

PI-ПРОЕКТЫ

Создадим на Python
видеоплейер, и не только!

МИНИ-СЕРВЕРЫ

Установим и настроим
крошечный web-сервер

LINUX FORMAT

Главное в мире Linux

ВИРТУАЛИЗУЕМ СВОЙ МИР

Освоим контейнеры и VirtualBox,
чтобы запускать Linux везде и всюду

**ДАРИМ
8-ГБ DVD!**

ГОТОВ К ЗАПУСКУ

НОВЕЙШИЕ РЕЛИЗЫ

- ELEMENTARY OS 5
 - UBUNTU 18.10
- & ЕЩЕ



68

**страниц
статей
и учебников**



Паскаль Геененс —
о борьбе с ботами методом
глубокого обучения

Получите новейшие сборки
приложений с Flatpak

Открытый код помогает
картографированию
3D-пространств

БЕЗОПАСНОСТЬ WEB

Вникните в самые современные
методы шифрования

ПОТРАСАЮЩИЕ ГРАФИКИ

Необходимейшее руководство
по могучему GNUplot

ДЕКАБРЬ 2018 LXF244



4 607130 825167



1 8012

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

Защита
для высокоскоростных
сетей Ethernet
100/1000 Мбит/с

Сертифицированы
ФСТЭК и ФСБ
(3-й класс защиты)

На базе
процессоров
с 64-разрядной
многоядерной
архитектурой

Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва
+7 (499)

283-86-06

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center



LINUX FORMAT



» КТО МЫ

В этом номере мы узнали все о ботнетах и о том, как люди могут с ними повеселиться. А какую подлянку вы бы учинили, окажись в вашем распоряжении ботнет?



Джонни Бидвелл

Мощь вычислений развращает. Поэтому моя рабочая машина — дряхлый ящик образца 2011 г. Было бы мило провести несколько безумных распределенных моделирований, этак для прогноза погоды. Но охотнее всего я бы забросил борьбу с каплями; помогай тут ботнет, я бы холил его, как свою плоть и кровь.



Лес Паундер

Будь в моем распоряжении ботнет, я бы заставил его прочесывать Интернет в поисках сделок со складами Dollar и магазинами Pound. Спрашиваете, зачем мне это надо? Так ведь эти магазины — золотая жила для взломанных частей проекта, особенно на Рождество и Хэллоуин.



Мэттью Хэнсон

Посидев недавно на ряде презентаций новых продуктов некой фруктоподобной компании, я возжелал ботнета, способного поразить ближайшие подключенные к Интернету кофемашины — пусть брызжут мне в лицо свежесваренным эспрессо, чтобы поддержать уровень моей энергии (и жажды жить).



Маянк Шарма

Как измученный бессонницей отец первенца, я уж точно употребляю свой домашний ботнет самым нетворческим образом — пусть делает то, что умеет лучше всего: запускает DDoS-атаку на Huggies, пока фирма не согласится назначить круглосуточный персонал по смене памперсов у моего новорожденного.



Крис Торнетт

Я твердо верю, что открытый исходный код может быть силой добра в мире. И именно поэтому собираюсь употребить мощь своего новообетенного ботнета, чтобы удалить из Интернета все фотографии Джастина Бибера и заменить их на снимки Линуса Торвальдса, с его «жестом» в сторону Nvidia.



Валентин Синицын

Ботнет — это просто распределенный суперкомпьютер из дешевого, а порой и не очень, железа. Облако, если хотите. Я бы и употребил его, как любое другое облако: установил бы какой-нибудь оркестратор, да и запускал бы приложения. Гляньте повнимательнее: не крутится ли у вас моя домашняя страничка?

До свидания!



В марте 2011 г., когда мы с Валентином Синицыным, первым редактором русского **LXF**, писали эту колонку, никто не предполагал, что когда-нибудь придется делать это в последний раз. К сожалению, издатель и владелец журнала Павел Фролов вынужден был объявить о закрытии издания по финансовым соображениям. Последние годы предприятие с трудом выходило в ноль, а после очередного повышения цен на доставку (и неизбежного вследствие этого уменьшения количества подписчиков) окончательно стало убыточным.

Деньги тем, кто успел подписаться на 2019 г., будут возвращены в самое ближайшее время. Мне остается только попросить у них прощения. «Невидимая рука рынка» оказалась сильнее нас.

От своего имени хочу сказать самые теплые слова благодарности всем авторам, переводчикам, редакторам диска и системным администраторам, кто участвовал в создании «бумажной» и электронной версий журнала. Извините, объем «Слова редактора» не позволяет перечислить вас всех поименно. Отдельное спасибо Елене Толстяковой и Сергею Рогожникову, на плечах которых в течение 10 лет и держался **LXF**. Было очень приятно и интересно работать с вами.

Самое главное «Спасибо!» хочу сказать читателям нашего журнала. Спасибо за доброе отношение и поддержку. Без вас журнала бы не получилось.

Надеюсь, что российский *Linux Format* возродится в версии 2.0. Скорее всего, уже в электронном формате. Поэтому — не прощайте, а до свидания...

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

> Тел.: (812) 309-0686 — Санкт-Петербург, (499) 283-8606 — Москва

Дополнительная информация на стр. 112

ИНЖЕНЕРНЫЙ ИННОВАЦИОННЫЙ КЛАСС

для обучения информатике, робототехнике и 3D-прототипированию
на базе свободного аппаратного и программного обеспечения



Комплектации и цены

НАИМЕНОВАНИЕ	ЦЕНА, руб.	КОМПЛЕКТАЦИЯ, шт.				
		Демонстрационный набор	Инженерный инновационный класс РОББО			
			Экспресс	Стандарт	Делюкс	Расширенный
РОББО Робоплатформа / СкетчДуино Робоплатформа	23 000	1	6	13	15	15
РОББО Лаборатория / СкетчДуино Лаборатория	11 500	1	6	13	15	15
РОББО Схемотехника стандартный набор	4 800	1	6	13	15	15
Миникомпьютеры Cubieboard	9 500	—	—	—	15	15
РОББО Цифровая Лаборатория с набором датчиков	90 000	—	—	—	1	1
РОББО Протос портативный центр прототипирования	400 000	—	—	—	1	1
3D-принтер РОББО Мини	36 000	1	1	1	1	5
Стартовый набор расходных материалов для 3D-принтера РОББО Мини (катушка пластика и синий скотч)	2 900	1	1	1	—	—
Набор аксессуаров и расходных материалов для 3D-печати (6 катушек пластика, синий скотч, инструменты для обслуживания принтера)	14 500	—	—	—	1	1
РОББО 3D-принтер: конструктор для самостоятельной сборки	30 000	—	—	—	—	6
СкетчДуино Робоплатформа конструктор для самостоятельной сборки	15 000	—	—	—	—	6
12 месяцев подписки на Электронный учебно-методический комплекс (ЭУМК)	5–6 лет	—	—	—	1	1
	7–10 лет	—	—	—	1	1
	11–15 лет	0,5	1	1	1	1
	РОББО Лагерь	—	—	—	—	1
Очное обучение преподавателя на курсе 72 часа	60 000	—	1	1	1	1
12 месяцев научно-методического сопровождения инновационного проекта в образовательном учреждении	40 000	—	—	—	—	1
Набор свободного ПО для образования, инженерного творчества и спортивного программирования	бесплатно	1	1	1	1	1
Онлайн обучение преподавателя	бесплатно	1	1	1	1	1
СТОИМОСТЬ КОМПЛЕКТА		98 200 руб.	374 700 руб.	649 800 руб.	1 452 500 руб.	1 946 500 руб.

Идея проекта

В основе проекта лежит идея создания школьного класса инноваций, где дети смогли бы обучаться робототехнике, программированию и конструированию роботов на свободном аппаратном и программном обеспечении в интерактивном режиме. Для того, чтобы ребенок был активно вовлечен в учебный процесс, все продукты компании оптимизированы для русского языка, встроены в прозрачные картриджи, демонстрируя детям микроэлектронную часть. Все компоненты можно разбирать, чтобы ребенок мог понять принцип устройства и самостоятельно создать собственное.

Дети воспринимают современные технологии как магию, а Россия практически полностью зависит от зарубежных архитекторов, разработчиков и поставщиков программного и аппаратного обеспечения. «РОББО» не только помогает разобрать оборудование и программное обеспечение и понять как оно устроено, но и улучшить его.

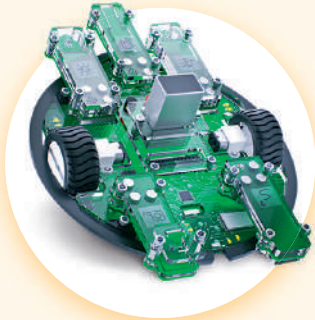
Это делает будущих отечественных инженеров настоящими инноваторами, которые будут конкурентоспособны и востребованы не только в России, но и на мировом рынке.

Оборудование



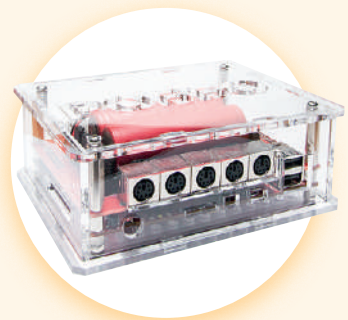
РОББО Лаборатория

Состоит из платы с датчиками и программируемыми сенсорами, которая упакована в прочный прозрачный картридж, чтобы дети могли видеть содержимое и разобраться в микроэлектронной части.



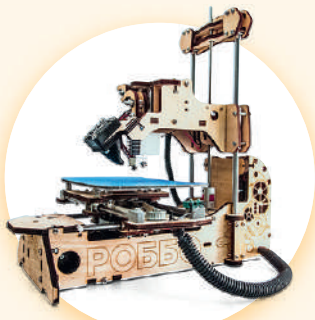
РОББО Робоплатформа

Модульная система, которую можно менять по своему усмотрению. На ней есть магнитные площадки, чтобы дети могли сами крепить разные датчики: фоторезисторы, датчики касания, линии и любые другие.



РОББО Автономная лаборатория Интернета вещей

Переносная модульная система, включающая в себя программируемый блок, который позволяет собирать и анализировать данные из окружающей среды. Цифровая лаборатория служит для работы вне класса, в рамках естественно-научных дисциплин.



РОББО 3D-принтер Мини

Устройство для создания трехмерного физического объекта на основе компьютерной модели.



Центр прототипирования РОББО Protos

Универсальная платформа, предназначенная для печати 3D-моделей из термопластов, 3D-фрезерования и резки, лазерной гравировки и резки.



РОББО Схемотехника

Набор состоит из различных электронных компонентов, которые легко подключаются к микроконтроллеру Arduino. Руководство «Конструируем роботов на Arduino» поможет разобраться с основами схемотехники и электроники самостоятельно.

Содержание

ОБЗОРЫ

Ubuntu 18.10 14

Как поживает дистрибутив после резкого разворота к Gnome? Да стабилизируется помаленьку...

Manjaro 18.0 16

Фаворит Distrowatch, Manjaro усердно работает над удобством пользователей, но настройка ему не повредит.

Fedora 29 18

Новый релиз позаботился о процессорах ARM (таких, как на Raspberry Pi) и приналег на Flatpak.

