрен ISO и всё чаще внедряется организациями государственного и частного секторов, которые хотят сделать данные более доступными для различных платформ и устройств.

Как мы упоминаем в основном тексте, LibreOffice использует расширенный стандарт ODF, и это означает, что в других офисных пакетах, использующих более строгие версии, некоторые второстепенные

функции должным образом не работают. Это не лень со стороны разработчиков этих пакетов: отчасти это вопрос, где они выделяют ресурсы, и в основном тот факт, что расширенная функциональность, возможно, еще не включена в стандарт. Однако в большинстве случаев вы должны иметь возможность открывать файлы в Microsoft Office, а LibreOffice всегда поддерживает обратную совместимость, позволяя вам читать и править документы, которые могли быть созданы много лет назад.

ODF изо всех сил боролся за поддержку на продуктах коммерческих поставщиков, но в настоящем

время ситуация значительно изменилась к лучшему, и даже Microsoft Office 365 поддерживает ODF 1.2 — после усиления давления со стороны потребителей, правительственных агентств и растущего количества бизнес-клиентов, которые предпочитают его внутреннему (проприетарному) формату Microsoft — OpenXML.

В маловероятном случае проблем с кросс-программной совместимостью вы можете изменить версию ODF, которую вы используете для создания файлов, выбрав **Сервис > Параметры > Загрузка/сохранение > Общие**.

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

Скорая помощь

В приступе ностальгии и для того, чтобы ваши листы выглядели красивее (или уродливее, в зависимости от ваших конкретных предпочтений), перейдите в раздел **Формат > Стили** и сделайте свой выбор.

для столбца 'Max Profit [Макс. прибыль]'. Вы также можете узнать, сколько у вас записей: '=COUNT()', а также определить максимальное, минимальное и среднее значения с помощью '=MAX()', '=MIN()' и '=AVERAGE()'.

Но это не всё. Вы можете установить значение на основе условия, используя функцию '=IF()'. Например, '=IF(B3="Bristol","yes","no")' вернет в таблицу из примеров "yes".

Другой ценной функцией является '=CONCATENATE()', которая бывает особенно полезна, когда требуется создать отчет, содержащий данные из разных частей вашего листа, таким образом, чтобы их можно было потом распечатать или отправить по электронной почте. В тестовой таблице '=CONCATENATE(B3," is selling ", C3)' вернет "Bristol is selling Keyboard Type A", но в качестве одного из значений, указанных в вашем предложении, также допускается добавлять функцию.

Правила форматирования

Было бы здорово выделять отрицательные значения красным фоном и помечать другие значения, которые требуют внимания. Хотя можно форматировать ячейки вручную, *LibreOffice Calc* с легкостью сделает это для вас автоматически встроенным механизмом под названием условное форматирование.

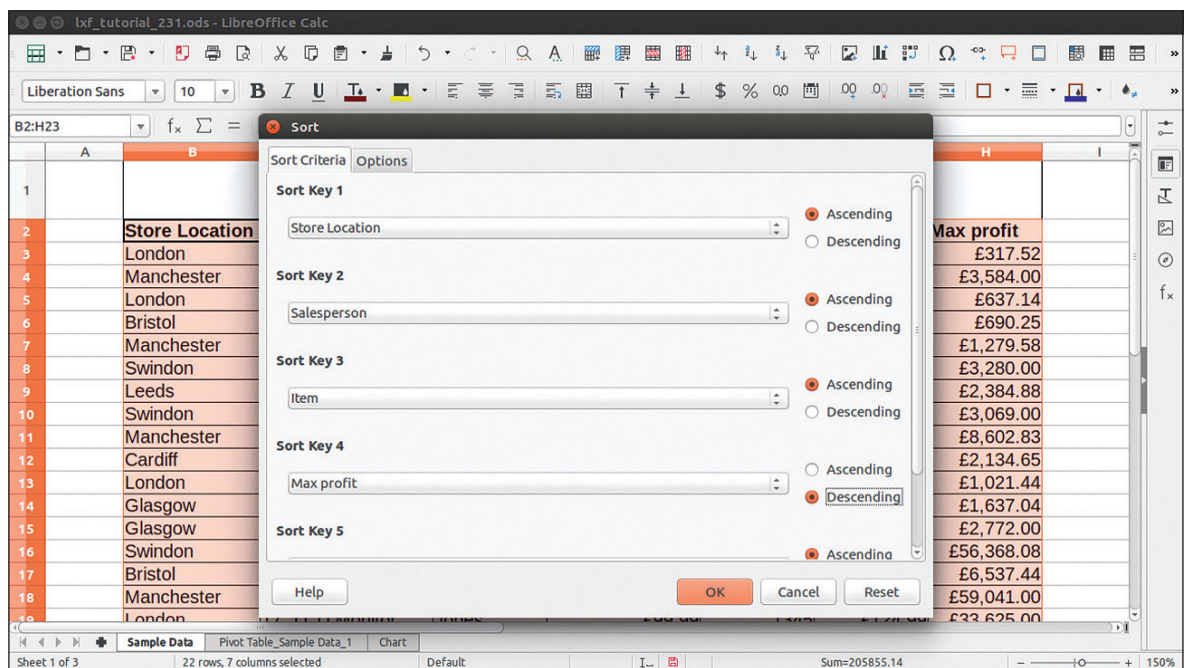
Как следует из названия, создав правила, определяющие ваши «условия», основанные на значении ячейки или листа, вы сможете динамически изменять такие вещи, как фон или цвет текста. Например, в случае отрицательных значений денежных сумм *Calc* уже делает шрифт красным по умолчанию; но если вы хотите сделать это вручную, то должны выбрать **Формат > Условное форматирование > Условие, задать условие «меньше»** и ввести в качестве правила '=0'.

Затем очень просто выбрать стиль форматирования, которое должно применяться к листам в вашей книге, если удовлетворены ваши условия. Возможно, вам захочется поэкспериментировать с установкой новых правил, созданием стилей и упорядочением правил, которые вы создаете, в соответствии с их приоритетом.

На ваше счастье, вам не придется каждый раз делать это вручную. В том же подменю **Условное форматирование**, где вы видели **Условие**, вы также увидите другие параметры, такие как **Цветовая шкала** и **Гистограмма**.

Цветовая шкала автоматически вычислит самые высокие и самые низкие значения и установит их фон по шкале цветов от красного через янтарный до зеленого. **Гистограмма** создаст во всех ячейках синюю полоску, указав их долю от расчетной суммы всех значений в выбранном диапазоне.

➤ **Всякий раз, когда вам понадобится расположить строки «именно так», вы можете установить несколько уровней сортировки в любом направлении по любому типу данных.**



Абсолютно изменяемые ячейки

В основном тексте мы упомянули, что можно щелкнуть и перетащить правый нижний угол ячейки, чтобы быстро повторить его содержимое или расширить вычисления, включив более широкий диапазон данных. По мере использования этого метода вы можете обнаружить, что хотите, чтобы одна конкретная ячейка или буква столбца остались неизменными.

Однако, поскольку все адреса ячеек по умолчанию являются переменными, потом вы с досадой обнаружите, что вам всегда приходится исправлять

их вручную. То же самое происходит, если вы копируете и вставляете ячейку, содержащую формулу. К счастью, есть решение этой проблемы. Добав-

Адрес ячейки	После переноса на строку ниже	После переноса на столбец правее
A1+A2	A2+A3	B1+B2
\$A1+A2	\$A2+A3	\$A1+B2
A\$1+A2	A\$1+A3	B\$1+B2
\$A\$1+A2	\$A\$1+A3	\$A\$1+B2

ление символа доллара \$ может сделать часть адреса ячейки «абсолютной», так что она не будет автоматически изменяться при нажатии и перетаскивании для расширения диапазона, как раньше.

В таблице показано, как влияет на ячейку A1 добавление символа \$ в разных местах формулы. Если поместить \$ перед буквой в адресе ячейки, то буква столбца останется неизменной; то же — после блокировки номера строки. При добавлении \$ в обоих местах адрес ячейки никогда не изменится, как вы ни расширяйте диапазон.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Рай для программиста

Есть известный афоризм: безумие — это делать одно и то же снова и снова и ожидать разных результатов. Чтобы помочь вам избежать этой позорной участи, в *LibreOffice Calc* предусмотрена подсистема макросов для автоматизации повторяющихся задач.

Простейший способ создать макрос — это вручную записать действие. Обычно это можно сделать в меню Сервис > Макросы > Записать макрос. Если в меню Сервис этого нет, такое можно активировать, отметив флажок Включить запись макросов в Сервис > Параметры > LibreOffice > Расширенные

возможности. Когда закончите, нажмите Завершить запись и сохраните действие, чтобы вы могли вызвать его позже.

Однако такой подход ограничен. Действия записываются только в текущем окне; движения мыши игнорируются (то есть никаким вам «щелкнул — потянул»); и изменение встроенных параметров *Calc* не повторяется в последующих запусках. В таких ситуациях поможет только код.

Если нажать Alt+F11, то откроется редактор кода *LibreOffice BASIC*. По умолчанию вы увидите пустую подпрограмму, в которую можно излить свой гений

кодирования. Как и в эквиваленте от Microsoft, можно программно манипулировать содержимым любого открытого документа и запускать эти скрипты заданным сочетанием клавиш или из пункта меню Сервис. Вы можете найти дополнительную информацию об этом на сайте www.bit.ly/lxf231libre.

Если BASIC вам не подходит, то, при наличии установленной Java Runtime Environment, вы также можете писать макросы на Python и JavaScript. Их можно организовать и запустить из меню Сервис > Макросы > Управление макросами. Выбор за вами...

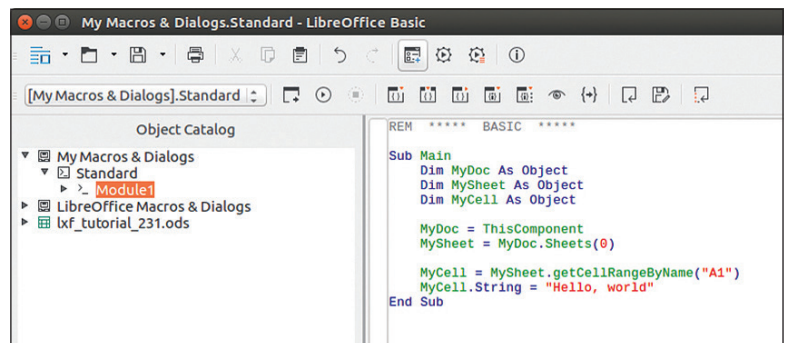
Секторы и столбики

Создание диаграмм в *OpenOffice* было достаточно сложным, чтобы его описание оказалось шестиступенчатой схемой с картинками. Вы будете рады узнать, что времена изменились, и теперь гораздо проще создать визуализацию ваших данных. Просто выберите интересующие вас столбцы с заголовками и перейдите во Вставка > Диаграмма. Вы сможете просмотреть свой график при выборе различных типов, таких как столбчатые и круговые диаграммы, диаграммы рассеяния и гистограммы.

Используя столбцы 'Store Location' и 'Max Profit' в образце электронной таблицы, вы заметите, что мастер попытается построить точку для всех строк, а не объединять их, как вы могли бы ожидать. Существует несколько способов решения этой проблемы, например, использование сводной таблицы для создания диаграммы на основе итогов для каждого местоположения.

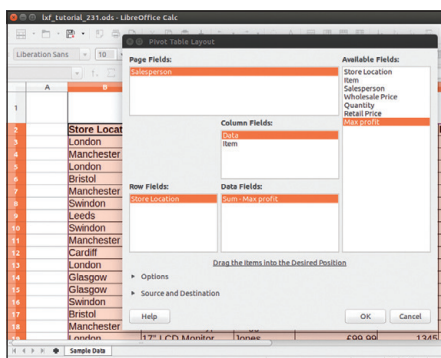
Кроме того, можно создать сводную таблицу на другом листе и использовать ее для построения диаграмм. Это можно сделать с помощью функции '=SUMIF()', или так называемой формулы массива. Самый простой способ создать последний — использовать мастер функций через функцию Вставка > Функция, установив флажок Массив в нижнем левом углу окна.

Создав график, вы обнаружите, что редактировать его можно, просто дважды щелкнув по той его части, которую следует изменить. Также можно щелкнуть правой кнопкой мыши по диаграмме, чтобы добавить новые элементы, такие как метки осей и дополнительные поля, или изменить масштаб при необходимости. Получайте удовольствие, резвясь в своих диаграммах! **LXF**



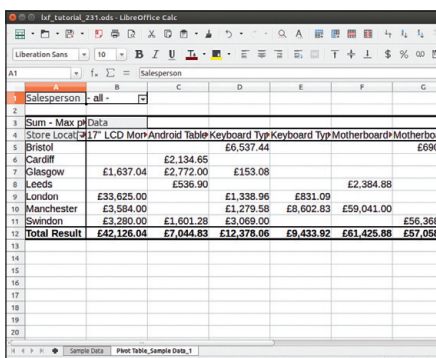
Для случаев, когда срабатывает только код, *LibreOffice* сохраняет редактор кода BASIC из родительского проекта *OpenOffice* от Apache.

Сводные таблицы



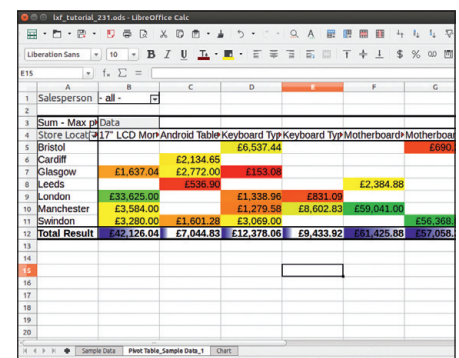
1 Компоновка

Для начала просто выделите какой-либо диапазон данных (вместе с заголовками столбцов) и выберите Вставка > Сводная таблица. Откроется окно, которое предлагает вам перетаскивать поля в требуемые области. Внутри Поля строки определяются критерии фильтра; Расчетные поля вам всё просуммируют; а Поля столбцов и Поля строк определяют общую компоновку таблицы.



2 Фильтрация результатов

Calc динамически создаст новый лист, содержащий вашу новую сводную таблицу. Затем вы просто используете серые поля со стрелками для активации раскрывающегося меню. В них можно назначить или отменить выбор данных для отображения в таблице. Это означает, что вы можете переключаться между просмотром обобщенной картины всех данных и цифрами, отражающими конкретный сценарий.



3 Форматирование данных

Как и в остальной части листа, правила условного форматирования применимы и к сводной таблице. В этом примере мы использовали Цветовая шкала, чтобы выделить самые высокие и самые низкие значения цветом в середине таблицы. В вычисленных полях мы применили Гистограмма, отражающую долю конечной суммы, которую составляет конкретный показатель.

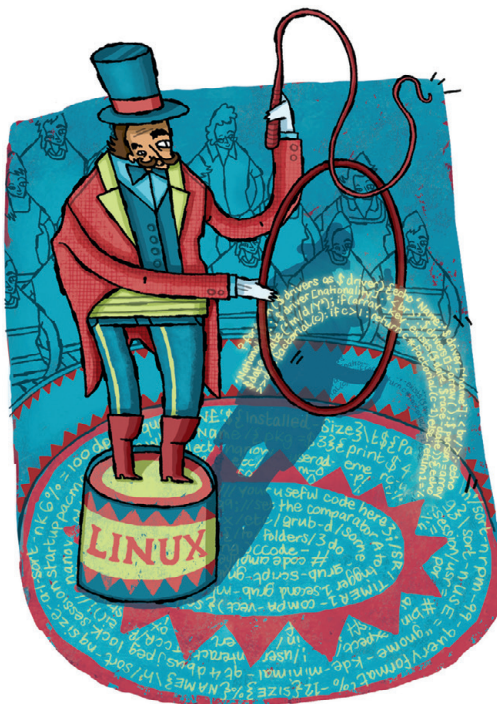
10 Гбит/с: Сети ускоряются

Темный Властелин Сетевых Операций, **Тим Армстронг** вернулся — пора применить его черную магию к вашим кабелям Cat6. Тащите лазеры!



Наш эксперт

Тим Армстронг — архитектор сети в Nerdalize. Он специализируется на разработке и внедрении центров обработки данных и сетей ISP и частенько тратит свое время на Twitter @HelpmAnExpert. «О-оо, зацените этот пакет!»



Кажется, не проходит и недели без оповещений о новых устройствах, совместимых с 10 Гбит/с — от популярных устройств NAS (таких как DS1817, рассмотренное в LXF228) до Asus, присоединяющихся к рынку Ethernet с их настольным коммутатором 10GbE (GigabitEthernet), он же 10G, больше

похожим на высокотехнологичную аудиосистему, чем на сетевой коммутатор. Даже несмотря на то, что мы слышим о подобных настройках со времен Netgear XS709E, появившегося несколько лет назад, 10G, возможно, наконец-то будет готов для использования в сетях Consumer и SoHo. Мы собираемся прорваться через дымовую завесу маркетинга и взглянуть на то, как перейти от малого до среднего офиса, не разорившись.

Кому это надо?

Прежде чем мы окунемся в самую гущу выбора оборудования и прокладки сети, возможно, стоит обсудить, кому действительно будет выгодна работа в сети с 10G и почему? Похоже, наибольшую выгоду от этой модернизации получат новые медиа-компании. Это не означает, что другим такое не светит — конечно, тоже светит — но есть нечто весомое в том, что несколько человек смогут редактировать черновые 4K-кадры из NAS одновременно.

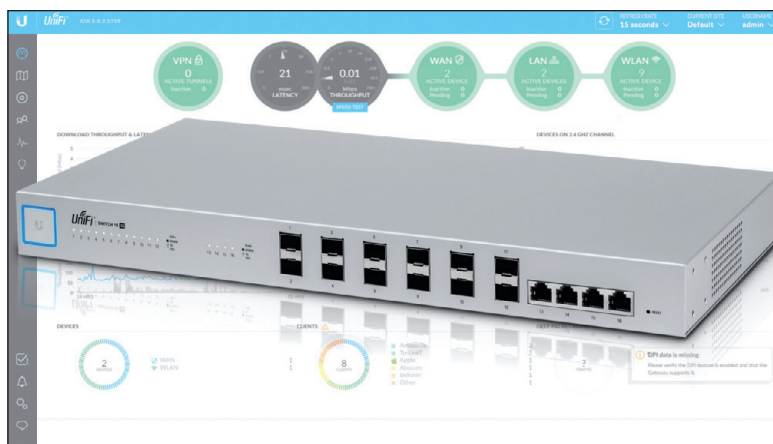
Например, маленькая студия или канал YouTube с двумя-тремя редакторами, выпускающими одно-два видео 4K в неделю — это уже типичный случай оправданности модернизации; но для более традиционной офисной среды, например, офиса бухгалтера, юридической фирмы или печатной медиа-компании, обновление, возможно, будет иметь смысл только при наличии около сотни рабочих станций. Может сложиться впечатление, что новые медиа-компании являются исключением, будучи единственными, для кого на данный момент имеет смысл офисная сеть 10G, но это — одно из самых больших заблуждений о 10G-сети.

Ключевой переменной здесь является пиковая, а не поддерживаемая пропускная способность. Неспособность сети в полной мере справиться со всплеском трафика убивает продуктивность. Мы все такое видели: сотрудники возвращаются с обеда, разблокируют свои ПК и ждут. Они могут даже оставить ПК разблокированным и пойти за кофе, поскольку ожидание, пока всё загрузится, отнимает очень много времени. С сетью 10G, при условии, что ваши устройства NAS и рабочие станции обновлены, эта проблема уходит в прошлое.

Подсчитаем

За основу мы возьмем архитектурную фирму примерно с 20 рабочими станциями. В сердце архитектурной фирмы находятся папки с проектами: для надежности, безопасности и управляемости они обычно хранятся в центральном NAS, затем (надемся) резервируются во вторичном NAS или «горячем резерве» и еженедельно архивируются в каком-нибудь холодном облачном хранилище.

Предположим, что объем проекта около 60 МБ; это включает чертежи, рендеринг, исследование участка и любые другие цифровые активы, необходимые для проекта. Цель заключается в том, чтобы все машины смогли загрузить весь проект с NAS почти с той же скоростью, как если бы проект содержался на локальном



➤ US-16-XG вносит в вашу сеть стиль, простоту и скорость.

Соединяемся с вашим коммутатором

Верите или нет, но 10 Гбит/с не всегда достаточно для поддержания потребностей компании: в результате вы часто видите сетевых инженеров, прибегающих к соединению двух или более интерфейсов, чтобы удвоить пропускную способность. Это дает бонус по обеспечению резервирования портов как для коммутатора, так и для подключаемого устройства, будь то другой коммутатор или конечная точка, требующая высокой пропускной способностью.

Однако существует множество условий для соединения, и, в зависимости от поставщика, также

есть перекрывающие условия, так что мы посчитали полезным сделать краткий обзор.

Поставщики коммутаторов часто ссылаются на соединение как на Link Aggregation Groups [Группы агрегации связи] (сокращенно LAG), Port Aggregation [Агрегация портов], Port Trunking [Объединение портов], Link Bundling [Соединение каналов], или Interface Teaming [Группирование интерфейсов]. MacOS называет это Link Aggregate Virtual Interface [Виртуальный интерфейс агрегации каналов]. Windows Server называет это NIC Teaming, а ПК с Windows не имеют встроенной поддержки

этого. Linux называет это просто Bonding [Соединение], что является коротким, точным и отражающим суть описанием без лишних слов.

Существует два способа настроить связь: статически на коммутаторе и другой конечной точке, или динамически, с использованием Link Aggregation Control Protocol [Протокол управления агрегацией ссылок] (LACP). В небольшой сети оба эти подхода имеют примерно одинаковую эффективность; но когда вы работаете с большим количеством серверов, применение LACP способно спасти жизни.

хранилище; это поможет избежать ситуаций, когда люди переносят проект из NAS на свои ПК, а затем возвращают обратно, создавая путаницу с версиями. Исходя из данных чисел, мы можем оценить ожидаемую пиковую загрузку NAS, разработать сеть и начать присматриваться к оборудованию.

Если мы ждем, что файл будет загружаться с той же скоростью, как и с локального диска, то на пике мы увидим 9,6 Гбит/с, что почти заполняет 10 Гбит/с соединение, а мы даже не рассмотрели резервное копирование Первичного NAS на Вторичное NAS.

Коммутация

Теперь мы осознали порядок чисел, и используем их для управления разработкой нашей сети и выбором оборудования. В данной среде есть три основных конечных точки: устройства NAS, рабочие станции и Интернет. Это легко описывается простой топологией «расширенной звезды», где серверы NAS прикреплены к коммутатору агрегации, а рабочие станции подключены к коммутатору распределения, уменьшая число портов, требуемых на нашем 10G коммутаторе — а порты являются основным фактором в его стоимости.

Сейчас на рынке коммутаторов существует путаница, поскольку на первый взгляд они выглядят однотипно, и люди принимают один из двух вариантов решений. Они либо уверены, что «чем дороже, тем лучше», либо приходят к выводу «раз они все одинаковые, лучше уж сэкономить деньги». Реальность заключается в том, что на рынке много хлама: вы ищете неограниченный алмаз, и, как и всегда в случае с сетевым оборудованием, цена — не показатель качества.

Когда дело доходит до выбора 10G-коммутатора для малого или среднего офиса, есть три ключевых момента: количество портов 10G, пропускная способность и управляемость. Для нашей демонстрационной сети нам нужно как минимум шесть портов 10G (два на NAS и два на коммутатор рабочей группы), и как минимум 41 порт 1G (два на стол, плюс один для роутера).

Для наших целей лучшим коммутатором-распределителем, отвечающим всем требованиям, является US-48 Unifi Switch от Ubiquiti: он доступен (£360/\$415 за базовую версию), имеет два порта 10G для соединения с центральным узлом, полнодуплексную пропускную способность, а также легко управляется через интерфейс Unifi. Раз мы работаем с коммутатором-распределителем Ubiquiti, логично, что мы будем использовать коммутатор агрегации Ubiquiti, поскольку он позволяет управлять всеми коммутаторами из одной точки через интерфейс Ubiquiti, Unifi.

Для нашей демонстрационной сети мы будем использовать US-16-XG (£520/\$599). Стоит отметить, что мы ссылаемся на серию коммутаторов Unifi от Ubiquiti, а не их печально известную серию EdgeSwitch, опороченную плохой прошивкой и плохим контролем качества, да вдобавок несовместимую с контроллером Unifi.

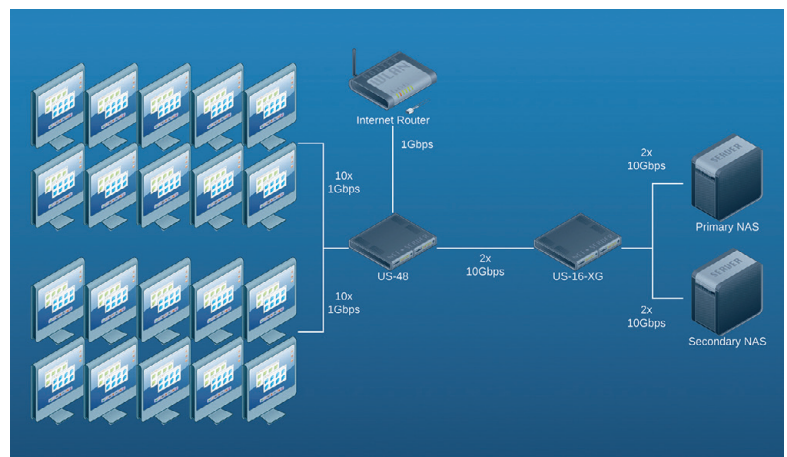
Я не буду вдаваться в подробности об устройствах NAS, поскольку они гораздо лучше рассмотрены в обзорах наподобие упомянутого ранее; однако мой личный опыт гласит, что QNAP обеспечивает хороший баланс между стоимостью и расширяемостью, тогда как Synology более дружелюбен к новичкам.

Компоновка

Теперь, когда всё оборудование выбрано, настало время немного подробнее поговорить о компоновке сети. Для офисов малого и среднего размера не найдешь более эффективной компоновки, чем расширенная звезда: она рентабельна, проста в управлении и легка в отладке. Конечно, расширенная звезда не годится для корпоративных сетей, где время безотказной работы стоит любых денег, но для офисов малых и средних размеров она более чем сойдет.

Наша компоновка в виде расширенной звезды имеет коммутатор агрегации в центре и коммутатор-распределитель (с двойным каналом связи) на первом луче звезды. Устройства NAS должны быть напрямую подключены к коммутатору агрегации (каждый с двойным интерфейсом). Каждая рабочая станция подключена к коммутатору-распределителю.

Эта конструкция не только обеспечивает достаточную пропускную способность для основного NAS с рабочих станций, но ее также достаточно для резервного копирования ваших данных на вторичный NAS, не влияя на доступ. Она даже легко поддается расширению без необходимости существенной реорганизации, если только вы не попадете куда-то, где от 60 до 80 рабочих станций, где устройства NAS сами по себе перегружены и требуют обновления. **LXF**



► При всего 20–40 рабочих станциях, это мало похоже на звезду, но масштабируется вдвое-втрое посредством добавления коммутаторов-распределителей.

OpenVPN: Безопасность онлайн

Как настроить бесплатную, быструю и полнофункциональную виртуальную частную сеть для удаленной работы? Ваш гид к знанию — Ник Пирс...



Наш эксперт

Ник Пирс активно интересуется безопасностью и конфиденциальностью, с учетом современных реалий. Но более чем счастлив делиться со всеми своей любовью к А-на!

Скорая помощь

Вам понадобится нацелить свой клиент на публичный IP-адрес вашего сервера *OpenVPN*. Рекомендуем вместо этого получить бесплатный динамический DNS-адрес, например, с NoIP (www.noip.com/free) и направить свой клиент на него.

Безопасно ли ваше интернет-соединение? Частый ответ — «Не очень». Одно из самых скользких мест для работы в Сети или проверки своей электронной почты — через общественную точку доступа Wi-Fi. В конце концов, вы не знаете, какие меры безопасности — если они вообще есть — были предприняты этими магазинами или компаниями. А вам нужна гарантированная форма шифрования, чтобы ваши личные дела оставались личными, и здесь-то и выходят на сцену виртуальные частные сети (VPN).

VPN созданы, чтобы добавить к Интернету и другим общественным сетям прослойку безопасности. По сути, они создают безопасный «канал», внутри которого ваши данные шифруются перед отправкой по сети. Их основное применение для домашних пользователей — обеспечение безопасного соединения с Интернетом, даже через сети, не являющиеся безопасными. Коммерческие VPN-сервисы предлагают бесплатные пакеты, но они медленные и ограниченные.

Гораздо лучше будет добавить собственную VPN на свой Linux-ПК совершенно бесплатно. Благодаря *OpenVPN* можно установить собственный сервер, затем настроить его на обеспечение одного или более устройств (мобильных или компьютеров) безопасной сетью VPN по требованию, когда это будет необходимо. Что ж, без лишнего шума, давайте приступим.

Что вам понадобится

На нашем уроке мы работаем с базовой установкой Ubuntu 16.04 — единственное реальное требование заключается в том, чтобы ваш ПК дома был включен и подключен к Интернету в тот момент, когда вы решите использовать сервисы VPN где-то в пути.

Сначала установим *OpenVPN Server*. Хотя это часть официальных репозитория Ubuntu, вы не сможете установить его через *Software Centre*, поэтому откройте Терминал и введите

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

```
$ sudo apt-get install openvpn easy-rsa
```

Установятся два пакета — второй (*easy-rsa*) настраивает и затем управляет сертификатами, необходимыми для шифрования

вашего сетевого трафика. Пакет *easy-rsa* позволяет использовать *OpenVPN* в PKI (public key infrastructure — инфраструктуре открытого ключа), которая состоит из открытого (публичного) и закрытого (частного) ключей для каждого сервера и клиента, где для идентификации доверенных клиентов и серверов друг для друга используются открытые ключи. Эти открытые ключи подписываются с помощью сертификатов для удостоверения их аутентичности, и ваш сервер должен быть настроен как удостоверяющий центр (certificate authority — CA) для создания, распределения и управления этими сертификатами.

Вашей первой задачей будет настроить свой CA и создать закрытый и открытый ключи для сервера — подробности вы найдете в пошаговом руководстве. По завершении этого процесса неплохо будет еще «укрепить» безопасность вашей VPN. Это делается такими командами Терминала:

```
$ cd ~/openvpn-ca
$ ./build-dh
$ openvpn --genkey --secret keys/ta.key
```

Настройте сервер OpenVPN

Пора приступить к настройке вашего *OpenVPN* сервера и подготовке его к работе. Сначала надо скопировать ключи шифрования и другие файлы, которые вы только что создали, в директорию */etc/openvpn*, опять же с помощью Терминала:

```
$ cd ~/openvpn-ca/keys
$ sudo cp ca.crt ca.key MyVPN.crt MyVPN.key ta.key dh2048.pem /etc/openvpn
```

Замените *MyVPN.crt* и *MyVPN.key* значениями, которые вы присвоите для *KEY_NAME* в пошаговой инструкции на стр. 68. Далее необходимо создать файл конфигурации вашего *OpenVPN*:

```
$ cd /usr/share/doc/openvpn/examples/sample-config-files
$ sudo cp server.conf.gz /etc/openvpn
$ sudo gzip -d /etc/openvpn/server.conf.gz
```

Теперь следуйте второму пошаговому руководству, на стр. 69, чтобы настроить свой сервер для базового использования. По завершении надо выполнить критически важное действие, чтобы

Скрипт make_config.sh

Откройте терминал и введите такую команду, чтобы создать пустой файл:

```
$ nano ~/openvpn-ca/client-configs/make_config.sh
```

Затем заполните этот файл следующими строками:

```
#!/bin/bash
# Первый аргумент: Идентификатор клиента
KEY_DIR=~/.openvpn-ca/keys
OUTPUT_DIR=~/.openvpn-ca/client-configs/files
```

```
BASE_CONFIG=~/.openvpn-ca/client-configs/client.conf
cat $(BASE_CONFIG) \
<(echo -e '<ca>') \
$(KEY_DIR)/ca.crt \
<(echo -e '</ca>\n<cert>') \
$(KEY_DIR)/${1}.crt \
<(echo -e '</cert>\n<key>') \
$(KEY_DIR)/${1}.key \
<(echo -e '</key>\n<tls-auth>') \
```

```
$(KEY_DIR)/ta.key \
<(echo -e '</tls-auth>') \
> ${OUTPUT_DIR}/${1}.ovpn
```

После создания сохраните и закройте файл.

Далее введите следующую команду, чтобы сделать его исполняемым:

```
$ chmod 700 ~/openvpn-ca/client-configs/make_config.sh
```

Ваш скрипт готов к запуску.

Настройка клиента

Вы можете получить доступ к вашему *OpenVPN* серверу через разные платформы — есть официальные клиенты для Windows, Android и iOS (более подробная информация — на www.openvpn.net), а пользователи MacOS могут использовать *Tunnelblick* (www.tunnelblick.net). Для переноса файла настройки *.ovpn* с одного устройства на другое вы можете просто скопировать файл на **C:\Program Files\OpenVPN\config** в Windows, например, или дважды щелкнуть по нему, чтобы импортировать его в *Tunnelblick* на MacOS.

Если вы работаете на клиенте Ubuntu, можете установить дополнительные пакеты и управлять

своим подключением через имеющийся GUI *Network Manager*.

Откройте терминал и введите

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

```
$ sudo apt-get install network-manageropenvpn-Gnome
```

После установки откройте *Network Manager* из строки меню и выберите *Edit Connections* [Редактировать подключения], затем нажмите на *Add* [Добавить]. Выберите *Import a saved VPN configuration* [Импортировать сохраненную настройку VPN]... из выпадающего меню и нажмите *Create* [Создать]. Выберите свой файл *.ovpn* и нажмите *Open*, затем проверьте правильность настроек в четырех

вкладках (нажмите на вкладку *Advanced > TLS Authentication* [Расширенные > Аутентификация TLS] для подтверждения распознавания ключа TLS). Дайте вашему подключению более подходящее имя — скажем, имя, которое вы присвоили вашему VPN в *KEY_NAME* — и нажмите *Save*.

Проделав это, подключитесь через подмену VPN в *Network Manager* — на значке строки меню вы должны увидеть появившийся замочек, который указывает на успешное создание безопасного подключения.

Последний шаг — открыть браузер и удостовериться, что подключение работает.

обеспечить вашим клиентам настоящий доступ в Интернет через ваш VPN. Сначала введите команду

```
$ ip route | grep default
```

Запомните интерфейс, указанный рядом с 'dev' — на нашем компьютере это *enp0s3*. Теперь введите следующую команду, заменяя *enp0s3* на интерфейс, указанный на вашем ПК, если это возможно:

```
$ sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o enp0s3 -j MASQUERADE
```

Запуск и проверка

Пора подготовить всё к работе. Введите следующую команду Терминала, которая запускает *OpenVPN* с помощью настроенного вами файла *server.conf*:

```
$ sudo systemctl start openvpn@server
```

Проверьте, правильно ли работает сервер:

```
$ sudo systemctl status openvpn@server
```

Вы должны увидеть список сообщений, включая самое важное — *active (running)*, вместе с *Initialization sequence completed* внизу; нажмите *Ctrl+C* для прерывания проверки состояния. Затем проверьте доступность интерфейса *OpenVPN*:

```
$ ip addr show tun0
```

Если вы видите настроенный интерфейс, заканчивающийся на *valid_lft forever preferred_lft forever*, значит, всё должно работать

корректно. Теперь можете настроить *OpenVPN* на автоматический запуск при начале работы, следующей строкой:

```
$ sudo systemctl enable openvpn@server
```

Всё для ключевых клиентов

Простейший способ настроить ваш клиент — с помощью специально созданных файлов *.ovpn*, внутри которых заключены требуемые файлы ключей вместе с настройками для подключения. Они создаются на вашем сервере, и вот что вам надо сделать.

Сначала создайте структуру директории, внутри которой хранятся ваши файлы настройки клиента. Затем заблокируйте разрешения директории и создайте файл настройки, чтобы из него работать:

```
$ mkdir -p ~/openvpn-ca/client-configs/files
```

```
$ chmod 700 ~/openvpn-ca/client-configs/files
```

Теперь откройте менеджер файлов *Nautilus*, нажмите на ссылку *Computer* слева и переберитесь в */usr/share/doc/openvpn/examples/sample-config-files/*. Скопируйте файл *client.conf* отсюда в *~/openvpn-ca/client-configs*. Дважды щелкните по скопированному файлу, чтобы отредактировать его в *gedit*. Сначала определите строку, отмеченную *remote my-server-1 1194*, и измените ее, чтобы она указывала на публичный IP-адрес вашего сервера или динамическое имя DNS (см. врезку *Скорая помощь* на стр. 66) — например, *remote 1.2.3.4 1194* или *remote dynamic.ddns.net 1194*.

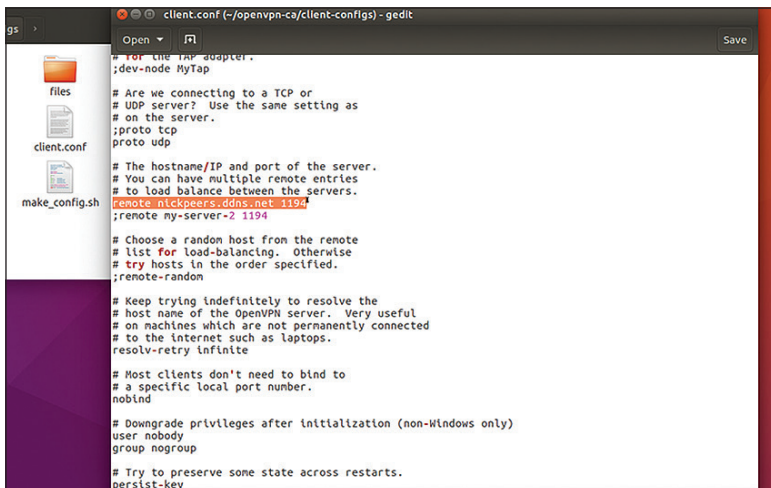
Скорая помощь

Укрепите безопасность вашего сервера *OpenVPN*, запустив вместе с ним брандмауэр *UFW* — загляните на <https://bit.ly/openvpn-ufw>, чтобы узнать, как его настроить (заменяя в инструкции папу на vim).

```
nick@UbuntuVM:~$ sudo systemctl status openvpn@server
[sudo] password for nick:
● openvpn@server.service - OpenVPN connection to server
   Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled; vendor preset:
   Active: active (running) since Wed 2016-07-13 13:39:29 BST; 2h 46min ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 4276 (openvpn)
     CGroup: /system.slice/system-openvpn.slice/openvpn@server.service
             └─4276 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/s

Jul 13 16:12:59 UbuntuVM ovpn-server[4276]: 92.26.250.142:34227 Control Channel:
Jul 13 16:12:59 UbuntuVM ovpn-server[4276]: 92.26.250.142:34227 [client2] Peer c
Jul 13 16:12:59 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 MULTI_sv
Jul 13 16:12:59 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 MULTI: L
Jul 13 16:12:59 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 MULTI: p
Jul 13 16:13:02 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 PUSH: Re
Jul 13 16:13:02 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 send_pus
Jul 13 16:13:02 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 SENT CON
Jul 13 16:22:31 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 [client2
Jul 13 16:22:31 UbuntuVM ovpn-server[4276]: client2/92.26.250.142:34227 SIGUSR1[
lines 1-20/20 (END)
```

» Настроив свой сервер, запустите эту проверку состояния, чтобы удостовериться в его готовности к работе.



➤ Создайте шаблон файла настройки клиента в *gedit*, он станет основой для всех файлов *.ovpn*, которые вы создадите.

Далее раскомментируйте `;user nobody` и `;group nogroup` (убрав символ `;` в начале этих строк) и, наоборот, закомментируйте строки `ca.crt`, `cert client.crt` и `key client.key`, поскольку они добавляются командным файлом. И напоследок — но это не менее важно — добавьте следующую строку под строкой `tls-auth ta.key 1` (она должна остаться комментированной):

```
key-direction 1
```

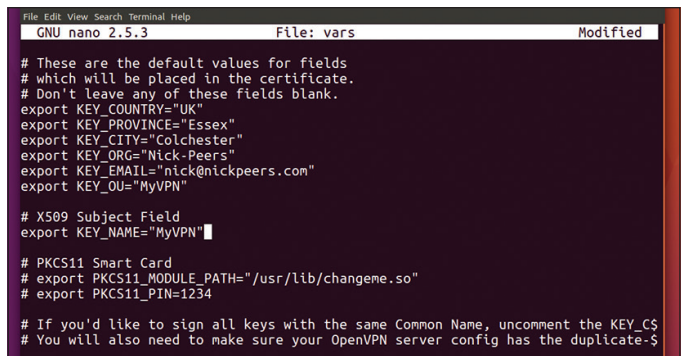
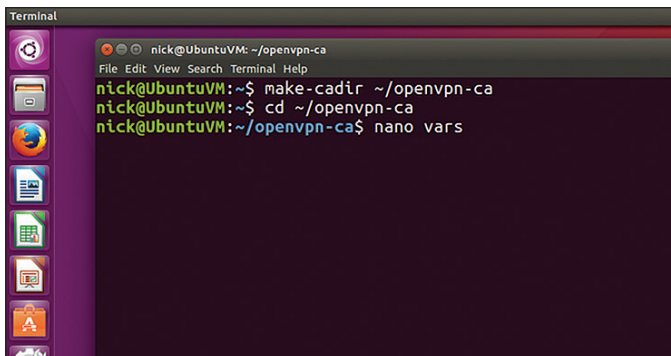
Сохраните и выйдите, затем следуйте инструкциям во врезке по созданию файла `make_config.sh`, который вы будете использовать для создания файлов настройки клиента, как только они потребуются для каждого клиента.

Создайте файл настройки клиента

Теперь у вас есть все инструменты, необходимые для создания ваших файлов настройки клиента. Опишем процесс, которому вы должны следовать, чтобы их создать. Сначала вам понадобится создать сертификат и ключ для вашего клиента, с помощью следующих двух команд:

```
$ cd ~/openvpn-ca && source vars
```

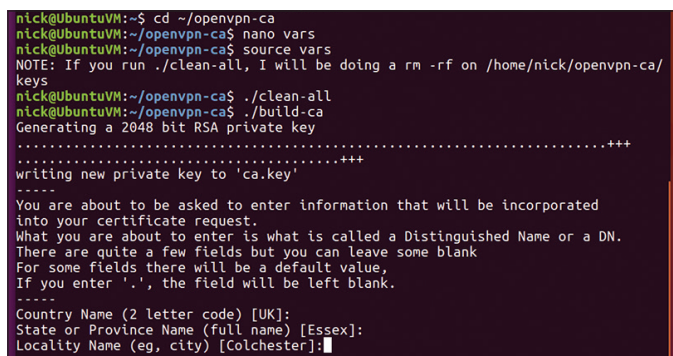
Все ко мне в центр (Сертификации)!



1 Центр сертификации

Сначала надо скопировать шаблон директории *easy-rsa* в свою домашнюю директорию, чтобы ваши индивидуальные сертификаты не переписывались при обновлении пакетов. Откройте Терминал и введите

```
$ make-cadir ~/openvpn-ca
$ cd ~/openvpn-ca
$ nano vars
```

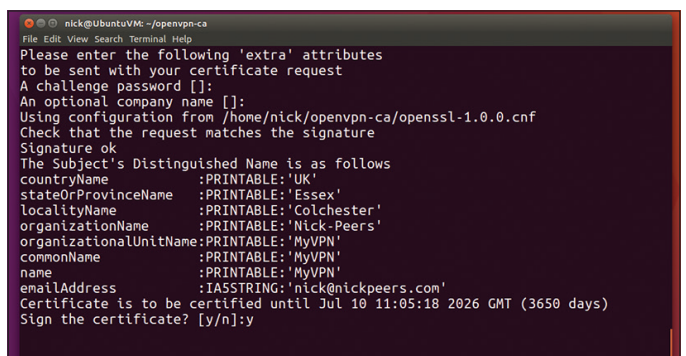


3 Создайте Центр сертификации

В Терминале проверьте, что вы по-прежнему находитесь в директории *openvpn-ca*, затем введите `source vars` и нажмите Enter. Вы должны увидеть уведомление (см. скриншот). Теперь введите `./clean-all` и нажмите Enter, а потом — `./build-ca`. После нажатия на Enter вы должны пройти через ключ и сертификат основного [root] центра сертификации. Просто нажмите Enter, когда вам это предложат, чтобы подтвердить правки, сделанные вами на предыдущем шаге.

2 Отредактируйте файл vars

Прокрутите до группы строк, начинающихся с `export KEY_COUNTRY`, и заполните их собственной личной информацией — в качестве примера см. скриншот. Обратите внимание, что `KEY_PROVINCE` заполняется названием графства, например, Эссекс или Мерсайд. Внесите свое имя или организацию в `KEY_ORG`, и заполните `KEY_OU` и `KEY_NAME` как хотите, например, внесите туда `MyVPN`. Сохраните файл (Ctrl+O) и выйдите.



4 Создайте ключи сервера

Введите `./build-key-server MyVPN` и нажмите Enter, заменив `MyVPN` на то значение, которое вы ввели для `KEY_NAME` на шаге 2. Нажмите Enter, чтобы принять выбор по умолчанию, как раньше. Когда вам предложат создать пароль запроса, нажмите Enter, чтобы оставить его пустым, и снова нажмите Enter, когда у вас спросят опциональное имя компании. Завершите процесс, нажав Y в ответ на два последние вопроса.

На сей раз примечание проигнорируйте — нам не надо ничего очищать — и введите

```
$. /build-key client1
```

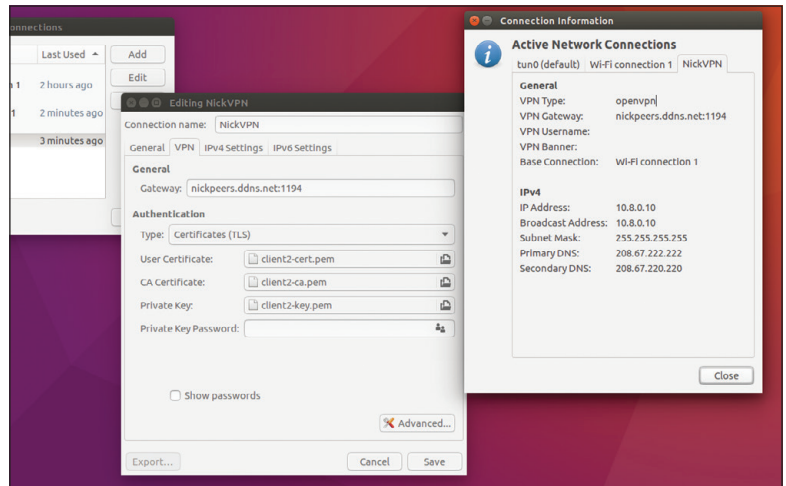
Вам понадобится уникальное имя для каждого сертификата клиента, который вы будете создавать — поэтому для последующих сертификатов вы могли бы использовать client2, client3 и т.д. И снова вы пройдете по процессу, похожему на тот, когда вы запускали команду *build-server-key*, и ответы в точности такие же — никакого пароля запроса и принятие выбора по умолчанию.

После создания следует настроить файл конфигурации:

```
$. cd client-configs
$. ./make_config.sh client1
```

Вы найдете файл, означенный как *client1.ovpn*, внутри подпапки *clientconfigs/files*. Теперь его надо безопасно переместить в ваш клиент — во врезке вы найдете разнообразные имеющиеся опции и способ настройки VPN-подключения к вашему серверу с другого Linux-ПК.

Поздравляем, вы только что сделали свое интернет-соединение более безопасным! **LXF**



Как только ваш сервер будет настроен, вы сможете безопасно выходить в Сеть удаленно через небезопасные подключения.

Настройте свой сервер

```
nick@UbuntuVM: /etc/openvpn
File Edit View Search Terminal Help
GNU nano 2.5.3 File: server.conf Modified
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert MyVPN.crt
key MyVPN.key # This file should be kept secret
#
# Diffie hellman parameters.
# Generate your own with:
# openssl dhparam -out dh2048.pem 2048
dh dh2048.pem
#
# Network topology
# Should be subnet (addressing via IP)
```

1 Определите ключи сервера

Откройте созданный вами файл настройки, набрав `sudo nano /etc/openvpn/server.conf` в Терминале. Мы сосредоточимся на минимально необходимых настройках для работы вашего сервера. Начните с того, что прокрутите до раздела `# SSL/TLS root certificate...` и измените строки `cert server.crt` и `key server.key`, чтобы они указывали на файлы, созданные вами на шаге 4 предыдущего руководства.

```
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
push "redirect-gateway def1 bypass-dhcp"
#
# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses. CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
```

3 Настройте направление Сети

Следующие несколько настроек дадут указание всем подключенным клиентам направлять их web-трафик через VPN. Нажмите `Ctrl+W` для определения `redirect-gateway` и раскомментируйте строку, где он включен — а именно, `'push...'`. Также раскомментируйте следующие два экземпляра `'push...'`, относящиеся к DNS-серверам, как показано на скриншоте. И наконец, найдите `'nobody'` и раскомментируйте строки `'user nobody'` и `'group nogroup'`. Сохраните и выйдите.

```
nick@UbuntuVM: /etc/openvpn
File Edit View Search Terminal Help
GNU nano 2.5.3 File: server.conf
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
# openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
tls-auth ta.key 0 # This file is secret
key-direction 0
#
# Select a cryptographic cipher.
```

2 Добавьте шифрование

Нажмите `Ctrl+W`, введите `tls-auth`, и вы перейдете на строку, отмеченную как `tlsauth ta.key 0`. Удалите ; в начале этой строки (этот процесс известен как раскомментирование), чтобы включить ее в дело. Ниже добавьте следующую строку, как показано на скриншоте: `key-direction 0`. Это обеспечит включение дополнительной TLS-аутентификации, настроенной ранее, для *OpenVPN*.

```
nick@UbuntuVM: /etc/openvpn
File Edit View Search Terminal Help
GNU nano 2.5.3 File: /etc/sysctl.conf Modified
#####3
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1
```

4 Включите переадресацию IP

Введите `sudo nano /etc/sysctl.conf` и раскомментируйте следующую строку, удалив из нее начальный символ `#`:

```
#net.ipv4.ip_forward=1
```

Сохраните и выйдите, затем введите следующую команду, чтобы перезагрузить `sysctl`: `sudo sysctl -p`. Вы должны увидеть `net.ipv4.ip_forward=1`, что говорит об успешном применении настройки. Вот и всё!

HTTPS: ЗАЩИТИМ СЕРТИФИКАТОМ

Шон Конвэй заглядывает за бархатный занавес сертификата безопасности из реализации протоколов SSL/TLS в командной строке.



Наш эксперт

Ночью Шон Конвей — репортер местной газеты. Днем — надзиратель за информационной безопасностью для канадского провайдера.



Скорая помощь
dnf — замена в области управления пакетами для yum (LXF205). Всё, что надо сделать пользователю для обеспечения совместимости — заменить yum на dnf, и результаты будут те же.

Работа в Сети, мы все надеемся на приватность и безопасность. В онлайн-транзакции должен быть четко определен пункт назначения, а информация должна быть доступна только двум сторонам. Протоколы наподобие SSL/TLS обеспечивают эти функции. Мы собираемся сделать обзор SSL/TLS — включая пакет ПО OpenSSL и общие сертификаты SSL/TLS — а также настроить web-сервер на использование сертификатов, для поддержки запросов HTTPS.

SSL (secure sockets layer [уровень защищенных сокетов]) — это протокол безопасности для создания зашифрованных защищенных каналов связи между сайтом и браузером. Рассматривайте его как тоннель, через который проходят данные и который не позволяет иным хостам, кроме отправителя и получателя, прочитать и понять данные.

TLS (transport layer security [безопасность транспортного уровня]) — новое имя для SSL. Когда SSL был в версии 3.0, появился TLS 1.0. Текущая версии — TLS 1.0 (SSL 3.1) и TLS 1.2. Каждая новая версия добавляет или изменяет функции. Это аббревиатуры иногда пишут вместе: SSL/TLS.

У HTTPS (hypertext transfer protocol secure [протокол защищенной передачи гипертекстовой информации]) есть довесок S, поскольку он использует SSL/TLS. HTTP — это протокол отправки пакетов запросов и получения пакетов ответов. Данные, отправленные через HTTP — это простой текст. Это означает, что любой, у кого есть доступ к каналу связи (такому, как кабель Cat5 или взломанное Wi-Fi-соединение), может прочитать данные. Когда запущен SSL/TLS, создается двунаправленный зашифрованный тоннель для передачи данных между двумя хостами. Данные HTTP в виде простого текста при использовании зашифрованного тоннеля становятся защищенными.

Сертификаты SSL

Сайт с поддержкой HTTPS будет использовать сертификаты SSL. Браузеры, получающие доступ к сайту, использующему SSL/TLS, могут быть уверены в подлинности сайта, а также в том, что передаваемые данные защищены посредством шифрования.

Установка пакета ПО *openssl* устанавливает инфраструктуру для внедрения HTTPS. Набор инструментов, содержащийся в пакете, позволяет создать запрос на подпись сертификата [certificate signing request] (CSR), который отправляется в удостоверяющий центр [certificate authority] (CA) для создания сертификата SSL.

Следуя схеме, приведенной внизу слева, процесс поддержки HTTPS на web-сервере начинается с генерации открытого (публичного) ключа криптографической пары ключей (1). Закрытый (приватный) ключ держится в безопасности и не разглашается. Открытый ключ обнаружится в ходе запроса CSR.

Пары ключей используются для создания CSR (2). CSR содержит, помимо других компонентов, открытый ключ для сайта. Рассматривайте открытый ключ как идентификатор ДНК для сайта. CSR отправлен в CA для подписи и создания сертификата (3).

CA создает сертификат, используя сведения из CSR (такие, как имя хоста, компания, страна, IP-адрес и пресловутый открытый ключ), подписывая его с помощью своей цифровой подписи. Процесс подписи аналогичен тому, что CA подтвердила: ДНК в запросе принадлежала запрашивавшему. CA возвращает сертификат SSL запрашивающему для использования при создании его поддерживающего HTTPS сайта (4).

Администратор сайта устанавливает сертификат и настраивает ПО web-сервера (в нашем случае — *Apache*) на использование сертификата SSL для сайта.

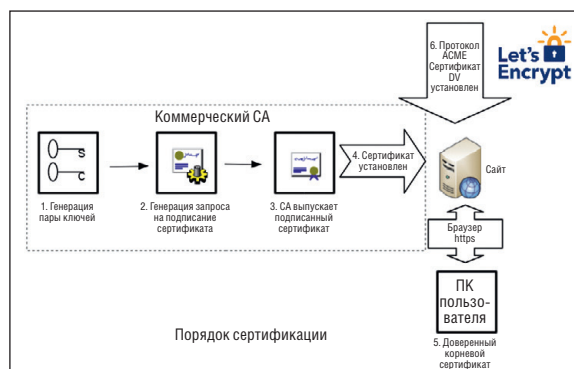
Браузер, получающий доступ к защищенному сайту (5) (другими словами, сайту с поддержкой HTTPS), проверяет сертификат от источника для установления доверительных отношений с сертификатом сайта. A=CA, B=браузер, C=защищен SSL. A доверяет C, B доверяет A, следовательно, B доверяет C.

Шифрование с открытым ключом, использованное при создании сертификата SSL, также входит в игру, устанавливая зашифрованный тоннель для перемещения данных между отправителем и получателем.

Сертификат SSL/TLS от солидной коммерческой CA может стоить от \$5 до \$1500. Поскольку сообщество Linux придерживается открытого кода, давайте рассмотрим альтернативу коммерческим сервисам.

Let's Encrypt (LsE) — это всемирный удостоверяющий центр (CA), выдающий сертификаты, которые могут быть использованы сайтами, чтобы разрешить использование защищенных HTTPS-соединений. Чтобы получить сертификат, запрашивающий должен продемонстрировать, что он владеет доменом (DNS-удостоверенный сертификат). Это позволяет осуществить Проверку Домена [Domain Validation] (DV). Let's Encrypt не предлагает проверку организации [Organisation Validation (OV)], расширенную проверку [Extended Validation (EV)] и сертификаты с поддержкой поддоменов.

Для работы LsE опирается на небольшую команду и автоматизирует процесс взаимодействия с пользователями, чтобы сделать выдачу сертификатов бесплатной. Здесь нет прямой поддержки.



➤ Спасибо \$IDEITY за блок-схему.

Что за формат?

Файлы SSL могут быть в разных форматах и предлагать ряд расширений. Было бы идеально, если бы расширение файла отражало используемый формат, но это не так. Не думайте, что если расширение правильное, то и формат правильный. Давайте рассмотрим форматы и выбранные расширения.

PEM: (privacy enhanced email — электронная почта с расширенной конфиденциальностью) это Base64-кодированный формат ASCII-файлов для хранения сертификатов и открытых ключей. Его предпочитают программы с открытым кодом. Формат определяется RFC 1421-24. Ему не удалось

достичь своей изначальной цели в качестве метода обеспечения безопасности электронной почты, но формат контейнеров по-прежнему используется. Файл содержит утверждения `-----BEGIN CERTIFICATE-----` <сертификат или ключ> и `-----END CERTIFICATE-----`. Файлы, содержащие этот формат, могут иметь расширение `.pem`, `.key`, `.csr` или `.cer`. DER — двоичная версия PEM-файла Base64. Формат обычно используется в Microsoft Windows. У файлов с этим форматом расширение `.der`. PKCS#7 — это Base64-кодированный формат ASCII-файлов, используемый обычно Java и Tomcat.

Он не содержит закрытых ключей. Это еще один формат, обычный для Windows. Файл содержит утверждение `-----BEGIN PKCS-----` <сертификат> `-----END PKCS7-----`. Файлы этого формата имеют расширения `.p7b` или `p7c`.

PKCS#12 — двоичный формат, шифрованный, который обеспечивает расширенную безопасность по сравнению с текстовым форматом PEM. Этот формат может содержать сертификаты и ключи. Обычно он используется системами Windows. Файлы, содержащие этот формат, имеют расширения `.pfx` или `p12`.

Поддерживаемые сообществом форумы являются единственной возможностью получить ответ. Процесс сертификации основывается на протоколе ACME, запущенном на вашем web-хосте для получения и установки сертификата.

Пользователи, посещающие одобренный LeS домен, получают шифрование, но не обеспечиваются таким же уровнем доверия, как сертификат с зеленым замком (расширенное подтверждение доменов [Extended Domain Validation]). Недобросовестные люди использовали сертификаты LeS, чтобы улучшать сайты, распространяющие вредоносные программы, обманом убеждая людей, что их сайты официальные. Люди не осознают, что зашифрованное соединение с ненадежным сайтом небезопасно и ведет к плохим последствиям.

На схеме внизу стр. 70 шаги 1, 2, 3 и 4 заменены на ручной процесс LeS (6) посредством поддержки протокола ACME, установленного на web-сервер. Инструкции по проведению пользователя через данный процесс вы найдете на <https://letsencrypt.org/getting-started/>.

Установка OpenSSL

Первое, что нужно сделать для поддержки SSL/TLS — установить пакет ПО, обеспечивающий набор инструментов для управления функциональностью протокола. Для данного руководства команда `openssl` будет обеспечена для вариантов операционной системы (ОС) Red Hat Linux и Debian для Raspberry Pi. На сервере с Red Hat Linux установка ПО выполняется от имени `root`, по следующей команде менеджера пакетов:

```
dnf install openssl
```

Поскольку нашим web-сервером будет Raspberry Pi, понадобится также установить ПО `Apache2`. Перед установкой любого нового ПО лучше всего убедиться, что текущее ПО обновлено:

```
sudo apt-get update
```

```
sudo apt-get install openssl
```

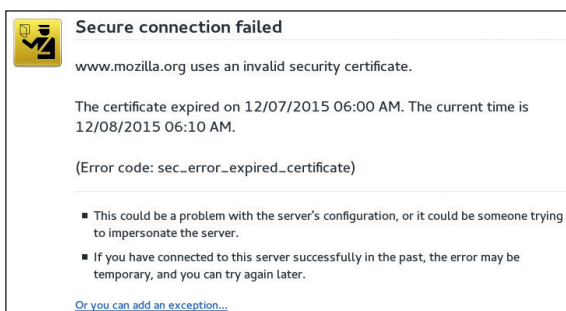
```
sudo apt-get install apache2
```

Первая команда `openssl`, которая нам понадобится, генерирует 2048-битный индивидуальный ключ по алгоритму шифрования RSA и прячет его в файл с именем `webserver.key`. Этот файл ключа критически важен для обеспечения целостности web-сервера, использующего SSL/TLS. Он содержит только что сгенерированные закрытый и открытый ключ для сайта. Желательно изменить права доступа, чтобы защитить его. Из командной строки выполните команды

```
sudo openssl genrsa -out /etc/ssl/private/webserver.key 2048
```

```
sudo chmod 400 /etc/ssl/private/webserver.key
```

Краеугольным камнем безопасной функциональности, обеспечиваемой сертификатами, является конфиденциальность закрытого ключа. Чтобы убедиться в безопасности сайта, важно, чтобы закрытый ключ никому не передавался. Будь он известен



злоумышленнику — скомпрометирован, посетители сайта подверглись бы угрозе нападения.

Используйте ваш любимый текстовый редактор, чтобы взглянуть на содержимое файла `webserver.key`. Содержимое между двумя заголовками будет разным для каждого сгенерированного файла:

```
-----BEGIN RSA PRIVATE KEY----- MIIE.... ..+xA/8zLlcw== -----
END RSA PRIVATE KEY----
```

Настройка OpenSSL

Добавьте следующие изменения в файл `/etc/ssl/openssl.cnf`, используя любимый текстовый редактор. Заключенный в скобки [текст] — это заголовки в файле настройки. Пункты DNS и IP приведены для лабораторного сайта, использованного для разработки нашего учебника:

```
[req]
req_extensions = v3_req
default_md = sha256
[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature,
keyEncipherment
subjectAltName = @alt_names
[alt_names]
DNS = mywebserver.inane.ca
IP = 192.168.100.102
```

Системный администратор может урезать некоторые детали настройки, поддерживая индивидуальный файл настройки `openssl`. Это избавляет от необходимости вводить ответы на вопросы. Большинство администраторов отказываются поддерживать файл настройки, потому что запросы на сертификацию делаются так редко — и зачем тратить время на поддержку файла?

Итак, наш файл настройки SSL/TLS оптимизирован; введите следующую команду для генерации запроса на подпись сертификата (CSR). Обратите внимание на ссылку на файл ключа web-сервера, который был сгенерирован для сайта:

› Дайте себе по рукам, если вы постоянно соглашаетесь на исключения.

Скорая помощь

Хотя 4096 бит надежнее, чем 2048, добавочные биты увеличивают нагрузку на процессор и замедляют процедуру кватирования [hand shake]. Большинство сайтов по-прежнему используют 2048-битные пары.



Данный автор предпочитает следующие расширения файлов: расширение `.csr` для подписи сертификата файла запроса, расширение `.cer` для файла сертификата и расширение `.key` для файла ключей.

```
openssl req -new -out /etc/ssl/certs/webserver.csr -key /etc/ssl/private/webserver.key -config /etc/ssl/openssl.cnf
```

После ввода команды для генерации CSR и нажатия Enter инструмент `openssl` будет ожидать ответов на ряд вопросов. Вот пример из лабораторной конфигурации.

```
Country Name (2 letter code) [XX]: ca
State or Province Name (full name) []: mb
.....
A challenge password []: 123456
An optional company name []:
```

Файл CSR создается в той же директории, где хранится сертификат. Если вы возьмете это себе в привычку при выпуске сертификатов, все члены службы поддержки будут знать, где искать, когда вас нет рядом. Теперь вас может сбить тележка для покупок, а ваши коллеги даже не заметят вашего отсутствия.

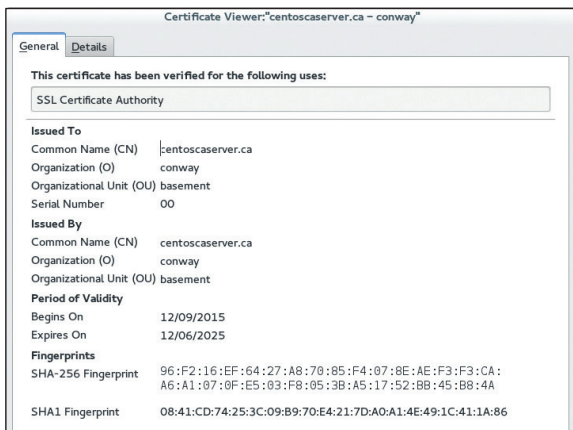
Отступите эту команду `openssl`, если вы хотите проверить содержимое бинарника (команда используется для показа содержимого файла, а не его двоичного формата — иными словами, PEM):

```
openssl req -text -noout -verify -in /etc/ssl/certs/webserver.csr
```

Ниже приведены некоторые строки из вывода, содержащие опции, представленные из командной строки, из файла настроек `openssl` или процесса QA.

Subject: C=ca, ST=mb, L=winnipeg, O=inane, OU=it, CN=webserver.inane.ca	Из командной строки
Public key algorithm: rsaEncryption	Определяет ключ шифрования
X509v3 Subject alternative name: DNS:mywebserver.inane.ca, IP Address:192.168.100.102	Из файла настройки SSL
Signature algorithm: sha256WithRSAEncryption	Из файла настройки и ключа SSL

Проверяя файл CSR в текстовом редакторе, мы заметили знакомые заголовки формата PEM. Важно учесть, что текст между заголовком и нижним колонтитулом у каждого файла свой.



Вот загрузчик сертификата для CA, так что браузеру знакома эта организация.

```
-----BEGIN CERTIFICATE REQUEST-----
MIIC.....6ck=
-----END CERTIFICATE REQUEST-----
```

CSR можно отправить в CA с документацией, которая поддерживает право собственности на домен. Перед созданием сертификата CA также захочет получить платеж. В ответ CA отправит веб-сертификат SSL/TLS, подписанный организацией.

Один из ключевых компонентов сертификата — срок его действия. CA поддерживает сертификат только до этой даты. Если сертификат не обновлен, CA добавляет его в свой список аннулированных сертификатов [certificate revocation list] (CRL), показывая, что сертификат ими больше не поддерживается. Использование устаревшего сертификата — это один из случаев, когда проверка сертификата в браузере может завершиться неудачей.

Новый файл сертификата или, выражаясь корректнее, ссылка на получение сертификата отправляется из CA. Используйте команду `openssl`, чтобы прочитать его содержимое. Открыв его в текстовом редакторе, вы увидите двоичный формат файла:

```
openssl x509 -text -noout -in /etc/ssl/certs/webserver.cer
```

Сравним сгенерированный файл `.csr` и отправленный из CA файл `.cer`. CA подписали сертификат, тем самым изменив подпись. Исходному файлу нельзя было доверять — в отличие от файла из CA или другой удостоверяющей организации. CA сообщает всем, что открытому ключу в данном сертификате можно доверять, поскольку CA имеет веские причины доверять ему.

Установка web-сервера Apache

Возможно, вы помните, что при установке пакета `openssl` на Pi мы установили `Apache2`. По умолчанию `Apache2` на Pi не настроен на поддержку SSL/TLS. Для поддержки SSL/TLS в настройку `Apache2` надо добавить соответствующий модуль. Затем создайте директорию, которую сайт будет использовать для файлов, и скопируйте в нее индексный файл. Содержимое этого файла отображается в браузере при запуске сайта:

```
sudo a2enmod ssl
sudo mkdir /var/www/html/
sudo cp /var/www/index.html /var/www/html/index.html
```

Сделайте копию файла настройки SSL/TLS по умолчанию, предоставленного с установкой `Apache2`, и измените эту копию для поддержки новой настройки:

```
cp /etc/apache2/sites-available/default-ssl /etc/apache2/sites-enabled/default-ssl.conf
```

Используя любимый текстовый редактор, внесите следующие четыре изменения в файл.

```
» Изменение 1: Вставьте в начало файла перед всем остальным
ServerName {укажите здесь имя сервера, убрав скобки}

» Изменение 2: Найдите и замените
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www
    <Directory />
```

Есть ли у вас проблемы с доверием?

Браузеры доверяют основным удостоверяющим центрам (CA). Чтобы получить право стать основным CA, компания должна соответствовать стандартам безопасности и аутентификации и выдержать проверку соответствия стандартам, установленным браузерами. У браузеров может быть список предустановленных надежных CA, известный как Хранилище доверенных корневых CA [Trusted Root CA store].

Когда CA выпускает цифровой сертификат, тот содержит открытый ключ владельца. Закрытый ключ, естественно, недоступен для публики. Сертификатом CA удостоверяется то, что открытый ключ, содержащийся в сертификате, принадлежит лицу или организации, запросившей сертификат. CA подтверждает информацию заявителя, так что пользователи и те, кто от них зависит, могут доверять информации, содержащейся в сертификатах CA.

Создать сертификат нетрудно. Браузеры доверяют только тем сертификатам, которые выданы организацией из их списка надежных CA. CA может удостоверить правильность информации в CSR. С помощью расширенного удостоверения, полного удостоверения организации или аутентификации домена CA старается верифицировать информацию, попросив заявителя предоставить дополнительную идентификацию.

Выбранный инструмент

Oracle имеет свой любимый инструмент для управления сертификатами под названием *keytool*, часть Java development toolkit (JDK), которая содержит этот инструмент. Их намерением было создать файл с именем **keystore**, который хранит ключи и все сертификаты, необходимые для установления доверия.