Synology DiskStation DS418 Play 19

Новое NAS — не просто хранилище: оно может вещать вашу коллекцию фильмов по всему дому.



Intel Core i9-9900K 20

Не появившись Ryzen, не было бы на свете и нового процессора от Intel. Вот что животворящая конкуренция делает!

Surviving Mars 21

Есть ли жизнь на Марсе? Нет — так будет! Если, конечно, вы обеспечите своих колонистов всем необходимым.

Megaquarium 22

Что за прелесть эти рыбки! И каждой обустрой среду обитания, накорми, защити от хищников...



Сделаем мир виртуальным

Забудем экзистенциальный страх, раскрутив новую виртуальную версию Linux с помощью глубокого машинного обучения нашего автора, на с. 32



СРАВНЕНИЕ



Всё для офиса 26

Некогда разрозненные программы сначала скучковались в настольные офисные комплекты, а потом даже взлетели в Интернет.

ИНТЕРВЬЮ



Ботолов 42

У злоумышленников есть несправедливое преимущество: они нападают первыми. Но Паскаль Геененс надеется на автоматизацию.

ПОЛЬЗОВАТЕЛЯМ RASPBERRY PI



Новости Pi 88

Raspberry Pi научился живому телевидению, умелец связал шерстяную карту звездного неба, а проект Pi-кота Petoï снова с нами!

HAT с питанием от Ethernet 89

Это, конечно, дело хорошее, но вы лишитесь доступа к GPIO — так стоит ли игра свеч?

Minecraft Zero 90

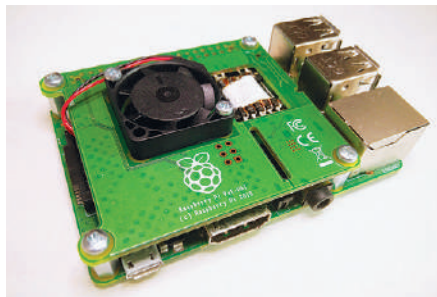
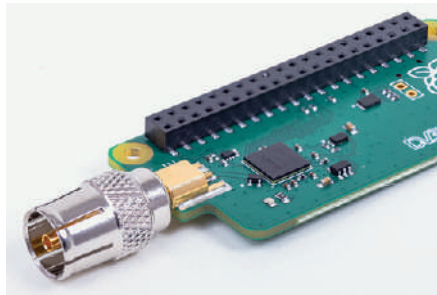
Алмазные блоки, взрывы под ногами, шумное падение с небес... дети будут рады новым командам в *Minecraft*.

Видеоплеер на Python 92

Что может показывать компьютер-крошка? Разумеется, мультики (на самом деле — не только).

Собираем печатную плату 94

Тем, кто не знает, с какого конца берутся за паяльник, мы окажем реальную помощь.



АКАДЕМИЯ КОДИНГА

Защищаем Web 76

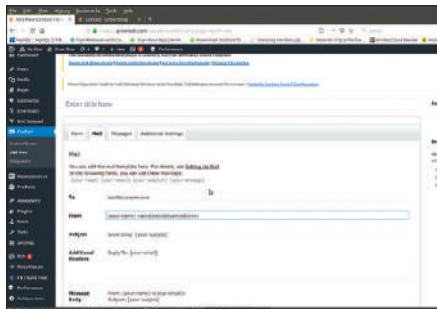
Сайты на HTTP вам практически любой браузер отметит как небезопасные. Пора уйти на HTTPS.

Web-формы без ботов 80

Минимизируем вредный трафик и ботов-автоматов, проникающих через пользовательские формы.

Головоломки под контролем 84

Пишем программу на Go, чтобы определить, правильно ли решена головоломка Судоку.



ПОСТОЯННЫЕ РУБРИКИ

Новости 6

Вышел новый Calculate, Ростех замахнулся на IoT и интернет-дронов, троянец для Linux удаляет антивирусы, AcroSser выдала сетевой сервер, Intel строит планы, твердотельный накопитель Samsung прибыл в Россию, а ОС АЛБТ — на Таволгу.

Вести мобильных ОС 24

5G проникает в дома, автономное вождение набирает обороты, а Samsung представила процессор для ИИ и переходит в миллиметровый спектр.

Сравнение 26

Нет бизнеса без инструментов офиса: *Calligra Suite*, Google Docs, *LibreOffice*, *OnlyOffice*, *WPS Office*.

Интервью LXF 42

Паскаль Геененс услышал, что по любому адресу в мире каждые две минуты стучится бот, и решил проверить это лично...

Рубрика сисадмина 50

Д-р Валентин Силицын занялся мониторингом систем и оосовременил работу в командной строке мультимплексором терминала.

Ответы 98

Учиняем смертоубийство задач, недоумеваем, куда подевались полосы прокрутки, и пытаемся приструнить свой компьютер, чтоб вел себя потише.

HotPicks 100

Отведаете горяченького! Лучшие в мире новинки свободного ПО: *Acid Cam*, *Audacity*, *Eolie*, *EtherApe*, *Gaupol*, *GrafX2*, *Haiku*, *Heimer*, *Treeline*, *uRacer*, *Zabuyaki*.

Диск Linux Format 106

Предыдущие номера 108

ЗАДАРОМ НА DVD
Elementary OS 5.0
Ubuntu 18.10
Manjaro 18.0
 ...и не только
 стр. 106

ЛINUX FORMAT **НОВЫЙ**
Elementary OS 5.0
 Красивейший и простейший среди дистрибутивов Linux! Пробуйте!
 • Уникальный рабочий стол Pantheon
 • Настраиваемые свободные программы
 • Простые и безопасные обновления
 • Без рекламы и шпионского ПО

Новинка! Ubuntu 18.10!
 Свежий релиз известнейшего в мире дистрибутива!
 64-битный

УЧЕБНИКИ

ТЕРМИНАЛ: Tomb 56

Вы можете легко укрыть все важные данные в зашифрованных, защищенных паролем каталогах.

LIBRE OFFICE: Правим PDFы 58

Вытворяем с PDFами такое, о чем и помыслить не могли, благодаря замечательному офисному пакету.

СЕРВЕР: Web-сервер-легковес 60

Крошечный специализированный сервер не будет перегружать ваше локальное оборудование.

FLATPAK: Интересное содержание ... 64

Flatpak предоставляет и приложение, и всё для него необходимое в независимом от дистрибутива формате.

GNU PLOT: Функциональные графики . 68

Если позарез нужно интерпретировать данные и ничто уже не помогает, обратитесь к нашему краткому курсу по *GnuPlot*.

SCRIBUS: Сам себе дизайнер 72

Не так страшен *Scribus*, как его малюют! Попривыкнув к нему, вы забудете об ужасах размещения картинок в текстовом редакторе.

УГЛУБЛЕННО...

Обследуем пещеры! 46

Облачаемся в каски и комбинезоны и спускаемся под землю, вооруженные лазерным измерителем и свободными программами.



Новости

В ЭТОМ НОМЕРЕ: Вышел Calculate » Чип IoT » Новый троянец-майнер » Обезопасим сеть » Наш интернет-дрон » Чиплеты от Intel » Сверхбыстрый SSD » Таволга + Альт

ДИСТРИБУТИВ МЕСЯЦА

«Gentoo — это свобода выбора...»

Вышел Calculate Linux Desktop 18 с рабочим столом LXQt.

Новый дистрибутив Calculate Linux Desktop с рабочим столом LXQt (CLDL) обратно совместим с Gentoo; для установки пакетов используется *Portage*, а собственный репозиторий насчитывает 13 033 двоичных пакетов. Менеджер загрузки — *OpenRC*, для настройки сети — *NetworkManager* или *OpenRC*, на выбор. Звуковая система по умолчанию — *ALSA*, без звукового сервера *PulseAudio*. В CLD сочетаются возможности библиотеки *Qt5*, на которой построен рабочий стол, с низкими системными требованиями оконного менеджера *Openbox*; из коробки доступна локализация для всех популярных европейских языков. В *Firefox* предусмотрен блокировщик рекламы и отключена телеметрия. Дистрибутив поддерживает единую систему авторизации

пользователей в *Calculate Directory Server*, перемещаемый профиль пользователя и общие с Windows клиентами права на сетевые диски.

Программное обеспечение включает рабочий стол *LXQt 0.13*, файловый менеджер *PCManFM-Qt 0.13*, браузер *Firefox 63.0.3*, офисный пакет *LibreOffice 6.0.6.2*, графический редактор *GIMP 2.10.4*, почтовый клиент *Claws Mail 3.17.1*, проигрыватели *SMPlayer 18.9.0* и *Clementine 1.3.1*, менеджер сообщений *Gajim 0.16.9*. Свежая версия утилит *Calculate 3.6.1.12* предоставляет установщик системы, утилиту обновления системы, утилиты для модификации ISO-образа. Дистрибутив доступен для архитектур i686 и x86_64. Для тестирования новой системы достаточно загрузиться с образа Live ISO.



Рабочий стол LXQt похож на упрощенный KDE, экономно потребляющий ресурсы компьютера.

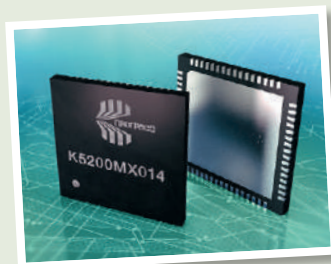
СДЕЛАНО В РОССИИ

Снижая импортозависимость

Ростех разработал чип для Интернета вещей.

Холдинг «Росэлектроника» Госкорпорации Ростех создал первую российскую микросхему для Интернета вещей (IoT). Чип размером 5×5 мм предназначен для приема и передачи информации в оптимальном для IoT диапазоне частот от 100 МГц до 2,5 ГГц. Это первая отечественная микросхема, обеспечивающая цифровую связь на расстоянии до 30 км. Устройство будет применяться в решениях IoT для промышленности, транспорта, связи, сельского хозяйства и в «умных» бытовых приборах. Основное преимущество — отсутствие незадекларированных функций, т. е. закладок, что является ключевым для стратегически важных предприятий и объектов инфраструктуры. На текущий момент создана и тестируется опытная партия изделий. «Электронная компонентная база — основной

элемент при построении систем Интернета вещей. Разработка отечественного чипа — необходимый шаг для снижения зависимости от импортных комплектующих. Это способствует масштабной цифровизации самых различных отраслей экономики и создает предпосылки для создания «умных» продуктов для массового рынка. Мы оцениваем сегодняшнюю потребность российского рынка не менее чем в 1 млн таких устройств в год», заявил индустриальный директор Ростеха Сергей Сахненко. Микросхема спроектирована ведущим дизайнерским центром микроэлектронных устройств в России НИИМА «Прогресс» холдинга «Росэлектроника» для серийного производства на предприятиях России, в соответствии с требованиями, предъявляемыми к отечественной микроэлектронной продукции первого уровня.



Главная особенность разработанной в «Ростехе» микросхемы — ее защищенность от внешнего вмешательства.

КИБЕРБЕЗОПАСНОСТЬ

«Люди, будьте бдительны!»

Новый троянец-майнер для Linux удаляет антивирусы.