Для начала создайте пару из закрытого и открытого ключей в файле, именуемом хранилищем ключей [keystore]:

```
/usr/bin/keytool -genkey -alias keypair -keyalg RSA
-keysize 2048 -storetype JKS -validity 99999 -key
pass Pa$$w0rd -storepass Pa$$w0rd -keystore /etc/
pki/java/private/webserver.keystore -dname
```

```
"CN=mywebserver.inane.ca, OU=inane, O=IT,
L=WINNIPEG, S=MB, C=CA"
```

Далее создайте CSR для сервера с помощью ключей из keystore. Обратите внимание на использованный здесь псевдоним. Этот псевдоним вам понадобится, когда вы будете импортировать сертификат:

```
/usr/bin/keytool -certreq -keyalg SHA2withRSA -alias
webserver -validity 3650 -storetype JKS -keypass
password -storepass password -file /tmp/webserver.
csr -keystore /etc/pki/java/private/webserver.keystore
-dname "CN=mywebserver.inane.ca, OU=inane, O=IT,
L=WINNIPEG, S=MB, C=CA" -ext SAN=dns:webserver,
ip:192.168.100.102
```

Затем создайте цепь доверия, импортировав цепь сертификата удостоверения CA в keystore.

```
/usr/bin/keytool -trustcacerts -importcert -alias cen
toscaserver -file /etc/pki/java/certs/centoscaserver.
cer -keystore /etc/pki/java/private/webserver.keystore
-storepass Pa$$w0rd
```

И, наконец, импортируйте сертификат web-сервера, созданный CA:

```
/usr/bin/keytool -trustcacerts -importcert -alias
webserver -file /etc/pki/java/certs/webserver.cer
-keystore /etc/pki/java/private/webserver.keystore
-storepass Pa$$w0rd
```

Подробности вы можете раздобыть на <https://docs.oracle.com>.

```
Options FollowSymLinks
```

```
AllowOverride None
```

```
</Directory>
```

На

```
<VirtualHost {укажите здесь IP-адрес своего web-сервера,
убрав скобки} :443>
```

```
ServerAdmin webmaster@localhost
```

```
<Directory /var/www/html/>
```

```
Options FollowSymLinks
```

```
AllowOverride All
```

```
</Directory>
```

» Изменение 3: Установите уровень журналирования, обеспечивающий более подробную информацию.

```
LogLevel debug
```

» Изменение 4: Обновите параметры, отразив следующее:

```
SSLCertificateFile /etc/ssl/certs/webserver.cer
```

```
SSLCertificateKeyFile /etc/ssl/private/webserver.key
```

Выйдите и перезагрузите сервер, введя новую конфигурацию:

```
/etc/init.d/apache2 restart
```

Если у вас возникли проблемы, проверьте файл журнала `/var/log/apache2/error.log`, чтобы отследить проблему. Запись для успешного перезапуска показана ниже:

```
[info] Loading certificate & private key of SSL-aware server
```

```
[debug] ssl_engine_pphrase.c(470): unencrypted RSA private key -
pass phrase not required
```

```
[info] Configuring server for SSL protocol
```

```
[debug] ssl_engine_init.c(365): Creating new SSL context
(protocol: SSLv3, TLSv1, TLSv1.1, TLSv1.2)
```

```
[debug] ssl_engine_init.c(603): Configuring permitted SSL ciphers
[HIGH:MEDIUM:!aNULL:!MD5]
```

```
[debug] ssl_engine_init.c(316): Configuring TLS extension handling
```

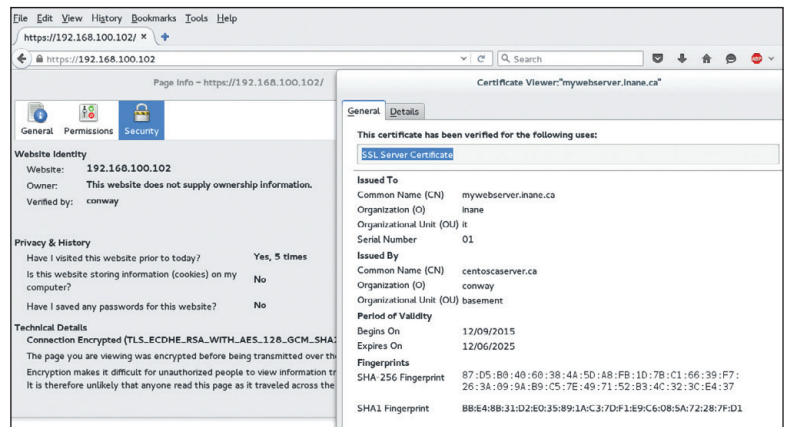
```
[debug] ssl_engine_init.c(736): Configuring RSA server certificate
```

```
[debug] ssl_engine_init.c(782): Configuring RSA server private key
```

```
[notice] Apache/2.2.22 (Debian) mod_ssl/2.2.22 OpenSSL/1.0.1e
configured -- resuming normal operations
```

Если у вас голова пошла кругом, вот вам краткая сводка. На нашем web-сервере Pi мы установили пакет *openssl* и использовали набор инструментов для создания пары открытого и закрытого ключа, которая использовалась для генерирования CSR. CSR был представлен уполномоченным лицам вместе с подтверждением права собственности на домен. В CA создали подписанный сертификат. После перемещения сертификата на web-сервер настройка сервера *Apache2* была изменена для поддержки сертификата SSL. Теперь мы готовы вкушать плоды нашего труда.

Если вы попытаетесь подключиться к web-серверу с текущей настройкой, браузер выдаст ошибку исключения. На нашем уроке



не использовался общедоступный CA для получения сертификата SSL.

Браузеры поддерживают подборку действующих организаций CA. Поскольку используемая в этом руководстве CA не входит в список, браузер показывает исключение. Пользователь вручную принимает сертификат, утверждая, что в нем всё хорошо — я называю это «exception-clickthru [прощелкивание исключений]» и не одобряю подобную практику. Бездумно соглашаясь на исключения, пользователь получает шанс зайти на потенциально опасный сайт, несмотря на то, что механизм SSL/TLS проверки выдал предупреждение. Не игнорируйте его!

Как мы докажем, что это работает? Чтобы преодолеть недостаток, заключающийся в неиспользовании надежной организации для создания сертификата для руководства, в браузер был загружен сертификат для CA, использованного при создании сертификата сервера. Браузер может подтвердить созданный сертификат, потому что у него есть доверие файла браузера.

Теперь при нашем доступе к сайту представляется сертификатированный SSL, подписанный нашей CA. Браузер его признает, и не надо куда-то щелкать.

Ну что, народ, вот и всё. Резюмируем наши достижения: мы создали пару ключей и использовали ее для создания CSR, где содержится открытый ключ нашего сайта. После получения нашего CSR CA выполнит аудит для установления доверенных отношений с заявителем посредством подтверждения информации. Затем CA выпустит сертификат, подписанный им, что создаст цепь доверия. Сайт введет сертификат в эксплуатацию, чтобы браузеры могли использовать HTTPS для подключения и подтверждения личности и установки канала шифрования для обеспечения безопасности данных. LXF

» Вы можете прочитать, что видит браузер.

IoT: Дом, который послушен нам

Завершая свою серию статей, Максим Черепанов помещает на web-панели слайдер для регулировки яркости светодиода.



Наш эксперт

Максим Черепанов познакомился с Linux в 2008 г. Через 9 лет сделал его своей профессией: поменял транспортную коммерцию на программирование. Интересуется Web, Asterisk и SoC. Отдыхает от них при написании прошивок для микроконтроллеров и сборке ламповых усилителей.



Д а пребудет с вами сила, и да будет Послушный Дом удобным! В прошлый раз мы научились «размножить» каналы, к которым вы сможете подключать новые и новые актуаторы и датчики. Это замечательно, и позволит вам немедленно опутать своими сетями управления всё, что придет вам в голову.

Однако про удобство можно покамест только мечтать. Вряд ли можно назвать комфортным способ управления посредством ввода команды `./PCA9685.sh pwm 1 60` по SSH. Так мы подаем на канал №1 ШИМ величиной в 60% от максимальной.

Но какой была ШИМ до нашей команды? Как можно выполнить желание ваших домочадцев «добавить чуть-чуть освещения»? Или плавно увеличивать накал лампочки и смотреть — хватит или нет? Только путем последовательного ввода всё больших и больших величин в команду... проще уж поставить ручной диммер.

Сегодня мы будем дорабатывать систему управления с точки зрения удобства (делать ее «юзер-френдли»).

От PHP к PHP + JS

Те, кто читает мои заметки, знает мою благосклонность к языку PHP. Он превосходит тем, что одинаково работает во всех браузерах, не нарушает приватность пользователя, лишен возможности вмешиваться в программы и безопасен. Поэтому первый вариант управления ПД был написан на нем.

Затем для индикации состояния наших нагрузок/актуаторов мы применили JavaScript и Ajax, но сделали это всего одной функцией, скромно вставив ее в основной файл, который был написан на PHP.

Пришло время сделать полноценную модель клиент-серверного приложения. Вы сможете развивать ее для своих нужд так, как вам захочется.

Для начала, как водится, немного теории.

Язык PHP предназначен для написания скриптов, которые выполняются на стороне сервера, а в браузер клиента отправляются не сам скрипт, а только результаты его работы. Это означает, что пользователь, загрузивший страницу вашего сайта, никогда не увидит исходный код скрипта и может даже не догадываться, что страница создана динамически.

В свою очередь, на языке JavaScript создаются исключительно клиентские скрипты, которые исполняются в браузере клиента. Для сервера скрипты JavaScript — это обычные текстовые данные, ничем не отличающиеся от прочего содержимого страницы. И написаны они могут быть как статически, в виде HTML, так и динамически, при помощи PHP, и затем уже в виде HTML-разметки отправляются в браузер клиента.

Чтобы выполнять скрипты, написанные на языке JavaScript, не требуется никаких дополнительных программ — всё необходимое для работы скрипта уже имеется в браузере (если, конечно, это не браузер пятнадцатилетней давности). Сразу нужно сказать, что в разных браузерах JavaScript ведет себя немного по-разному. Это не касается самого языка — одни и те же конструкции будут выполняться одинаково. Всё дело в средствах, которые браузер предоставляет скрипту: так, например, многие скрипты работают в браузере *Mozilla Firefox*, но выдают ошибки в других браузерах, например, в *Apple Safari*.

Одним из самых важных вопросов является вопрос взаимодействия сервера и клиента, т. е. передача данных между PHP на сервере и JavaScript в браузере пользователя. Надо понимать, что браузер соединяется с сервером только в момент получения или передачи данных и не имеет постоянного канала связи. И если за время, когда пользователь рассматривал страницу, на сервере что-то изменилось, то пользователь этого не увидит, если не предпринять для этого специальных действий.

Вспомните, у нас была функция, которая 1 раз в секунду опрашивала сервер на актуальность данных:

```
$(document).ready(function(){
    show();
    setInterval('show()',1000);
});
```

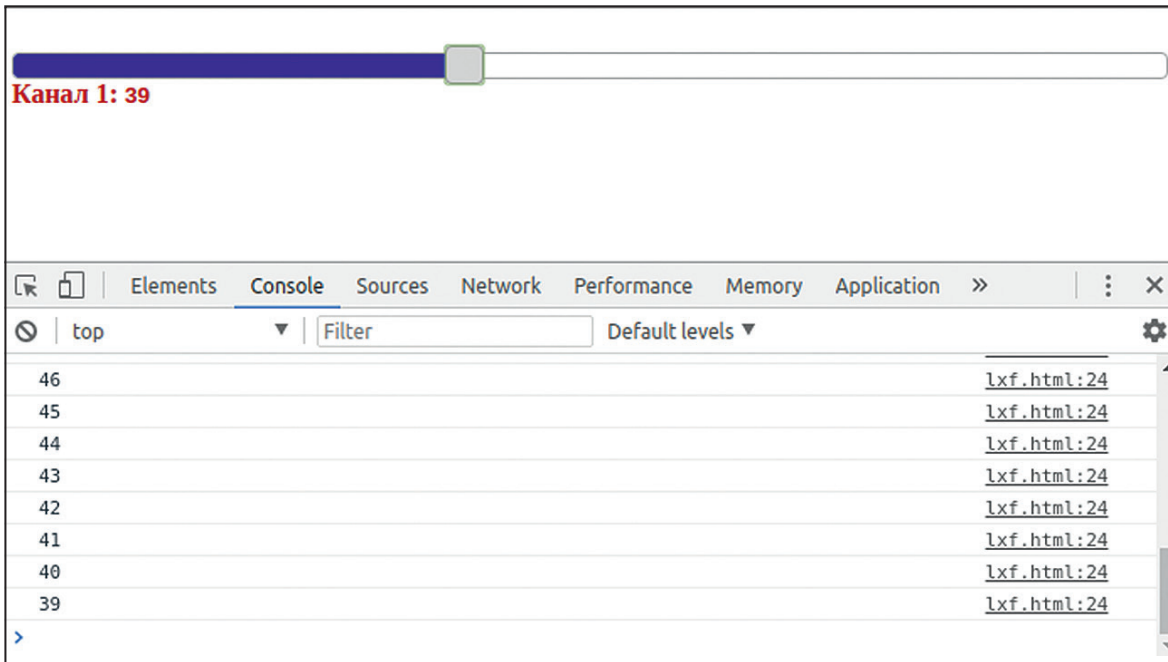


Рис. 1. Слайдер и консоль JS.

А сама функция, вызываемая каждую секунду, имела следующий вид:

```
function show()
{
$.ajax({
// тут тело функции,
})
});
```

`$.ajax` — краеугольный камень всей конструкции, про него мы тоже уже рассказывали.

Итак, вооружившись этими знаниями, приступим.

Техническое задание

Не будем ставить себе больших задач, сделаем просто и эффективно: на web-панели должен быть слайдер (или несколько, в зависимости от желания), передвигая который мы сможем регулировать интенсивность свечения светодиода, подключенного к плате с PCA9865.

В этих нескольких словах заключен достаточно большой объем работы:

- 1 Нарисовать сам слайдер. Это как минимум линия и какой-то графический объект на ней, который можно «ухватить» мышью и двигать вправо-влево (вверх-вниз).
- 2 Браузер должен отслеживать состояние слайдера, и при его изменении отправлять данные на сервер.
- 3 Сервер должен принять данные от клиента-браузера и изменить данные в регистрах PCA9865 так, чтобы они соответствовали присланному.
- 4 После изменения нужно считать записанные данные и передать обратно в браузер для контроля исполнения.

Не мало, правда? Меня особенно смутила изначально необходимость рисования графического объекта. Статичный-то нарисовать не проблема, а как его менять?

Благо, что всё уже нарисовано до нас. Знакомьтесь:

JQuery UI

jQuery UI — это библиотека на основе jQuery, реализующая более 20 гибких плагинов и предоставляющая продуманную систему

оформления и средства для создания собственных плагинов с нуля или на основе уже существующих (что такое jQuery, мы с вами уже знаем). Среди всего прочего в этой библиотеке присутствует и слайдер (ползунок).

Я надеюсь, что вы уже освоились с написанием HTML-страниц, и позволю себе обойтись без написания заголовочных тегов типа `<head>` и `<body>`.

Итак, для включения jQuery UI в вашу страницу надо разместить в начале (можно в заголовке, можно в теле) страницы (файл с расширением `.html`) следующие строки:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"></script>
<link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" rel="stylesheet" type="text/css">
<script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js"></script>
```

Разумеется, к моменту вашего чтения версии могут измениться, так что это надо будет проверить. Возможно также заранее скачать все файлы и грузить их локально (лично я именно так и поступаю).

После этого вставьте в тело (внутри тега `<body>`) следующий текст:

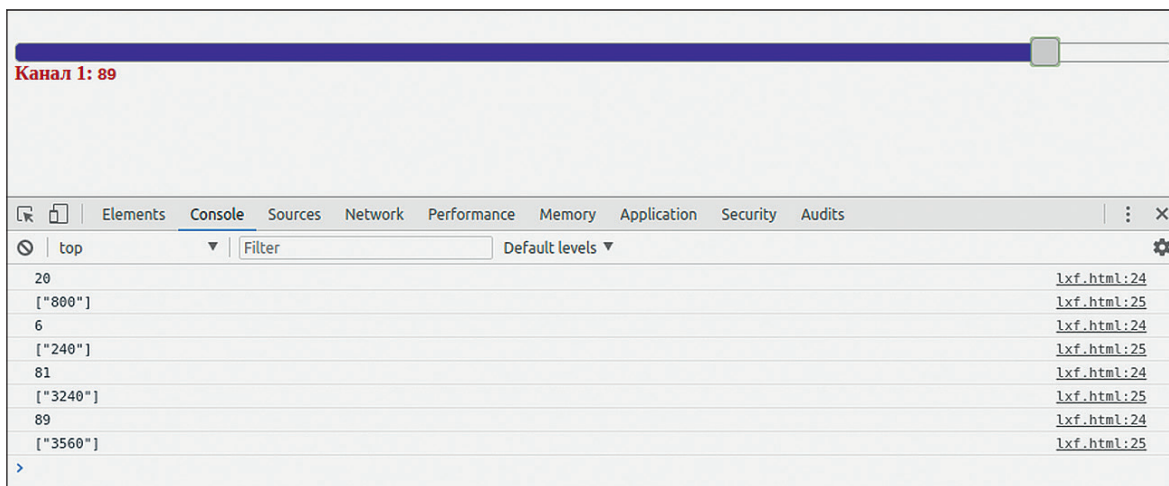
```
<div id="slider-range1"></div>
<label for="amount1" style="border:0; color:#ff0000; font-weight:bold;">Канал 1: </label><input type="text" id="amount1" style="border:0; color:#ff0000; font-weight:bold;"><br>
```

Таким образом мы задали в теле HTML-страницы место (слой, `div`), в котором будет красоваться наш слайдер, и место для отображения текста и численной величины. Если вы посмотрите на эту страницу в браузере, то пока что увидите только текст «Канал 1» красного цвета.

Дальше вставьте такой текст:

```
<script>
$("#slider-range1").slider({
range: "min",
value: 50,
min: 0,
max: 100,
```

Рис. 2. Ответ с сервера в консоли.



```
animate: "slow",
slide: function (event, ui) {
    $("#amount1").val(ui.value);
    console.log(document.getElementById('amount1').value); //
отладка
    $.get('server.php', 'volume1=' + document.
getElementById('amount1').value);
}
});
$("#amount1").val($("#slider-range1").slider("value"));
</script>
```

Ого-го! Мы уже начали пользоваться jQuery UI, сами того не замечая. Теперь, если вы откроете свою страницу, то, как по волшебству, на ней появится анимированный слайдер! Параметры его настроек вы поймете сами; где изменить цвета, тоже должны догадаться.

Вставленный текст состоит из нескольких функций jQuery, каждая из которых делает свою определенную задачу. Первая задает слайдер, его параметры и реакцию на изменение его величины (`$.get`). Вторая вписывает значение слайдера в `div amount1`.

Обратите внимание на строку с пометкой «отладка». Она выводит в консоль JavaScript значение, которое в текущий момент имеет слайдер. Чтобы эту консоль увидеть, нажмите (в *Google Chrome*) одновременно `Ctrl + Shift + J`, и внизу браузера появится окошко, в котором при движении слайдера будет появляться упомянутое значение (рис. 1)

Идем дальше. Вернемся к строке, начинающейся с `$.get`. Это функция jQuery, которая асинхронно посылает на сервер указанное значение, а конкретно в файл `server.php`, который должен быть расположен в той же директории, что и ваш заглавный HTML-файл. В него методом `get` мы посылаем строку `volume1='значение слайдера'`, например, `volume1=32`.

Немного про красивое слово «асинхронно»: «синхронный запрос» означает, что после отправления данных на сервер и до ответа сервера главный поток будет «заморожен»: посетитель не сможет взаимодействовать со страницей — прокручивать, нажимать на кнопки и т. п. После получения ответа выполнение продолжится со следующей строки. «Асинхронный запрос» означает, что браузер отправит запрос, а далее результат нужно будет получить через обработчики событий. Пока приходит ответ, страница будет жить в обычном режиме.

По умолчанию все запросы, выполняемые при помощи jQuery, асинхронные. Это можно изменить, применив следующий код перед вызовом функции:

```
$.ajax({ async: false });
```

Имейте в виду, что после этой директивы все запросы будут выполнены синхронно, и если вам нужен будет асинхронный запрос, предварите его такой же директивой с параметром, установленным как `true`.

Пора взяться за серверную часть.

Назад к PHP

Создайте в той же директории web-сервера, что и первый, файл `server.php`, и наполните его следующим содержанием:

```
<?php
$volume1 = $_GET['volume1'];
if (isset($volume1) && $volume1 != '') (`sudo /home/
bananapi/9685.sh pwm 0 $volume1 2>&t1`);
exec("sudo /home/bananapi/9685.sh get 0", $channel1);
echo json_encode($channel1);
```

Пробежимся построчно: забираем из массива `GET` значение переменной `volume1`; если она установлена и не пустая — запишем в наш чип PCA9865 то, что пришло. Затем считаем фактическое значение и выведем в стандартный поток строку, имеющую декодированное в JSON представление этого значения, который потом подхватит клиентская часть.

Давайте проверим, что нам ответит сервер; пока достаточно это сделать в консоли JS. Функция `$.get` принимает несколько параметров; одним из них является функция обратного вызова, в нее мы отправим данные с сервера. Меняйте строку:

```
var jqxhr = $.get('server.php', 'volume1=' + document.
getElementById('amount1').value, function () {console.log(jqxhr.
responseText)});
```

Ух ты, какой кошмар — на первый взгляд! Но ничего кошмарного: мы добавили переменную, которая получит значение, прилетевшее с сервера, и функцию-обработчик (она только выводит в лог величину переменной (через свойство `responseText`)).

Пробуйте. Если вы следовали за мной по пятам, то у вас тут должна возникнуть ошибка — у скрипта `9685.sh` нет опции `get`, мы ее не предусмотрели в прошлой статье, поэтому сервер ничего не отправит.

Исправляемся, добавив в скрипт `9685.sh` еще одну опцию в `case` (надеюсь, что вы меня поняли):

```
get)
PIN=$2
VALUE=$((256*$(($16#`i2cget -y 0 0x40 $((($PIN*4+9)) | cut -c
3-4`))+$(($16#`i2cget -y 0 0x40 $((($PIN*4+8)) | cut -c 3-4`))))
echo "$VALUE"
```

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Да-да, именно так! Эти сложности связаны с преобразованием величин между шестнадцатеричной и десятичной системами счисления, а также особенностями работы с переменными в интерпретаторе *Bash*.

Не пропустите знаки, и обратите внимание — кавычки тут обратные, а не апострофы!

После этого всё должно заработать. Ну, как всё — вывод в консоль (рис. 2).

Несмотря на кажущиеся простоту и бесполезность, только что я дал вам (тем, у кого его не было) в руки мощнейшее средство: вы можете влиять на состояние сервера из web-консоли, не перезагружая страницу, и получать ответ на свое действие!

На основе этого вы можете полностью пересмотреть концепцию своего ПД, и выполнить всё в клиент-серверном стиле, не заставляя браузер перезагружать страницу.

Но давайте сделаем нашу «мордочку» чуть симпатичнее.

Добавьте после `div`'ов для слайдера еще один:

```
<div id="display" style="border: groove">Сервер не был опрошен</div>
```

При начальной загрузке страницы вы увидите фигурирующее в вышеприведенной строке сообщение. Оно не несет особо важной информации, но наглядно покажет нам изменения в процессе регулировки.

Измените функцию обратного вызова:

```
function () {
    console.log(jqxhr.responseText)
    document.getElementById('display').innerText = 'С сервера получено значение: ' + (JSON.parse(jqxhr.responseText))[0];
}
```

Видите, ее пришлось записать уже в несколько строк, хотя первая только информационная. Теперь попробуйте: у вас должна получиться картинка как на рис. 3.

Собственно говоря, с этого момента вы можете делать с данной переменной всё, что вам угодно.

Обратите внимание также на то, что я не стал приводить ее значение к процентам, она получена в чистом виде 12-битной ШИМ. Для приведения к процентам это значение потребуется разделить на 460.

Вкусности JQuery UI

А ведь вовсе не зря я в последнем примере задал рамочку в слое (`style="border: groove"`). Мне нужно, чтобы границы слоя были заметны. Потому что этот слой «легким движением руки превращается» в перемещаемый!

Впишите в любое место внутри тегов `<script></script>` следующую строку:

```
$("#display").draggable();
```

Перегрузите страницу. И попробуйте мышью перетаскать слой с информацией от сервера в другое место. Он перемещается (рис. 4)!

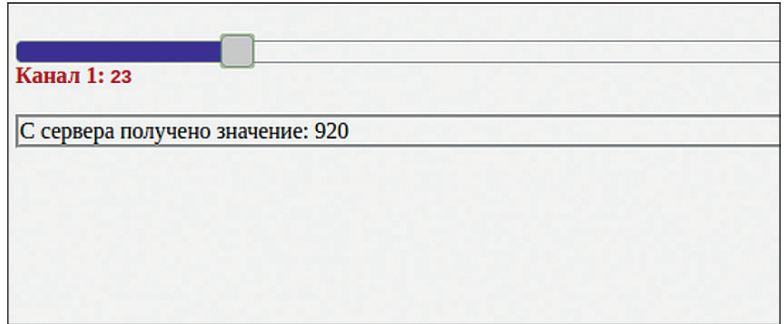
Если честно, когда я попробовал это в первый раз, то был просто потрясен. Теперь можно перенастраивать вид панели прямо в браузере, не прерывая работы.

У JQuery UI еще очень много плюшек, каждая из которых заслуживает отдельной статьи. Но все они оправдывают девиз JQuery «Кода меньше, дела больше!»

Запуск PCA9865

Хочу напомнить вам о том, что PCA9865 перед использованием следует запустить. А для этого надо запустить скрипт с аргументом `start`. Это мы делали в прошлый раз из консоли, и всё прошло гладко.

Но есть одни грабли, на которые я наступаю с регулярностью раз в год или два — запуск скриптов из *crontab*. Казалось бы,



► Рис. 3. Ответ с сервера в слое.

чего там такого, вписал строчку с параметром `@reboot` — и дело в шляпе.

Итак, пишем:

```
crontab -e
Добавляем строку для запуска:
@reboot /home/bananapi/start_9865.sh
Впишем в скрипт
#!/bin/bash
/home/bananapi/9865.sh start
```

Перезагружаем — и с важностью думаем, что всё запущено. Ан нет. А причина проста: в скрипте `9865.sh` команды написаны без абсолютного пути к исполняемым файлам. И хоть строка в *crontab* и указывает абсолютный путь к скрипту, но если в самом скрипте есть хотя бы одна команда без этого пути, такая команда будет проигнорирована.

Поэтому мне проще оказалось написать скрипт первоначального запуска с абсолютными путями:

```
nano /home/bananapi/start_9865.sh
#!/bin/bash
/usr/sbin/i2cset -y 0 0x40 0x00 0x00
/bin/sleep 1
/usr/sbin/i2cset -y 0 0x40 0x00 0xa0
exit 0
```

Теперь всё будет работать нормально.

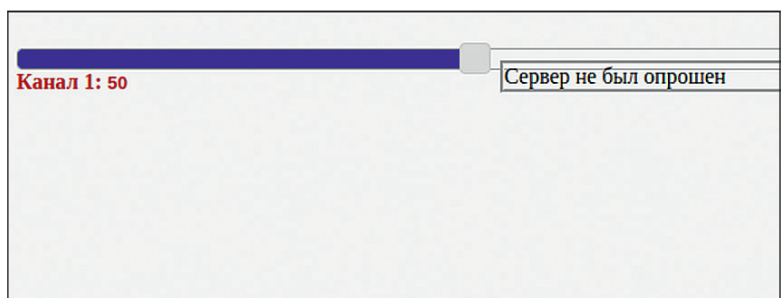
Друзья в минуту расставанья...

...с надеждой шепчут «До свиданья!»

Пришла пора заканчивать серию статей по Послушному Дому. Мне было приятно их писать. Надеюсь, что кому-то эти заметки открыли путь к тому, как можно делать сложные вещи просто, а кто-то познакомился с новыми понятиями и технологиями передачи информации.

И еще, что хотелось мне напомнить вам: если чего-то нет в операционной системе, то это можно написать самому. Для этого не надо иметь многолетнего образования и сверхмощного компьютера. Достаточно читать **LXF** и иметь одноплатник типа Raspberry/Banana Pi.

И да пребудет с вами Linux! **LXF**



► Рис. 4. Перетаскиваемый объект.

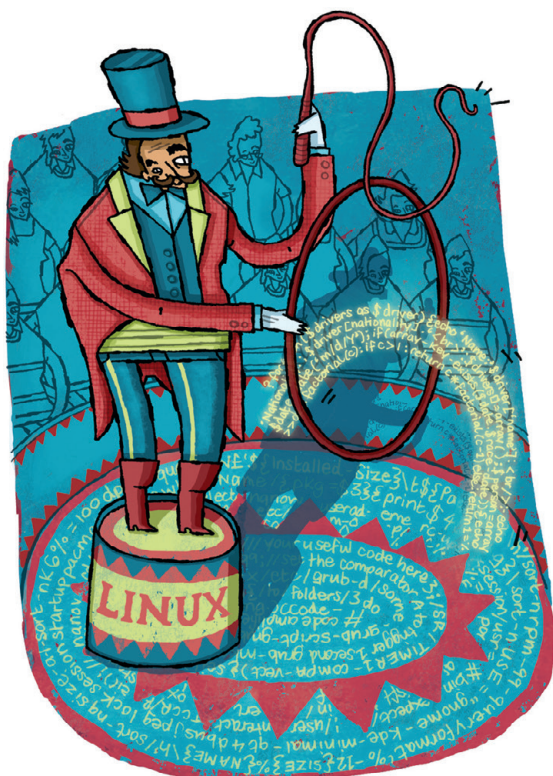
BASIC: ОЖИВИМ КОД BBC

Нейт Дрейк помогает вам вновь пережить славные дни программирования на BASIC для микрокомпьютера BBC, наскоро зачесав свою лысину.



Наш эксперт

Нейт Дрейк посвящает эту статью своему отцу, который когда-то сказал ему, что никто не будет платить ему за то, что он сидит и играет в видеоигры.



дни текстовые игры кажутся несовременными, некоторые наши читатели, вероятно, помнят эти бурные деньки, когда угодливые виртуальные слуги жаждали доложить вам о положении дел в Шумере.

Игра *Hammurabi* была создана на языке программирования BASIC. BASIC (Beginners All-Purpose Symbolic Instruction Code — Универсальная система символического кодирования для начинающих) изначально появился в 1964 г. и стал исключительно популярен на таких домашних компьютерах, как ZX Spectrum, Commodore 64 и BBC Micro. В противоположность жестким требованиям к квалификации, предъявляемым к тем, кто допускался к работе с компьютером в те дни, BASIC был написан специально для пользователей без технической подготовки, и его на удивление легко освоить.

На нашем уроке мы начнем исследовать запуск простых команд на BBC BASIC, с перспективой создания вашей собственной игры в стиле *Hammurabi*. Сюда будет входить ввод и вывод текста, внесение случайности и объявление переменных, а также выяснение того, как определить процедуры, чтобы они выполняли определенные группы задач.

Хотя частично наш проект предназначен для старшего поколения читателей, у которых слезы наворачиваются при воспоминании о своем дебюте, этот учебник также станет отличным выбором для родителей и учителей, желающих познакомить детей с программированием. Некоторые команды, например, для определения процедур, совершенно идентичны командам в Python, что делает BASIC идеальным тренировочным полигоном. Весь приведенный ниже код можно скачать с <https://github.com/natedrake/bbcbasicvsamples>.

Глоток Brandy

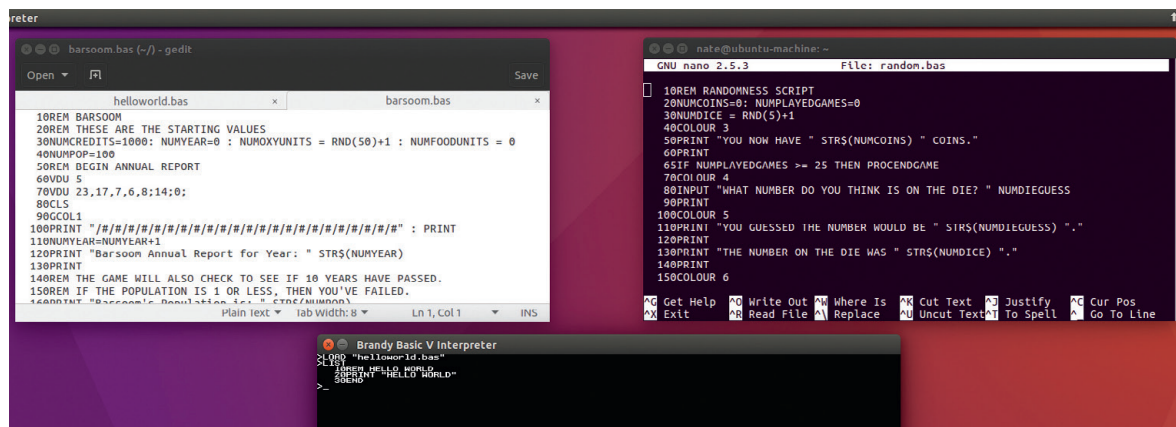
В этом проекте мы сосредоточились на использовании Brandy BASIC V Interpreter (<http://sourceforge.net/projects/brandy>), который

Скорая помощь

В *Brandy* введите 'LIST', чтобы просмотреть весь написанный код. Вы также можете ввести 'CLS', чтобы удалить текст с экрана, если вы запутались.

Файлы BASIC можно редактировать в текстовом редакторе, Терминале или прямо в *Brandy*.

Как бы вам понравилось денек побыть царем — а может, лет десять? Это сценарий, в который вы попадаете, начиная играть в текстовую стратегию *Hammurabi*, где благополучие ваших подданных зависит от ваших способностей собирать пуды зерна и кормить им голодающее население. Хотя в наши



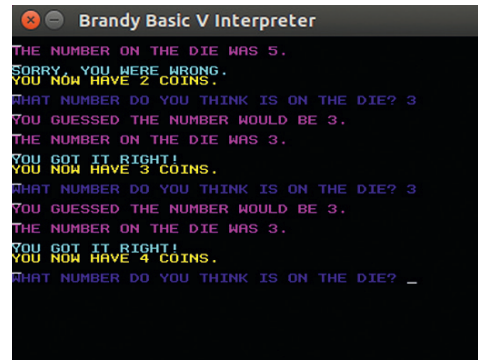
Плеснем краски

Хотя интерпретатор BASIC довольно громоздкий и текстовый, при желании вы можете добавить немного цвета. Простейший способ сделать это — использовать команду `COLOUR`. В Терминале введите `brandy`, а затем, например, `LOAD myname.bas`, чтобы загрузить свой скрипт. Введите `15 COLOUR 1`, чтобы дать указание интерпретатору выводить весь текст красным цветом, начиная со строки 15. Введите `RUN`, чтобы увидеть результат. Возможно, по окончании работы программы вы увидите, что и весь вводимый вами текст стал красного цвета. Подумайте о том, чтобы ближе к концу вашей программы (перед оператором `END`) вернуть тексту исходный белый цвет, введя строку `COLOUR 7`. Если вы хотите, чтобы ваши строки аккуратно нумеровались через 10, скомандуйте `RENUMBER`. Желая сделать

текст более красочным, воспользуйтесь командой `VDU`, чтобы рисовать текст как графику. Вначале убедитесь, что в вашей программе есть команда `VDU 5`. На следующей строке поместите команду `VDU 23,17,7,6,8;X;0;`, где 'X' — количество пикселей в высоту, заданное для ваших букв. Код для проекта `Barsoom`, который мы обсуждаем, в основном тексте имеет высоту букв 14 пикселей. При использовании режимов `VDU` вы должны вместо `COLOUR` использовать `GCOL` — например, `90 GCOL 1`.

Полный список цветовых кодов — на сайте www.bbcbasic.co.uk/bbcwin/tutorial/chapter07.html.

Те, кто говорит на американском английском: обратите внимание на британский вариант написания 'COLOUR', который используется в BBC BASIC. В конце концов, это BBC.



► Программа оценки шансов `random.bas` использует команду `COLOUR`, чтобы отобразить количество монет и ваших предположений по отдельности.

запускает BBC BASIC (или BASIC V, его менее известное имя) под разными операционными системами. BBC BASIC — язык программирования, созданный для компьютера Acorn Archimedes, и он может также работать на определенных версиях BBC Micro. Самое важное — что интерпретатор *Brandy* доступен в репозиториях Debian и Ubuntu. Этот проект был создан на компьютере с Ubuntu 16.04, но *Brandy* также доступен в репозиториях Raspbian, и при желании вы можете начать программировать на BBC BASIC на своем Pi.

Выбрав свою платформу, откройте терминал и запустите `sudo apt-get update`, затем `sudo apt-get install brandy`. Когда *Brandy* запустится, создавайте себе свою первую программу "Hello World". Как большинство версий BASIC, *Brandy* требует, чтобы каждая строка кода начиналась с номера. При запуске программы эти строки будут выполняться по порядку, если только вы не дадите программе других указаний. По условию первая строка должна начинаться с номера 10, а номера следующих строк увеличиваются с шагом 10. Это позволит вам позднее редактировать код, вставляя при необходимости новые строки.

Используйте `Enter`, чтобы перейти к новой строке. Начните со ввода:

```
10 REM HELLO WORLD
20 PRINT "HELLO WORLD"
30 END
```

Если вы сделаете ошибку, просто перепишите код правильно в следующей строке. Интерпретатор BASIC обновит ее для вас. Введите `RUN`, чтобы увидеть свою программу в действии.

Обратите внимание, что слово `REM` в начале любой строки дает указание интерпретатору игнорировать всё, что написано в остальной части строки. Это отличный способ пояснять ваш код BASIC.

Сохраните свою первую программу BASIC, введя `SAVE <имя-файла>` — например, `SAVE helloworld.bas`.

Теперь выйдите из *Brandy*, введя `QUIT`. Вы увидите ранее сохраненный файл в домашней папке. Можете открыть этот файл в Терминале с помощью `папа <имяфайла>` или в своем любимом текстовом редакторе, например, *Gedit*. Это значительно упростит изменение и редактирование новых строк кода.

Зная, как отображать текст, вы готовы создать интерактивную программу, которая может принимать вводимую информацию.

Сначала создайте файл с именем `myname.bas`, например, запустив `sudo nano myname.bas` в Терминале. Далее введите

```
10 REM КАК ВАС ЗОВУТ?
20 INPUT "Как вас зовут?" STRNAME$
30 PRINT "Привет," STRNAME$ ". Как дела?"
```

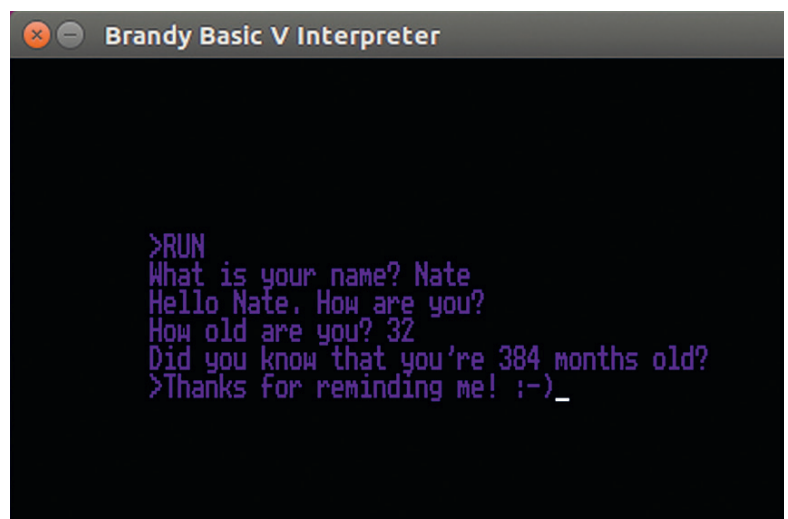
Команде `INPUT` здесь велено предложить вам ввести данные. Имя переменной `STRNAME$` можно затем включить в оператор `PRINT`. Значения строковых переменных всегда заканчиваются символом '\$'; чтобы развить правильную манеру программирования, сделайте это явным, поместив буквы `STR...` перед остальной частью имени переменной.

Сохраните свой текстовый файл и выйдите, а затем заново откройте Терминал. Скомандуйте `brandy myname.bas` для запуска вашей программы. Введите свое имя и нажмите `Enter`, затем ностальгически улыбнитесь, когда интерпретатор BASIC вас поприветствует.

Ввод чисел работает так же, как и для строк. Чтобы это продемонстрировать, заново откройте `myname.bas` и добавьте следующие строки:

Скорая помощь

Используйте команду `PRINT` без параметров, чтобы вставить новую пустую строку перед выводом остального текста. Это облегчит чтение вашей программы.



► Эта программа принимает и строковый, и числовой ввод и вывод данных. Ваш возраст в годах умножается на 12, чтобы вывести возраст в месяцах.

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

```
40 INPUT "Сколько вам лет?" NUMAGE
50 NUMAGEINMONTHS = NUMAGE*12
60 PRINT "А вы знаете, что вам "
STR$(NUMAGEINMONTHS) " месяцев?"
70 END
```

Как видите, единственная разница между запросом цифрового и строкового значений заключается в отсутствии символа '\$' в конце переменной в строке 40. Ради простоты мы также приставили буквы NUM в начале переменной, показав, что это числовая величина; но опять же, это дело личного вкуса.

В строке 50 переменная NUMAGEINMONTHS выражена как умножение на 12 введенного вами возраста. Затем в строке 60 это значение печатается. Вы можете заметить, что числовая величина NUMAGEINMONTHS заключена в выражение STR\$(). Это не обязательно, но указывает интерпретатору считать число текстом, отчего вывод на экран выглядит красивее.

Уделите некоторое время изучению кода, введенного на данный момент, и затем снова запустите программу, как описано выше. На этой стадии поздравьте себя, потому что хотя вы написали всего 10 строк кода, вы уже на полпути к созданию собственной игры на BASIC!

Дело случая!

Большинство игр используют некое количество непредсказуемости. Хотя функция RND в BASIC, возможно, и не обеспечивает чистой форму энтропии, ее вполне хватит для наших целей. Случайная переменная объявляется так же, как вы видели в скрипте 'mupate'; например, A = RND(9)+1 дает указание интерпретатору присвоить переменной A случайное число от 1 до 10.

Вы можете использовать это для создания очень простой игры «Угадай число». Создайте новый текстовый файл с именем **random.bas** и введите следующий код:

```
10 REM RANDOMNESS SCRIPT
20 NUMDICE = RND(5)+1
30 COLOUR 3
```

```
40 INPUT "КАКОЕ ЧИСЛО, ПО-ВАШЕМУ, ВЫПАЛО НА КУБИКЕ?"
NUMDIEGUESS
50 PRINT
60 COLOUR 4
70 PRINT "ВЫ ДУМАЛИ, ЧТО ЧИСЛО БУДЕТ "
STR$(NUMDIEGUESS) "."
80 PRINT
90 PRINT "А ВЫПАЛО ЧИСЛО " STR$(NUMDICE) "."
100 PRINT
110 COLOUR 5
120 IF NUMDIEGUESS = NUMDICE THEN PRINT "ВЫ УГАДАЛИ!"
ELSE PRINT "УВЫ, ВЫ ОШИБЛИСЬ."
130 COLOUR 7
140 GOTO 10
```

Из Терминала запустите `brandy random.bas` для выполнения. Строка 20 объявляет случайную переменную NUMDICE. Далее в строке 40 программа попросит вас угадать число. Затем выведется ответ, и очень простой оператор IF... THEN ... ELSE в строке 120 отобразит результат, показав, правы или нет.

Строка 140 использует ранее не встречавшийся оператор GOTO, чтобы дать указание интерпретатору вернуться к строке 10 и начать программу заново. Будучи предоставлена самой себе, **random.bas** будет продолжать спрашивать у вас случайные числа бесконечно. Используйте клавишу Esc и введите QUIT, чтобы избежать вечного угадывания.

Хотя операторы IF ... THEN... ELSE в строке 120 тоже служат своим целям, они реально полезны для выполнения только одного действия — в данном случае, для сообщения, правы вы или нет в своей догадке.

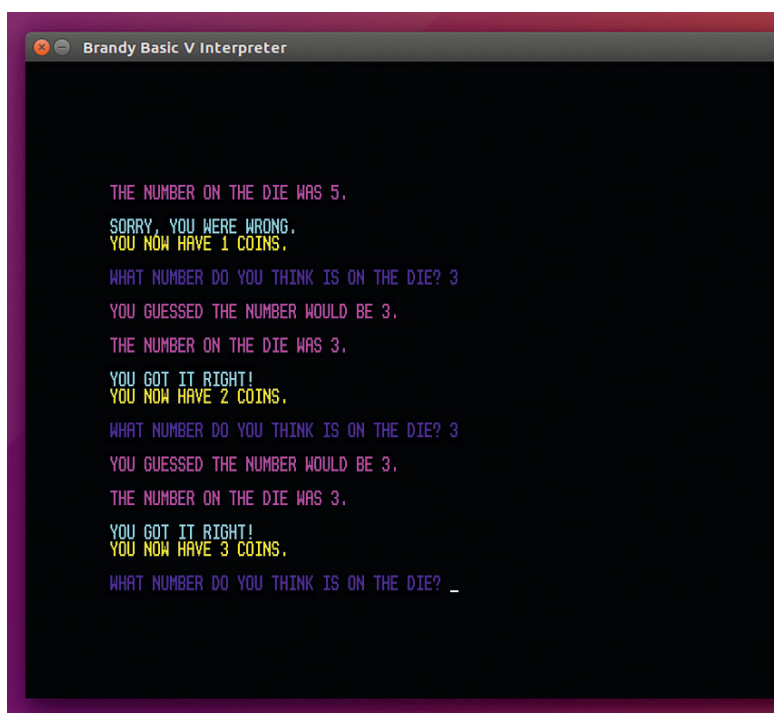
Здесь-то и пригодятся процедуры: с ними вы сможете дать указание интерпретатору выполнить несколько команд и затем вернуться назад по их завершении. Процедуру можно определить, начав любую строку с команды DEF PROC; например, DEF PROCRIGHT. Введите команды для этой процедуры на следующих строках и закончите ENDPROC.

Удалите существующий файл **random.bas** и создайте новый с таким же именем. Вставьте следующий код:

```
10 REM RANDOMNESS SCRIPT
20 NUMCOINS=0
30 NUMDICE = RND(5)+1
40 COLOUR 3
50 PRINT "СЕЙЧАС У ВАС " STR$(NUMCOINS) " МОНЕТ."
60 PRINT
70 COLOUR 4
80 INPUT "КАКОЕ ЧИСЛО, ПО-ВАШЕМУ, ВЫПАЛО НА КУБИКЕ?"
NUMDIEGUESS
90 PRINT
100 COLOUR 5
110 PRINT "ВЫ ДУМАЛИ, ЧТО ЧИСЛО БУДЕТ "
STR$(NUMDIEGUESS) "."
120 PRINT
130 PRINT "А ВЫПАЛО ЧИСЛО " STR$(NUMDICE) "."
140 PRINT
150 COLOUR 6
160 IF NUMDIEGUESS = NUMDICE THEN PROCRIGHT ELSE
PROCWRONG
170 REM *** ПРОЦЕДУРА, КОГДА ДОГАДКА ПРАВИЛЬНАЯ ***
180 COLOUR 7
190 GOTO 30
200 DEF PROCRIGHT
210 PRINT "ВЫ УГАДАЛИ!"
220 NUMCOINS = NUMCOINS+1
```

Скорая помощь

В отличие от других языков программирования, таких как Python, вам не надо определять процедуру до ее вызова: интерпретатор найдет ее в коде для вас.



➤ Каждый раз, когда вы угадываете число, значение NUMCOINS увеличивается на 1.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

```
230 ENDPROC
240 REM *** ПРОЦЕДУРА, КОГДА ДОГАДКА НЕПРАВИЛЬНАЯ ***
250 DEF PROCWRONG
260 PRINT "УВЫ, ВЫ ОШИБЛИСЬ."
270 ENDPROC
280 END
```

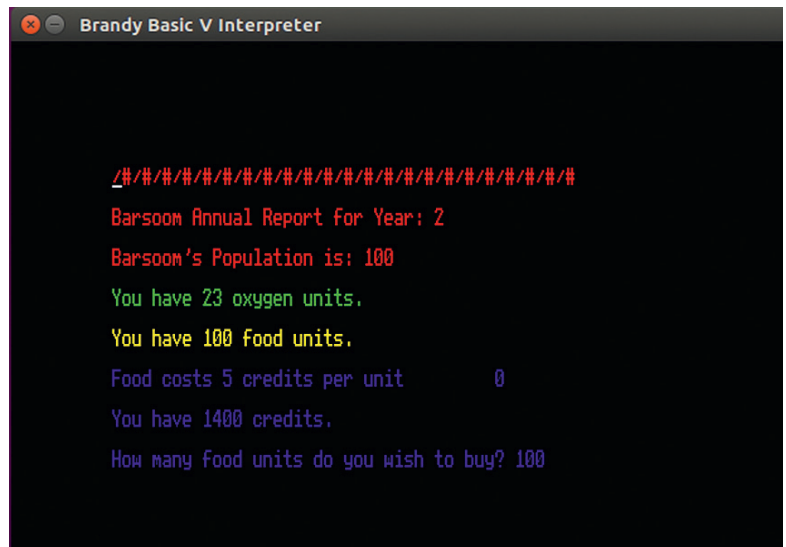
Хотя пока это ваша самая длинная программа BASIC, в ней нет скрытых сюрпризов. Строка 160 содержит оператор IF ... THEN ... ELSE, запускающий PROCRIGHT, если ваш ответ верный, или PROCWRONG, если неверный. Как только процедуры будут выполнены, если ответ верный, игрок получает виртуальную монету. Затем игра перезапустится и покажет, сколько монет вы использовали. Поскольку игра всё еще ведется в бесконечном цикле, как и ваш первый проект BASIC, вы, возможно, захотите немного изменить код, создав новую процедуру, которая будет выполняться при достижении NUMCOINS определенного значения. Вы можете найти одну из реализаций на Github с помощью предоставленной ранее ссылки.

Игры на BASIC

Если вы следовали тому, что написано в предыдущих разделах, вы уже должны запросто вводить и выводить текст плюс строковые и числовые переменные, а также работать со случайными числами. Эти элементы и создают игры вроде *Hammurabi*, которую теперь вы можете скачать, проанализировать и сыграть в нее (см. врезку внизу «Донос на *Hammurabi*»). В порядке альтернативы можете использовать взятую нами в качестве примера игру *Barsoom*, которая сочетает изученные вами элементы кодирования.

Откройте терминал на своем компьютере и запустите `wget https://raw.githubusercontent.com/nate-drake/bcbasicvsamples/master/barsoom.bas[1]`, чтобы скачать игру. Для просмотра исходного кода запустите `nano barsoom.bas`.

Игра *Barsoom* названа в честь планеты Марс из знаменитого романа Эдгара Райса Берроуза про *Джона Картера*, и в ней вы на десять лет делаетесь руководителем Марсианской базы. Вашему начальному населению в количестве 100 человек нужна как минимум одна пайка еды на каждого, чтобы протянуть год. В обмен на свой хлеб насыщенный они будут добывать для вас железную руду,



➤ Код *Barsoom* использует команду 'CLS' для отображения отдельных экранов, чтобы купить еду, накормить население и продать избытки кислорода.

пригодную для производства избыточного кислорода на продажу. Цена на кислород (переменная NUMOXYPRICE) и пищу (NUMFOODCOST) меняется из года в год. Если вы решите продавать или покупать ресурсы, игра запустит специальную процедуру. Чем больше еды вы выделите населению, тем быстрее оно будет расти, и тем больше кислорода вы сумеете добыть. Если вы успешно будете делать это в течение 10 лет, вы получите поздравление.

На данном этапе игры ее очень легко закончить. Но если этот проект вас заинтересовал, можете попробовать изменить начальные значения, чтобы чуть усложнить игру, или ввести случайные события. Например, марсианские жуки-долгоносики могут каждый год портить ваши запасы еды. В порядке альтернативы, сепаратистская часть марсианского населения может решить отделиться от вашей базы, с катастрофическими экономическими последствиями. Не стесняйтесь дать волю своему воображению! **LXF**

Донос на Hammurabi

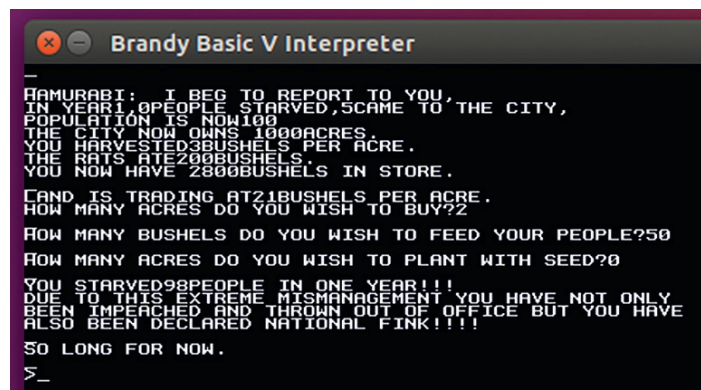
Hammurabi — стратегия с добычей ресурсов. В этой игре вы — правитель древневавилонского города, и вы должны управлять страной, выращивать зерно на прокорм населения, а заодно разбираться со случайными событиями, например, нашествием крыс, которые уничтожают ваш урожай. В игре есть несколько возможных финалов, включая ваш возможный импичмент и объявление вас государственным преступником.

Скачайте игру, открыв Терминал и запустив `wget http://vintage-basic.net/bcg/hammurabi.bas`. Перед началом игры откройте ее в своем любимом текстовом редакторе, чтобы увидеть, как используются команды, которые вы выучили на нашем уроке. События в игре определяются случайными числами; например, переменная 'Y' определяет, что каждая сотка земли будет стоить от 17 до 26 пудов зерна. Вы также должны выбрать, сколько засеять соток, хозяйственно отложив достаточно пудов, чтобы накормить свой народ. Несколько путает то, что значения ввода для нескольких из этих переменных — 'Q'. Местами следовать коду довольно сложно из-за того, что автор поместил несколько

команд в одну строку, разделив их двоеточием (:), а также использовал GOSUB для перемещения в разные места кода. Это менее эффективно, чем применение процедур. Однако по мере следования структуре встречаются полезные и иногда остроумные аннотации — например, строка 532, которая определяет рост населения из года в год,

читается: `532 REM *** LET'S HAVE SOME BABIES [Заведем-ка детей].`

Если вы новичок в программировании, попробуйте изменить исходный код *Hammurabi*, употребив процедуры и присвоив значимые имена переменным, чтобы попрактиковаться в изученном материале.



➤ Попробуйте сделать лучше, чем здесь, когда будете играть в *Hammurabi*.

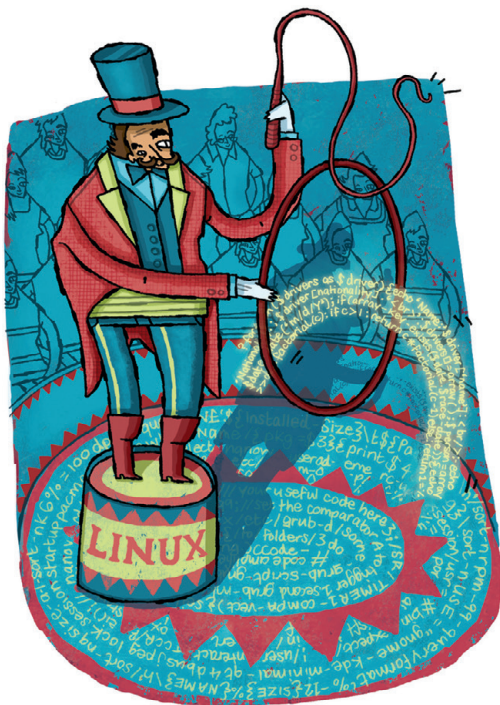
Kotlin: Управляем данными и потоком

Во второй части серии о Kotlin **Михалис Цукалос** учит использовать массивы, списки и циклы `for` и `while` с блоками и диапазонами.



Наш эксперт

Михалис Цукалос — администратор UNIX, программист, администратор баз данных и математик, который любит писать статьи и узнавать что-то новое. Он — автор книги «Системное программирование на Go». С ним можно связаться через сайт www.mtsoukalos.eu и Twitter — @mactsouk.



Пора заняться Kotlin всерьез! Сегодня мы познакомимся с двумя важнейшими темами: типами данных и управлении потоком в Kotlin. Эти темы включают массивы и списки, а также циклы `for`, `while` и `do...while`. Кроме того, мы подробнее познакомимся с массивами, обработкой исключений и null-безопасностью. Вскоре вы сможете написать собственные утилиты командной строки на Kotlin!

Типы данных

Kotlin поддерживает стандартные типы данных: различные типы целых чисел и чисел с плавающей запятой и с двойной точностью, а также логические переменные и строки. Ключевой момент состоит в том, что вы можете определять два типа переменных — изменяемые и неизменяемые. Первые определяются ключевым словом `var`, вторые — `val`. Обратите внимание, что неизменяемые переменные нужно обязательно инициализировать при их создании.

На экранном снимке вверху справа (рис. 1) показан код Kotlin файла `dataTypes.kt`; в нем более подробно иллюстрируется использование типов данных Kotlin. Так как Kotlin поддерживает определение типов, компилятор может «угадать» тип выражения самостоятельно. При желании тип переменной можно и указать. После

```
dataTypes.kt — code
1 | fun main(args: Array<String>) {
2
3 |     var i = 10
4 |     println("i initial value: $i")
5 |     val k = 100
6 |     i = i + 1
7 |     println("i is now: " + i)
8
9 |     var aString: String = "string"
10 |    var anotherString = "string"
11
12 |    if (anotherString == aString)
13 |        println("Equal!")
14
15 |    var aLong: Long = 1231234332
16 |    var aShort: Short = -123
```

► Рис. 1. Код файла `dataTypes.kt`, иллюстрирующий использование изменяемых и неизменяемых переменных с помощью простых операторов Kotlin.

компиляции и запуска файла `dataTypes.kt` мы получим следующий результат:

```
$ java -jar dataTypes.jar
i initial value: 10
i is now: 11
Equal!
aLong = 1231234332
```

Управление потоком и циклы

Пора рассказать об управлении потоком и о циклах в Kotlin. Это оператор `if` и циклы `for`, `while` и `do...while`.

Оператор `if` работает в точности так, как вы и подумали, но заодно обладает несколькими умными функциями:

```
>>> var h = 10
>>> var myVariable = "Сейчас " + if ( h < 12 ) { "рано" } else {
"поздно!" }
>>> println(myVariable)
Сейчас рано
```

То есть, как вы здесь видите, операторы `if` допускаются включать в выражения!

Следующий блок кода Kotlin выполняется в REPL. В нем показано использование цикла `for` с набором значений:

```
>>> val aSet = setOf(1, 2, 5, 6, 8)
>>> for (i in aSet) { print (" $i ") }
1 2 5 6 8
```

Примите участие!

Полный код к этому учебнику можно найти на DVD номера или в архивах на сайте www.linuxformat.com/archives.

Null-безопасность

На прошлом уроке по Kotlin (LXF229/230) мы упоминали «исключения от нулевых указателей».

Теперь пришло время подробнее познакомиться с важной характеристикой Kotlin — null, с которой ваш код станет нагляднее и гораздо надежнее. Для начала рассмотрим такой пример кода на Kotlin:

```
var aString = "Tsoukalos"
aString = null
```

Этот код не скомпилируется, потому что по умолчанию в Kotlin нельзя присваивать переменной значение null. Появится сообщение об ошибке `cannot be a value of a nonnull type String [Значение null нельзя присвоить переменной типа String]`,

которое поясняет, что происходит. Однако это поведение можно изменить, употребив оператор доступа с проверкой на null, который изображается в виде вопросительного знака:

```
>>> var anotherString: String? = "Tsoukalos"
>>> anotherString = null
```

Kotlin забегает на шаг вперед и требует проверить переменную на null перед ее использованием. Поэтому следующая программа на Kotlin не скомпилируется:

```
fun printA(printMe: String) {
    println(printMe)
}
```

```
fun main(args: Array<String>) {
    val a: String? = "Tsoukalos"
    printA(a)
}
```

А если заменить `printA(a)` на следующий фрагмент кода, программа `nSafety.kt` прекрасно скомпилируется:

```
if (a != null)
    printA(a)
```

Если вы программист-любитель, то, возможно, оцените null-безопасность в Kotlin не сразу. Но с ней вы избавитесь от исключений `NullPointerException`... и это замечательно!

Обратите внимание, что если в цикле `for` есть только один оператор, фигурные скобки не требуются. Однако их отсутствие может вызывать неприятные ошибки. Поэтому указывайте скобки всегда, если у вас достаточно места. Кроме того, цикл `for` работает с каждым объектом, у которого есть итератор. В этом он аналогичен циклу `each` `for` из других языков программирования.

Ключевое слово `break` используется для выхода из цикла `for` или `while`, а ключевое слово `continue` — для пропуска текущей итерации и перехода к следующей.

В следующем REPL (Read–Eval–Print Loop — цикл «чтение–вычисление–вывод») показан оператор `while` в действии:

```
>>> println(i)
10
>>> while ( true ) {
...     if ( i == 12 ) { break }
...     i = i + 1
...     println(i)
... }
11
12
```

И последнее: объекты класса `String` можно использовать внутри циклов `for`, так как у этих объектов есть итераторы. Следующая программа на Kotlin, сохраненная в файле `countChar.kt`, подсчитывает количество символов в каждом из аргументов командной строки:

```
fun main(args: Array<String>) {
    if (args.size == 0) {
        println("Нужен хотя бы один аргумент!")
        return
    }
    for (k in args) {
        var sum = 0
        for (c in k)
            sum = sum + 1
        println("$k содержит $sum символов.")
    }
}
```

Как видите, в `countChar.kt` используется два цикла. Первый перебирает все аргументы командной строки, второй подсчитывает количество символов в аргументах.

При запуске файла `countChar.kt` мы получим следующий результат:

```
$ kotlinc countChar.kt -include-runtime -d countChar.jar
$ java -jar countChar.jar 123 12.3.4 abcd
123 содержит 3 символов.
12.3.4 содержит 6 символов.
abcd содержит 4 символов.
```

Вот цикл `do...while` в следующей итерации с REPL:

```
>>> var i = 10
>>> do { println(i) } while ( i < 5 )
10
```

Основное различие между циклом `do...while` и остальными двумя циклами в том, что с циклом `do...while` вы можете быть уверены в том, что получите хотя бы одну итерацию. Это удобно, когда надо что-то инициализировать. Хотя циклы `for`, `while` и `do...while` можно сделать эквивалентными, блоки кода в циклах `do...while` смотрятся нагляднее. Это основная причина, по которой их следует использовать.

Масса массивов

Давайте посмотрим, как в Kotlin используются массивы.

Два следующих оператора создают два одинаковых массива:

```
>>> var a1 = arrayOf("1", "4", "9", "16")
>>> var a2 = Array(4, { i -> (i * i).toString()})
```

Т.е. для создания элементов массива можно использовать функцию, как с `a2`. Строго говоря, `Array()` представляет собой конструктор класса `Array`. В нем надо указать размер массива и функцию, которая будет использоваться для создания элементов массива.

Рассмотрим следующие команды, которые будут выполнены в REPL:

```
>>> var strangeArray = arrayOf(1, 0, "aString")
>>> var thisWillNotWork = intArrayOf(1, 0, "aString")
error: type mismatch: inferred type is String but Int was expected
[ошибка: несовпадение типов: передан String, а д.б. Int]
var thisWillNotWork = intArrayOf(1, 0, "aString")
```



Скорая помощь

Если у вас есть несколько вариантов действий, всегда используйте тот, который проще и понятнее. Это означает, что не следует использовать оператор `When` вместо `If` с одной веткой, и не стоит определять массив целых чисел, если можно обойтись двумя целочисленными переменными!

»

► Рис. 2. Код Kotlin из файла `usingArrays.kt`. В нем показано два способа доступа ко всем элементам массива: `array`, `indices` и `array.size`.

Первый оператор показывает, что в массиве Kotlin можно использовать элементы разных типов. Однако если тип массива объявлен (как в случае с `thisWillNotWork`), вы не сможете добавлять в этот массив элементы других типов.

Еще один удобный метод, который применяется к массивам — `joinToString()`. Вот как он работает:

```
>>> println(a1.joinToString(" + "))
1 + 4 + 9 + 16
>>> println(a1.joinToString(" + ", "->", "<-"))
->1 + 4 + 9 + 16<-
```

Другие примеры использования массивов можно найти в коде файла `usingArrays.kt`, который показан на рис. 2 (стр. 83). После компиляции и запуска файла `usingArrays.kt` мы получим следующий результат:

```
$ kotlinc usingArrays.kt -include-runtime -d usingArrays.jar
$ java -jar usingArrays.jar
Size: 5
0 -2 3 100 2
0 1 2 1500 4
```

Также допускается создавать двумерные массивы. Следующая команда создает массив с четырьмя строками и семью столбцами, причем начальное значение каждого из элементов массива равняется -1:

```
>>> var array2D = Array(4) { Array(7) { -1 } }
```

Для обращения к первому элементу массива `array2D` используйте следующую конструкцию:

```
>>> array2D[0][0]
-1
```

Получить доступ ко всем элементам двумерного массива можно, используя два цикла `for`:

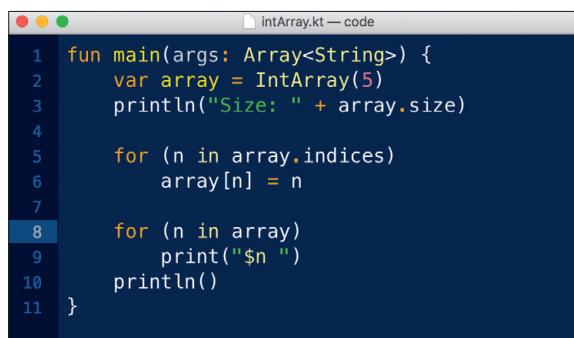
```
for (r in array2D) {
    for (c in r) {
        print("$c ")
    }
    println()
}
```

Массивы для примитивных типов

В Kotlin есть следующие реализации массивов для примитивных типов: `ByteArray`, `CharArray`, `ShortArray`, `IntArray`, `BooleanArray`, `DoubleArray` и `FloatArray`. Это очень много, поэтому мы проиллюстрируем только использование массива `IntArray`. Для этого воспользуемся файлом `intArray.kt`, показанным на экранном снимке внизу (рис. 3). Запустив файл `intArray.kt`, мы получим следующий результат:

```
$ kotlinc intArray.kt -include-runtime -d intArray.jar
$ java -jar intArray.jar
Size: 5
0 1 2 3 4
```

Если не считать различий в способе создания массива, то как `IntArray()`, так и `arrayOf()` возвращают объект `Array` и предоставляют одни и те же методы для работы с массивами. Полный список



```
1 fun main(args: Array<String>) {
2     var array = IntArray(5)
3     println("Size: " + array.size)
4
5     for (n in array.indices)
6         array[n] = n
7
8     for (n in array)
9         print("$n ")
10    println()
11 }
```



Скорая помощь

Чем важнее ваше приложение, тем большее значение имеет обработка исключений. В своих программах всегда проверяйте входные данные и их формат: пользователям доверять нельзя.

► Рис. 3. Код Kotlin из файла `intArray.kt` показывает, как использовать встроенный тип данных `IntArray` для создания массива целых чисел.