Специалистами компании «Доктор Web» обнаружен майнер-кукушонок, способный инфицировать устройства под управлением ОС Linux. Троянец Linux.BtcMine.174 — большой скрипт, написанный на языке командной оболочки *sh* (более 1000 строк кода) и состоящий из нескольких компонентов — при запуске проверяет доступность сервера, с которого он впоследствии скачивает другие модули, и подыскивает на диске папку с правами на запись, куда эти модули будут затем загружены. После этого скрипт перемещается в ранее подобранную папку с именем **diskmanagerd** и повторно запускается в качестве демона. Установившись в системе, Linux.BtcMine.174 разыскивает конкурирующие майнеры, и, при обнаружении, завершает их процессы.

Если Linux.BtcMine.174 не был запущен от имени суперпользователя (*root*), для повышения своих привилегий в зараженной системе он использует набор эксплойтов; при этом загруженные из Интернета исходники DirtyCow троянец компилирует прямо на зараженной машине. После этого вредоносная программа пытается отыскать работающие сервисы антивирусных программ с именами *safedog*, *aegis*, *yunsuo*, *clamd*, *avast*, *avgd*,

cmdavd, *cmdmgd*, *drweb-configd*, *drweb-spider-kmod*, *esets* и *xmirrord*. В случае их обнаружения троянец не просто завершает процесс антивируса, но с помощью пакетных менеджеров удаляет все его файлы и директорию, в которой был установлен антивирусный продукт. Затем Linux.BtcMine.174 регистрирует себя в автозагрузке, после чего скачивает и запускает на инфицированном устройстве руткит; троянец собирает информацию о сетевых узлах, к которым ранее подключались по протоколу SSH, и пробует заразить их.

Выполнив все эти действия, Linux.BtcMine.174 запускает в системе майнер, предназначенный для добычи криптовалюты Monero (XMR). С интервалом в минуту троянец проверяет, запущен ли этот майнер, и при необходимости автоматически перезапускает его. Также он в непрерывном цикле соединяется с управляющим сервером и скачивает оттуда обновления, если таковые доступны.



Новый троянец Linux.BtcMine.174 является значительным прорывом — по стандартам вредоносных программ...

ОБОРУДОВАНИЕ

Сетевая платформа-микробокс

Acrosser представила новый компактный сервер сетевой безопасности.

Компания Acrosser (партнер Intel IoT Solutions Alliance) представила работающее под управлением GNU/Linux настольное сетевое устройство-сервер Micro Box AND-DNV3N2, предназначенное для защиты безопасности.

Как и решение с шестью портами FWA-1012VC от компании-конкурента Advantech, новинка комплектуется 4-ядерным «серверным» SoC Atom C3558 «Denverton» 2,2 ГГц, и поставляется либо с шестью «медными» портами Gigabit Ethernet, либо с четырьмя «медными» портами GbE и двумя оптоволоконными портами SFP (Advantech также предлагает модель с восемью портами и 8-ядерным чипом C3758), имеет по два порта mini-PCIe и USB 3.0, 2,5-дюймовый отсек SATA-III для подключения SSD, 8-битный GPIO, аппаратно реализованную схему контроля над зависанием системы (сторожевой таймер) и (опционально) TPM 2.0. Управление портами GbE осуществляется сетевыми контроллерами Intel I210 и Marvell 88E1543.

«Бюджетный» AND-DNV3N2 поддерживает типовые приложения, включая системы предотвращения сетевых вторжений, фильтрацию контента, предотвращение утечки данных, VPN, UTM-системы и SD-WAN, его размеры 230×105×44 мм. Два сокетa DDR4 позволяют получить минимум 8 ГБ ОЗУ (напомним, что большинство устройств Atom C3000 поддерживают до 48 ГБ ОЗУ, а недавно анонсированный Congatec модуль Conga-B7AC поддерживает 3 планки DDR4 по 32 ГБ — всего до 96 ГБ DDR4). Имеются также порт 2-pair LAN bypass и последовательный консольный порт WAN.

Наличие 2 слотов mini-PCIe и слота для SIM-карт позволяет использовать беспроводное расширение, обеспечивающее 4 антенных порта с поддержкой LTE и WiFi до 802.11ac. Поддерживаются Intel QuickAssist Technology (Intel QAT) и open source Data Plane Development Kit (DPDK). Устройство поставляется с адаптером питания DC 12 В 40 Вт и батареей Li-Ion 3V 220 мА·ч; диапазон его рабочих температур — от 0°C до 40°C.



Сервер сетевой безопасности Acrosser AND-DNV3N2 идет на смену популярному решению AND-G420N1.

ЗНАЙ НАШИХ!

Отечественная радиоэлектроника

Ростех разрабатывает интернет-дронов «Скайер».

Холдинг «Росэлектроника» Госкорпорации Ростех разрабатывает Интернет дронов (Internet of Drones, IoD) — высокоскоростную систему связи для беспилотников «Скайер». Комплекс представляет собой сеть распределенных наземных станций, которая обеспечивает обмен информацией и управление большими группами беспилотников, одновременно находящимися в воздухе. Начало производства «Скайера» запланировано на конец 2020 г. Беспилотники могут поддерживать связь со «Скайер» на расстоянии до 300 км от ближайшей наземной станции. Система связи является модульной, что позволяет создавать сети любого размера с неограниченным числом дронов, а также управлять ими из единого пункта управления. Комплекс может быть адаптирован под требования конкретного заказчика — количество наземных станций и схема их расположения зависят от поставленной задачи, числа беспилотников, типа и объемов передаваемых данных, площади и особенностей ландшафта. «Для создания Интернета дронов наши разработчики применили технологию, которая ранее хорошо зарекомендовала себя при производстве продукции военного назначения. Система связи

обеспечит высокоскоростной обмен информацией между группами беспилотников и центром управления, что повысит эффективность их эксплуатации на большой территории. «Скайер» может применяться для решения широкого круга задач. Это системы видеонаблюдения на территориально распределенных объектах, экологический надзор, контроль состояния объектов нефтегазового комплекса, энергетической и транспортной инфраструктуры», сказал исполнительный директор Госкорпорации Ростех Олег Евтушенко. В зависимости от полезной нагрузки дрона — видеокамера, тепловизор, датчик поля — «Скайер» может собирать информацию различного рода, от аэрофотосъемки до измерения магнитного поля. На сегодняшний день создан прототип наземной станции. Разработчик решения — АО «НПП «Полет» (входит в холдинг «Росэлектроника») — произведет опытные образцы в 2019 г.



Главным преимуществом программно-аппаратного комплекса «Скайер» является большая зона покрытия наземных станций.

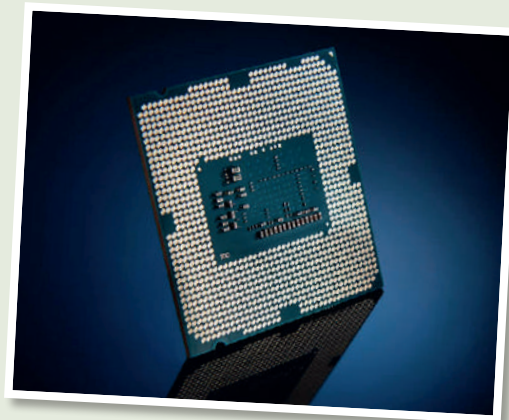
НОВОСТИ INTEL

«Что год грядущий мне готовит?»

Intel провела традиционный Architecture Day.

Процессоры Core и Xeон следующего поколения будут выполнены на представленной 11 декабря в рамках «Architecture Day 2018» новой архитектуре CPU, получившей название Sunny Cove. Кроме того, Intel обещает в конце 2019 г. вывести на рынок свои 10-нанометровые чипы. Сведений о возможностях Sunny Cove и тактовых частотах новых чипов нет пока никаких, однако Intel обещает, что Sunny Cove позволит параллельно выполнять больше операций и что будут сокращены задержки при выполнении команд. Кроме того, Intel говорит об «архитектурных расширениях для конкретных сценариев использования и алгоритмов. Например, новые инструкции по повышению производительности для криптографии, такие как вектор инициализации AES и SHA-NI, и другие критически важные сценарии, в том числе сжатие и декомпрессия». Чипмейкер также прорекламировал интегрированную графику Gen11, которая «предназначена для преодоления барьера 1 TFLOPS», и представил свою 3D-технологию «Foveros» для размещения логических чипов в несколько слоев,

охарактеризовав ее как «радикальную реорганизацию однокристалльных систем». В этой объемной архитектуре физический чип состоит из собранных в стек логических «чиплетов», что позволяет создавать процессоры для определенных задач. «Транзистор, хорошо подходящий для настольного игрового CPU, не обязательно будет лучшим транзистором для GPU. Для работы 5G и для межсетевого взаимодействия тоже нужны разные транзисторы, — пояснил руководитель архитектурных проектов Intel Раджа Кодури [Raja Koduri]. — Раньше мы выбирали компромисс из имеющихся процессоров. Теперь мы можем взять наиболее подходящие функциональные процессы и объединить их в один пакет. А благодаря высокой пропускной способности между «чиплетами» они будут работать, как если бы были одним чипом».



Количество вычислительных блоков в iGPU Gen11 увеличится до 64 (в нынешней Gen9 их 24).



Unix Education Center

Москва

+7 (495) 369-46-38

Санкт-Петербург

+7 (812) 611-15-75



UNIX EDUCATION CENTER

ПРЕВРАЩАЕМ ИНЖЕНЕРОВ В ЭКСПЕРТОВ!

КУРСЫ ДЛЯ ИТ-СПЕЦИАЛИСТОВ

ТЕХНОЛОГИЧЕСКИЕ НАПРАВЛЕНИЯ:

Серверные решения
Red Hat
Хранение данных
Java
Безопасность
Защита данных
ПО для нефтегазовой отрасли
(Aspen HYSYS, Aspen PIMS)

НАПРАВЛЕНИЕ SOFT SKILLS:

Техническая презентация
Мастерство презентаций
Мастерство продаж в ИТ
Искусство переговоров
Управление ИТ-проектами

ПОЛНЫЙ СПИСОК КУРСОВ — WWW.UNIXEDU.RU

ХРАНИЕНИЕ ДАННЫХ

Результат синергии технологий

Сверхбыстрый накопитель Samsung SSD X5 поступил в продажу в России.

В России начались продажи портативного твердотельного накопителя Samsung Portable SSD X5, созданного на базе спецификации NVMe и использующего современный интерфейс Thunderbolt 3 (40 Гбит/с). Накопитель получил металлический ударопрочный глянцевый корпус с нескользящим основанием, который выдерживает падение с высоты до 2 метров, а динамический алгоритм защиты от высокой температуры защищает X5 от перегрева и позволяет поддерживать оптимальную температуру для надежной работы. Устройство обладает мощной защитой данных благодаря ПО «Samsung Portable SSD Software», основанному на 256-битном алгоритме шифрования AES, включая возможность защиты паролем и легкую настройку параметров безопасности. Прочие технические характеристики: размер 119×62×19,7 мм; скорость передачи до 2800 МБ/с, а записи до 2300 МБ/с (500 ГБ — до 2100 МБ/с); сертификации CE, BSMI, KC, VCCI, C-tick, FCC, IC, UL, TUV, CB; соответствие RoHS2; гарантия на 3 года.