```
1 fun main(args: Array<String>) {
2
3     var myList = mutableListOf<Int>(0,1,2,3,4,5,6,7)
4     for (v in myList)
5         print("$v ")
6     println()
7
8     println("Index for value 4: ${myList.indexOf(4)}")
9     println("Index for value 14: ${myList.indexOf(14)}")
10    println("The size of the list: ${myList.size}")
11    myList.add(1, -1)
12    myList.add(1)
13    println("Reversed: ${myList.asReversed()}")
14    println("Index for value 1: ${myList.indexOf(1)}")
15    println("Remove 1: ${myList.remove(1)}")
16    println("Remove 1: ${myList.remove(1)}")
17    println("Remove 1: ${myList.remove(1)}")
18    println("Reversed: ${myList.asReversed()}")
19 }
```

► Рис. 4. Код Kotlin файла `useLists.kt`. Этот файл иллюстрирует работу со списками.

методов можно найти на страницах <https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-int-array/> и <https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-array/>. В Kotlin есть и более сложные типы массивов, которые представляют собой объекты класса `Array<T>`, где `T` — любой тип данных.

Тут уместно спросить: когда следует использовать массивы, предлагаемые Kotlin, и когда создавать свои? Ответ простой: если то, что предлагает Kotlin, соответствует вашим потребностям, используйте именно этот вариант. Он будет отличаться более высокой скоростью, потому что все структуры данных, предлагаемые языком программирования, будут быстрее по сравнению с другими реализациями. В противном случае создайте свой собственный массив.

Списки

Kotlin поддерживает списки (`List`), которые представляют собой упорядоченные коллекции. Наряду со списками стандартная библиотека Kotlin поддерживает сопоставления (`Map`), наборы (`Set`), массивы (`Array`) и последовательности (`Sequence`), которые также представляют собой коллекции — все они основаны на фреймворке коллекций Java. Это означает, что не всё в Kotlin так просто, как два типа массива, которые вы видели чуть выше.

Создать неизменяемый и изменяемый списки целых чисел можно следующим образом:

```
var myList1 = listOf<Int>(0,1,2,3,4,5,6,7)
var myList2 = mutableListOf<Int>(0,1,2,3,4,5,6,7)
```

На рис. 4 (вверху) показан код Kotlin из файла `useLists.kt`. Из этой программы вы сможете подробнее узнать об использовании списков в Kotlin.

Скомпилировав и запустив файл `useLists.kt`, мы получим следующий результат:

```
$ kotlinc useLists.kt -include-runtime -d useLists.jar
$ java -jar useLists.jar
0 1 2 3 4 5 6 7
Index for value 4: 4
Index for value 14: -1
The size of the list: 8
Reversed: [1, 7, 6, 5, 4, 3, 2, 1, -1, 0]
Index for value 1: 2
Remove 1: true
Remove 1: true
Remove 1: false
Reversed: [7, 6, 5, 4, 3, 2, -1, 0]
```

Метод `indexOf()` проверяет, есть ли аргумент в списке, и возвращает его индекс или `-1`, если аргумента нет в списке. Кроме того, вы видите, что элементы списка всегда выводятся в одном и том же порядке и что метод `add()` позволяет добавить новый элемент в указанное место списка или в конец списка. В списках возможны повторы значений. Первое вхождение элемента можно

удалить с помощью метода `remove()`. Если элемент встречается в списке несколько раз, надо будет вызвать метод `remove()` несколько раз.

Итерация по диапазонам

Ранее в файле `usingArrays.kt` вы видели, как использовать диапазон для обращения ко всем элементам массива. Строго говоря, диапазон — это интервал с начальным и конечным значениями. Он удобен тем, что для создания диапазона можно использовать любой совместимый тип.

Существует множество способов использования диапазонов, например:

```
val zTo10 = 0..10
val step3 = zTo10.step(3)
val down = 10.downTo(0)
val reverse = (0..10).step(2).reversed()
```

Первый диапазон создается с помощью оператора `..`. Во втором диапазоне используется шаг, а в третьем примере создается диапазон в порядке убывания. Поскольку отрицательное значение шага использовать нельзя, на помощь приходит метод `reversed()`.

Проверить, входит ли указанное значение в диапазон, всегда можно с помощью оператора `in`:

```
>>> println(1 in down)
true
>>> println(1 in step3)
false
```

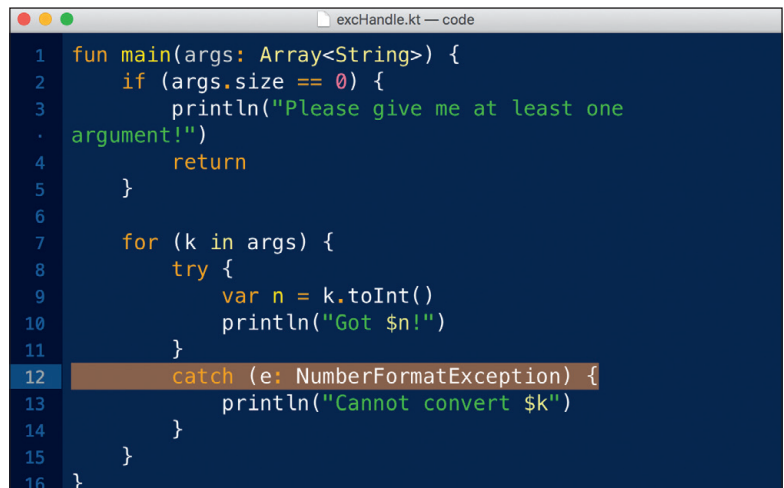
Однако в циклах `for` можно использовать только диапазоны целых чисел. Это типы `int`, `long` и `char`. Рассмотрим пример, в котором некоторые ранее созданные диапазоны используются в двух циклах `for`:

```
>>> val myArray = Array(20, { i -> i })
>>> for (i in reverse)
... print("${myArray[i]} ")
10 8 6 4 2 0
>>> for (i in step3)
... print("${myArray[i]} ")
0 3 6 9
```

Обработка исключений

Одна из ключевых характеристика Kotlin — обработка исключений с использованием блоков `try` и `catch`. При желании можно указать в конце блок `finally` и использовать несколько блоков `catch` для перехвата различных исключений.

В этом разделе вы увидите программу `excHandle.kt`, которая проверяет, является ли введенное число целым. Для демонстрации обработки исключений используется функция `toInt()`. Эта функция



```
1 fun main(args: Array<String>) {
2     if (args.size == 0) {
3         println("Please give me at least one
4         argument!")
5         return
6     }
7     for (k in args) {
8         try {
9             var n = k.toInt()
10            println("Got $n!")
11        }
12        catch (e: NumberFormatException) {
13            println("Cannot convert $k")
14        }
15    }
16 }
```

Рис. 4. Код Kotlin из файла `excHandle.kt`, иллюстрирующий перехват исключения `NumberFormatException` из метода `toInt()` с помощью блока `try...catch`.


выдает исключение `NumberFormatException`, когда строка, которую вы хотите преобразовать в целое число, имеет неправильный формат. Обратите внимание, что в документации для каждой функции должны указываться названия создаваемых ею исключений, а также причины создания каждого исключения. На экранном снимке (рис. 4) показан код Kotlin файла `excHandle.kt`, в котором можно увидеть блок `try...catch` в действии.

Запуск файла `excHandle.kt` даст следующий результат:

```
$ kotlinc excHandle.kt -include-runtime -d excHandle.jar
$ java -jar excHandle.jar -2 123.2 a
Got -2!
Cannot convert 123.2
Cannot convert a
```

С помощью ключевого слова `throw` можно создавать собственные исключения, которые в основном используются в ваших собственных функциях и методах.

Чтобы понять, хорошо ли вы разобрались в нашем уроке, можете проделать следующие два упражнения. Первое упражнение — создание программы, которая считывает вводимые пользователем целые числа до тех пор, пока он не введет значение 0. Затем эта программа выводит сумму всех прочитанных целых чисел. Попробуйте реализовать эту программу тремя различными способами: с помощью циклов `for`, `while` и `do...while`. Второе упражнение — создание утилиты, которая проверяет, являются ли аргументы командной строки действительными целыми числами, а также являются ли они знаковыми или беззнаковыми. **LXF**

Скорая помощь 

При разработке программ на Kotlin помните, что всё в Kotlin является объектом. Так вы сможете понять, как перебирать строковые переменные и как реализовать массивы в своем коде.

Оператор when

В Kotlin нет оператора `switch`, так как он заменен оператором `when`. Использование оператора `when` показано в программе `useWhen.kt`. В файле `useWhen.kt` наиболее важен следующий фрагмент:

```
when (x) {
0 -> println("x is zero!")
1 -> println("x is one.")
in 2..100 -> println("x is between 2 and 100!")
else -> {
    println("Unknown x value: $x")
}
}
```

С помощью оператора `when` можно делать много интересных вещей — например, использовать

диапазоны и блок по умолчанию, объявляемый с помощью ключевого слова `else`. Кроме того, допускается использовать выражения, как показано ниже:

```
when {
(x % 2 == 0) -> println("x is odd!")
(x % 2 == 1) -> println("x is Even!")
else -> {
    println("$x is neither odd nor even!")
}
}
```

В данном случае оператор `when` вычисляет первое и второе логическое выражения, и ветвь выполняется, если условие истинно. Обратите

внимание, что выполнится только самая первая ветвь, для которой условие будет истинным.

При запуске файла `useWhen.kt` будет получен следующий результат:

```
$ kotlinc useWhen.kt -include-runtime -d useWhen.jar
$ java -jar useWhen.jar 0
x is zero!
x is odd!
$ java -jar useWhen.jar 200
Unknown x value: 200
x is odd!
$ java -jar useWhen.jar 75
x is between 2 and 100!
x is Even!
```

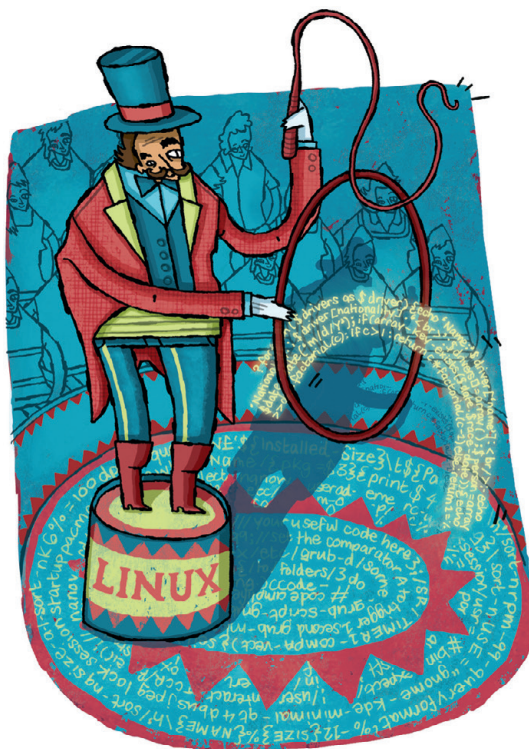
Kotlin: Системные программы

Михалис Цукалос разрабатывает в Kotlin системные утилиты и создает потоки и случайные числа. Он — душа компании!



Наш эксперт

Михалис Цукалос — администратор UNIX, программист, администратор баз данных и математик, а также автор книги «Системное программирование на Go». С ним можно связаться через сайт www.mtsoukalos.eu и Twitter — @mactsouk.



```

1 import java.util.Random
2
3 fun ClosedRange<Int>.random() = Random().nextInt(endInclusive - startInclusive + 1) + startInclusive
4
5 fun main(args: Array<String>) {
6     if (args.size != 3) {
7         println("Please give me exactly three arguments!")
8         return
9     }
10
11     for (k in args) {
12         try {
13             var n = k.toInt()
14         }
15         catch (e: NumberFormatException) {
16             println(k + " is not an integer")
17             return
18         }
19     }
20
21     val n = args[0].toInt()
22     val min = args[1].toInt()
23     val max = args[2].toInt()
24 }

```

► Рис. 1. Код файла random.kt, иллюстрирующий создание случайных чисел в Kotlin.

При запуске файла **checkFile.kt** с различными типами аргументов мы получим следующий результат:

```

$ kotlinc checkFile.kt -include-runtime -d checkFile.jar
$ java -jar checkFile.jar /tmp/mysql.sock
/tmp/mysql.sock существует!
$ java -jar checkFile.jar /usr/bin/
/usr/bin/ существует!
Но это каталог!
$ java -jar checkFile.jar /tm

```

Кроме того, **checkFile.kt** использует свойство `isDirectory`, чтобы сообщить, является ли путь каталогом или файлом. Последняя команда показывает, что если такого пути не существует, программа не выведет никакого результата.

```

Следующий фрагмент кода Kotlin показывает, как прочесть текстовый файл построчно с помощью списка и двух библиотек Java:
import java.io.File
import java.io.BufferedReader
fun main(args: Array<String>) {
    val bufferedReader = File("tmp/aFile").bufferedReader()
    val lineList = mutableListOf<String>()
    bufferedReader.useLines { lines -> lines.forEach { lineList.add(it) } }
    for (l in lineList) { println(l) }
}

```

После чтения файла его строки сохраняются в списке `lineList`. Затем они поочередно выводятся на экран с помощью цикла `for`. Имя всей программы — **lineByLine.kt**. Она принимает один аргумент командной строки — имя файла, который требуется прочитать. В качестве бонуса **lineByLine.kt** также выводит количество

Системное программирование на Kotlin обязательно включает методы чтения и записи файлов. Помните, что если вам когда-нибудь придется писать серьезные программы на Kotlin, то вы почти наверняка столкнетесь с обработкой файлов и каталогов. Так что внимательно изучите наш урок. Кроме того, здесь приведены примеры кода для генерации случайных чисел и создания потоков в Kotlin.

Время читать

Когда вы хотите обработать файл, первым делом надо выяснить, существует ли он. Следующий код на Kotlin, который сохранен в файле **checkFile.kt**, проверяет, существует ли файл по указанному пути:

```

var input = args[0]
val k = File(input)
if (k.exists())
    println(input + " существует!")

```

Функция `File()` считывает свойства для указанного пути, даже если путь не существует. Затем используется метод `exists()`, который возвращает `true`, если указанный путь существует.

Примите участие!

Как и в большинстве языков программирования, в Kotlin есть несколько способов чтения и записи. Выбор того или иного способа зависит от решаемой задачи. Если скорость и использование памяти не критичны, выбирайте любой понравившийся способ.

Генерация случайных чисел

Строго говоря, генерация случайных чисел не относится к системному программированию. Однако возможность создания случайных чисел на Kotlin полезна, когда надо что-то проверить или создать случайный пароль. Далее будет использована программа **random.kt**. Она принимает три параметра: количество случайных целых чисел, которые надо создать, а также минимальное и максимальное значения создаваемых целых чисел. Наиболее интересен следующий фрагмент кода:

```
val myRange = min..max
for (i in 1..n)
    println(myRange.random())
```

В файле **random.kt** определена новая функция **random()**, которую вы можете просмотреть в коде на рис. 1 (стр. 86). Функция **random()** применяется к диапазону чисел и выполняет за нас всю необходимую работу.

Обратите внимание, что для генерации случайных чисел файл **random.kt** использует библиотеку

```
java.util.Random в Java. После компиляции и запуска файла random.kt мы получим следующий результат:
```

```
$ kotlinc random.kt -include-runtime -d random.jar
$ java -jar random.jar 5 10 100
80
15
...
```

Последняя команда создает пять случайных целых чисел в диапазоне от 10 до 99, и вы можете адаптировать ее к своим нуждам.

прочитанных строк. Все, что вам надо сделать — скомпилировать и запустить файл **lineByLine.kt**:

```
$ kotlinc lineByLine.kt -include-runtime -d lineByLine.jar
$ java -jar lineByLine.jar /tmp/aFile
```

Файл **lineByLine.kt** будет основой для многих других утилит этого учебника, так что не стоит его недооценивать!

Время записывать

Теперь посмотрим, как открыть текстовый файл для записи и как записать в него данные, используя код из файла **writeToFile.kt**. Утилита **writeToFile.kt** требует, чтобы на момент ее запуска выходной файл еще не существовал, поскольку мы, разумеется, перезаписывать его не хотим. Так что мы по-прежнему будем использовать часть кода из файла **checkFile.kt**, но другим способом.

В файле **writeToFile.kt** наиболее важен следующий код, выполняющий собственно запись:

```
val writer = PrintWriter(output)
val message = "Writing to " + output + "\n"
writer.append(message)
writer.close()
```

Здесь создается объект **PrintWriter**, далее используемый для записи данных с помощью метода **append()**. По окончании записи вызывается метод **close()** для закрытия файла.

Скомпилировав и запустив файл **writeToFile.kt**, мы получим следующий результат:

```
$ kotlinc writeToFile.kt -include-runtime -d writeToFile.jar
$ java -jar writeToFile.jar /tmp/test
$ java -jar writeToFile.jar /tmp/test
/tmp/test already exists!
```

Следующий пример кода, сохраненный в файле **writeError.kt**, покажет, как эффективно обрабатывать ошибки при записи в файл. В нем наиболее интересен следующий фрагмент кода:

```
try {
    val writer = PrintWriter(output)
    val message = "Writing to " + output + "\n"
    writer.append(message)
    writer.close()
} catch (e:Exception) {
    println(e.message)
}
```

Вы видите, что для обработки ошибок можно просто встроить код Kotlin в блок **try**. При запуске файла **writeError.kt** будет получен следующий результат:

```
$ java -jar writeError.jar /usr/bin/myfile
/usr/bin/myfile (Operation not permitted)
```

В данном случае проблема заключается в том, что вы пытаетесь записать в каталог **/usr/bin**, не имея необходимых разрешений.

Попытка сделать то же самое в файле **writeToFile.kt** приведет к гораздо более жуткому сообщению об ошибке:

```
$ java -jar writeToFile.jar /usr/bin/myfile
Exception in thread "main" java.io.FileNotFoundException: /usr/bin/myfile (Operation not permitted)
at java.io.FileOutputStream.open0(Native Method)
at java.io.FileOutputStream.open(FileOutputStream.java:270)
at java.io.FileOutputStream.<init>(FileOutputStream.java:213)
at java.io.FileOutputStream.<init>(FileOutputStream.java:101)
at java.io.PrintWriter.<init>(PrintWriter.java:184)
at WriteToFileKt.main(writeToFile.kt:18)
```

Однако существенная разница между двумя подходами состоит в том, что вариант **try/catch** позволяет обрабатывать ошибки так, как вы хотите, тогда как в файле **writeToFile.kt** просто выдается ошибка без возможности что-либо сделать!

Флаги

Почти все утилиты Unix поддерживают несколько параметров, позволяющих изменить способ работы утилит. В этом разделе вы узнаете, как добавить поддержку нескольких параметров в свои утилиты командной строки на Kotlin. Однако с помощью пакета стандартной библиотеки Kotlin этого сделать нельзя, так что вам понадобится написать собственный код. Это означает, что у вас будет больший контроль над процессом, но для этого придется написать больше кода.

Скорая помощь

Лучшее, что можно сделать для изучения системного программирования — попробовать реализовать основные системные утилиты командной строки UNIX. Начните с таких инструментов, как **rm(1)**, **find(1)**, **ls(1)** и т.д. Когда вы почувствуете себя более уверенно, реализуйте некоторые параметры этих утилит.

```
options.kt — KotlinP3.LXF231
1 fun main(args: Array<String>) {
2     if (args.size == 0) {
3         println("Please give me at least one argument!")
4         return
5     }
6
7     var iOption = false
8     var cOption = false
9
10    for (i in args) {
11        if (i == "-i")
12            iOption = true
13        else if (i == "-c")
14            cOption = true
15        else
16            break
17    }
18 }
```

► Рис. 2. Здесь показан код Kotlin файла **options.kt**. Этот файл иллюстрирует обработку флагов и аргументов командной строки в Kotlin.

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

Скорая помощь



Kotlin формирует информативные сообщения об ошибках, которые проясняют, что пошло не так. Эти сообщения помогут понять, что надо исправить для решения проблемы — поэтому читайте их внимательно!

Следующая программа предполагает, что сначала указываются параметры командной строки. Если вы обнаружите нечто, не являющееся параметром, это означает, что вы перешли к аргументам, и параметры командной строки больше не проверяются.

Далее представлена программа **options.kt**. В ней наиболее интересен следующий фрагмент кода:

```
for (i in args) {
    if (i == "-i")
        iOption = true
    else if (i == "-c")
        cOption = true
    else
        break
}
```

Программа поддерживает два параметра (флажка) командной строки: **-i** и **-c**. Весь код файла **options.kt** можно увидеть на рис. 2 на стр. 87. Если требуется поддержка более трех параметров, вам, скорее всего, придется искать внешнюю библиотеку Kotlin, которая это сделает.

После компиляции и запуска файла **options.kt** мы получим следующий результат:

```
$ kotlinc options.kt -include-runtime -d options.jar
```

```
$ java -jar options.jar -c -i 1
-i is set!
-c is set!
1
$ java -jar options.jar -c 1 -i
-c is set!
1
-i
```

Копируем cp

Попробуем реализовать на Kotlin базовые функции утилиты **cp**. Программа находится в файле **cp.kt**, и наиболее важен в нем следующий код:

```
File(input).copyTo(File(output), false);
```

Здесь мы пользуемся одной функцией, которая позволяет скопировать файл в другой файл. Второй параметр метода **copyTo()** указывает, следует ли переписать файл назначения. **False** означает, что не следует переписывать файл назначения, если он уже существует.

Хотя копировать с помощью одной функции очень удобно, отсутствие полного контроля над процессом не всегда бывает предпочтительно.

Ниже показано, как скомпилировать и запустить файл **cp.kt**:

```
$ kotlinc cp.kt -include-runtime -d cp.jar
$ java -jar cp.jar 123
Usage: program input output
$ java -jar cp.jar /tmp/adobegc.log /tmp/adobegc.log.copy
```

Если всё работает, программа не выводит никакого результата. А если файл назначения существует, **cp.kt** выдаст следующую ошибку:

```
Exception in thread "main" kotlin.io.
FileAlreadyExistsException: /tmp/adobegc.log -> /tmp/
adobegc.log.copy: The destination file already exists.
```

Возможны варианты

Теперь создадим альтернативную версию программы копирования и в качестве стартовой точки воспользуемся кодом Kotlin из файла **lineByLine.kt**. Эта программа **cpLine.kt**, и она считывает входной файл строка за строкой, чтобы затем записать его в выходной файл.

В файле **cpLine.kt** наиболее важен следующий код, выполняющий само копирование:

```
val writer = PrintWriter(output)
for (l in lineList) {
    writer.append("$l\n")
}
writer.close()
```

Весь код файла **cpLine.kt** показан на рис. 3 слева внизу. Программа записывает в выходной файл каждую строку, считанную из входного файла. Для больших входных файлов этот процесс может занять много времени. После компиляции и запуска файла **cpLine.kt** мы получим следующий результат:

```
$ kotlinc cpLine.kt -include-runtime -d cpLine.jar
$ java -jar cpLine.jar wc.kt /tmp/test.kt
$ diff wc.kt /tmp/test.kt
$ java -jar cpLine.jar wc.kt /tmp/test.kt
/tmp/test.kt already exists!
```

Проблема заключается в том, что хотя **cpLine.kt** отлично работает с текстовыми файлами, он не может работать с двоичными. Однако эта программа может быть очень удобной, если нужно прочесть текстовый файл построчно, заодно внести в него требуемые изменения и сохранить результаты в новом файле.

```
7      println("Please give me exactly two arguments!")
8      return
9  }
10
11  var input = args[0]
12  val i = File(input)
13  if (i.exists() == false) {
14      println(input + " does not exist!")
15      return
16  }
17
18  if (i.isDirectory) {
19      println(input + "cannot be a directory!")
20      return
21  }
22
23  var output = args[1]
24  val o = File(output)
25  if (o.exists()) {
26      println(output + " already exists!")
27      return
28  }
29
30  val bufferedReader = File(input).bufferedReader()
31  val lineList = mutableListOf<String>()
32  bufferedReader.useLines {
33      lines -> lines.forEach {
34          lineList.add(it)
35      }
36  }
37
38  val writer = PrintWriter(output)
39  for (l in lineList) {
40      writer.append("$l\n")
41  }
42  writer.close()
43 }
```

➤ **Рис. 3.** Программа **cpLine.kt** копирует только текстовые файлы, считывая их построчно. При попытке скопировать двоичный файл результирующий файл будет отличаться от исходного.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Итак, реализовав основные функции утилиты *wc*, займемся кое-чем поинтереснее: напишем на Kotlin утилиту *wc*.

Слово за слово

Утилита *wc* выполняет три важные задачи: подсчет строк, подсчет слов и подсчет символов. Самая важная задача, которую вам надо решить — это чтение текстового файла по строкам (так вы сможете подсчитать количество строк в файле). Две остальные задачи намного проще — особенно подсчет символов в файле, который можно выполнить, определив длину текущей строки с помощью функции Kotlin или путем перебора символов строки. Определить длину с помощью функции намного быстрее, но перебор символов строки даст вам большую гибкость, если вы хотите внести изменения в содержимое файла.

Имя файла, который реализует утилиту *wc* — **wc.kt**, а логика этой программы основана на коде в файле **lineByLine.kt**. Однако код из файла **wc.kt** может не слишком напоминать код в файле **lineByLine.kt**.

В файле **wc.kt** наиболее важен следующий фрагмент:

```
for (l in lineList) {
    nLines++
    nChars = nChars + l.length
    if (l.trim() != "") {
        val splitLine = l.trim().split(Regex("\\s+"))
        nWords = nWords + splitLine.count()
    }
}
```

Файл считывается построчно и обрабатывается каждая строка. Однако есть особый случай, когда строка файла содержит одни пробелы. В данном случае программа ничего не добавит к количеству слов — это реализовано с помощью метода `trim()`, который возвращает входную строку без ведущего и завершающего пробелов. Если строка не пуста, вы разделяете ее на слова с помощью регулярного выражения, а затем подсчитываете слова с помощью метода `count()`.

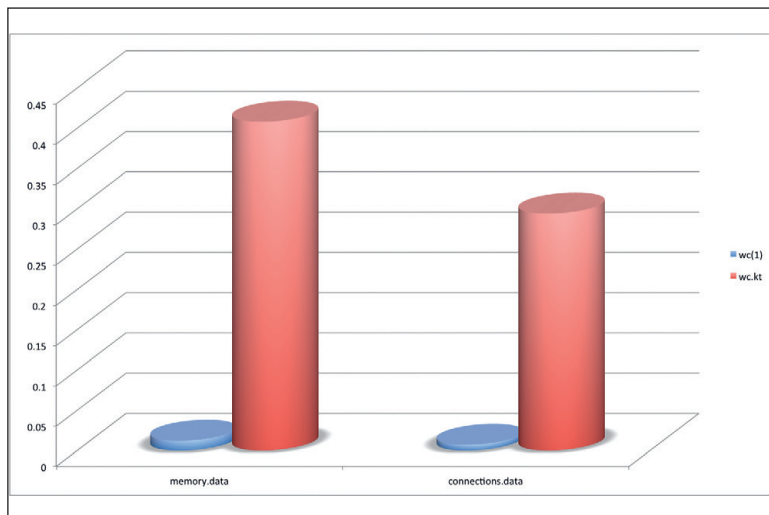
При запуске файла **wc.kt** будет получен следующий результат:

```
$ kotlinc wc.kt -include-runtime -d wc.jar
$ java -jar wc.jar /tmp/adobegc.log
6776 128048 809229
$ wc /tmp/adobegc.log
6776 128048 809229 /tmp/adobegc.log
```

Для проверки работы программы **wc.kt** используется вывод утилиты *wc*. Желая улучшить программу, можете добавить поддержку нескольких имен файлов. Это не так уж сложно, если вы разобрались в программе **countChar.kt** из прошлого урока по Kotlin (стр. 82) — в основном это относится к поочередной обработке аргументов командной строки.

Потоки

Наконец, кратко проиллюстрируем возможности параллельной обработки в Kotlin, написав программу, которая создает два потока.



► Рис. 4. На этом графике сравниваются быстродействия утилиты *wc(1)* и программы *wc.kt*, с помощью утилиты *time(1)*. Как видите, *wc(1)* намного быстрее!

В Kotlin есть стандартная функция потока, позволяющая создавать новые потоки. В общем случае потоки завершаются после выполнения всех действий. Однако, если вы захотите остановить поток досрочно, нужно вызвать функцию `stop` класса `Thread`. Здесь мы ее рассматривать не будем.

Эта программа называется **threads.kt**, и в ней нам наиболее интересен следующий фрагмент:

```
class thread():Thread() {
    override fun run() {
        for (i in 0..99) {
            print(". ")
            Thread.sleep(1000)
        }
        println()
    }
}
```

Программа создает новый класс потока, перегружая метод `run()` класса `Thread`. Каждый поток выводит 100 точек на экране и ожидает в течение некоторого времени после окончания каждой операции вывода. После создания потока и его назначения переменной надо выбрать метод `start()`, чтобы поток начал выполняться:

```
var t1 = thread()
t1.start()
```

После компиляции и запуска файла **threads.kt** мы получим следующий результат:

```
$ java -jar threads.jar
.....
```

Если вы захотите подробнее узнать о программе **threads.kt**, внимательно изучите ее код и поэкспериментируйте с ним. **LXF**

Производительность Kotlin

Хотя создание утилиты *wc* на Kotlin — это хорошее упражнение, столь же важно иметь возможность проверить производительность программы и сравнить ее с производительностью утилиты *wc*, которая входит в состав Linux.

Для измерения времени выполнения утилиты *wc* и программы **wc.kt** мы воспользуемся удобной утилитой *time*. Размеры двух используемых файлов таковы:

```
$ wc /tmp/memory.data /tmp/connections.data
121143 484572 1658375 /tmp/memory.data
40381 201905 565389 /tmp/connections.data
161524 686477 2223764 total
```

Вывод утилиты *time* будет походить на нижеследующий:

```
$ time java -jar wc.jar /tmp/memory.data
121143 484572 1658375
real 0m0.425s
```

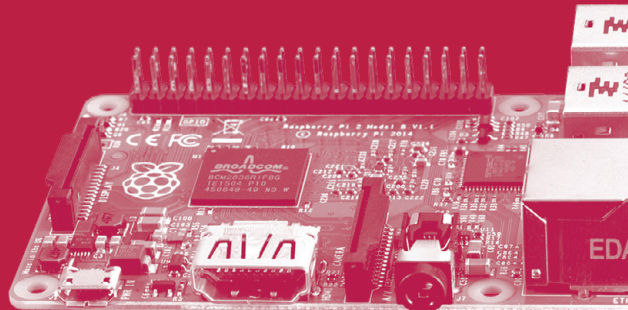
```
user 0m1.058s
sys 0m0.081s
```

На рис. 4 (вверху) показана производительность программы **wc.kt** и утилиты *wc*, которые запускались дважды, при обработке двух сравнительно больших файлов.

Как видите, наша программа на Kotlin работает довольно-таки быстро. Но с *wc* ей тем не менее не сравниться!

LINUX FORMAT Пользователям

Pi



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

**РЕЙЧЕЛ
ЛАНКАСТЕР**
глава отдела
продаж и сбыта
в micro:bit
Educational
Foundation.



Привет...

Пару лет назад моя жизнь изменилась. По работе я стала заниматься одноплатами компьютерами и инструментами для любителей электроники и создателей цифровых проектов. Поначалу было страшновато, ведь я не имела дела ни с техникой, ни с программированием и не отличала бы свой же Raspberry Pi от своего же Arduino.

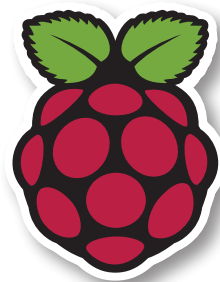
Сейчас, три года спустя, я пока не претендую на звание эксперта в любой из этих областей, но у меня уже есть опыт разработки новых продуктов для BBC micro:bit и Codebug, проведения тренингов по использованию BBC micro:bit для физических вычислений, я уже устраивала мастер-классы для детей, помогла организовать Maker-фестиваль.

Всего этого я добилась благодаря огромной помощи со стороны своих друзей. Умельцы — это самое крутое, самое открытое и дружелюбное сообщество, с которым я когда-либо сталкивалась. И в более официальном формате, к примеру, на форумах продуктов для таких плат, как Raspberry Pi, и в менее — в Twitter или лично на мероприятиях вроде Makerfaire, отношение всё то же: общая радость от того, что кто-то хочет чему-то научиться, и готовность делиться своим временем и знаниями.

Это сообщество открыто всем, что ни спросить или попросить. Нужна ли помощь в отладке фрагмента кода или хороший поставщик виджетов, кто-нибудь всегда поможет. А взамен от вас ждут лишь ответного энтузиазма и готовности делиться тем, что умеете вы. Сообщество — уникальный источник бесплатной техподдержки, вдохновения и мотивации. Абсолютно для всех.

Pi на ниве просвещения

Компьютерных классов на базе Pi в Западной Африке становится всё больше.



История Raspberry Pi являет нам невероятные и потрясающие чудеса, которые никогда бы не стали реальностью, если бы не это доступное, компактное и многофункциональное устройство. В 2016 г. мы рассказывали об образовательном проекте в Западной Африке по созданию компьютерных лабораторий в Того, где Raspberry Pi 2 используются для запуска Ubuntu Mate (см. LXF216). Данный проект был основан в 2012 г. Домиником Лалу [Dominique Laloux] и имел своей целью привлечь к ИКТ студентов и преподавателей в тех областях, где 75% учителей никогда не использовали компьютер. Спустя пять лет строится уже третья очередь компьютерных лабораторий, и более 1000 студентов смогут теперь получить доступ к ИТ-оборудованию. Подробнее об этом читайте на сайте www.initic.be.



BrailleBox

Система чтения по Брайлю на базе Pi.

Джо Берча [Joe Birch], инженера из Android, к созданию BrailleBox подтолкнула плохая семейная наследственность по зрению. Его идея в общем проста: машина преобразует новостные статьи из Интернета в шрифт Брайля. Устройство использует соленоиды для физического создания движущихся точек Брайля и AndroidThings для работы ПО. Подробнее на www.hackster.io/hithere-joe/braillebox-braille-news-reader-e86060.



М-мм, JAM!

Получите свою порцию авторских цифровых проектов.

В декабре проходили:

6 декабря

Leeds JAM Колледж Суоллоу-Хилл

9 декабря

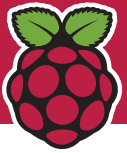
Bognor Regis JAM Богнор Реджис Кампус

Cornwall Tech JAM Библиотека Бодмина

Torbay Tech JAM Библиотека Пейгнттона

Чтобы узнать о других JAM'ах недалеко от вас, посетите www.raspberrypi.org/jam/calendar.





MotoZero

Злой гений Лес Паундер разбирается с простеньким контроллером двигателя, который потянет все четыре колеса! Для его армии роботов нет преград!

Вкратце

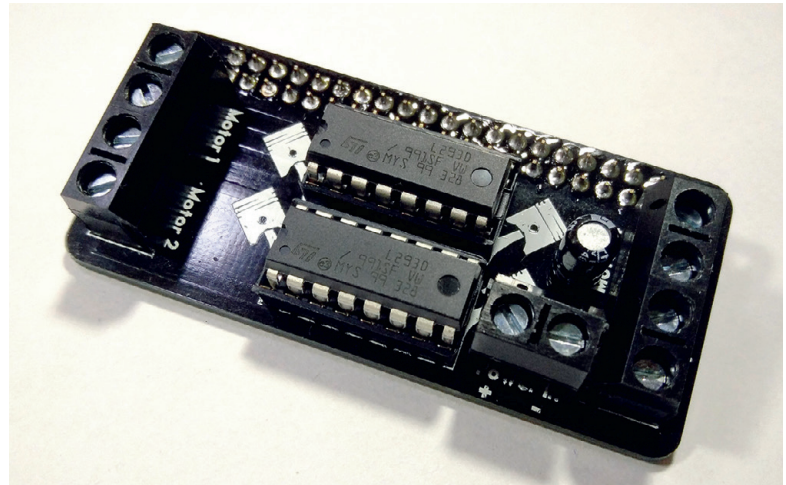
» Разработанный для линейки Pi Zero, но совместимый со всеми 40-контактными GPIO Pi, это набор для начинающих, на базе двух надежных контроллеров мотора L293D. Это значит, что он более чем потянет для малогабаритных и средних роботов, требующих полного привода. Набор совместим с многими языками программирования благодаря простому соединению с GPIO.

Создавать роботов с Raspberry Pi стало очень модно; каждый день являет миру что-то новое. Но продвигаться дальше простого двухколесного робота, нередко умеющего вращать головой и приводимого в действие одним H-мостовым контроллером, считается уже непросто, ведь нужны дополнительные двигатели и еще один H-мост.

Контроллер MotoZero представляет собой удобную плату как раз для этих целей. Изначально разработанный для плат Pi Zero, MotoZero поладит и с другими 40-контактными GPIO Pi, но помните об ограниченном пространстве между винтовыми клеммами двигателя 3 и 4 и портами USB и Ethernet.

Благодаря двум микросхемам L293D, MotoZero способен управлять четырьмя двигателями одновременно. L293D присутствуют на рынке уже довольно давно и ценятся благодаря низкой стоимости и встроенной защите. L293D в состоянии управлять двигателями постоянного тока на 600 мА и поддерживает напряжение от 4,5 до 36 В. Так что мощности и возможностей для начинающих изобретателей хоть отбавляй! Радует и то, что L293Ds поставляется в формате DIP, благодаря чему замена неисправного L293D относительно безболезненна.

На создание MotoZero из комплекта для самостоятельной сборки опытному умельцу понадобится минут 10. Имеющиеся в Сети инструкции понятны и дают краткое пошаговое руководство по сборке платы; объясняется даже, почему комплект поставляется в виде набора. Конструкция



» MotoZero — простая, но эффективная плата, дающая свободу в выборе оборудования и способов ее использования.

платы проста, винтовые клеммы моторов расположены на противоположных концах, обеспечивая понятное и надежное соединение с вашими моторами. Поскольку Raspberry Pi не рассчитан на такое потребление мощности, есть дополнительное соединение с винтовыми клеммами, предназначенное для внешнего источника питания моторов.

Обойдемся без инструкций

Но плата — ничто без программного обеспечения, а специального ПО с MotoZero не поставляется. Зато MotoZero можно использовать с любым языком, обеспечивающим доступ к GPIO. Так, на Python мы можем использовать для управления двигателями RPi, GPIO и GPIO Zero, и даже работать с Node-RED, при его библиотеке GPIO.

Управлять двигателем через GPIO Zero оказалось легко, понадобилось только настроить контакты, используемые для моторов, и еще один «пусковой» для их включения/выключения. Для всех соединений с мотором задействованы только 12 контактов GPIO, а значит, большинство остаются свободными, в особенности, SPI, I2C и UART. Как же обидно, что они не выведены на плате отдельно, для более легкого доступа.

MotoZero — отличная плата, которая легко собирается и встраивается в ваш робот-проект. Хорошо, что плата выполнена в том же форм-факторе, что и Pi Zero, и для

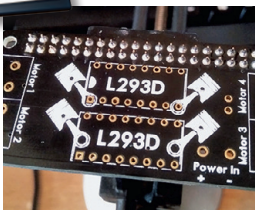
большой надежности конструкции можно использовать изоляционные прокладки между MotoZero и Pi Zero. Выбор контроллера мотора, L293D, обеспечивает достаточное количество запасных частей, если вы умудритесь сломать чип.

Отсутствие специальной библиотеки программирования никак не сказывается, поскольку компенсировано тем, что плата поддерживает множество языков. То, что MotoZero можно использовать с GPIO Zero и RPi.GPIO, позволяет легко «вклинить» плату в существующий проект.

Проект MotoZero широко используется в роботах, которые участвуют в ежегодном мероприятии Pi Wars. И это надежная основа для вашего следующего боевого робота — сто пудов! LXF

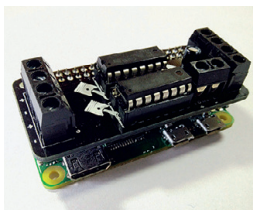


Свойства навскидку



Простота сборки

Пятью этот комплект — сплошное удовольствие. Нужны только понятные инструкции и немного терпения.



Форм-фактор Pi Zero

Конструкция MotorZero проста, хорошо продумана и симметрична, чтобы наш робот хорошо держал равновесие.

LINUX FORMAT Вердикт

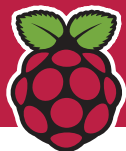
MotoZero

Разработчик: The Pi Hut
Сайт: <http://bit.ly/moto-zero>
Цена: £ 10

Функциональность	8/10
Производительность	9/10
Удобство в работе	9/10
Справданность цены	9/10

» Маленький, мощный и стоит своих денег. Прекрасно собирается и подойдет пользователям разных уровней.

Рейтинг 9/10



Node-RED: Контакты GPIO

Кому он нужен, этот Python? Лес Паундер создает простой web-интерфейс для управления GPIO на Raspberry Pi, используя Node-RED.



Наш эксперт

Лес Паундер помогает таким организациям, как Raspberry Pi Foundation и micro:bit Foundation, продвигать в народ навыки умельцев. Он ведет блог на www.bigles.es.

Вам нужны

- » Любая модель Raspberry Pi
- » Последняя версия Raspbian
- » Два светодиода (зеленый и красный)
- » Два резистора на 220 Ом (красный, красный, коричневый, золотой)
- » Кнопка или мгновенный переключатель
- » Макетная плата
- » Шесть коннекторов-перемычек мама-папа
- » Подключение к Интернету

Когда речь заходит об использовании GPIO на Raspberry Pi, первое, что приходит на ум — это Python. Однако на нашем уроке мы расскажем, как управлять и взаимодействовать с вводом/выводом с GPIO с помощью Node-RED — специального графического языка для устройств IoT.

Схема для этого проекта довольно проста; его суть в том, чтобы использовать Node-RED для управления светодиодами (выходы) и реагировать на нажатия кнопок (входы). Постройте схему согласно диаграмме и затем включите свой Raspberry Pi.

Node-RED присутствует в Raspberry Pi по умолчанию, но прежде чем начинать писать код, надо установить библиотеку *dashboard* для Node-RED. Тогда мы получим еще несколько узлов, требуемых для написания нашего кода. Для этого надо попасть в терминал; соответствующий значок находится в верхнем левом углу рабочего стола Raspbian. Чтобы установить библиотеку *dashboard*, перейдите в каталог Node-RED, скрытый внутри нашего домашнего каталога, по команде

```
$ cd ~/.node-red/
```

Из этого каталога введите команду для установки библиотеки с помощью Node Package Manager (*npm*):

```
$ npm i node-red-dashboard
```

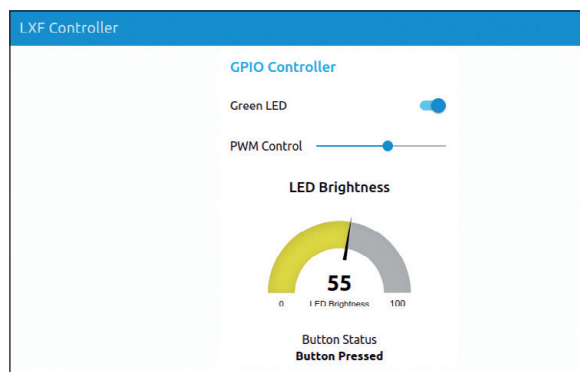
По окончании установки перезагрузите свой Raspberry Pi и убедитесь, что установка завершена. Затем откройте новое окно терминала. В этом новом окне терминала мы запускаем Node-RED в виде службы, работающей в фоновом режиме:

```
$ node-red start
```

Через несколько минут вы увидите нечто вроде

```
30 Nov 10:09:30 - [info] Server now running at http://127.0.0.1:1880/
```

Скопируйте URL-адрес и вставьте его в окно браузера *Chrome*, затем нажмите Enter. Вы увидите редактор Node-RED. Нужные



» Web-интерфейс прост и ясен. То, что нужно для управления устройствами с экрана, благодаря несложному, но функциональному фреймворку Bootstrap.

нам узлы находятся в левой части экрана. В центре мы создаем "Flows [Блок-схемы]", это наши алгоритмы или кодовые последовательности. Справа расположены три вкладки: *Info* [Информация] предоставляет информацию об узле; *Debug* [Отладка] сообщает выходные данные из узлов по мере продвижения потока; а вкладка *Dashboard* [Приборная панель] используется для создания и изменения информационной панели.

Приступаем к построению

Начнем с создания панели управления, которая будет нашим GPIO-интерфейсом.

Перейдите на вкладку *Dashboard*, выберите + *group* и нажмите *Edit* [Редактировать]. Назовите группу **LXF Controller** и нажмите *Update* [Обновить].

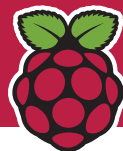
Далее наша задача — создать группу, где будут храниться все элементы нашего пользовательского интерфейса. Наведите указатель мыши на вновь созданную **LXF Controller**, и вы увидите пункт + *groups*. Щелкните по нему, чтобы создать новую группу. Теперь наведите указатель мыши на группу 1 и выберите *Edit*, затем переименуйте группу в контроллер GPIO и нажмите *Update*, чтобы сохранить установки. Со вкладкой *Dashboard* на этом всё, так что давайте добавим наш первый переключатель для управления зеленым светодиодом.

Прокрутите список доступных узлов — вы увидите набор под названием *Dashboard*. Там будет узел под названием *Switch*. Перетащите этот узел в поток 1, затем дважды щелкните по нему. Измените настройки переключателя следующим образом (остальные параметры оставьте, как есть):

- » **Group** [Группа]: GPIO Controller [**LXF Controller**]
- » **Label** [Метка]: Green LED [Зеленый светодиод]
- » **On Payload** [Активен] [**Число 0-9**]: **1**
- » **Off Payload** [Неактивен] [**Число 0-9**]: **0**
- » **Name** [Название]: Green LED

Ввод у нас теперь есть, и надо подключить его к выводу, нашему зеленому светодиоду. В списке узлов Raspberry Pi найдите *gr1 gpio* — точнее, узел с серой точкой слева. Это узел вывода. Перетащите его в поток, а затем щелкните и перетащите коннектор с узла переключателя зеленого светодиода на узел *gr1 gpio*. Их мы соединили, теперь перейдем к редактированию узла *gr1 gpio*. Щелкните по узлу дважды, и появится новое диалоговое окно. Задайте контакт как GPIO27 и тип как цифровой выход, а узел назовите *Green LED*.

Для проверки, что он работает, нажмите на кнопку *Deploy* [Развернуть], она находится в правом верхнем углу экрана. Чтобы увидеть панель управления вашего проекта, откройте новую вкладку браузера, зайдите на <http://127.0.0.1:1880/ui>, найдите переключатель зеленого светодиода и нажмите на него. Лампочка должна включиться и выключиться.



Теперь давайте создадим что-нибудь посложнее. Мы создадим новую последовательность кода во Flow 1, отдельно от нашего предыдущего кода. Начнем с узла slider [ползунок] на панели инструментов. Настройте это следующим образом (прочие настройки, опять же, оставьте по умолчанию):

- » **Group:** GPIO Controller [LXF Controller]
- » **Label:** PWM Control
- » **Range** [Диапазон]: от 0 до 100 шаг 1
- » **Name:** PWM Control

Нажмите Done [Готово], чтобы сохранить сделанное. Наш следующий узел — еще один узел вывода gpio из класса Raspberry Pi. Соединим эти узлы вместе.

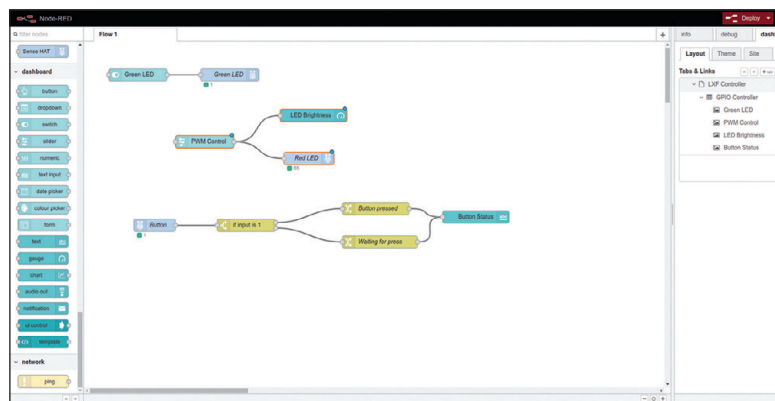
На сей раз мы установим вывод GPIO в GPIO17 с типом PWM. Назовите узел Red LED [Красный светодиод]. Нажмите Deploy, вернитесь на вкладку Dashboard, и вы увидите ползунок. Сдвиньте его, и красный светодиод будет реагировать соответствующим образом. Чтобы добавить на Dashboard немного дополнительной информации, давайте вернемся к нашему потоку и используем узел gauge [калибровка] из панели управления. Перетащите его в поток PWM Control и подключите узел slider PWM Control к датчику. Затем дважды щелкните по узлу датчика и отредактируйте, как показано далее. Остальные настройки, как раньше, оставляем по умолчанию:

- » **Group:** GPIO Controller [LXF Controller]
- » **Label:** LED Brightness [Яркость светодиодов]
- » **Units** [Единицы измерения]: LED Brightness
- » **Range:** min 0 max 100
- » **Name:** LED Brightness

Нажмите Done, чтобы сохранить настройки, а затем Deploy, чтобы произвести изменения. Вернитесь на вкладку Dashboard, и вы увидите датчик, который будет реагировать при перемещении ползунка PWM Control.

Всегда на связи

Наша последняя задача — включить физическую кнопку для обновления информации на Dashboard. Начните новую последовательность кода во Flow 1, и на сей раз используйте узел ввода gpio. Установите его в GPIO22 и присоедините Resistor. Назовите узел Button [Кнопка]. Теперь у нас настроен физический вход; давайте создадим поток, который будет реагировать на нажатие. Для этого нам надо использовать узел Switch [Коммутатор] из класса функции. Подключите его к нашему узлу кнопки. Дважды щелкните по узлу Switch и назовите его «если ввод равен 1»; внизу таблицы должно добавиться «==». Сохраните это как значение по умолчанию и измените текстовое поле так, чтобы в нем стоял 0



» Node-RED использует «узлы», объединенные в «поток»; число потоков, выполняющих параллельные или последовательные задачи, не ограничено.

(ноль). В нижней левой части таблицы нажмите + add [добавить]. Тем самым вы добавите еще одно правило. В раскрывающемся списке для этого нового правила выберите Otherwise [Иначе]. Это означает, что для этого узла у нас есть два выхода. Если отправленное значение равно 0, будет переход к выводу 1, в противном случае — к выводу 2. Из узлов функций перетащите Change [Сменить] и подключите его к выводу 1 (вверху) нашего предыдущего узла. Дважды кликните по узлу и задайте имя Button Pressed [Кнопка нажата], далее установите msg.payload в Button Pressed и нажмите Done для сохранения. Теперь перетащите еще один узел Change и подключите его к выводу 2. Отредактируйте этот новый узел, назовите его Waiting for Press [Ожидание нажатия] и установите msg.payload на Waiting for Button Press [Ожидание нажатия на кнопку], затем нажмите Done.

Последним узлом в этой последовательности будет Text [Текст] из Dashboard. Дважды щелкните по узлу Text и отредактируйте его следующим образом (прочие параметры оставляем по умолчанию):

- » **Group:** GPIO Controller [LXF Controller]
- » **Label:** Button Status [Статус кнопки]
- » **Layout** [Расположение]: Метка над значением
- » **Name:** Button Status

Нажмите Done, чтобы сохранить свои настройки, затем Deploy, чтобы внести изменения. Вернитесь на вкладку Dashboard — там появится новый виджет, под названием «Статус кнопки, ожидание нажатия кнопки». Нажмите кнопку на плате, и текст изменится.

Итак, всё получилось; теперь мы немного знаем о том, как использовать GPIO для ввода и вывода с Node-RED. **LXF**

Скорая помощь

Не уверены, что получится в выводе? Возьмите его узел отладки и подключите к потоку. Вы увидите вывод из потока на вкладке отладки, справа от интерфейса Node-RED. Очень удобно, если надо проверить, правильно ли считывается RSS-канал и срабатывает ли датчик.

Использование Node-RED с другого устройства

На нашем уроке мы создали панель инструментов с помощью браузера Raspberry Pi и кода, написанного с Node-RED. Однако последнее можно сделать и с другого компьютера: всё, что нам нужно — это IP-адрес нашего Raspberry Pi.

Проще всего открыть терминал на вашем Pi и ввести

```
$ hostname -I
```

Вы увидите однострочный IP-адрес.

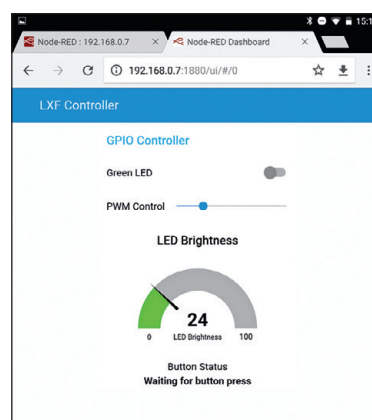
Если IP-адресов несколько, Raspberry Pi подключается через Ethernet и Wi-Fi одновременно, затем выбирает один IP-адрес и использует его.

Зная IP-адрес, зайдите на него с другого компьютера или планшета, приставив

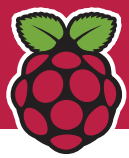
к нему порт:1880, чтобы увидеть интерфейс Node-RED. Например, наш Pi был на 192.168.0.7:1880.

Итак, теперь мы в состоянии написать код Node-RED с помощью нашего компьютера или планшета, а также можем использовать интерфейс панели управления, просто добавив /ui в конец адреса. Например, наша панель размещалась на 192.168.0.7:1880/ui.

С Node-RED можно работать через браузер Chromium на самом Raspberry Pi, но делать это через сеть с более мощной машины гораздо приятнее. Кроме того, такой подход позволяет группам совместно работать над проектом.



» Благодаря структуре Node-RED, панелью управления легко пользоваться с планшета или мобильного устройства.



RFID: Контроллер

Ночной дозорный Лес Паундер показывает, как создать собственную систему безопасности, используя Raspberry Pi, магнитный замок и читалку RFID.



Наш эксперт

Лес Паундер является независимым разработчиком и хакером, специализирующимся на Raspberry Pi, micro:bit и Arduino. Лес путешествует по Великобритании, помогая людям научиться умению разработчика. Он пишет блог на <http://bigl.es>.

В проекте этого месяца мы построим собственную систему безопасности дверей, пригодную для регистрации входящих на территорию, употребив недорогие карты с RFID (радиочастотной идентификацией). Карты RFID читаются устройством RFID-RC522, которое выдает каждой карте уникальный идентификатор. Он сверяется со словарем допустимых карт, и при совпадении срабатывает реле, отключающее цепь 12 В, питающую магнитный замок. Однако берегите пальцы: магнит могуч и может создавать усилие в 280 кг!

Высокое и низкое

Этот проект состоит из двух частей: низковольтной цепи для Raspberry Pi и высоковольтной цепи 12 В для дверного магнитного замка. Pi подключен к читалке RFID через интерфейс SPI на GPIO. Кнопка открывания двери тоже включена в эту цепь. Мы подключаем плату реле к Pi, давая ей 5 В и Землю [GND], а затем подключаем GPIO17 Pi к IN1 на плате реле. Это позволяет Pi запускать реле и оставаться изолированным от цепи 12 В.

Цепь реле 12 В получает питание через 12V+ от источника постоянного тока 12 В. Он подключен к центральному клеммнику. И подключен через контакт NC (нормально замкнутый) к магнитному замку двери. GND источника постоянного тока 12 В подключена к GND магнитного замка. Поэтому, когда реле срабатывает, цепь DC12V+ замыкается на замок двери, включая магнит и блокируя дверь. См. схему вверху справа.

Для этого проекта требуется установить довольно много библиотек. Во-первых, SPI-Py — особая ветка, созданная Саймоном Монком [Simon Monk]. Эта библиотека требует ручной установки через `git clone`, поскольку ее нет в репозиториях `pip`, поэтому нам надо также установить `logzero`, простой инструмент ведения журнала Python, который умеет записывать внешние файлы и обеспечивает легко узнаваемый вывод в оболочке Python. Поэтому, чтобы упростить установку программного обеспечения, мы написали небольшой скрипт `Bash`, который загружается и запускается в терминале следующим образом:

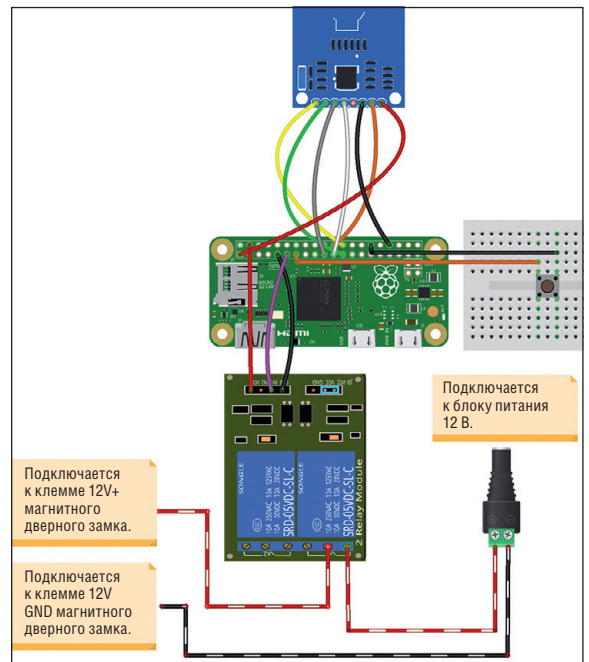
```
$ curl https://raw.githubusercontent.com/lesp/LXF230-RFID-Door-Lock/master/install.sh | bash
```

Нам также необходимо включить интерфейс SPI. Итак, на рабочем столе Raspbian перейдите в главное меню, затем в Preferences, затем выберите Raspberry Pi Configuration. В новом меню найдите Interfaces и включите интерфейс SPI. Нажмите OK, чтобы принять изменения. Сейчас самое время перезагрузить Pi и вернуться за рабочий стол.

Пишем код

В `Thonny` выберите File > Save, вызовите проект `RFID-security.py` и сохраните файл в каталоге `LXF230-RFID-Door-Lock`. Не забывайте сохраняться.

Наши первые несколько строк Python — это библиотеки. Сначала мы импортируем из GPIO Zero классы `Button` и `OutputDevice`. С классом `Button` [англ. кнопка] всё ясно, а `OutputDevice` используется для подключения Raspberry Pi к реле. Следующие две библиотеки — `SimpleMFRC522` и `time`. Опять же, `time` очевидна, а другая используется для подключения и получения данных от читалки RFID-RC522. Последние два импорта относятся к `logzero`. Это



► Схема разделена на две части: цепь низкого напряжения, подключенная к Pi, и цепь с более высоким напряжением, подключенная к реле и магнитному дверному замку.

библиотека Python, обеспечивающая простой способ создания лог-файлов и форматирование данных, которые выводятся в оболочку Python:

```
from gpiozero import Button, OutputDevice
import SimpleMFRC522
import time
import logzero
from logzero import logger
```

Теперь давайте создадим три объекта, которые мы применим для подключения нашей платы реле к GPIO 17 Raspberry Pi; затем создадим кнопку открывания на GPIO27 и соединение с читалкой RFID:

```
relay = OutputDevice(17)
release = Button(27)
reader = SimpleMFRC522.SimpleMFRC522()
```

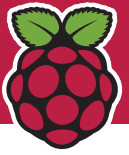
Следующая строка кода использует свеже созданный объект `relay`. Она включает контакт GPIO, что заставит включиться реле. С другой стороны реле находится наша цепь 12 В, в данном случае — контакт «+» 12 В, подключенный к реле как «нормально замкнутый». Это означает, что когда реле включено, цепь замкнута, и на магнитный замок двери может поступать питание, вызывая его срабатывание.

```
relay.on()
```

Теперь, чтобы регистрировать все события входа и выхода из здания, создадим файл журнала, используя `logzero`. Он называется `access.log` и имеет максимальный размер 1e6, 1 МБ. Также создается до трех резервных копий журнала, с использованием ротации файла журнала:

Вам нужны

- Любой Pi, Raspbian
- RFID-RC522 (The Monk makes Clever Card Kit)
- Плата реле
- Кнопка
- Макетная плата
- 10 соединителей мама—мама
- 2 соединителя папа—мама
- Блок питания 12 В, 2 А
- Гнездовой соединитель для постоянного тока
- Клеммник, например, Wago
- Магнитный дверной замок (<http://bit.ly/2yEfqR0>)
- Код: <https://github.com/lesp/LXF230-RFID-Door-Lock/archive/master.zip>



Считывание ID

Мы сделали отличный инструмент, который откроет дверь по предъявлении действительной карты; но как узнать свой идентификационный номер, чтобы добавить его в словарь? Решение состоит в том, чтобы создать еще один проект Python в *Thonny*, и на сей раз мы назовем его **card-check.py**. Начнем код импортом библиотеки *SimpleMFRC522*, а затем создадим объект, который позволит подключиться к читалке.

```
import SimpleMFRC522
reader = SimpleMFRC522.SimpleMFRC522()
```

Затем надо вывести некоторые инструкции пользователю.

```
print("Для проверки ID подержите карту около читалки")
print("Закончив, нажмите Ctrl+C для выхода")
```

После этого начнем цикл, который попробует запустить код внутри себя:

```
try:
    while True:
```

Внутри цикла мы считываем параметры карты и выводим их в оболочку Python:

```
id, text = reader.read()
print("ID этой карты",id)
```

Две последние строки сообщают пользователю о выходе из приложения:

```
except KeyboardInterrupt:
    print("Выход")
```

Сохраните код и нажмите Run > Run Script. Теперь попробуйте помахать карточкой перед считывателем — надеюсь, она опознается. Не забудьте записать идентификатор карты, а затем добавить его в словарь в главном скрипте.

```
logzero.logfile("access.log", maxBytes=1e6, backupCount=3)
```

Чтобы открыть магнитный дверной замок, нам надо написать функцию, которая будет принимать идентификатор пользователя карты, предъявленной считывателю, и записывать его в журнал. Наша функция `opendoor` принимает один аргумент, идентификатор карты. Он передается регистратору и записывается в журнал с использованием формата `info`, который в выводе оболочки окрашен в зеленый цвет.

Затем мы записываем имя пользователя в журнал, используя словарь с именем `cards` и ключ, который является идентификатором карты. Подробнее об этом позже. Следующий шаг — выключить реле, разорвав цепь 12 В, отключить магнитный замок и разрешить пользователю войти. Затем мы выводим в оболочку Python, что дверь открыта, после чего код ждет пять секунд, прежде чем снова включить цепь 12 В и заблокировать дверь.

```
def opendoor(id):
    logger.info(id)
    logger.info("Пользователь "+str(cards[id]))
    relay.off()
    print("ДВЕРЬ ОТКРЫТА")
    time.sleep(5)
    relay.on()
    print("ДВЕРЬ ЗАПЕРТА")
```

Теперь создадим еще одну функцию для обработки ситуации при нажатии кнопки открывания. Она зарегистрирует, что дверь была открыта, и выключит реле, чтобы отключить магнит так же, как с помощью карты RFID.

```
def button_open():
    logger.info("Нажата кнопка открывания")
    relay.off()
    print("ДВЕРЬ ОТКРЫТА")
    time.sleep(5)
    relay.on()
    print("ДВЕРЬ ЗАПЕРТА")
```

Чтобы иметь список авторизованных пользователей, мы создадим словарь, называемый `cards`. Словарь — это тип хранилища данных, в котором используется отношение ключ-значение. Поэтому, когда мы видим идентификатор пользователя с карты RFID, он используется как ключ для поиска в словаре и возвращает значение для этого ключа, в данном случае — имя пользователя. Идентификационные номера в этом словаре найдены сканированием тегов через читалку, о чем вы можете узнать больше во врезке вверху.

```
cards = {642581926087: "Вася Иванов", 572050203913: "Маша Петрова"}
```

Перейдем к основному циклу нашего проекта. Мы используем цикл `while True` для постоянного запуска кода внутри него. Первая задача — проверить, нажата ли кнопка открывания, и если да, то запустить функцию `button_open`, которую мы создали ранее.

Обратите внимание, что вызов функции не имеет круглых скобок, потому что это вызывало бы функцию при каждом повторе цикла, а не при нажатии кнопки:

```
while True:
```

```
    release.when_pressed = button_open
```

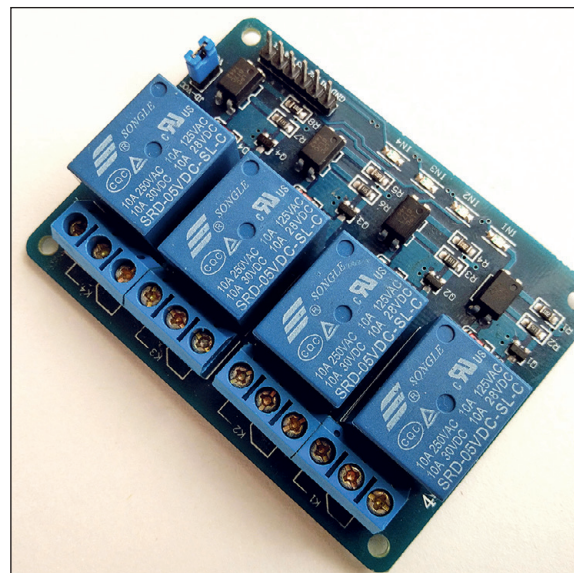
Далее мы вводим задержку 0,1 секунды для выполнения кода и создаем две переменные: `id` и `text`. Переменная `id` используется для хранения идентификатора карты RFID, предъявленной читалке. Затем мы выводим идентификатор в оболочку:

```
    time.sleep(0.1)
    id, text = reader.read()
    print(id)
```

Чтобы проверить идентификатор предъявленной карточки в нашем словаре, надо использовать проверку условия `if`. Она проверит, «есть ли `id` в словаре `cards`»; и если есть, то запустится функция `opendoor(id)`, открывающая дверь и записывающая данные пользователя в журнал доступа. Но если пользователь неизвестен, дверь останется закрытой, а данные запишутся в журнал:

```
    if id in cards:
        opendoor(id)
    else:
        print("Карта неизвестна")
        logger.error(id)
```

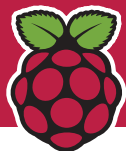
Завершив код, сохраните работу и нажмите Run > Run Current Script, чтобы запустить проект. Теперь предъявите считывателю одну из карточек, и если она есть в словаре, дверь откроется. Если нет, вы можете ее добавить — просто прочитайте врезку вверху для получения справки. **LXF**



Скорая помощь

Желая сделать свой код Python исполняемым, добавьте `#!/usr/bin/env python3` в его начало. Затем в терминале скопировать `chmod +x NAME OF FILE`, чтобы сделать его исполняемым. Наберите `./NAME OF FILE`, и ваш код запустится.

➤ Реле бывают различных видов и размеров. Мы использовали четырехканальную плату реле стоимостью около \$4.



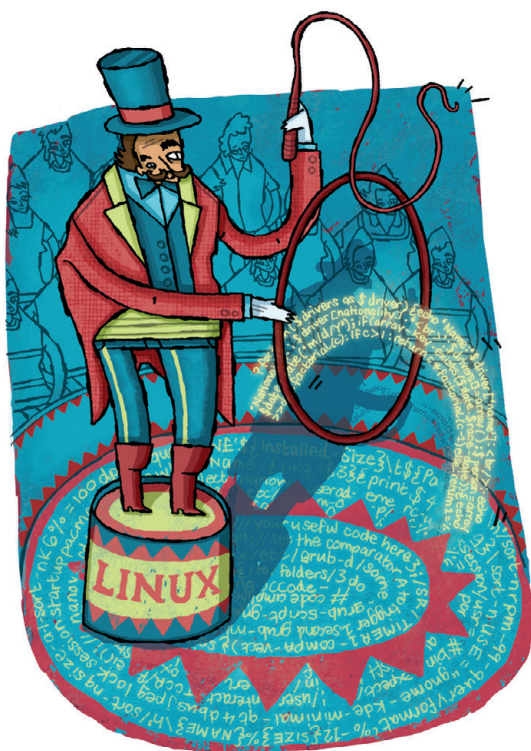
Wi-Fi: Ставим GPS на карту

Стюарт Бернс настраивает Raspberry Pi 3, чтобы по дороге сканировать и наносить на карту сети, для удовольствия и ради выгоды. Довольны все!



Наш эксперт

Стюарт Бернс — специалист по виртуализации для компании Fortune 500. Когда он не занят созданием облачных серверов, его можно найти копящимся в системах Linux и создающим интересные штуки.



Смонтируйте Pi как можно выше, чтобы ничего не перекрывало обзор неба. Это улучшит прием GPS и Wi-Fi.

Пароль для входа в Raspberry по умолчанию — “raspberrry”. Первым делом надо разрешить Pi загружаться без подключения HDMI-кабеля. Вам понадобится войти в Pi напрямую. Сделайте это, отредактировав файл настройки /boot/config.txt (sudo nano /boot/config.txt).

Найдите строку hdmi_force_hotplug=1 и раскомментируйте ее, затем сохраните изменения и выйдите. Это лучший способ обновления ПО Pi.

Полностью проводной

Чтобы настроить удаленный доступ, следующим шагом будет установка проводного Ethernet-соединения для установки всех пакетов, необходимых для этого проекта. Если вы ограничены во времени, можно использовать встроенный в плату Wi-Fi, но позднее через встроенный Wi-Fi мы будем напрямую подключаться к Pi, и если вы используете беспроводное соединение, его придется перенастроить. Измените настройки проводной сети, скомандовав

```
sudo nano /etc/network/interfaces
auto eth0
iface eth0 inet static
address 10.0.0.177
netmask 255.255.255.0
gateway 10.0.0.254
dns-nameservers 8.8.8.8
```

Подключите кабель, если вы еще не сделали этого. Внесите изменения в режиме live с помощью команды

```
sudo service networking restart
```

Проверьте, работает ли IP-адрес: используя ifconfig eth0, убедитесь, что изменения внесены. Raspbian предлагает инструмент

Среди преимуществ Raspberry Pi — портативность и компактность. Чтобы продемонстрировать этот факт, заодно сделав что-то интересное, наш учебник расскажет, как настроить Pi 3, чтобы запускать его от автомобильного прикуривателя, одновременно позволяя водителю собирать местоположение сетей Wi-Fi и прочие данные с помощью USB GPS и внешнего модуля USB Wi-Fi. А затем мы объясним, как нанести эти данные на карты Google.

Сперва скачаем файл Raspbian Lite Image с www.raspberrypi.org/downloads/raspbian. Скачайте ZIP-файл и распакуйте его. Убедитесь, что распакован только файл образа. На данный момент должен быть только один файл, заканчивающийся на .img. Используйте Etcher, чтобы записать образ (скачивается с www.etcher.io). Вставьте карту MicroSD и откройте Etcher. Вначале выберите карту назначения и нажмите Start. На запись и удостоверение образа уйдет несколько минут.

По завершении размонтируйте карту MicroSD, а затем вытащите ее и вставьте снова. Смонтируйте ее как диск FAT32. На данном этапе у вас должен быть смонтирован раздел /boot. Чтобы включить удаленный доступ, создайте пустой файл с именем ssh в папке загрузки. Сделайте это командой

```
touch /media/boot/ssh
```

Теперь размонтируйте карту, вставьте ее в Pi и включите питание.

Вам нужны

- » Raspberry Pi 3
- » Адаптер USB Wi-Fi с режимом мониторинга:
- » TP-Link TL-WN72N
- » Модуль USB GPS: U-Blox 7
- » USB-прикуриватель
- » Обратите внимание: адаптеры должны стоить менее £20



для управления Pi, *raspi-config*. Кроме обновления, у него есть еще несколько применений. Чтобы использовать его, скомандуйте

```
sudo raspi-config
```

Выберите опцию 8 (обновление). Находясь в *raspi-config*, задайте новый пароль и, при желании, имя хоста с помощью соответствующих опций. Закончив, перейдите к кнопке завершения и нажмите Enter. В этом пункте может понадобиться перезагрузка. Тогда перезагрузитесь и заново подключитесь через SSH, как это делалось выше.

Теперь пойдет самое интересное, а именно, настройка GPS и USB Wi-Fi. Подключите приемник GPS. Через несколько секунд он должен зарегистрироваться. Чтобы проверить, виден ли он, как положено, скомандуйте

```
sudo lsusb
```

Теперь устройство GPS отобразится в списке (в нашем учебнике устройство показывается как U-blox AG). Если вы не уверены, удалите устройство и повторите команду выше, сравнив полученные результаты. Чтобы устройство GPS работало, надо, чтобы оно распознавалось как последовательное [serial] устройство. Увидеть, какое устройство было зарегистрировано, поможет команда

```
sudo dmesg | grep -i usb
```

С нашей текущей настройкой устройство показывается как `ttyACM0`:

```
cdc_acm 1-1.2:1.0: ttyACM0: USB ACM device.
```

Для проверки, что принимается должное устройство, используйте следующую команду для вывода данных GPS в консоль:

```
sudo cat /dev/ttyACM0
```

Есть несколько способов посмотреть данные GPS, включая команду *gpsd*. У нее есть ряд зависимостей; некоторые из них она делит со сканером Wi-Fi, *Kismet*. Вначале установите следующие зависимости:

```
sudo apt-get install -y screen gpsd libncurses5-dev libpcap-dev tcpdump libnl-dev gpsd-clients python-gps ntp
```

Чтобы сервис GPS работал нормально, необходимы несколько опций. Отредактируйте файл `/etc/default/gpsd` (`sudo nano /etc/default/gpsd`) и измените файл в соответствии с показанным ниже. Некоторые строки комментариев удалены для краткости...