Благодаря интерфейсу Thunderbolt 3 и спецификации NVMe, SSD X5 обеспечивает впечатляющие скорости передачи данных.

С 19 ноября устройство можно приобрести в фирменных магазинах и в точках продаж партнеров Samsung по рекомендованной розничной цене 28990 руб. (версия с емкостью 500 ГБ), 49990 руб. (версия с емкостью 1 ТБ), 99990 руб. (версия с емкостью 2 ТБ).

ИМПОРТОЗАМЕЩЕНИЕ

«Рождая собственных Платонов...»

Компьютеры «Таволга Терминал» будут поставляться с предустановленной ОС АЛЪТ.

Компании «Базальт СПО» и «Т-Платформы» подписали соглашение о поставке компьютеров «Таволга Терминал» с предустановленной ОС «Альт Рабочая станция», потребители которых смогут быстро развернуть готовое рабочее место, оснащенное ОС и большим набором офисного ПО. Вычислительная техника серии «Таволга Терминал» производится компанией «Т-Платформы» на базе российского процессора «Байкал-Т» с архитектурой MIPS (разработка «Байкал Электроникс»). В зависимости от конфигурации, компьютеры могут использоваться либо как энергоэффективная рабочая станция, либо как «тонкий клиент» в терминальном режиме. Устройства предназначены, в первую очередь, для корпоративных заказчиков, которым необходимы отечественные решения с высоким

уровнем защиты. Для оснащения устройств «Таволга Терминал» компания «Базальт СПО» создала специальный дистрибутив ОС АЛЪТ, который обеспечивает высокую производительность и надежность ОС. Дистрибутив включает ОС и набор офисного ПО для полноценной работы: редакторы текста, таблиц, презентаций, графики и видео, средства для сканирования документов, web-браузер, почтовое приложение и др. Дистрибутивы АЛЪТ создаются на базе отечественного репозитория «Сизиф», обеспечивающего поддержку практически неограниченного жизненного цикла ОС. Помимо хранилища программных пакетов, репозиторий содержит оригинальную отечественную инфраструктуру разработки, позволяющую одновременно собирать дистрибутивы под разные аппаратные платформы. **LXF**

Короткой строкой

- » Canonical увеличила срок поддержки дистрибутива Ubuntu 18.04 LTS до 10 лет.
Источник: www.serverwatch.com
- » Samsung открыла лабораторию проекта «IoT Академия Samsung» в Челябинске на базе Южно-Уральского государственного университета.
Источник: <https://news.samsung.com>
- » Версии игры *Total War: Warhammer II* выпущены для Linux и MacOS.
Источник: www.gamingonlinux.com
- » Эксперты выявили не менее семи новых вариантов атак класса Meltdown/Spectre на процессоры Intel, AMD и ARM.
Источник: <http://safe.cnews.ru>
- » Пакет *Microsoft Office* ведет сбор данных пользователей со всего мира без возможности отключения этой функции.
Источник: <http://safe.cnews.ru>
- » Корректирующий релиз офисного пакета Apache OpenOffice 4.1.6 ввел 13 исправлений.
Источник: www.mail-archive.com
- » Свою первую конференцию команда Devuan проведет в Амстердаме 5–7 апреля 2019 г.
Источник: <https://lists.dyne.org>
- » Microsoft планирует заменить Edge в Windows 10 браузером на свободном движке Chromium.
Источник: www.windowscentral.com
- » Последнее кумулятивное обновление Windows 10 вызывает на флагманских устройствах Microsoft Surface Book 2 «синий экран смерти».
Источник: <http://safe.cnews.ru>
- » Основан проект libcamera, создающий библиотеки для унификации поддержки и упрощения работы с видекамерами, фотокамерами и TV-тюнерами в Linux, Android и ChromeOS.
Источник: <http://libcamera.org>
- » Компания SUSE представила релиз промышленного дистрибутива SUSE Linux Enterprise 12 SP4.
Источник: www.suse.com

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!

ГНУ/Линуксцентр

*Ваш поставщик свободного программного
и аппаратного обеспечения*

**Комплекты
легализации СПО**

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

**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



Обзоры



Алексей Федорчук
Тэг <сарказм>
по умолчанию,
смайлики по вкусу.

» СВЯТАЯ ПРОСТОТА, ИЛИ БРАУЗЕР SURF

Как известно, браузеры, подобно попкам из рекламы фирмы Huggis, становятся всё здоровей. Постепенно превращаясь в здоровенные... сами знаете что. Браузер *Surf* ломает эту традицию, и потому заслуживает отдельной колонки.

Разработчики определяют его специфику «от противного»: чего в нем нет. А нет в нем поддержки закладок, сохранения истории посещений и плагинов. В нем не предусмотрена даже пиктограмма для запуска: он стартует из командной строки терминала (или из мини-терминала по Alt+F2) в форме *surf* <имя домена>. После чего возникает окно с указанной страницей и без всяких элементов управления.

Для навигации служат комбинации клавиш в стиле Vi: Ctrl+j и Ctrl+k — вверх и вниз по текущей странице, Ctrl+h и Ctrl+l — назад и вперед в текущей истории, и так далее (см. *man surf*, там все написано). А если запускать *Surf* с опцией -N, то доступно еще и контекстное меню с инспектором кода.

Иными словами, в *Surf*е ничто не отвлекает от вдумчивого чтения материалов, одного заслуживающих.

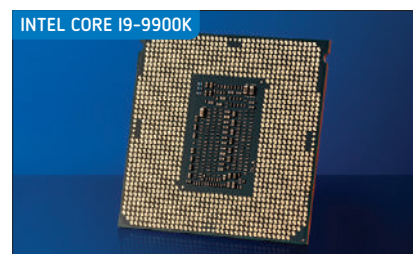
alv@posix.ru

Сегодня мы рассматриваем:

- | | | |
|--|----|---|
| Ubuntu 18.10 | 14 | на любое устройство в вашей сети. Да еще создавать резервные копии. |
| Дистрибутив продолжает продвигаться к интеграции Snap-пакетов с рабочим столом и полирует Gnome Shell. Впрочем, пользователям 18.04 хвататься за новинку не стоит. | | |
| Manjaro 18.0 | 16 | |
| Самообновляемый дистрибутив на базе Arch, в отличие от своего родителя, беспокоится об удобстве пользователей. | | |
| Fedora 29 | 18 | |
| Графический сервер Wayland включен по умолчанию, обновлена версия ядра, Gnome повысил производительность — итоги неплохие. | | |
| Synology DiskStation DS418 Play ... | 19 | |
| NAS позволяет хранить огромное количество видео и фотографий и передавать их все | | |
| Intel Core i9-9900K | 20 | |
| Новый, самый быстрый процессор для настольных ПК обеспечивает невероятную однопоточную производительность. | | |
| Surviving Mars | 21 | |
| Воевать вам в этой игре не придется: выживание на Марсе — и без того задача непростая. Битва идет между людьми и природой, но у вас есть союзник — технологии. | | |
| Megaquarium | 22 | |
| Еще одна мирная игра — организация огромного аквариума. Необходимо оборудовать и резервуары для рыбок, и буфеты-туалеты для посетителей... | | |

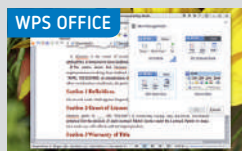
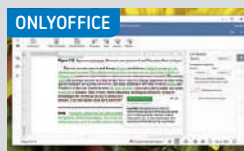
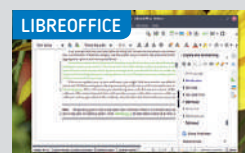


В Manjaro вы получите массу приложений прямо из коробки.



Процессор 9900K задает новый стандарт для основных чипов.

» СРАВНЕНИЕ: ОФИСНЫЕ КОМПЛЕКТЫ СТР. 26



Linux позаботился об офисном планконе, упаковав текстовые редакторы, электронные таблицы и приложения для презентаций в удобные комплекты.

Ubuntu 18.10

Откинувшись на спинку стула во время тестирования последней итерации, **Маянк Шарма** размышляет: не пора ли Ubuntu выпускать одни LTS-релизы?

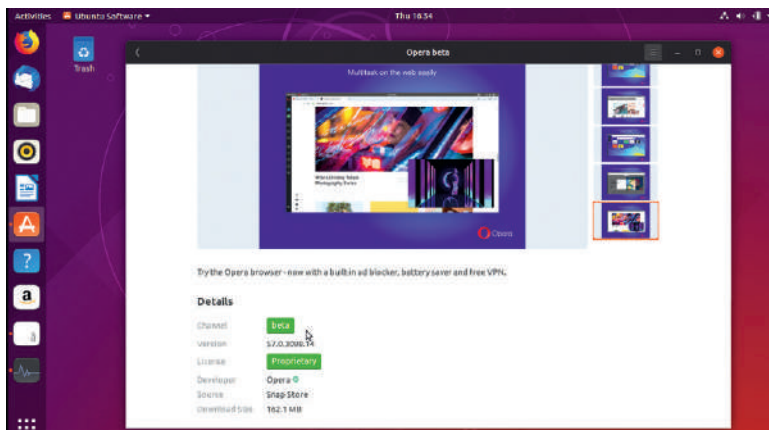
ВКРАТЦЕ

Один из самых популярных дистрибутивов Linux, благодаря которому Linux прорвался в мейн-стрим. Проект недавно прикрыл изрядную долю своих начинаний, касающихся рабочего стола. Наиболее заметно, что после нескольких лет отпугивания пользователей своим рабочим столом Unity, Ubuntu переключился обратно на интерфейс рабочего стола Gnome, который продолжает полировать, чтобы отвечать его требованиям.

Вам не придется пристально вглядываться, чтоб обнаружить шаблон для релизов Ubuntu. После нескольких релизов, несущих бунтарские изменения, дистрибутив в течение нескольких циклов выпуска выдает инкрементные обновления, а потом снова перемешивает котелок. Сейчас Ubuntu вступает в этот тряский период — со своим первым LTS-релизом без рабочего стола Unity и с заменой широко разрекламированной графической системы Wayland на X.org. В данном релизе Ubuntu 18.10, который окрестили Космической Каракатицей [Cosmic Cuttlefish], кривые начинают снова приходить в норму, по мере того, как дистрибутив входит в ритм.

Наиболее заметное изменение в Ubuntu 18.10 — новый слой краски. С Gnome, вернувшимся на рабочий стол Ubuntu, дистрибутив был непрост в работе, приходя в соответствие с его нуждами и требованиями. Темам Ambiance и Radiance, наконец, пришла на смену новая, нареченная Yaru, которая выглядит более современно и плоско. Тема хорошо дополняется набором иконок Suru.