```
START_DAEMON="true"
```

```
USB_AUTO="true"
```

```
DEVICES="/dev/ttyACM0"
```

```
GPSD_OPTIONS="-n"
```

```
GPSD_SOCKET="/var/run/gpsd.sock"
```

Для включения местоположения GPS при загрузке используйте команду `sudo systemctl enable gpsd.socket`.

Чтобы убедиться, что всё работает как надо, читатель может использовать команду *cgps*. Она должна показать экран консоли в приятном формате, со множеством данных GPS, включая ваши текущие координаты и высоту над уровнем моря. Используйте Ctrl+C для выхода. Если всё не работает, мы бы посоветовали в первую очередь проверить, работает ли сервис (и потенциально перезагрузиться, если налицо проблемы). Используйте команду `systemctl status gpsd`. Вдобавок, если устройство GPS не определяет никаких координат, выведите его из окна. Мы обнаружили, что двойной стеклопакет препятствует получению сигнала!

Настройте интерфейс

Настало время настроить интерфейс WLAN1 Wi-Fi. Вначале убедитесь, что новый контроллер виден, с помощью той же команды `lsusb`, что и раньше. К несчастью, встроенный Wi-Fi, поставляемый с Pi 3, пока не работает для наших целей, поскольку не поддерживает режим Monitor. Более подробную информацию о том, почему это так, см. во врезке на стр. 99.

Следует отметить, что режим Monitor [мониторинг] именно это и делает: выполняет мониторинг сети, но не подключается к беспроводным сетям — даже не может этого! Затем вам понадобится установить ПО *Kismet*. Это инструмент сканирования Wi-Fi, который мы будем использовать. *Kismet* надо компилировать из исходника — его нельзя просто установить, поскольку его нет в репозиториях или пакетах. Мы используем последний релиз на момент написания. Следуйте командам ниже для скачивания и установки *Kismet*:

```
wget http://www.kismetwireless.net/code/kismet-2016-07-R1.tar.xz
```

```
tar -xvf kismet-2016-07-R1.tar.xz
```

```
cd kismet-2016-07-R1/
```

```
make dep
```

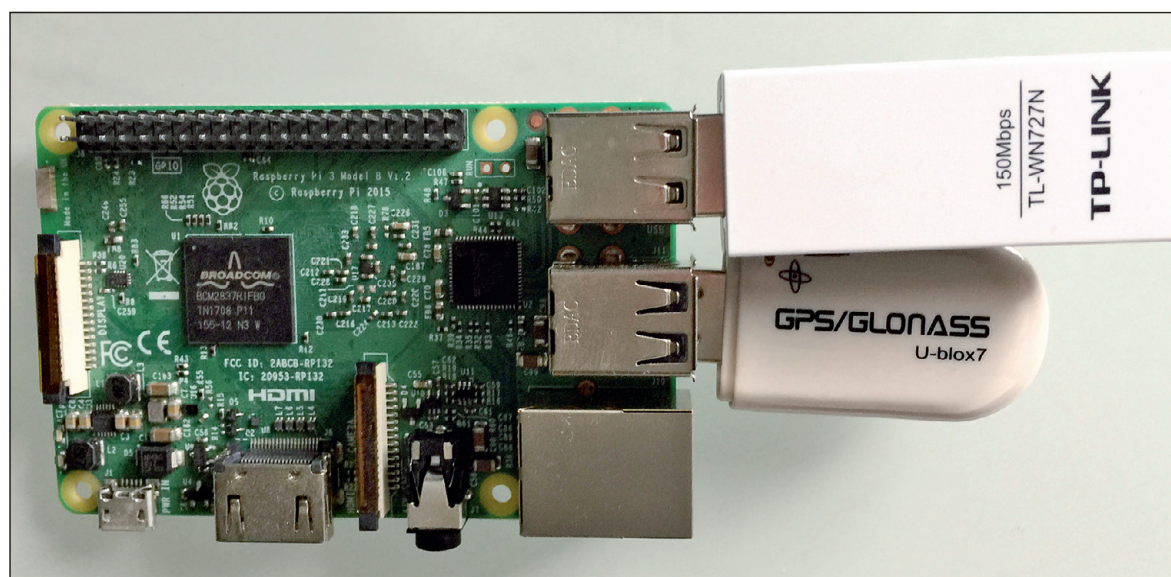
```
sudo make install
```

Этот процесс займет немало времени (30, а то и более минут), поскольку все программы компилируются из исходника. *Kismet* можно скомпилировать для работы без прав root, но это за рамками нашего вводного проекта. Будет множество предупреждений, но волноваться не о чем. В конечном итоге система вернется в командную строку. *Kismet* может принимать аргументы командной

Скорая помощь

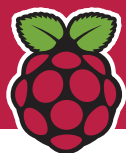


Хотя наш учебник занимается Raspberry Pi, ту же самую настройку можно с легкостью использовать на ноутбуке. Большинство команд, обсуждаемых здесь, будет работать на любой системе Debian. Примечание: сначала убедитесь, что ваша карта Wi-Fi поддерживает режим мониторинга.



» Вот полностью настроенный Pi с добавленными функциями Wi-Fi и GPS. Не громоздкий, и вы видите его внутреннюю красоту!

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



строки, но в данном сценарии статья использует файл настройки *Kismet*. Единственное, что нужно задать — это интерфейс для использования. Отредактируйте файл *Kismet conf*:

```
sudo nano /usr/local/etc/kismet.conf
```

Раскомментируйте строку `pcsource=wlan0`, а также измените `wlan0` на `wlan1`.

Теперь можно запустить *Kismet*, и он обнаружит все точки локальных сетей и их геолокацию. Хотя запускать *Kismet* от имени `root` не рекомендуется, для наших целей это сойдет. Запустите его командой `sudo kismet`. Она загрузит GUI *Kismet* и задаст ряд вопросов. Для навигации можно использовать клавишу `Tab`.

Лучшая процедура запуска — оставить настройки в покое. Когда у вас спросят, желательно ли соединиться с сервером *Kismet*, нажмите `Yes`. Надо надеяться, что появится лог консоли, отображающий все сети, которые он может на данный момент обнаружить. За сценой сервер *Kismet* также создает несколько групп лог-файлов. Если тестовый прогон прошел как ожидалось, и в `/home/pi` есть лог-файлы, пора выключить Pi и сделать всё по-настоящему. Для выхода из *Kismet* используйте `Ctrl+C`.

Общедоступно

Последний большой пункт настройки — настройка `wlan1` в качестве точки доступа. Эти шаги настройки основаны на официальном руководстве по Raspberry Pi (www.raspberrypi.org/documentation/configuration/wireless/access-point.md). Это означает, что ваш ноутбук может видеть и подключаться к Pi, не вникая в IP-адреса. Чтобы настроить эту конфигурацию, требуется две программы: *dnsmasq* и *hostapd*. И снова, установите их с помощью

```
sudo apt-get install -y dnsmasq hostapd
```

dnsmasq позаботится о DHCP, а *hostapd* обеспечит всё необходимое для настройки точки беспроводного доступа. Вначале нам понадобится изменить способ работы `wlan0`. Отредактируйте файл настройки DHCP (`sudo nano /etc/dhcpd.conf`) и добавьте следующий пункт внизу файла настройки:

```
denyinterfaces wlan0
```

Вам также понадобится задать для `wlan0` конфигурацию интерфейса. То, что делалось раньше для настройки статического IP-адреса для проводной сети, необходимо и для интерфейса `wlan0`. Отредактируйте его (`sudo nano /etc/network/interfaces`), чтобы получилось следующее:

```
allow-hotplug wlan0
iface wlan0 inet static
address 172.16.0.1
netmask 255.255.255.0
network 172.16.0.0
```

После настройки сохраните изменения и выйдите. Перезапустите сервисы, как показано ниже. Если вы не используете проводное соединение, то перезапуск интерфейса вызовет проблемы. Лучше перезапустить машину (`sudo reboot now`).

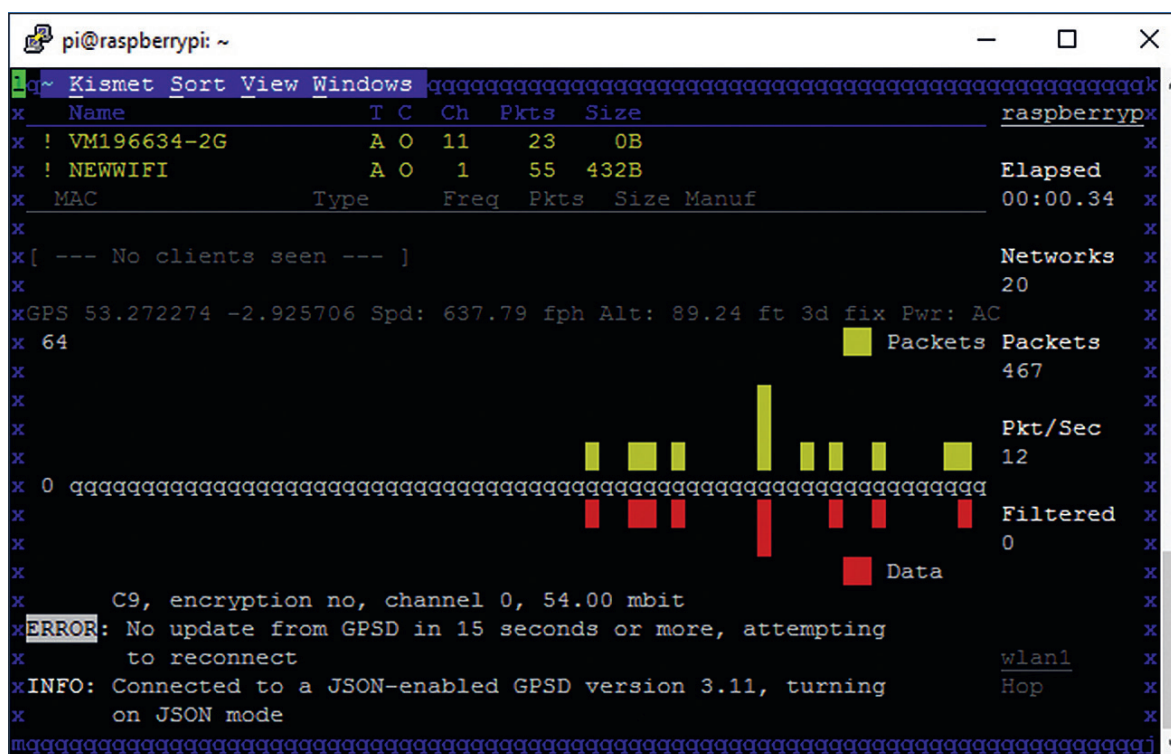
```
sudo service dhcpd restart
sudo ifdown wlan0
sudo ifup wlan0
```

Задайте настройки DNS, отредактировав файл *dnsmasq* (`sudo nano /etc/dnsmasq.conf`) с тем, чтобы добавить информацию по интерфейсу, как показано ниже:

```
interface=wlan0 # Используйте требуемый беспроводной интерфейс - обычно wlan0
dhcp-range=172.16.0.2,172.16.0.20,255.255.255.0,24h
```

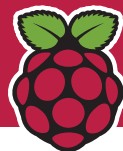
И, наконец, настройте файл конфигурации точки доступа (ТД) (`sudo nano /etc/hostapd/hostapd.conf`), добавив показанные ниже пункты:

```
interface=wlan0
driver=nl80211
ssid=MyNetwork
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```



➤ Консоль *Kismet* показывает все доступные сети и их мощность.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



Освойте режим мониторинга

Из-за чего вся суета с добавлением карты Wi-Fi? Вкратце, есть несколько режимов, потенциально доступных на сетевых картах. Эти режимы включают Monitor [Мониторинг], Infrastructure [Инфраструктура], Mesh [Сеть] и Managed [Управляемый], а также несколько других. По большей части пользователям не придется понимать или использовать режимы. В результате производители оборудования не прилагают особых усилий по созданию карт, хорошо работающих в режиме мониторинга.

Во время написания этой статьи было предпринято несколько попыток найти USB-карту,

работающую в Raspbian. Можно выяснить, с какими режимами работает ваша сетевая карта, по команде `iwconfig info wlan0` (или другой беспроводной интерфейса). Это предоставит вам всюкую информацию, включая подробности по режиму Supported [Поддерживаемый]. По умолчанию *Kismet* настраивает сетевую карту на использование режима Monitor, так что вручную это задавать не надо.

Однако можно вручную настроить карту на переключение в режим Monitor, используя следующие команды:

```
sudo ifdown wlan1
```

```
sudo iwconfig wlan1 mode Monitor
sudo ifup wlan1
```

Как отмечалось ранее, в данном случае сеть перестанет отвечать, если вы используете это для сетевого интерфейса, через который подключаетесь. Он настроен на пассивный мониторинг и отчет обо всех видимых сетях.

Беспроводная карта, которая применялась для нашего учебника — TP-LINK TL-WN727N, приобретенная онлайн примерно за £9. Обязательно проверьте ее спецификации — есть множество похожих карт, которые работать не будут.

```
wpa=2
wpa_passphrase=MyPassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Отредактируйте файл `hostapd` и строку `DAEMON_CONF`, отразив показанное ниже (`sudo nano /etc/default/hostapd`):

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Чтобы включить сервисы при загрузке, используйте такой код:

```
sudo service hostapd start
sudo service dnsmasq start
```

Перезапустите сервис, чтобы изменения вступили в силу:

```
sudo service ntp restart
```

Для проверки среды выключите Pi и отключите кабели Ethernet и HDMI. Включите его, подключите к свежесозданной ТД и получите доступ с вашего ноутбука к удаленной оболочке на ТД. Выключите Pi и поместите в автомашину. Помните, что если Pi не хватит мощности источника питания, он может загрузиться не полностью, хотя и покажет, что питание есть. Прежде чем что-то делать, проверьте требования по питанию. При необходимости можете запитать Pi от USB-порта своего ноутбука. Можно также использовать внешний аккумулятор — обязательно прочтите инструкцию, чтобы убедиться, что он сможет обеспечить необходимое питание.

Предположим, что у Pi нормальное питание и вы можете получить к нему доступ через SSH; запустите приложение *Kismet* с помощью

```
ssh pi@192.168.0.1
sudo kismet
```

Следуйте ранее описанной процедуре для доступа к консоли *Kismet*.

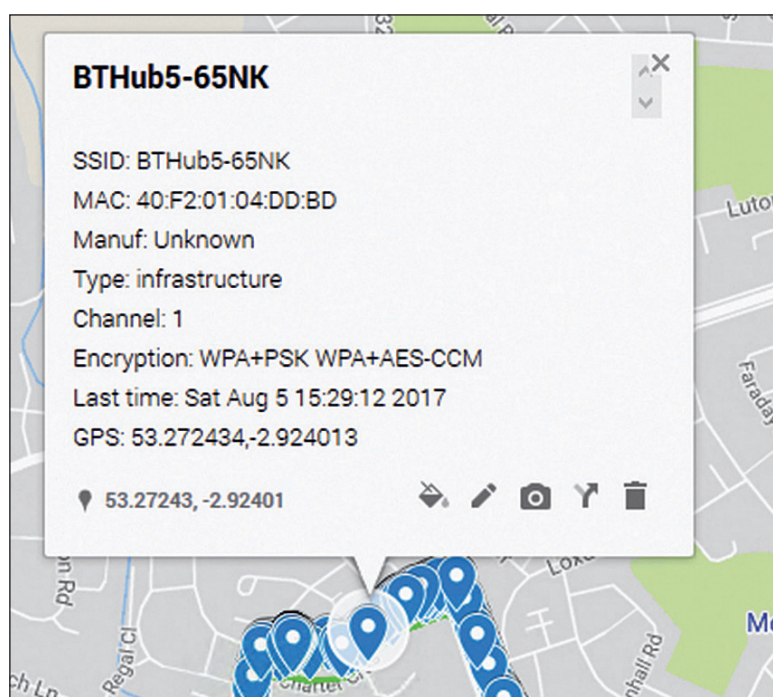
Отправляйтесь в небольшую поездку и возвращайтесь. Все данные будут собраны автоматически и помещены в файлы внутри папки `/home/pi`. С помощью ноутбука вы сможете элегантно выключить Pi, используя команду `sudo shutdown -h now`. Эти файлы содержат данные GPS и базовых станций Wi-Fi, но в настоящий момент они в формате, который непонятен картам Google Maps. Мы воспользуемся скриптом Скотта Хелма [Scott Helme], чтобы конвертировать файл в картографический. Для этого скопируйте

```
wget https://gist.githubusercontent.com/ScottHelme/5c6869e17c3e9c8b2034dc8fc13e180b/raw/31c2d34f66748b6bd26415fd7d120c06b3d92eaf/netxml2kml.py -O netxml2kml.py (1)
```

Для создания картографического файла запустите команду

```
cd /home/pi
python netxml2kml.py --kml -o output *.xml
```

Это может занять одну-две минуты, но в конечном итоге покажет информацию обо всех обнаруженных базовых станциях Wi-Fi. (1) <http://bit.ly/LXF230netxml> -O netxml2kml.py



Очевидно, данные всё еще находятся на Pi 3, так что их надо скопировать. Скопируйте файл с помощью `scp` на свой локальный ПК, заменяя при необходимости значения. Используйте следующую команду для копирования данных с Pi:

```
scp /home/pi/*.xml myuser@myworkstation:/home/myuser
```

На данном этапе файл данных должен быть в текущей директории, из которой запускалась команда. Финальный шаг — импортировать файл `xml` в Google Maps (<https://www.google.com/maps/d/home>). Выберите `Create New Map` [Создать новую карту]. Добавьте слой. Отобразится окно, позволяющее добавить вновь созданный файл. Это может занять несколько минут. По завершении читатель должен увидеть на карте все эти точки Wi-Fi, а также информацию по поддерживаемым протоколам и прочие интересные сведения.

Создайте эскиз Google Map. Для импорта данных выберите `Add New Layer` [Добавить новый слой]. Далее нажмите `Import`. Перетащите сохраненный файл. Загрузка и импорт данных займет буквально момент. После этого карта покажет местоположение, имя станции Wi-Fi, схему шифрования и прочие данные.

Мы надеемся, что наш урок был полезным. Есть множество других дополнений и настроек, применимых при выполнении данного проекта, но с ними пусть читатель экспериментирует сам. Однако сообщите нам, что у вас получилось — это ваш шанс вдохновить сообщество Raspberry Pi! [LXF](#)

► Конечный результат — карта Google со всеми собранными данными и с возможностью полной навигации.



LXF Hot Picks



Александр Толстой

укрываясь от сурового дыхания зимы, греет руки над экраном накалившегося ноутбука и исследует лучшее на сегодняшний день OSS...

Storyboarder » Borg » Calibre » Gnome Commander » MConnect » Geary » Zrandr » Glances » Klavaro » BeFive Smartball » ChickenPeppa

Приложение для рисования

Storyboarder

Версия: 13.2 Сайт: <http://bit.ly/story-b>

Будем честны: у большинства из нас нет художественных способностей, как бы нам того ни хотелось. Но иногда и пользователи Linux поддаются творческому порыву и жаждут выразить свои эмоции в виде рисунка. Конечно, в Linux предостаточно клонов *Microsoft Paint* — и у нас даже есть великолепный *GIMP*, конкурирующий с промышленным стандартом в лице *Adobe Photoshop*; но в данном обзоре мы сосредоточимся на создании композиции с нуля, а не на работе с умными технологиями обработки изображения.

Storyboarder — это программа, созданная, как понятно из ее названия, в помощь созданию собственной раскадровки или

комикса. Она даже справится с базовой анимацией. *Storyboarder* проста в работе и определенно понравится тем, кто считает нелинейные видеоредакторы слишком сложными и трудоемкими.

Пяти разных ручек и карандашей и ластика должно вполне хватить для ваших творческих потребностей. Рядом с ними вы найдете указатель цвета с палитрой из трех цветов и опцией управления размером



» Это сетевое приложение подстегнет вашу креативность.

Storyboarder поможет создать раскадровку или комикс.

кисти (используйте [and] для настройки размера). Есть несколько инструментов для масштабирования и включения/выключения направляющей решетки, чтобы помочь вам с перспективой и композицией. Конечно, лучше употребить планшет для рисования или, по крайней мере, сенсорный экран, но и мышь вполне сойдет. Можно добавить фреймов с помощью строки внизу изображения, а также добавить снимки и скрипты и настроить длительность кадра через правую боковую панель.

Уже через несколько минут рисования вы, вероятно, освоите большинство функций *Storyboarder* благодаря интуитивно-му и четкому интерфейсу. Кнопка *Photoshop* на панели инструментов дает ссылку на *GIMP* в Linux, тогда как экспорт в *Final Cut Pro X* просто делит сцену на серию PNG, добавляя описание в XML-файле. Вы также можете сохранить свою работу в виде GIF, PDF или ZIP-архива.

Storyboarder основана на *Electron*, и вам надо проверить, установлена ли в вашей системе команда менеджера пакетов *npm*. Затем установите *Electron* (www.npmjs.com/package/electron) и запустите такую команду в главной директории *Storyboarder*:

```
$ npm build && electron
```

У программы красочный экран приглашения, функция автосохранения и масса полезных подсказок, раскиданных по интерфейсу. Спешите видеть!

Исследуем интерфейс Storyboarder

Стационарный набор

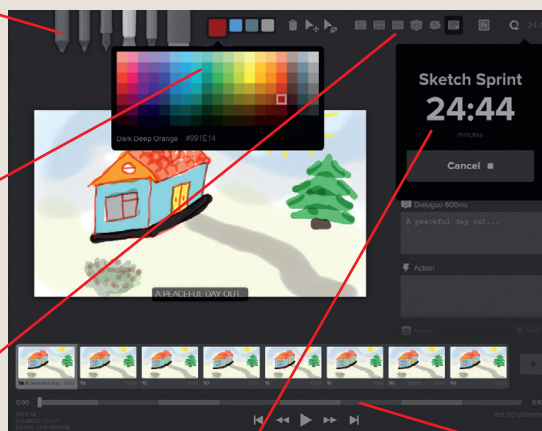
Пять основных инструментов рисования позволяют рисовать в разных стилях.

Указатель цвета

Нажмите на любой из маленьких образцов, чтобы задать цвет для основного (большого) образца.

Используйте решетку

Выберите любую из удобных решеток *Storyboarder* для исправления своей композиции или построения перспективы при рисовании.



Ускоренное рисование

Простая технология производительности, позволяющая сделать больше за меньшее время. По умолчанию у вас есть 25 минут.

Оживите свою историю

Добавьте кадры с разным содержанием и настройте их тайминг. Очень похоже на изготовление слайд-шоу.

Система резервного копирования

Borg

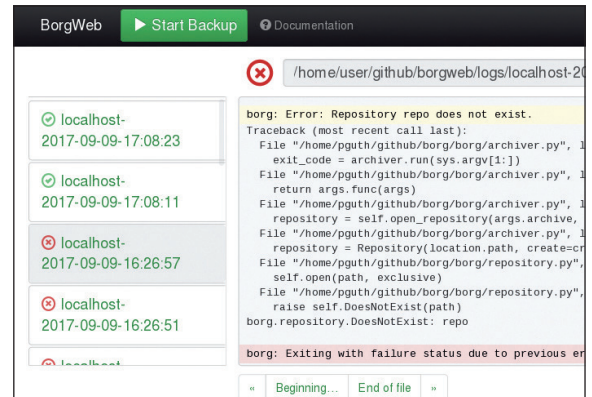
Версия: 1.1 Сайт: <http://bit.ly/borg-b>

В Linux существует целый ряд решений для резервного копирования. Вероятно, их можно сравнить с несметным количеством текстовых редакторов, браузеров или музыкальных плееров. Однако резервирование данных — важная компьютерная стратегия, затмевающая в реальном мире стоимость любых программ или оборудования, и нам отнюдь не стоит отмахиваться от подобных программ.

На сей раз мы рассматриваем *Borg*, универсальное решение резервирования данных для предприятия и домашнего использования. Это приложение командной строки, на Python, с достойными функциями — например, создание экономичного хранения резервных копий и аутентифицированное шифрование. Можно запустить инкрементное резервное копирование своих данных, сжать их с помощью *LZ4*, *zlib* или *LZMA*, обеспечить их безопасность с помощью пароля и затем управлять ими удаленно.

Последнее утверждение следует пояснить. Вы можете смонтировать свои резервные копии как тома FUSE и получать доступ к ним через свой любимый менеджер файлов. Мы не хотим продвигать один инструмент резервного копирования перед другими, так что не обязательно отказываться от своего решения резервного копирования на основе *rsync* в пользу *Borg*; но мы должны отметить, что последний очень хорош в работе с большими и постоянно изменяющимися массивами данных.

Для сокращения объема файла резервного копирования (объедините это со сжатием LZMA, чтобы еще уменьшить размер копии) *Borg* применяет дедупликацию, и отлично справляется с этой задачей. Утилита основана на Cython (это расширения C для



» **Borgweb — то, что надо, если вы хотите избежать возни с Borg в командной строке.**

Python) и оптимизирована для быстрой обработки синхронизированного кэша. Кроме этого, она потребляет ничтожно мало памяти.

Если вы пожелаете попробовать *Borg*, не пропустите сетевой GUI, разработанный вместе с основным кодом *Borg*. *Borgweb* — это web-сервер, который предполагается запускать на том же компьютере, чью резервную копию создает *Borg*. Привлекательный интерфейс предлагает большую зеленую кнопку *StartBackup* и простой для чтения лог. Поищите в менеджере пакетов своего дистрибутива Linux нечто вроде 'borgbackup' и не забудьте настроить *Borgweb* для простого просмотра файла резервной копии и управления архивом.

Хорош в работе с большими изменяющимися массивами данных.

Менеджер книг

Calibre

Версия: 3.9 Сайт: <http://bit.ly/calibre-book>

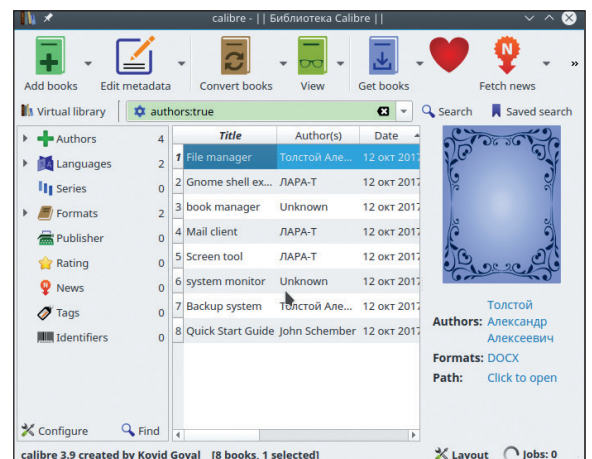
Когда автор был еще ребенком, приобретение новой книги было долгим приключением — надо было пойти в магазин и провести там часть дня в поисках нужного тома, а потом в переговорах с продавцами и кассирами. Сегодня всё по-другому, и хотя кое-кто до сих пор предпочитает ходить в книжные лавки, есть отличная альтернатива и возможность собирать книги как вам удобно благодаря специальным устройствам, именуемым электронными книгами.

Calibre — личный менеджер книжной полки, с поддержкой практически всех типов электронных книг. При первом запуске появляется мастер с несколькими вопросами, в т. ч. с диалоговым окном выбора устройства. Даже если у вас нет электронной книги, через настройку *Generic* можно использовать *Calibre* со смартфоном — конечно, если вы захотите с него читать!

Calibre — нечто большее, чем очередная «менеджер чего-то», т. е. программа,

относительно простая в разработке и эксплуатации. *Calibre* позволяет импортировать все виды документов в собственную базу данных (с хорошо документированным API), а также экспортировать в разные форматы, если вам такое понадобится. Это означает, что если у вас есть некие случайные файлы PDF, DOCX, DJVU, EPUB, FB2, HTML, разумно будет рассортировать их с помощью *Calibre*.

Приложение весьма солидное и зрелое, благодаря не одному году разработки. Через несколько часов работы становится ясно, что программа работает с книгами не хуже, чем *Digikam* с фото. Она также напоминает нам iTunes благодаря своей интеграции с физическими устройствами



» **Сортируйте свою виртуальную библиотеку по любым критериям, потом добавляйте теги и создавайте собственные рейтинги книг!**

и режимом отображения обложки 'cover flow'. *Calibre* также позволяет проверить, сколько свободного места осталось на диске вашего устройства, и умеет легко перемещать книги между ним и вашим компьютером.

Calibre — качественное приложение и отличный способ собирать книги, журналы, комиксы и прочие публикации. Оно сопровождается солидной документацией с логично расположенными разделами и даже демонстрационными видео. Не пропустите его, когда приметесь формировать свою библиотеку!

Calibre — зрелое приложение, благодаря годам разработки.

Менеджер файлов

GCommander

Версия: 1.8 Сайт: <http://gcmd.github.io>

Работы крадут у нас рабочие места, корпорации вкладывают средства в растущий рынок криптовалют, ученые строят планы колонизации Марса... и при этом миллионы людей всему предпочитают классические двухпанельные менеджеры файлов. Успех *Total Commander* в Windows неоспорим, но в Linux далеко не один двухпанельный менеджер, и нам есть из чего выбирать. Мы уже писали, например, о *Krusader*; а теперь предлагаем вам *Gnome Commander*, отличное приложение, во многом сходное с *Total Commander*.

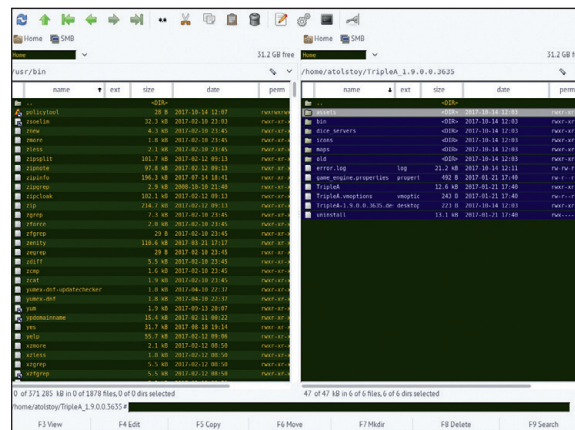
Gnome Commander следует установившейся договоренности по привязке клавиш и комбинациям и предоставляет вам знакомые клавиши для работы с файлами (F5 — копировать, F8 — удалить и т.д.), как в *Norton Commander*. Приложение по-прежнему использует библиотеку *GTK2*, и это также означает, что раскладка GUI весьма компактна и экранное место не тратится попусту (в противоположность

крупным кнопкам управления *GTK3*). Поддерживается ряд функций, что просто отлично, учитывая, насколько проста и легка в работе эта программа.

Gnome Commander служит отличной административной утилитой удаленного управления файлами. Вы можете настроить и сохранить подключения FTP, SSH, Web-Dav и даже Samba, и легко получать доступ к удаленным файлам. Программа предлагает разумные опции по умолчанию для интеграции с другими аксессуарами рабочего стола.

При двойном щелчке по файлу тот открывается в соответствующем приложении, однако если ваш менеджер архивов по умолчанию — *File Roller*, вы можете

Отлично работает как утилита удаленного управления файлами.



Gnome Commander уместает солидный объем информации по вашим файлам в приятное компактное окно.

употребить плагин *File Roller* в *Gnome Commander* и при создании новых архивов воспользоваться шаблонами имен. Менеджер файлов поддерживает вкладки, закладки, фильтрацию файлов, и предлагает обширный набор объектов для настройки по вашему личному вкусу.

Новый релиз *Gnome Commander* был в разработке около года, и наблюдать, как сильно и хорошо поддерживается это прекрасное приложение — определенно приятное зрелище.

Расширение Gnome Shell

MConnect

Версия: v15 Сайт: <http://bit.ly/m-connect>

Мы не стали бы писать об очередном расширении для рабочего стола Gnome 3, поскольку, как вы, конечно, знаете, рабочий стол поставляется только с основными функциями. А всё сверх того выводится в отдельные расширения. Однако данный экземпляр заполняет весьма важный функциональный пробел в Gnome: интеграцию со смартфонами Android.

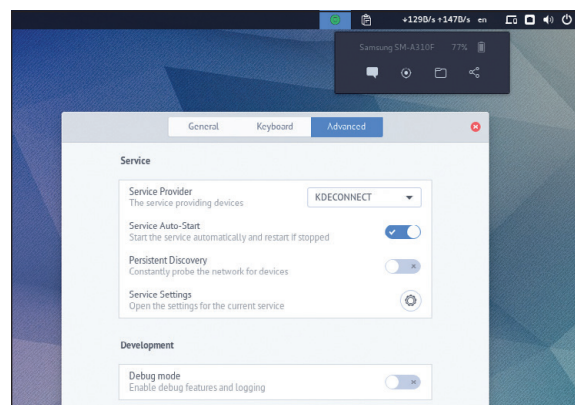
Ранее мы восхищались прекрасным *KDE Connect*, позволяющим подключить ваш смартфон на Android к рабочему столу KDE Plasma и наслаждаться такими функциями, как копирование и вставка текста, удаленный контроль, просмотр файловой системы телефона и многими другими. В Gnome у нас был *Nuntius*, приложение, пытавшееся внести такие же функции в меню HUD. *Nuntius*, похоже, уже некоторое время не обновляется; но это не беда, ведь у нас теперь есть *MConnect*, наводящий мост между *KDE Connect* и Gnome. *MConnect* позволяет получать доступ к телефону из Gnome Shell

и выполнять большинство задач, которые поддерживает *KDE Connect*, и всё это выглядит на 100% родным для Gnome.

MConnect показывает имя вашего телефона, а также кнопки для диалогового окна SMS и поиска вашего телефона (нажмите на нее, и ваш телефон зазвонит). Вы можете просмотреть файловую систему своего телефона в *Nautilus* и отправить файлы с вашего ПК на телефон. Верхняя часть всплывающего окна *MConnect* также отображает текущий уровень заряда батареи, что для некоторых линуксоидов крайне существенно.

Чтобы всё заработало, следует установить приложение *KDE Connect* для Android на свой телефон (поищите его в Google Play), но вам еще надо, чтобы ваш телефон

MConnect наводит мост между KDE Connect и Gnome.



MConnect делает больше, чем просто показывает уровень заряда батареи вашего телефона.

был в той же подсети, что и ваш ПК. Простейший способ этого достичь — подключить оба устройства к своей сети Wi-Fi. После этого вы должны быть в состоянии подключить ваше устройство к любому компьютеру Linux, где работает Gnome и *MConnect*. Последний втянет много зависимостей KDE Frameworks и *Qt5*, но нам это кажется простительным.

Почтовый клиент

Geary

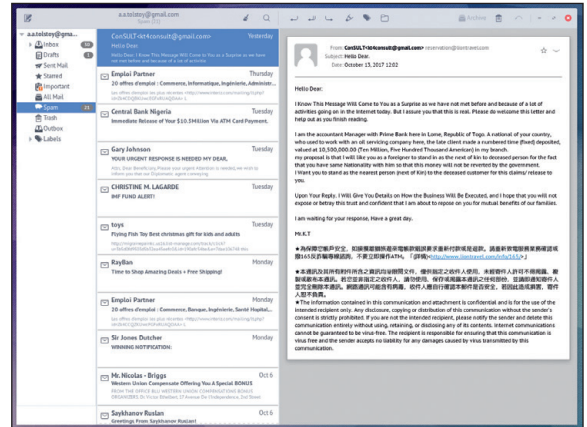
Версия: 0.12 Сайт: <https://github.com/GNOME/geary>

Geary — многообещающее приложение электронной почты, на GTK3, со множеством удобных функций и специфическим графиком выхода релизов. Данная версия вышла спустя полтора года полной тишины (это почти равно времени разработки прошлой версии) и наконец-то развеяла наши страхи по поводу судьбы этого проекта.

Основная работа проводилась над методикой создания сообщений: теперь там имеется форматирование текста, встроенные изображения и сообщения, отображенные в виде нитей беседы. В текущем релизе Geary исправлено немало ошибок; прочие изменения являются практическими и демонстрируют попытки разработчиков создать удобный и полнофункциональный почтовый клиент. Например, теперь можно выбирать более одного словаря для проверки орфографии сообщений — отличная функция для текста, включающего сообщения на разных языках. Перетащите

на сообщение изображение, и оно вставится туда, а не добавится в виде вложения.

Geary когда-то был тесно связан с проектом Elementary, но в настоящий момент перебрался в Gnome. Geary по-прежнему придерживается стиля Elementary: это простое приложение, не перегруженное расширенными функциями. В прошлом нарекания по поводу Geary касались отсутствия в нем адресной книги. Заинтригованные, мы решили посмотреть, как Geary ведет себя в домашней обстановке — иными словами, в среде Gnome — и вскоре обнаружили, что как только вы входите в одну из ваших учетных записей онлайн в Gnome, Geary автоматически собирает информацию о ваших контактах.



➤ Geary отличается быстро-той и отлично поддерживает Google Gmail.

Большинство используют Geary как интерфейс к Gmail, но помните, что Geary ладит и с любыми другими серверами POP3 или IMAP. Однако в Gmail вам надо выдать сторонним приложениям доступ к входящим сообщениям и отключить двухфакторную авторизацию. Это требование Google, а не Geary.

Вы можете установить Geary для Ubuntu из PPA ([ppa:geary-team/releases](https://launchpad.net/~ppa:geary-team/releases)) или через Flatpak:

```
$ flatpak install flathub org.gnome.Geary.
```

Другие дистрибутивы вскоре должны обновить Geary в своих репозиториях.

Перетащите на сообщение изображение, и оно вставится туда...

Инструмент экрана

Zrandr

Версия: 0.2.2 Сайт: <http://bit.ly/z-randr>

Cо временем у всех нас накапливается коллекция утилит, аксессуаров и даже рабочих приложений. Это позволяет создать собственный индивидуальный и легковесный рабочий стол с малым потреблением ресурсов и отличным набором функций.

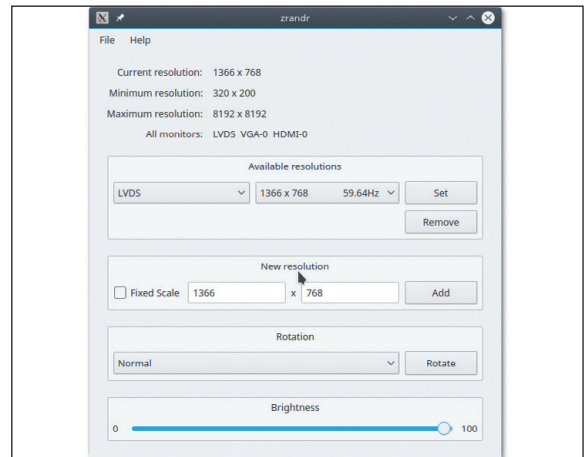
Хорошим примером является Zrandr, компактная утилита управления экраном. Для вас не секрет, что настройки экрана X11 можно контролировать из пространства пользователя по протоколу RandR, и наиболее популярнейшее приложение для данной цели — Xrandr. Героем Hotpicks этого месяца становится более современная и элегантная альтернатива Xrandr, под названием Zrandr. Она не делает ничего сверх уже достигнутого Xrandr, но позволяет создать интегрированный рабочий стол, где разрешение экрана, повороты и прочие функции будут управляться этим приложением через GUI профессионального вида.

Zrandr отображает полезную информацию: скажем, текущее разрешение экрана,

а также минимальное и максимальное разрешение, поддерживаемое вашим экраном. Ниже помещены два выпадающих меню для индивидуальной настройки разрешения, и это очень удобно, если у вас более одного монитора. Можно даже настроить собственный горизонтальный и вертикальный размер в пикселях и добавить индивидуальные разрешения в основной список. Zrandr также может повернуть экран на 90° в любом направлении и изменить его яркость (независимо от кнопок яркости на ноутбуке).

Вот в основном и все функции Zrandr, и попробовав его, мы считаем, что в сервере отображения Wayland даже близко нет ничего похожего на Zrandr, и он не вытеснит X11 полностью, просто потому что

Zrandr может повернуть экран на 90° в любом направлении.



➤ Вам не нужен полнофункциональный рабочий стол для настройки вашего экрана через аккуратный GUI как у Zrandr.

Zrandr идеален для выяснения типа подключения вашего монитора и его текущего разрешения.

Компиляция Zrandr из исходника должна быть простой, поскольку нужен только Xrandr (ищите в своем менеджере пакетов нечто типа x11-xserver-utils) и базовые заголовки Qt5. Затем введите

```
$ qmake-qt5 && make
```

Запустите приложение в папке компиляции как `./zrandr`.

HotGames Развлекательные приложения

Игра в шарики

BeFive Smartball

Версия: GIT Сайт: <http://bit.ly/be-five>

Как ни удивительно, в Linux довольно мало игр типа пинбола. *BeFive Smartball* появился, чтобы исправить баланс.

Это не прямой клон пинбола, но здесь имеются вполне реалистичный плоский стол, стальные шары, падающие мишени и целая коллекция шаров внизу. Когда игра начинается, вы должны запустить шар так, чтобы он попал в цель, и за это вы получите больше шаров для игры.

В *BeFive Smartball* нет флипперов — только пусковое устройство, которое выстреливает по шару за раз. Чем дольше вы удерживаете пусковое устройство, тем с большей скоростью из него вылетает шар. Каждая мишень защищена колючками, которые мешают шару поразить цель, так что для успешного попадания вам надо постараться рассчитать точное время пуска и его силу.

На первый взгляд у вас всегда имеется множество шаров в запасе, но *BeFive Smartball* использует таймер обратного отсчета, и хочет, чтобы вы покупали дополнительное время за запасные шары — это называется оплата кредита. Чтобы оплатить кредит, щелкните на панели управления по кнопке Payout [Выплата] либо нажмите на клавишу P.

Во время выплаты один шар обменивается на один кредит. Эта несложная экономика добавляет в игровой процесс некий баланс и делает игру более захватывающей. Вы также можете изменить вид с помощью клавиш со стрелками.

Вы должны запустить шар так, чтобы он попал в цель...



» Для простенькой браузерной аркады у этой игры на удивление привлекательная графика.

Игра написана на HTML5 и для рендеринга 3D-графики использует библиотеку Three.js на JavaScript. Установка игры не требует компиляции кода, но вам надо будет опубликовать ее на web-сервере, чтобы игра заработала. Используйте любой web-сервер, который вам нравится, например, Apache Tomcat, сбросьте папку игры в директорию web-приложений и затем зайдите в нее через свой браузер.

Игра во славу свинок, где несут яйца

ChickenPeppa

Версия: GIT Сайт: <http://bit.ly/chick-pep>

Как и все остальные коммерческие ТВ-сериалы, *Свинка Пеппа* сопровождается традиционными брендовыми товарами — мягкими игрушками, пеной для ванн, наборами бирюлек и т. п. — а также простыми настольными логическими играми и аркадами, где фигурируют любимые персонажи. Конечно, мы не можем писать о таких вещах в *Hot-picks*, но мы были счастливы откопать в сумрачных глубинах Github чудесную игру с открытым кодом под названием *ChickenPeppa*.

Это достойная реализация игры *Happy Mrs Chicken*, где используется оригинальный мультфильм, и создал ее Алексей Пономарев [Alexey Ponomarev]. И стили, и правила намеренно упрощены до предела, в духе игры по телешоу. Во время игры курица откладывает яйцо и воспроизводит забавный звук при каждом нажатии на кнопку мыши. Вам дается 30 секунд,

чтобы снести как можно больше яиц, и когда время истекает, из этих яиц вылупляются цыплята и разбегаются. Вот, собственно, и всё!

Мы признаем, что в игре есть нечто увлекательное — потому что вы можете предложить своим друзьям посоревноваться, кто снесет больше яиц за определенное время; однако главная ценность *ChickenPeppa*, конечно же, в ее схожести с мультфильмом. *Свинку Пеппу* любят не только дети: некоторые взрослые (включая автора) тоже ею очарованы, поэтому наличие на вашем компьютере любимой игры Джорджа [Джордж — младший брат Пеппы, — прим. пер.] — это удовольствие,

Вам дается 30 секунд, чтобы снести как можно больше яиц...



» Нажмите на мышшь, отложите яйцо, повторите.

которое ни за что не пропустят истинные фанаты.

Для установки игры требуется web-сервер. Поместите *ChickenPeppa* в директорию web-приложений и запустите браузер, чтобы в нее зайти. Альтернатива — установка онлайн с <http://chickenpeppa.github.io>, и доступ туда кажется таким же простым, как сама игра. Наслаждайтесь, и постарайтесь не слишком часто хрюкать!

Монитор системы

Glances

Версия: 2.11.1 Сайт: <http://bit.ly/sys-glances>

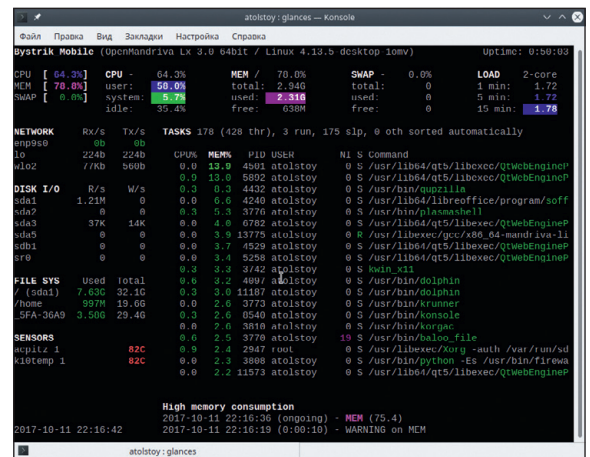
Возможно, вы помните, как *Linux Format* погрузился в технодраму а-ля Голливуд благодаря программе *Hollywood*, созданной Дастином Керклендом [Dustin Kirkland] в начале 2015 г. (см. LXF195). Программа была забавной, хотя и абсолютно бесполезной, и нам хотелось это исправить; и наконец-то — со вздохом облегчения — мы предлагаем вам *Glances*. Это приложение командной строки, которое не менее интересно, но при этом вносит в ваш эмулятор терминала огромное количество полезных вещей.

По сути, *Glances* — это расширенная замена команд *top* и *htop*, которые вам, вероятно, знакомы. Кроме того, *Glances* показывает текущие процессы вашей системы с их PID, владельцами, потреблением памяти и CPU и прочими удобными деталями. Но в то же время вы можете наблюдать за сетевыми интерфейсами с их статистикой, вводом-выводом с диска, свободным местом на ваших разделах диска, информацией с датчиков температуры ноутбука,

временем его работы и многим другим. Мы не преувеличиваем, заявляя, что *Glances* следит за вашей системой чрезвычайно внимательно.

Программа написана на Python и будет отлично работать на многих платформах, как только вы установите Python (*Glances* запускается на Windows, Linux, macOS, Android и т.д.). Файл *Readme.md* на Github содержит слишком уж много слов по установке *Glances*, потому что в большинстве случаев всё сводится к простой команде `$ pip install glances`. Однако советуем изучить документацию *Glances*, поскольку она рассказывает о многих функциях, о которых вы, возможно, даже не знаете. По крайней мере, вывод от `$ glances --help` показывает десятки функций. Например,

Glances — это расширенная замена команд *top* и *htop*.



▶ Считайте *Glances* умной альтернативой виджетам *Conky* и похожим утилитам рабочего стола.

запуск программы как `$ glances -w` выведет web-интерфейс, доступ к которому можно получить на <http://0.0.0.0:61208>.

Вы также можете использовать разные выходные шлюзы и экспортировать статистику своей системы в виде файлов CSV или JSON, или прямо на Cassandra, Elasticsearch, CouchDB, InfluxDB и множество других серверов. Обычно для большинства подобных целей вам нужны соответствующие модули Python, но это решается с помощью установщика Python — *pip*.

Учитель печати

Klavaro

Версия: 3.0.3 Сайт: <http://klavaro.sourceforge.net>

Большинства пользователей хотели бы освоить технику быстрой печати. Те, кто способен заполнить лист A4 за пару минут, чувствуют себя крутыми и продвинутыми. Если вы умеете быстро печатать, для вас не проблема записать длинные заметки или перевести бумажный документ в цифровую форму, не метаясь в поисках решения OCR.

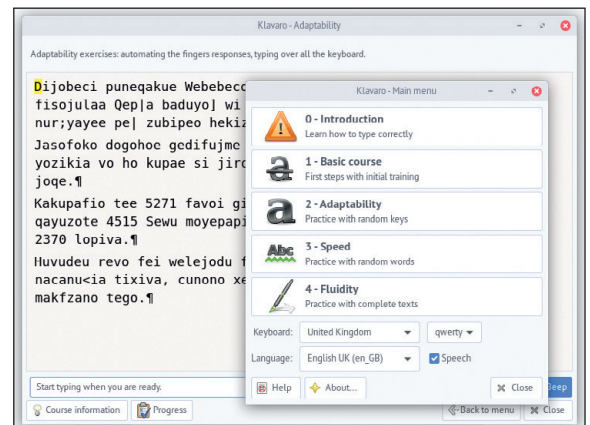
Klavaro — обучающая программа: клавиатурный тренажер, который поможет вам улучшить навыки печати. Ключ к успеху кроется в теории постановки рук, которая дается перед тем, как постепенно вас проведут по разным заданиям, неуклонно усложняемым. Первый экран *Klavaro* приглашает прочитать вводную информацию, из которой вы узнаете, что на каждой клавиатуре есть небольшие выступы на клавишах F и J, которые служат своеобразными якорями для ваших указательных пальцев и помогают сориентировать остальные пальцы по соседним клавишам; и что

оба больших пальца должны располагаться над пробелом.

Затем *Klavaro* приглашает вас пройти курс и выполнить задания по печати вслепую. Всё начинается с повторения разных комбинаций пар клавиш и продолжается печатью случайных комбинаций клавиш. Приложение фиксирует ваши очки и отображает ваше соотношение времени и ошибок после каждого задания, так что у вас есть стимул побить собственный рекорд, точно так же, как когда вы начали играть в *Angry Birds*.

Мы воспользовались шансом проверить свои способности набора и сочли *Klavaro* весьма эффективным. Уже через какие-то 10 минут у вас адски заболят пальцы

Уже через какие-то 10 минут у вас заболят пальцы и запястья.



▶ Пора поскорее напечатать какую-нибудь бессмыслицу — это единственный способ освоить искусство клавиатур-фу!

и запястья, но это цена, которую придется заплатить за приобретение ценных навыков в столь короткий срок. Скорость печати приходит после долгих дней постоянных усердных занятий!

Klavaro — кросс-платформенное приложение, доступное на многих языках, во многих случаях — с качественной локализацией и переводом. По нашему мнению, это бесценный инструмент для личного роста — и кроме того, поможет вам быстрее писать код! Загляните в свой менеджер пакетов в поисках недавнего релиза *Klavaro* и попробуйте его. LXF

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Дистрибутивы

Большая новость, сопровождающая релизы дистрибутивов в этом месяце — Canonical оказалась от Unity как рабочего стола по умолчанию для Ubuntu в пользу Gnome 3. Unity появился в Ubuntu 11.04 и был частью плана Canonical создать унифицированный интерфейс для всех устройств, настольных ПК, нетбуков и телефонов. В реальности Unity не оправдал свое имя [Unity — англ. Единство], приведя к раздорам и разделению, с массой возражений насчет нового интерфейса, и к выходу ответвлений Ubuntu с другими рабочими столами. Однако компания уперлась, продолжая совершенствовать Unity.

Конечно, любые подобные изменения влекут бурную реакцию, но это не обязательно мнение большинства. Gnome 3, на котором основан Unity, вызвал такую же бурю, впервые появившись в Fedora. Возможно, противники Ubuntu шумели громче, однако база его пользователей, несомненно, самая большая.

Хорошо, что Ubuntu не склонился перед громогласным меньшинством, но в итоге Unity провалился, а Gnome 3 развился; и хорошо, что им хватило отваги на смену курса. Беспристрастному пользователю KDE было очень забавно наблюдать за этим!

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.

Девять ремиксов рабочих столов

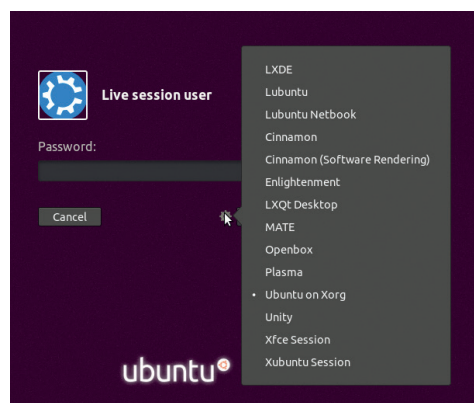
Ubuntu 17.10 Remix

Ночи в Великобритании становятся всё темнее, и это означает две вещи: начало бесконечной рождественской рекламы и новый релиз Ubuntu. С первым мы вряд ли сможем что-либо поделать; ну разве что вручить вам подарок в виде одной из наших экстравагантных вариаций Ubuntu с несколькими рабочими столами.

Этот релиз Ubuntu уже значителен — по нескольким причинам. Первая — больше нет 32-битного релиза. 32-битные пакеты по-прежнему доступны, и поэтому вы можете установить его с установочного диска, но при таком старом оборудовании вам, возможно, лучше будет выбрать более легкий рабочий стол, например, версию с LXDE (см. стр. 107). Еще одно важное изменение — рабочий стол. После стольких лет с рабочим столом Unity, Ubuntu переключился на более привычный Gnome 3. Оба работают на Gnome Shell, поэтому внутри всё одно и то же, однако само взаимодействие с рабочим столом больше похоже на другие дистрибутивы с Gnome, например, Fedora.

Однако нас не радует наличие одного рабочего стола при таком их количестве вокруг, и вот вам еще один из наших ремиксов с девятью рабочими столами. Кроме стандартного Gnome, вы можете выбирать между KDE Plasma, Xfce, Cinnamon, Mate, Enlightenment, LXDE, LXQt и даже Unity. Есть два способа выбрать рабочий стол для знакомства с ним. Можно выйти из системы и вновь в нее зайти как “Live session user”, после выбора другого рабочего стола по небольшому значку рядом с кнопкой Sign in, или перезагрузиться и выбрать рабочий стол из загрузочного меню DVD.

Установка из ремикса рабочих столов невозможна. Мы убрали значок Install, чтобы избавить вас от проблем. Выберите опцию Standard Ubuntu внизу меню



» Благодаря нашим ухищрениям, вы можете перепробовать множество рабочих столов. Расскажите нам, что у вас получилось!

загрузки Ubuntu и запустите установщик оттуда, или загрузитесь прямо в установщик с помощью опции Install Standard Ubuntu.

После установки с рабочим столом по умолчанию Gnome вы сможете установить другой рабочий стол через менеджер пакетов. Подключите репозитории universe и multiverse, затем установите *lubuntu-desktop* для LXDE, *kubuntu-desktop* для KDE, *cinnamon-desktop-environment* для Cinnamon, и т. д. Пакет, который вам требуется установить, всегда содержит рабочий стол под его именем, поэтому не пытайтесь установить компоненты рабочего стола, если не уверены в том, что делаете.

Аутентификационная информация: имя пользователя — **ubuntu**, пароля нет.



» Теперь рабочий стол Ubuntu по умолчанию — Gnome; он унаследовал прежние цвета, но в остальном совсем не похож на Unity. Придется привыкать!



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?

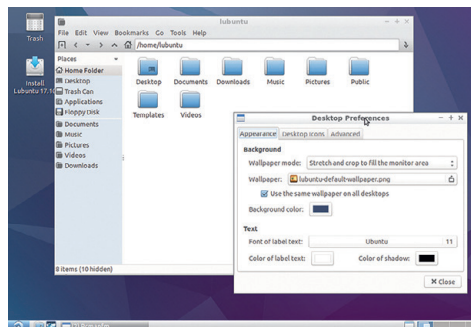
Ответы приводятся в [Index.html](#) на диске.

Дистрибутив-легковес для старых ПК

Lubuntu 17.10

На удивление много людей по-прежнему работают в 32-битных системах. Для них у нас — самый последний релиз Lubuntu. Lubuntu — это Ubuntu с рабочим столом *LXDE*, легковесная версия, отлично подходящая для менее мощного оборудования. Вы можете запустить его на 64-битной системе, но у вас будут проблемы при попытке загрузить этот дистрибутив на прошивке UEFI (UEFI на самом деле только для 64-битных систем). Желая попробовать Lubuntu на системе UEFI, включите эмуляцию BIOS или используйте 64-битный ремикс рабочих столов.

Аутентификационная информация: имя пользователя — **lubuntu**, пароля нет.



» С Lubuntu у вас всё будет быстрым и легким.

Дистрибутив на базе Trinity, идеал для новичков

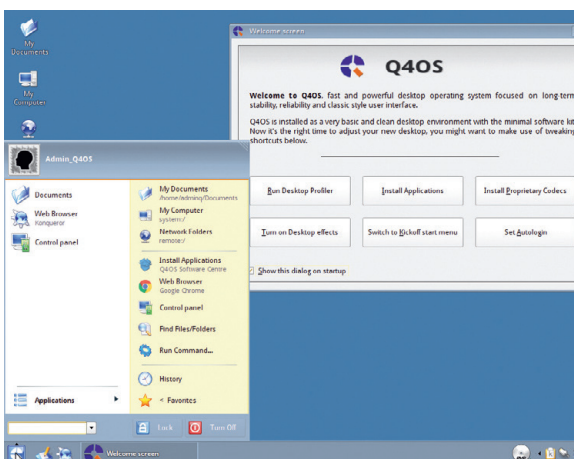
Q4OS 2.4

Когда у Gnome вышла версия 3, а Ubuntu выпустил версию с Unity, раздался вопль горя тех пользователей, которым нравился Gnome 2.x. Это привело к появлению Mate и Cinnamon — рабочих столов, которые постарались продолжить традицию Gnome 2. Когда KDE выпустил версию 4, раздался практически такой же вопль, но, похоже, особых мер не последовало: большинство в итоге просто примирилось с KDE 4.

Кроме тех, кто относится к проекту Trinity — эти люди трудятся над тем, чтобы сохранить стиль работы, присущий KDE 3.

Внимание на него обратили немногие, но у нас здесь есть дистрибутив с рабочим столом *Trinity*. Q4OS предназначен для тех, кто практически новичок в Linux (у него даже есть значок My Documents на рабочем столе), но не стоит думать, что этот дистрибутив предназначен только для беглецов с Windows. Рабочий стол KDE 3.5 был зрелым и многими любимым, и его реинкарнация *Trinity* сильно порадует тех, кто тоскует по старым добрым дням (хотя слово «добрый» здесь спорно).

Аутентификационная информация: имя пользователя — **adminq**, пароля нет.



» Самая свежая версия Q4OS подойдет тем, кто не так давно перешел на Linux.

И еще!

Системные инструменты

Главное

Checkinstall Установка tar-архива с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, обязательные быть в каждой операционной системе.

Hardinfo Инструмент тестирования системы.

Kernel Исходный код самого последнего стабильного релиза ядра.

Memtest86+ Проверьте ОЗУ на предмет ошибок.

Plop Простой менеджер загрузки для запуска ОС с CD, DVD и USB.

RaWrite Создавайте загрузочные дискиеты в MS-DOS в Windows.

SBM Независимый от ОС менеджер загрузки с несложным интерфейсом.

Wvdial Соединяйтесь с Интернетом через телефонный модем.

Чтение

Книжная полка

Расширенное руководство по скриптам Bash Изучите написание скриптов еще лучше.

Руководство Bash для начинающих Осваивайте написание скриптов Bash.

Руководство по созданию скриптов Bourne Shell Начните осваивать скрипты оболочки.

The Cathedral and the Bazaar [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

Справочник администратора Debian Базовое руководство для системных администраторов.

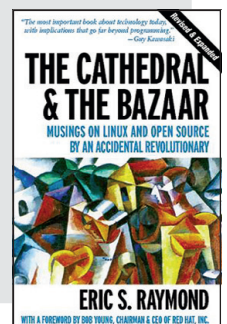
Введение в Linux Удобное руководство со множеством подсказок для начинающих пользователей Linux.

Словарь Linux Linux от А до Я.

Linux Kernel in a Nutshell [Ядро Linux в Двух Словах] Введение в ядро, написанное великим магистром ядра Грегом Кроа-Хартманом [Greg Kroah-Hartman].

Руководство системного администратора Linux Контролируйте свою систему.

Обзор инструментов Полный обзор инструментов GNU.



Пропустили номер?



Закажите его на сайте www.linuxformat.ru в «ГНУ/Линуксцентре»!
Журналы доставляются и в печатной, и в электронной форме,
так что получение нужного вам выпуска LXF может занять всего
пару минут с момента открытия браузера!



LXF229/230 Ноябрь 2017

- » Raspberry Pi как телохранитель Защищает нашу сеть
- » Шифруемся Иначе теперь не проживешь
- » Немного о сборках Дове-ряем только воспроизводимым
- » Пакеты будущего Независимо от дистрибутивов

LXFDVD: ArchLabs, Backbox, Bodhi Legacy, Sparky, LinHES, OpenELEC PC и RPi, Rescatux, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_229-230/
PDF-версия: shop.linuxformat.ru/elxf_229-230/



LXF228 Октябрь 2017

- » Linux в школе Учиться будет весело!
- » Синхронизация файлов Без третьих сторон
- » Спасите наши данные Инструментом хирурга
- » Брандмауэр для «чайников» Естественно, на Raspberry Pi

LXFDVD: CAINE, Fedora 26 Gnome, Fedora 26 LXQt, Mageia Xfce, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_228/
PDF-версия: shop.linuxformat.ru/elxf_228/



LXF227 Сентябрь 2017

- » Польза виртуальности Машины в машине
- » Бизнес-серверы Даже на дому
- » Ноутбук с Linux Как не купить кирпич
- » Биткойны и блокчейны Станьте эмитентом

LXFDVD: ClearOS, Debian, Koozali, NethServer, Voyager, Zentyal, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_227/
PDF-версия: shop.linuxformat.ru/elxf_227/



LXF226 Август 2017

- » Уходим на Linux Выбираем свободу
- » Раскидываем сети Ну-ка, проследи!
- » Анимация Почувствуйте себя Диснеем
- » Меньше ада Doom на шоколадном движке

LXFDVD: BackBox Linux, Elementary OS, Manjaro, Tails, Zorin, горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_226/
PDF-версия: shop.linuxformat.ru/elxf_226/



LXF225 Июль 2017

- » Притворимся хакерами Но не корысти ради
- » Мультки и не только Программы анимации
- » Параллельный Интернет Сеть для избранных
- » Лучше перебить Двухфакторная аутентификация

LXFDVD: Ubuntu, Solus, Android-x86, 12 книг о Linux (на английском), горячие новинки, архив LXF за 2005–2016 гг. и прочее...

Печатная версия: shop.linuxformat.ru/lxf_225/
PDF-версия: shop.linuxformat.ru/elxf_225/



LXF224 Июнь 2017

- » Конец Unity Ubuntu перешел на Gnome
- » Где KDE краше Лучший дистрибутив с KDE
- » Боронися, бабка Брандмауэр + роутер = защита
- » Кому верить? Надежные ключи шифрования

LXFDVD: Ubuntu, Linux Lite, Manjaro, Netrunner, 12 книг о Linux (на английском), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_224/
PDF-версия: shop.linuxformat.ru/elxf_224/

Подпишитесь на печатную версию журнала на www.linuxformat.ru/subscribe
или www.linuxcenter.ru, и получите электронную версию в подарок!

Телефоны отдела подписки

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



Linux Format ВКонтakte:
vk.com/linuxform

Страница 1

» Содержание



АУСТРИГУТИВЫ

Ubuntu 17.10 (64-битный)
Популярная ОС Linux для настольных компьютеров, свободно доступна с поддержкой сообщества и профессионалов. Сообщество Ubuntu постоянно на идеалах, закрепленных Манифестом Ubuntu. Ремикс с 9-ю рабочими столами.
Ubuntu 17.10 (32-битный)
Легковесная версия Ubuntu с рабочим столом LXDE.
Ubuntu 2.4 (32-битный)
Дистрибутив на базе Debian, с рабочим столом Trinity, идеальный для начинающих.

НОТРІСКС

VeGie Smartball (GIT) Игра в шарки
by:UvE-Live
Box 1.1 Система резервного копирования
by:KdGrGb
Сайте 3.9 Менеджер книг
by:KdGrGb
ChickenPerra (GIT) Вдохновленная связными игра, где несут яйца
by:KdGrGb
GSommander 1.8 Менеджер файлов
src:dfhbbdo

Безру 0.12

Почтовый клиент
dfhbbdo.com/SDOME/Beazru
Beazru 2.11.1 Монитор системы
by:UvE-Live
Калваро 3.0.3 Учебный печатки
Калваро.scribblecode.net
Миссестив 1.5 Расширение Blopple Shell
by:UvE-Live
Sturdrander 13.2 Приложение для рисования
by:UvE-Live
Zrand 0.2.2 Инструмент экрана
by:UvE-Live

УЧЕБНИКИ

ВЕСАСАС
Kolpin
Spreadsheet
ТОМОШЬ
Руководство новичка
Руководства
Ответы
ЧаВо (FAQ)

Описание 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!

ЗАГРУЗИ » ОПРОБУЙ » УСТАНОВИ!

НОВИНКА!

UBUNTU

64-БИТНЫЙ

» Ядро 4.13

» Рабочий стол Gnome

» Тысячи приложений!



Ubuntu 17.10
Ubuntu, собранный для старых ПК, с рабочим столом LXDE. **64-битный**

СВЕЖИЙ РЕЛІЗ

Содержание



Страна 2

ГЛАВНОЕ

- Checkinstall
- Coreutils
- Hardinfo
- Kernel
- Mentest86+
- Plop
- SBM
- WVDial

ДОКУМЕНТАЦИЯ: 12 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

- The Cathedral and the Bazaar** Классический текст Эрика Реймонда [Eric S Raymond] «Собор и базар»
- The Debian Administrator's Handbook** Руководство администратора, написанное разработчиками Debian
- Dive Into Python** Учебник по программированию на Python
- Intro to Linux** Начальное руководство по Linux
- Linux Dictionary** Словарь Linux, объясняющий специфическую терминологию
- Linux Kernel in a Nutshell** Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом [Greg Kroah-Hartman]
- Route** Знаменитый справочник администрирования Linux
- System Administrators Guide** Руководство по базовому администрированию Linux
- GNU Tools Summary** Руководство по работе в командной строке и обзор основных утилит GNU

ДИСТРИБУТИВЫ

- RevengeOS (64-битный)** Настольный дистрибутив на базе Arch Linux
- Системный менеджер оконным менеджером Orpexbox и панелью Xfce с меню Whisker для входа по умолчанию, графическим менеджером пакетов Patas; имеет Live-DVD.
- Robolinux 8.10 (64-битный)** Дистрибутив на базе Debian с преднастроенными виртуальными машинами, помогающими пользователям запускать ПО Windows со своего рабочего стола Linux. Поставляется с более чем 120 собственными драйверами Wi-Fi, видео и принтеров, и может запускать Windows XP, 7 и 10 без вирусов внутри. Многие популярные приложения устанавливаются одним щелчком мыши: ZIP, несколько мультимедиа-приложений, Google Chrome, Google Планета Земля, Skype и VirtualBox.

СРАВНЕНИЕ: СРЕДЫ РАБОЧЕГО СТОЛА

- Budgie 10.3.1** www.budgie-desktop.org
- Deepin Desktop Environment 15.4** www.deepin.org
- Enlightenment 0.2.21** www.enlightenment.org
- LXQt 0.12** www.lxqt.org
- Pantheon** www.elementary.io

Все дистрибутивы представлены ISO-образами, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXFDVD. У всех присутствует возможность установка на жесткий диск.

Пожалуйста, перед использованием Аджнотного Диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДВОЕКЛЕТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех этапах производства, однако, как и в случае с любым другим ПО, мы рекомендуем вам использовать апликационный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какие-либо ПО на компьютер, пожалуйста, скачайте программу для проверки целостности данных с нашего сайта www.linuxformat.ru

Тираж издательства ООО «Марком», 186852, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а, Лицензия ИДТР 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.

LINUX FORMAT

Подписывайтесь и читайте Linux Format на iPad или iPhone!

Доступно
в AppStore



А если у вас Android, подпишитесь
на Linux Format через Zinio!

 zinio™
Доступно в Google Play





В январском номере

Только лучшее... Легкие и быстрые

Не алчные до памяти и диска, быстрые на ходу...
как настроить дистрибутивы для работы
на маломощных системах.

Лучший с Gnome

Gnome вам ненавистен? Может, и зря! Сравниваем дистрибутивы,
на которых наследил этот рабочий стол.

Социальная инженерия

Обводите всех вокруг пальца, применив Raspberry Pi —
мы расскажем о надувательских трюках.

NextCloud в облаке

Стоит ли пачкать предательским ответвлением свой сервер,
когда его можно развернуть в облаке Amazon?

Откроем Red Alert!

Соберем версию классической игры *Red Alert* с открытым кодом,
готовясь к *World War III: Trumpageddon*.

Содержание будущих выпусков может меняться — вдруг мы второпях перескочим через номер...

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 1200 экз.,
распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстакова

Переводчики

Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина,
Валентин Развозжаев, Валерий Смирнов, Елена Толстакова,
Ирина Шулакова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

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 16902

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Научный редактор Джонни Бидвелл [Jonni Bidwell]

jonni.bidwell@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Клиффорд Хоуп [Cliff Hope]

clifford.hope@futurenet.com

Директор редакции Грэм Барлоу [Graham Barlow]

Старший художественный редактор Джо Гулливер [Jo Gulliver]

ПОДГОТОВКА МАТЕРИАЛОВ

Тим Армстронг [Tim Armstrong], Джонни Бидвелл [Jonni Bidwell],
Нейл Ботвик [Neil Bothwick], Стюарт Бернс [Stuart Burns], Шон Конвей
[Sean D. Conway], Нейт Дрейк [Nate Drake], Энди Келли [Andy Kelly],
Джон Найт [John Knight], Рейчел Ланкастер [Rachel Lancaster],
Бобби Мосс [Bobby Moss], Ник Пирс [Nick Peers], Лес Паундер
[Les Pounder], Шашанк Шарма [Shashank Sharma], Валентин Синицын
[Valentine Sinitzyn], Мэтт Свайдер [Matt Swider], Александр Толстой
[Alexander Tolstoy], Михалис Цукалос [Mihalis Tsoukalos], Джарред Уолтон
[Jarred Walton], Евгений Балдин, Андрей Гондаренков, Алексей Федорчук,
Максим Черепанов

Иллюстрации Шейн Коллинж [Shane Collinge]

Иллюстрация с обложки www.magictorch.com

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 (0) 344 848 2852, email: linuxformat@myfavouriteimagazines.co.uk

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 лицензируются Future Publishing Ltd (Future plc group
company). Все права зарегистрированы. Никакая часть данного журнала не может быть
повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публи-
кации, если иное не указано явно. Редакция оставляет за собой право корректировать
присланные письма и другие материалы. Редакция Linux Format получает эксклюзив-
ное право на публикацию и лицензирование всех присланных материалов, если не было
оговорено иное. Linux Format стремится оставить уведомление об авторских правах
всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора
предложенных 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 повреждение или
ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех
или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].
“GNU/Linux” заменяется на “Linux” в целях сокращения. Все остальные товарные
знаки являются собственностью их законных владельцев. Весь код, опубликованный
в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html
За информацией о журналах, издаваемых Future plc group company, обращайтесь
на сайт www.futureplc.com.

РОББО КЛУБ™

Образовательные кружки для детей

Робототехника ● Программирование ● 3D-моделирование ● 3D-печать



Обучение: от игры к профессии

Для дошкольников

- ★ **Игровая робототехника**
Возраст — от 5 лет.
Требуется уверенный навык чтения

Для школьников

- ★★ **Стартовый**
Основы знаний
- ★★★ **Базовый**
Специальность
- ★★★★ **Углубленный**
Профессия, соревнования,
собственные проекты

А ТАКЖЕ

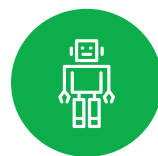
- Праздничные мастер-классы
- Городской лагерь на время каникул
- Курсы повышения квалификации для педагогов



Инновационная программа обучения

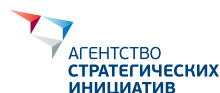


Собственная методическая база



Российские технологии и оборудование

ВООБРАЖАЙ! ПРОГРАММИРУЙ! СОЗДАВАЙ!



www.robboclub.ru

8 800 777 2985

vk.com/robboclub ● info@robboclub.ru

HETZNER
ONLINE

БЫСТРЫЙ КАК МОЛНИЯ

ОСТРЫЙ КАК ЛЕЗВИЕ

НОВИНКА



Выделенный сервер EX51-SSD-GPU

Intel® Core™ i7-6700 Quad-Core Skylake Processor
64 ГБ DDR4 RAM
2 x 500 ГБ SATA 6 Gb/s SSD
Видеокарта GeForce® GTX 10800
100 ГБ место для резервных копий
50 ТБ трафик*
Без минимального контракта
Установка 7200 рублей



7200 рублей в месяц

Идеальное решение для ресурсоёмких вычислений.

Новый выделенный сервер EX51-SSD-GPU содержит невероятно мощную графическую карту GeForce® GTX 1080 для ускорения графических приложений и быстрого 3D-рендеринга.

www.ru.hetzner.com

* Нет платы за превышение. При превышении 50 ТБ/месяц скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 85 руб. за каждый дополнительный ТБ.

Все цены вкл. НДС 18%. Цены могут измениться без уведомления. Все права защищены соответствующими производителями. Intel, логотип Intel, Core и Core Inside являются товарными знаками корпорации Intel в США и других странах.