Однако подавляющее большинство достойных улучшений в этом выпуске скрыто от глаз. Релиз работает на ядре Linux 4.18, главные функции которого включают полную поддержку Raspberry Pi моделей 3 В и 3 В+, наряду с усовершенствованной поддержкой устройств Thunderbolt и USB-C. Другими примечательными особенностями этого ядра являются его многочисленные улучшения для геймеров. В него добавлена поддержка графических процессоров AMD Radeon RX Vega M, а также



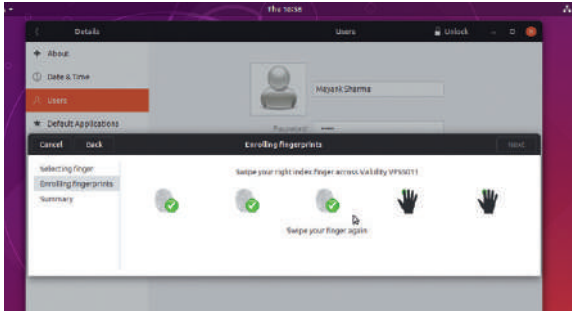
Теперь вы можете использовать программное обеспечение Ubuntu для установки приложений Snap из разных каналов, таких как Stable и Beta.

реверсивный драйвер контроллера Steam, который позволяет контроллеру Steam работать как обычное устройство ввода HID из коробки. Ubuntu 18.10 также обновляет сервер X.org до значительно улучшенной v1.20.1. Еще одним закулисным изменением является использование сжатия LZ4 вместо более старого сжатия gzip для ram-диска initramfs. Это изменение даст о себе знать в виде более быстрой загрузки. Разработчики утверждают, что сокращение составляет до 10%, хотя у нас нет никаких точных показателей.

Рабочий стол Ubuntu 18.10 также выигрывает от усовершенствований оболочки Gnome Shell. Разработчики рабочего стола полировали его, улучшая производительность различных компонентов. Ubuntu 18.10 включает Gnome 3.30, который потребляет меньше ресурсов системы, чем когда-либо прежде,



Рабочий стол функционирует плавнее благодаря улучшению производительности Gnome.



Разработчики Ubuntu считают, что отпечатки пальцев эквивалентны именам пользователей, и потому должны использоваться для разблокировки сеанса, в который уже выполнен вход.

значительно улучшая опыт взаимодействия. Дистрибутив также выигрывает от улучшений де-факто приложений GNOME. Наиболее важным из них является утилита *Disks*, которая теперь может расшифровывать и монтировать тома хранения, зашифрованные посредством *VeraCrypt*, благодаря работе дистрибутива *Tails*.

Thunderbolt пошел!

Есть также некоторые улучшения, связанные с периферийными устройствами, которые очевидны в приложении *Settings* [Настройки] версии 18.10. Оно обзавелось новым специальным разделом для управления устройствами Thunderbolt 3. Кроме того, в итоге попытки разгрузить интерфейс в *Settings* появились динамические панели, отображающиеся только тогда, когда они могут найти совместимое оборудование. Вы также можете использовать для аутентификации на вашем ноутбуке сканер отпечатков пальцев, без дополнительных усилий или привлечения библиотек.

Одним из основных компонентов GNOME, оставленным без обновления, является файловый менеджер *Nautilus*. Ubuntu 18.10 продолжает работу со старым v3.26.4 поскольку это была последняя версия файлового менеджера с поддержкой значков на рабочем столе. Работа над расширением для предоставления значков рабочего стола уже ведется, и мы подозреваем, что оно скоро появится на рабочем столе Ubuntu, ведь нельзя же вечно игнорировать удобство использования и визуальные улучшения, вводимые в более новых выпусках файлового менеджера.

Еще одна вещь, не попавшая в последний релиз — это интеграция с Android при помощи GSConnect, расширения GNOME Shell. Оно по-прежнему доступно в репозитории Universe, и хотя разработчики Ubuntu пока не сочли его готовым для включения, оно остается одним из лучших приложений, которые вы должны установить поверх первоначальной установки 18.10. Точно так же разработчики Ubuntu хотели добавить возможность потоковой передачи через медиа-сервер DLNA на рабочий стол, но не смогли из-за сжатых сроков.

По части программного обеспечения, дистрибутив продолжает продвигаться к интеграции пакетов Snap с рабочим столом. Интерфейс *xdg-desktop-portal* теперь также поддерживает Snaps, что рекламируется как улучшение удобства использования инструментов песочницы на рабочем столе. Версия 18.10 также поставляется с тремя программами Snap по умолчанию (*Calculator*, *System Monitor* и *Characters*), и в официальных репозиториях есть еще более 4000 программ.

Тем не менее, несмотря на все эти изменения, Ubuntu 18.10 — это не вариант автоматического обновления для всех. Поскольку, если вы уже пользуетесь Ubuntu 18.04 LTS, вам обеспечена официальная поддержка до начала 2023 г. Напротив, Ubuntu 18.10 будет получать обновления для системы безопасности, исправления ошибок и другие обновления только в течение девяти месяцев, что заставит вас обновить ее до нового

» И ДРУГИЕ НОВОСТИ ОТ UBUNTU...

Релиз Ubuntu — это больше, чем просто один дистрибутив. Наряду с официальными разновидностями, новый релиз Ubuntu подпитывает ряд производных дистрибутивов, многие из которых выпускают обновления в кратчайшие сроки. Возможно, одним из самых больших изменений в этой экосистеме является отказ от 32-разрядного ISO. Хотя многие версии Ubuntu все еще предоставляют для скачивания 32-разрядные ISO-образы для 18.04 LTS релиза, для текущей версии 18.10 некоторые из них ограничились лишь 64-разрядными. К ним относятся Kubuntu, Ubuntu Mate, Ubuntu Budgie и Ubuntu Studio. Конечно, отсутствие 32-разрядного установочного носителя не мешает существующим 32-разрядным установкам обновиться до более новой версии через репозитории. Помимо настольных релизов, другая крупная разработка Ubuntu, Ubuntu Server, также получает несколько обновлений. В своем анонсе к релизу Canonical указала на то, что она сосредоточена на рынке финансовых услуг: «В этом году индустрия финансовых услуг интенсивно взаимодействовала с Canonical и Ubuntu для повышения эффективности инфраструктуры своих ЦОД и ускорения их перехода к облаку. Рынок в сторону аналитики, машинного обучения и технологий финтех вокруг блокчейна, приложений для распределенных реестров и криптовалют являются движителями инвестиций и внедрений Ubuntu на данный момент». По сути, это означает, что Ubuntu Server теперь может также обслуживать финансовые учреждения вместе с другими корпоративными клиентами для включения частных облачных хранилищ в их инфраструктуру. Образы Ubuntu Server 18.10 доступны во всех основных общедоступных облаках. Для размещения частных облаков 18.10 включает последнюю версию OpenStack релиза Rocky, которая предназначена для работы с зарождающимися сценариями использования, такими как пограничные вычисления, виртуализация сетевых функций (NFV) и машинное обучение. Существует также LXD 3.0 — первый релиз, способный объединить идентично настроенные серверы LXD в кластер, и Kubernetes 1.12 для развертывания контейнерных приложений.

релиза где-то в середине 2019 г. Кроме того, многие улучшения рабочего стола и обновления программ в 18.10, такие как новая тема, уже появились в 18.04, а другие, такие как поддержка *xdg-portals*, в конечном итоге просочатся после более тщательного тестирования.

Всё это, наряду с более коротким сроком поддержки, означает, что, если вам действительно не нужны улучшенные обновления поддержки и производительности оборудования, нынешние пользователи 18.04 могут продолжать придерживаться своих LTS-установок и ждать, когда обновления попадут в репозитории 18.04. Если, с другой стороны, вы пока не пользователь Ubuntu и желаете прощупать обстановку, Ubuntu 18.10 не хуже любого другого выпуска Ubuntu. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Canonical

САЙТ: www.ubuntu.com

ЛИЦЕНЗИЯ: GPL и др.

ФУНКЦИОНАЛЬНОСТЬ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО В РАБОТЕ	8/10
ДОКУМЕНТАЦИЯ	8/10

Заурядный не-LTS релиз Ubuntu с кучей обновлений, которые со временем попадут в релиз LTS.

» Рейтинг 7/10

Manjaro Linux 18.0

Маянк Шарма тестирует удобный дистрибутив на базе Arch, игнорируя насмешки некоего адвоката, обзывающего его псевдопользователем Arch.

ВКРАТЦЕ

Основанный на Arch Linux, Manjaro является одним из самых популярных производных, ориентированных на удобство пользователя. Он поддерживает широкий спектр рабочих столов в официальных и поддерживаемых сообществом версиях. Это самообновляемый дистрибутив, который также изредка выпускает установочные носители (только для 64-битной архитектуры), как рассматриваемый.

Мanjaro уже довольно давно занимает первое место на странице рейтинга популярности distrowatch.com. Возможно, это не точное отражение популярности проекта, но, по крайней мере, свидетельство постоянного интереса к дистрибутиву.

Manjaro — антипод своего прародителя. Он спроектирован ради удобства пользователя и нацелен на обеспечение полной работоспособности прямо из коробки — обе эти характеристики у Arch противоположны. Если Arch известен как дистрибутив для самостоятельной сборки, позволяющий опытным пользователям создавать свои установки с нуля, то Manjaro поставляется с удобным графическим установщиком, который позволит всем — независимо от уровня их квалификации — попробовать необычность Arch в роли повседневной настольной системы.

Manjaro доступен в трех официально поддерживаемых версиях, а редакция с Xfce является флагманским предложением. Последний выпуск украшен новой темой под названием Adapta-Maia, а также некоторыми другими мелкими визуальными изменениями в системе для улучшения взаимодействия с пользователем. Во всех выпусках используется модифицированный установщик *Calamares*, который, помимо косметических отличий, также позволяет шифровать раздел, где планируется установить дистрибутив.

Опытные пользователи по достоинству оценят *Manjaro-Architect*, установщик командной строки, предназначенный помочь пользователям установить полностью настраиваемую сборку со своим выбором компонентов, включая ядро, систему инициализации, среду рабочего стола и пр., и получить свежие

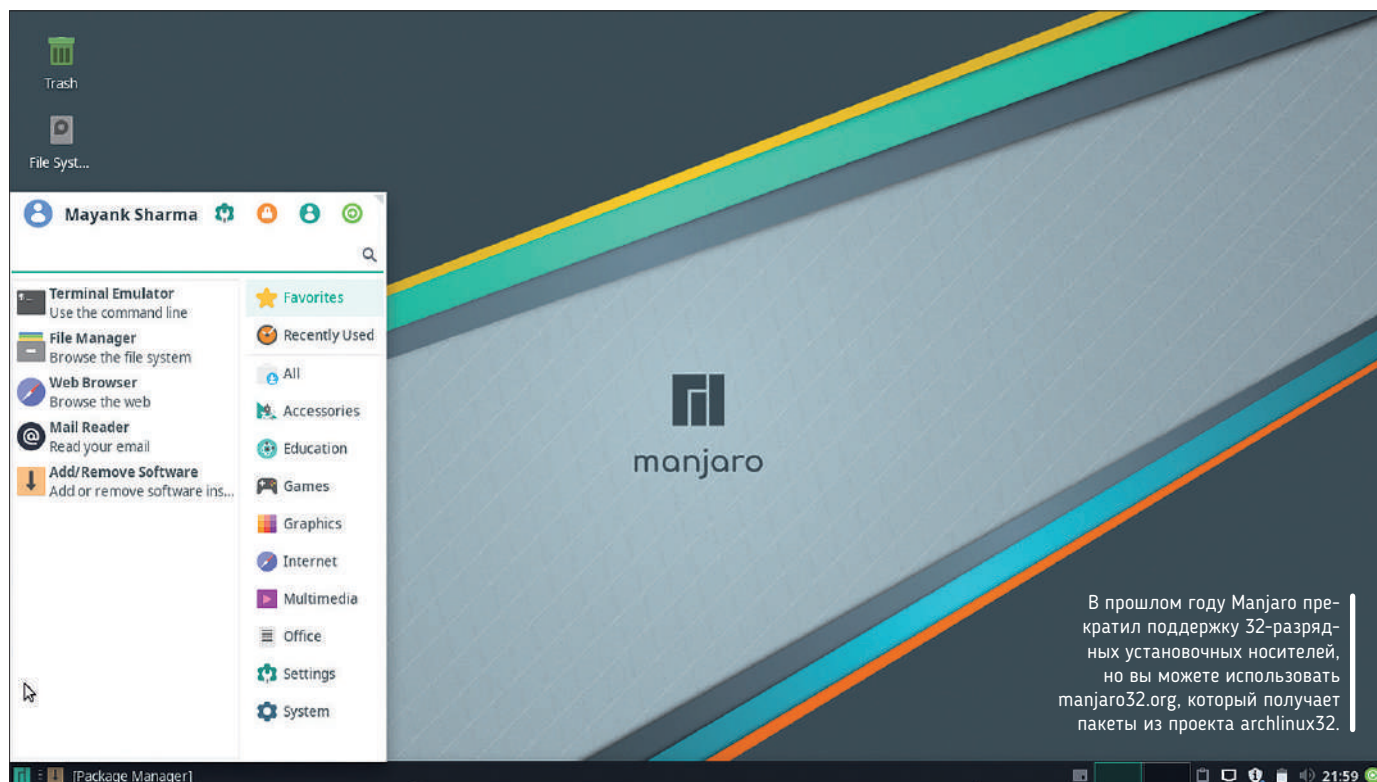
версии всех пакетов из репозитория. Помимо трех официальных редакций, проект также привлекает множество поддерживаемых сообществом выпусков с разными рабочими столами, включая Budge, Cinnamon, Deepin, Mate, LXDE и др.

Меню загрузки среды Live дает возможность загрузить дистрибутив, используя для вашей видеокарты драйверы с открытым исходным кодом либо проприетарные. Позже вы можете использовать панель *Hardware Detection* в *Manjaro Settings Manager* для установки и переключения между драйверами графического оборудования с открытым исходным кодом и проприетарными.

Версия Xfce Manjaro 18 включает последнюю версию рабочего стола 4.13. Облегченный рабочий стол следует обычной схеме рабочего стола и подойдет большому количеству пользователей. Тем не менее, несмотря на улучшения, удобство рабочего стола всё еще может выиграть от настройки. Многие пункты меню Whisker, а также и на рабочем столе имеют неполные имена, которые царапают глаз.

Проблемы видимости

Некоторые приложения Xfce не были правильно настроены, особенно терминал. Он прозрачный, и это уместно только в случае единственного приложения, открытого на рабочем столе: прозрачность делает текст в терминале совершенно неразборчивым, особенно если у вас есть другие окна программ, открытые в фоновом режиме. Неспособность Xfce решить давние проблемы также не способствует удобству использования Manjaro. Одной из таких проблем в текстовом редакторе *mousepad* Xfce является возможность случайной перезаписи файла, поскольку



он открывает разные файлы в разных окнах (не во вкладках) и не предупреждает пользователей при сохранении изменений в файле, открытом в другом окне.

Но не всё так мрачно на фронте удобства использования. Одной из новых функций этого выпуска являются Display-Profiles, которые позволяют пользователям сохранять различные настройки для всех подключенных мониторов. Кроме того, вы можете попросить Manjaro 18 автоматически переключиться на сохраненный профиль монитора, как только он подключится. Вроде не так уж и круто, но избавляет от многих проблем настройки для пользователей с несколькими мониторами, которые регулярно подключают различные внешние дисплеи, как, например, выездные работники, таскающие свой ноутбук между мониторами дома и в офисе.

После загрузки дистрибутив приветствует пользователей утилитой *Hello*, которая предоставляет прямые ссылки на документацию проекта и ресурсы поддержки. Новым в этом выпуске является добавление кнопки Applications, отображающей обширный список популярных программ в различных категориях, которые можно использовать для уточнения вашей установки. Тем не менее, если у вас нет особых требований, вы можете спокойно пропустить установку любых дополнительных приложений, поскольку Manjaro 18 поставляется с полным набором программ и может использоваться прямо из коробки. Вы найдете все обычные популярные приложения, такие как пакет *LibreOffice*, *GIMP*, *VLC*, *Firefox*, *Thunderbird*, клиент *Steam* и многое другое.

Одним из интересных инструментов, который привлек наше внимание, был *Microsoft Office*. В меню *Office* объединены ссылки на *Microsoft Office Online*, который работает как собственная программа, помимо других приложений Linux, благодаря оболочке JAK. Хотя поддержка *LibreOffice* закрытого офисного формата росла как на дрожжах, доступ к самому проприетарному инструменту сделает Manjaro более привлекательным. Основные программы хорошо дополняются удобными утилитами из стабильной Xfce.

Пакеты на любой вкус

Если вам необходимо выполнить установку из репозитория, можете использовать менеджер пакетов Manjaro *Pamac*, который также выполняет функции менеджера обновлений. Последняя версия менеджера пакетов *Pamac v7.2* включает новый инструмент командной строки с именем *pamac-cli*, который похож на *apt-get* в Debian. По какой-то причине KDE не использует *Pamac*. Он использует *Octopi*, популярный интерфейс для *pacman*. И — да, вы всё еще можете использовать *pacman* в командной строке во всех редакциях Manjaro.

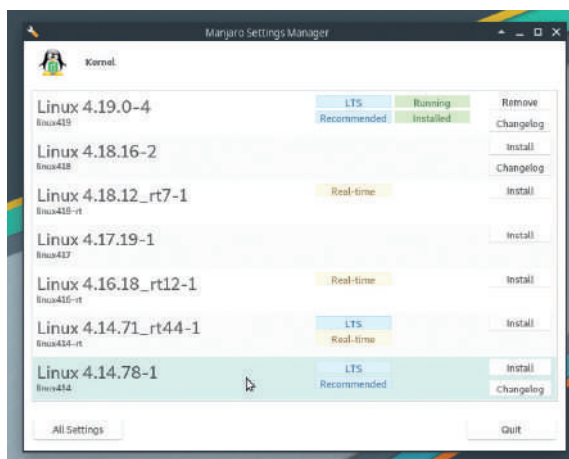
Кроме того, хотя Manjaro основан на Arch, в дистрибутиве есть собственные репозитории приложений, которые поставляют стабильные программное обеспечение. Дистрибутив раскладывает программы в три набора репозиториях. Есть стабильные репозитории тщательно протестированного программного обеспечения, а также тестовые репозитории, содержащие пакеты, синхронизируемые с нестабильными репозиториями каждую неделю. Наконец, в нестабильных репозиториях размещаются новейшие пакеты Arch, которые не прошли тестирования у составителей пакетов Manjaro. Кроме того, можно также включить пользовательский репозиторий AUR Arch для доступа ко многим новейшим инструментам.

Еще одним важным и уникальным предложением Manjaro являются пользовательские инструменты дистрибутива, встроенные в Manjaro *Settings Manager*. Одним из лучших инструментов в этом комплекте является переключатель ядра. Manjaro 18 позволяет пользователям выбирать из восьми различных ядер — от последней v4.19 вплоть до v3.16, включая три ядра реального времени. Широкий спектр поддержки ядер гарантирует, что

» MANJAROWARE

Проект Manjaro время от времени выпускает ограниченную серию оборудования для своего пылкого сообщества пользователей. Последним из этих сертифицированных Manjaro устройств является Bladebook Fall 2018. Эти устройства оснащены 13,3-дюймовым дисплеем FHD IPS и процессором Intel Celeron N3450 с Intel HD Graphics 500. В качестве основного хранилища они используют диск eMMC емкостью 64 ГБ и поставляются с 6 ГБ оперативной памяти DDR3L и аккумулятором емкостью 8000 мА·ч, который может обеспечить работу портативного компьютера до восьми часов. Manjaro выставил 100 штук Bladebook в своем интернет-магазине, и вы можете получить его за 259,05 швейцарских франков, что составляет всего £200.

Удивительно, но вместо знакового Xfce, Bladebook работает под управлением KDE Manjaro. Эта версия KDE поставляется с Plasma v5.14 и набором программ KDE v18.08. Он получает новый инструмент Display Configuration, который будет полезен тем, кто регулярно использует свои ноутбуки для подключения к проекторам для проведения презентаций. Хотя рабочий стол KDE выглядит немного лучше, чем Xfce, использование прозрачного фона в панели запуска приложений Kickoff выглядит неуместно.



Проект имеет обширные ресурсы помощи и поддержки. Их wiki довольно подробны и хорошо классифицированы, а в дистрибутив входит подробное руководство пользователя в формате PDF.

Manjaro 18 может работать как на новейшем, так и на устаревшем оборудовании. Преимущество дистрибутива в том, что он устраняет сложность двух наиболее распространенных, но громоздких процессов администрирования Linux: компиляция и установка ядер и переключение между свободными и проприетарными графическими драйверами. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Manjaro Team

САЙТ: www.manjaro.org

ЛИЦЕНЗИЯ: GPL и др.

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО В РАБОТЕ	8/10
ДОКУМЕНТАЦИЯ	9/10

Благодаря собственным инструментам и имеющимся приложениям, настольная система достойна высокой оценки и за форму, и за содержание.

» Рейтинг 8/10

Fedora 29

Маянк Шарма ждал фейерверка, когда на Gnome перешел Ubuntu. Тем не менее, Fedora бесспорно остался ведущим дистрибутивом с Gnome...

ВКРАТЦЕ

Один из знаковых настольных дистрибутивов на RPM, который продолжает предлагать лучшие наработки для рабочего стола Gnome. Многие из его штатных разработчиков работают в Red Hat, которая использует проект Fedora для внедрения и полировки практически всех компонентов, в конечном итоге попадающих в корпоративные предложения.

Fedora 29 является первым из крупных дистрибутивов с Gnome, выпускаемых в этом месяце. Тем не менее, оба они (второй — Ubuntu) имеют мало общего, за исключением среды рабочего стола. Fedora — это испытательный полигон для Red Hat Enterprise Linux (RHEL). Возможно, именно поэтому он использует более современный подход, чем другие настольные монстры наподобие Ubuntu. Например, хотя Beta-версия Fedora 29 работает на ядре Linux 4.18, окончательный релиз будет базироваться на v4.19, проходившем тестирование на момент написания статьи.

Между ними имеются и другие различия. В отличие от Ubuntu 18.10, Fedora Workstation 29 по умолчанию использует графический сервер Wayland. Эта версия также использует Gnome 3.30 с повышенной производительностью и обладает всеми усовершенствованиями данного релиза, упомянутыми в обзоре Ubuntu 18.10. Улучшения производительности очевидны при выполнении задач с интенсивной нагрузкой во время работы с дистрибутивом на виртуальной машине.

Пока не меркнет свет Gnome

Однако, в отличие от Ubuntu 18.10, Fedora 29 стремится поддерживать наработки Gnome по умолчанию. Самым заметным различием между двумя дистрибутивами с Gnome 3.30 является отсутствие поддержки значков рабочего стола в Fedora 29. Дело в том, что значки рабочего стола никогда не входили в цели дизайна Gnome 3, и, начиная с v3.28, разработчики решили убрать 6 лет не использовавшийся код из файлового менеджера *Nautilus*.

Как упоминалось в обзоре Ubuntu 18.10, Ubuntu обходит проблему значков на рабочем столе, включив более старую версию *Nautilus*. С другой стороны, Ubuntu теряет все усовершенствования *Nautilus 3.30*, такие как новая навигационная панель «хлебные крошки [breadcrumbs]», интеграция панели поиска, динамическое изменение размера и другие. Расширение Gnome Shell для отображения значков рабочего стола, которое находится в стадии разработки, обеспечит паритет между обоими дистрибутивами в этом аспекте.

Еще одно различие между двумя дистрибутивами заключается в предпочтительном выборе платформ виртуализации приложений. Для Ubuntu это Snaps, для Fedora — Flatpaks (см. стр. 64). Многие программы Gnome, такие как *Games* [Игры], *Recipes* [Рецепты] и новое приложение *Podcasts* [Подкасты], даже и не доступны в официальных репозиториях Fedora, и вместо этого предполагается их установка в качестве Flatpaks. Вы можете установить Flatpaks из приложения *Software*, включив поддержку Flathub. Кстати, *Software* в Fedora 29 теперь позволяет автоматически обновлять и приложения Flatpak.

Интересной новинкой Fedora 29 является наличие модульных репозиториях для разновидностей Fedora. Концепция модульности была введена в предыдущем цикле релизов, но только для редакций Fedora 28 Server. Модульность помогает запускать несколько



Fedora 29 пропустила свою первоначальную дату релиза из-за нескольких обнаруженных ошибок, но должна выйти к тому времени, как этот номер попадет на прилавки.

версий одного и того же программного обеспечения в изолированных контейнерах. Эта схема позволит использовать в работе тщательно протестированные версии программ и библиотек, и в то же время играть с их передовыми версиями, не беспокоясь о том, что одна из них мешает другой. А в остальном это довольно стандартное обновление с инкрементными изменениями. Если Fedora 29 является единственной ОС, установленной на компьютере, *Grub* загрузится прямо в нее, не отображая меню загрузки. Вдобавок обновленный рабочий стол *Xfce 4.13* на *GTK3* доступен из официальных репозиториях Fedora 29.

Выходя за рамки рабочего стола, Fedora 29 — это не только рабочая станция. Есть также Fedora Server и Fedora Atomic Host для размещения контейнеров, который теперь переименован в Fedora Silverblue. Разработчики также утверждают, что Fedora 29 дружелюбнее к ARM, чем раньше. Включение поддержки zRAM для подкачки на ARMv7 улучшит производительность и надежность дистрибутивов на устройствах ARM с небольшим объемом ОЗУ, типа Raspberry Pi. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Red Hat
САЙТ: www.fedoraproject.org
ЛИЦЕНЗИЯ: GPL и др.

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО В РАБОТЕ	8/10
ДОКУМЕНТАЦИЯ	9/10

Инкрементное обновление семейства релизов Fedora. Одни только улучшения, связанные с Gnome 3.30, уже делают его достойным рекомендации для пользователей настольных ПК.

» **Рейтинг 8/10**

DiskStation DS418play

Наслаждаясь своей обширной коллекцией 4K-медиа, Линдси Хандмер задается вопросом: ну что в жизни лучше, чем потоковое вещание кинофильмов?

СПЕЦИФИКАЦИИ

- **Процессор**
двухъядерный Intel Celeron J3355, 2 ГГц
- **ОЗУ** 2 ГБ (до 6 ГБ)
- **Отсеки**
4×3,5/2,5-дюймовые (горячая замена)
- **Емкость** 56 ТБ (4×14 ТБ)
- **Сеть** 2×Gigabit Ethernet
- **Разъемы** 2×USB 3.0
- **Мощность** 29 Вт в работе, 10 Вт в простое
- **Габариты**
166×199×223 мм
- **Вес** 2,23 кг
- **Гарантия** 2 года

DS418play позволяет хранить огромное количество видео и фотографий, а затем легко передавать их все на любое устройство в вашей сети (или даже удаленно, если у вас хватает пропускной способности), не влияя на ваше соединение с Интернетом. Он также предоставляет дополнительную опцию резервного копирования для этих повседневных воспоминаний и имеет кучу других встроенных функций.

DS418play построен на базе слегка устаревшего двухъядерного процессора Intel Celeron J3355, способного достигать частоты до 2,5 ГГц. Хотя процессор включает Intel Graphics, выходов HDMI нет. NAS поставляется с 2 ГБ оперативной памяти плюс пустой слот, который может принять дополнительные 4 ГБ; итого 6 ГБ.

Сзади 418play имеет два гигабитных порта LAN, которые поддерживают агрегацию каналов и аварийное переключение. Агрегирование каналов особенно удобно в этом случае: оно означает, что при правильной настройке сети вы можете удвоить доступную пропускную способность для NAS. 418play также имеет два порта USB 3.0 (один спереди и один сзади), которые можно использовать для подключения внешних устройств хранения данных, но без eSATA.

Четыре отсека для дисков не требуют инструментов и могут быть заменены в горячем режиме, и совместимы с 2,5- и 3,5-дюймовыми дисками. Они также запираются с помощью прилагаемого ключа, но это не особо надежно, отсеки легко открыть с помощью небольшой отвертки с плоской головкой. NAS поддерживает файловые системы Btrfs и Ext4, а с поддержкой до четырех дисков по 14 ТБ может иметь в общей сложности 56 ТБ емкости.

Лучший расчет развеваться

Несмотря на хорошую вентиляцию, двойные 92-мм вентиляторы DS418play могут быть шумными под нагрузкой по сравнению с одним большим вентилятором. Имеется возможность их запуска в тихом режиме, хотя и за счет более высоких температур ваших жестких дисков.

Настроить и запустить устройство Synology очень просто. Вставьте несколько дисков, подключите его к источнику питания и локальной сети, а затем укажите браузеру на <http://find.synology.com>. Чтобы максимально упростить обслуживание, DS418play включает распечатанное руководство по быстрому запуску с инструкциями и схемами.

Операционная система Synology NAS является одной из наших любимых, она хорошо продумана и проста в использовании, а также предоставляет кучу программ, которые можно загружать и запускать на NAS. Хотя веб-интерфейс является отличным способом управления NAS, Synology также предлагает приложения для управления со смартфона или планшета. Вместо того, чтобы объединять все функции в одном приложении, Synology создала более 10 отдельных приложений, которые охватывают всё, от управления файлами до создания заметок.



Имея весь ваш контент в DS418play, вы можете управлять им либо через веб-интерфейс, либо со своего дивана, используя смартфон или планшет.

Ключевые особенности DS418play сосредоточены на работе с медиа. Самое главное, что устройство способно перекодировать два видеопотока 4K одновременно. Поскольку видеоконтент 4K становится всё более популярным, мы считаем, что это, безусловно, отличный вариант на будущее.

Мы протестировали Synology DS418play с 2 ГБ оперативной памяти, четырьмя жесткими дисками Seagate IronWolf Pro и гибридным RAID Synology. DS418play работал хорошо, и у нас не возникло проблем с потоковой передачей мультимедиа на несколько устройств одновременно. Транскодирование также работает безупречно, хотя для запуска и работы Plex потребовалось немного повозиться. На передаче файлов большого объема при скорости чтения 108,5 МБ/с и скорости записи 109,2 МБ/с DS418play использовал нашу сеть по полной.

Synology DS418play — интересное устройство NAS, так как ориентировано на любителей 4K-мультимедиа, но здесь отсутствуют связанные с этим функции, такие как порт HDMI или продвинутое расширение, такие как eSATA. **LXF**

ВЕРДИКТ

ПРОИЗВОДИТЕЛЬ: Synology
САЙТ: www.synology.com
ЦЕНА: £420 (без дисков)

ФУНКЦИОНАЛЬНОСТЬ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
УДОБСТВО В РАБОТЕ	8/10
ОПРАВДАНОСТЬ ЦЕНЫ	7/10

Идеален для обычного пользователя мультимедиа, но отсутствие расширения eSATA и поддержки BM, устаревший процессор и отсутствие HDMI оставляют желать лучшего.

» **Рейтинг 8/10**

Intel Core i9 9900K

Еще больше основных ядер от Intel для **Джереми Лэйрда**, чтобы добывать монеты Monero в его постоянно растущую коллекцию.

СПЕЦИФИКАЦИИ

- Разъем LGA1151
- Тип 64 бит
- Техпроцесс 14 нм
- Ядер 8
- Потоков 16
- Тактовая частота 3,6 ГГц (5 ГГц турбо)
- Кэш 16 МБ L3
- Память DDR4-2666, 2 канала, 64 ГБ макс.
- PCIe v3.0, 16 линий
- Графика Intel UHD Graphics 630
- Частота графики 350 МГц (1,2 ГГц макс), OpenGL 4.5
- Тепловыделение 95 Вт
- Виртуализация VT-x, VT-d

Еще в конце 2006 г. Intel представила свой первый настольный четырехъядерный процессор. Перенесемся вперед почти на 11 лет в лето 2017 г., и что изменилось? С точки зрения количества ядер, не так много. Основной сокет Intel для настольных ПК по-прежнему ограничен четырьмя ядрами.

Хорошо, вы могли бы купить так называемые настольные чипы с более чем четырьмя ядрами, но они были частью того, что было и остается сегодня серий серверных платформ со сменой торговой марки, оснащенных дорогостоящими многоканальными контроллерами памяти и другими функциями, которые увеличивают цену, но мало что делают для повышения производительности настольных компьютеров. Тем не менее, в конце 2018 г. мы наблюдаем основной восьмиядерный процессор Intel, всего через год после того, как компания представила свои первые шестиядерные процессоры. Что, собственно, происходит?

Ответ — Ryzen. Новому Core i9-9900K не бывать бы без процессора AMD Ryzen в прошлом году. Ryzen не идеален, но его хватило, чтобы пробудить Intel от спячки. В результате — быстрое ускорение планов Intel и удвоение количества основных ядер настольных ПК чуть более чем за год.

Новый 9900K подходит для большего обновления процессоров для основного сокета LGA1151. С этой новой серией 9000 ЦП Intel пересмотрел свою реализацию HyperThreading. Проще говоря, HyperThreading исчез, за исключением топового восьмиядерного процессора 9900K с 16 потоками. Что расстраивает, поскольку эта функция имеется во всех чипах: Intel просто отключила ее, чтобы создать искусственную иерархию продуктов.

Эпическая производительность

Какова производительность этого флагманского представителя новой серии девятого поколения? Intel устанавливает его базовую частоту в 3,6 ГГц, а максимальную частоту турбо в 5 ГГц (примечание: последняя не является частотой всех ядер). Тем не менее, гарантируется, что новый 9900K обеспечивает невероятную однопоточную производительность. Она находится на одном уровне с существующим однопоточным королем Core i7-8086K.

Как и ожидалось, 9900K задает новый стандарт для основных чипов, когда речь идет также и о многопоточной производительности. Да, он вполне «рвет» топовый чип AMD Ryzen 2700X, новый во всем. На самом деле, в контексте настольных ПК



Ценовая категория делает этот чип недоступным для большинства домашних пользователей, но те, кто может себе такое позволить, выиграют от отличной производительности.

достаточно быстро сделать подобные 10-ядерному процессору 7900X Intel, производные от серверного ЦП для сокета LGA2011, выглядят избыточными.

Разумеется, одна из слабостей 9900K заключается в разгоне. Со стандартными настройками и при большой нагрузке он обычно работает на 4,7 ГГц всех восьми ядер. Нам удалось довести его до 5 ГГц на всех ядрах. Это разгон на шесть с небольшим процентов, и этого недостаточно, чтобы повлиять на ваш субъективный опыт вычислений в реальном мире. Учитывая количество ядер и без того высокие тактовые частоты, которые Intel включила как стандарт для 9900K, ограниченная область для разгона не удивительна.

9900K и остальная часть нового набора девятого поколения совместимы с существующими чипсетами материнских плат серии 300, хотя различается от платы к плате.

Таким образом, сам по себе новый 9900K — фантастический чип. Это самый быстрый настоящий процессор для настольных ПК. Но относительно других представителей новой линейки девятого поколения — сомнительно. Как сравнить восьмиядерный восьмипоточный чип, такой как новый Core i7-9700K, с шестиядерным 12-поточным Core i7-8700K? Вот это интересный вопрос. **LXF**

ВЕРДИКТ

ПРОИЗВОДИТЕЛЬ: Intel
САЙТ: www.intel.com
ЦЕНА: £600

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
УДОБСТВО В РАБОТЕ	9/10
ОПРАВДАНОСТЬ ЦЕНЫ	7/10

Огромная однопоточная и многопоточная производительность, благодаря использованию основного сокета, но не имеет запаса по разгону, и не так уж доступен по цене.

» **Рейтинг 9/10**

Тест	Intel Core i9-9900K	Intel Core i7-8086K	AMD Ryzen 7 2700X
Cinebench R15 Single (Index)	214	212	180
Cinebench R15 Multi (Index)	1994	1630	1814
Кодирование x264 (кадр/с, в среднем)	42,69	35,98	31,71
Задержка памяти (нс)	47,7	49,1	68,4
Пропускная способность памяти (МБ/с)	46144	46451	46506
Потребляемая мощность работа/простой (Вт)	65/185	69/200	91/221
Цена (£)	600	490	300

Surviving Mars

Мы часто говорим, что менеджмент заперт в своем собственном закрытом пузыре. Знать бы, что **Фрейзер Браун** поймал их в ловушку на Марсе...

СПЕЦИФИКАЦИИ

Минимальные

- **ОС** Ubuntu 14 64-разрядная
- **Процессор** Intel Core i3 4-го поколения
- **ОЗУ** 4 ГБ
- **На диске** 6 ГБ
- **Графика** OpenGL 4.5 (GeForce 600 / AMD Radeon 7700), 1 ГБ видеопамяти

Рекомендуемые

- **Процессор** 4-ядерный
- **ОЗУ** 8 ГБ
- **Графика** GeForce 970-level, 4 ГБ видеопамяти

Выживание в этом негостеприимном мире — подвиг не слабый, но стоящий усилий.

Люди не станут прибывать на Марс, пока не смогут там жить, поэтому сперва надо создать всю инфраструктуру. К тому времени, когда первый человек ступил на планету, у нас имелась сложная сеть жизнеобеспечения, подающая кислород и электричество во всё, от куполов до дронов, а также множество идей о том, что делать дальше. Список потенциальных задач ошарашивает, но игра *Surviving Mars* [Выживание на Марсе] не принуждает вас сразу беспокоиться о потребностях колонистов: начало у нее щадящее.

Несмотря на свой уклон в сторону выживания, *Surviving Mars* следует той же схеме, что и *Tropico* от Haemimont, превращая ресурсы в готовые продукты и создавая на их базе целые отрасли промышленности, пытаясь при этом всем угодить. Это некая знакомая основа и поддержка, когда начинаются неожиданности.

Механические помощники

Среди пыли, метеоров и торнадо, воссоздавать жизнь на Марсе — это прорва работы. Преодоление всяких бедствий и наблюдение за тем, как армия дронов всё исправляет — опыт, доставляющий невероятное удовлетворение. Если вы подготовились к худшему, пополняете свои запасы и правильно разместили свои дроны, то вам предстоит наблюдать механический балет, по мере того как прилежные сборщики подбирают ресурсы и затем в считанные секунды берут всё под контроль, ремонтируя машины и дроны во всей колонии.

Тем не менее, даже с учетом автоматизации всё равно кажется, что катастрофа идет за вами по пятам. Это битва между людьми и природой, и при всех хитроумных технологиях именно упорством создаются успешные колонии.

Surviving Mars постоянно заставляет вас расширяться, будь тому причиной иссякающие ресурсы или колонисты, требующие новых услуг. Неудивительно, что экспансия предъявляет еще больше требований к вашей колонии.



Купола бывают одной формы, но разных размеров. Не нужно много времени, чтоб заполнить их тружениками-колонистами и их пожитками.

Поскольку эти сложно организованные колонии могут разрастись до гигантского размера, *Surviving Mars* необходим надежный интерфейс пользователя, чтобы во всем разобраться. К сожалению, тот, который есть, не справляется с задачей. Имеются функции качества жизни, такие как возможность закреплять вещи на панели быстрого доступа, но меню запутанные, и много чего не хватает. Дается широкий обзор колонии, но нужно больше способов покопаться в деталях.

Результатом является излишний микро-менеджмент, который кажется неуместным в игре, где вы командуете армиями автоматизированных помощников. Нам в общем-то нравится, что даже когда вы построите довольно развитую колонию, всё равно нужно не дремать, но часто бывает слишком уж много всего, чтоб одновременно этим жонглировать.

При всей канительности и напряженности *Surviving Mars*, ничто другое так умело не сочетает выживание и градостроительство. Это непростая, но приносящая удовлетворение космическая аварийная ситуация, хотя всё-таки жаль каждого колониста, втопанного в красную пыль... **LXF**

Пускай Солнце и зашло, но жизнь на Марсе всё еще продолжается до рассветных часов.



ВЕРДИКТ

РАЗРАБОТЧИК: Paradox Interactive

САЙТ: www.paradoxplaza.com

ЦЕНА: £26,99

ИГРОВОЙ ПРОЦЕСС	7/10
ГРАФИКА	7/10
СРОК СЛУЖБЫ	8/10
ОПРАВДАНОСТЬ ЦЕНЫ	8/10

Выживание на Марсе — тяжкий труд, но управление процветающей колонией никогда не перестает быть притягательным.

» **Рейтинг 8/10**

Megaquarium

Налицо рыбка в мутной воде, и в кои-то веки управление — это не макиавеллиевские махинации с дерганием за нитки, как разобралась **Филиппа Уорр!**

СПЕЦИФИКАЦИИ

- **Минимальные**
- **ОС** Ubuntu 64-разрядная
- **Процессор** Intel Core i5-4690 или AMD Ryzen 7 2700U
- **ОЗУ** 8 ГБ
- **Графика** GeForce GTX 560, AMD Radeon HD 7770
- **На диске** 1 ГБ свободного места

Это игра для фанатов благоустройства. Время так и утекает, пока вы расширяете свой план этажа, размещаете новые резервуары, возитесь с насосами и фильтрами, регулируете температуру и вообще всячески стараетесь осчастливить и рыбок, и посетителей.

Миссии кампании постепенно знакомят вас с концепциями и ходовыми валютами. Например, есть регулярные наличные деньги, которые вы зарабатываете на входных билетах и продаже закусок и напитков, а также в сувенирных магазинах. Но вы еще и накапливаете баллы экологии и науки, когда гости наслаждаются вашими экспонатами. Они тратятся на поиск новых видов и новых технологий, соответственно. Престиж выступает как общий показатель привлекательности вашего аквариума. Наберите достаточно, и ваш рейтинг поднимется, разблокировав больше лакомых кусков для вашего аквариума.

Рыбоснабжение

Каждое существо имеет конкретные потребности, так что в конструировании аквариума есть и логический элемент головоломки. Рыбы-хулиганы будут гонять рыбешек послабее, а некоторые рыбы злятся, заведя других представителей своего вида. Морские звезды не станут принимать пищу напрямую; вместо этого им надо, чтобы вы кормили другое существо в резервуаре, а морская звезда получит свое питание, подбирая объедки.

Некоторые системы трудновато собрать. Скажем, мы не сразу сообразили, как подключить несколько резервуаров к одному насосу. А это важно, потому что насосы позволяют размещать неказистое оборудование аквариумов на некотором расстоянии, вне поля зрения гостей. И когда насос обслуживает несколько емкостей, экономится ценное пространство.

Было бы удобно иметь опции меню для сортировки рыб по различным категориям, и не помешали бы несколько настроек качества жизни. Одна из важных — это чтоб не надо было удалять сегмент стены вручную, когда требуется проделать дверь в помещение для персонала. Возможность увидеть размер



Продвигайтесь от простых стеклянных резервуаров к более замысловатым, сквозь которые можно проходить. Вы ведь помните, что нельзя стучать по стеклу, привлекая внимание экспонатов?

резервуара в информационном окне аквариума помогла бы с размещением некоторых из активных рыб, чьи потребности в просторе для плавания выше определенного минимума. Заимствование из недавно выпущенной *Two Point Hospital* (см. **LXF243**) и использование тепловых карт для оценки опыта посетителей было бы полезно при размещении туалетов и буфетов.

Ох, а выяснение различий в настройках сложности потребовало от нас загрузки одного и того же уровня в разных режимах и проверки, какие числа не совпали. Насколько мы можем судить, это вопрос чисто денежный — персонал нанимается за более высокую плату, оборудование и рыбки тоже дороже. Так что трудность, похоже, только в уменьшении финансового буфера.

Вишенка на этом рыбном пироге — то, что при включении масштаба на максимум камера переключается на ближний план, и вы и впрямь оказываетесь в аквариуме. Мало того, что можно обустроить пространство — вы можете совершить путешествие по нему, как если бы вы были посетителем, гордясь своей работой и любуясь очередной коллекцией угрей. **LXF**

ВЕРДИКТ

РАЗРАБОТЧИК: Twice Circled

САЙТ: www.twicecircled.com/2c/megaquarium

ЦЕНА: £ 19,49

СЮЖЕТ	8/10
ГРАФИКА	8/10
УВЛЕКАТЕЛЬНОСТЬ	8/10
ОПРАВДАНОСТЬ ЦЕНЫ	9/10

Очаровательная игра в управление тематическим водным парком, где рыбы — друзья, а не поджарка в масле.

» **Рейтинг 8/10**

Позаботьтесь, чтобы ордам ваших посетителей хватало туалетов.



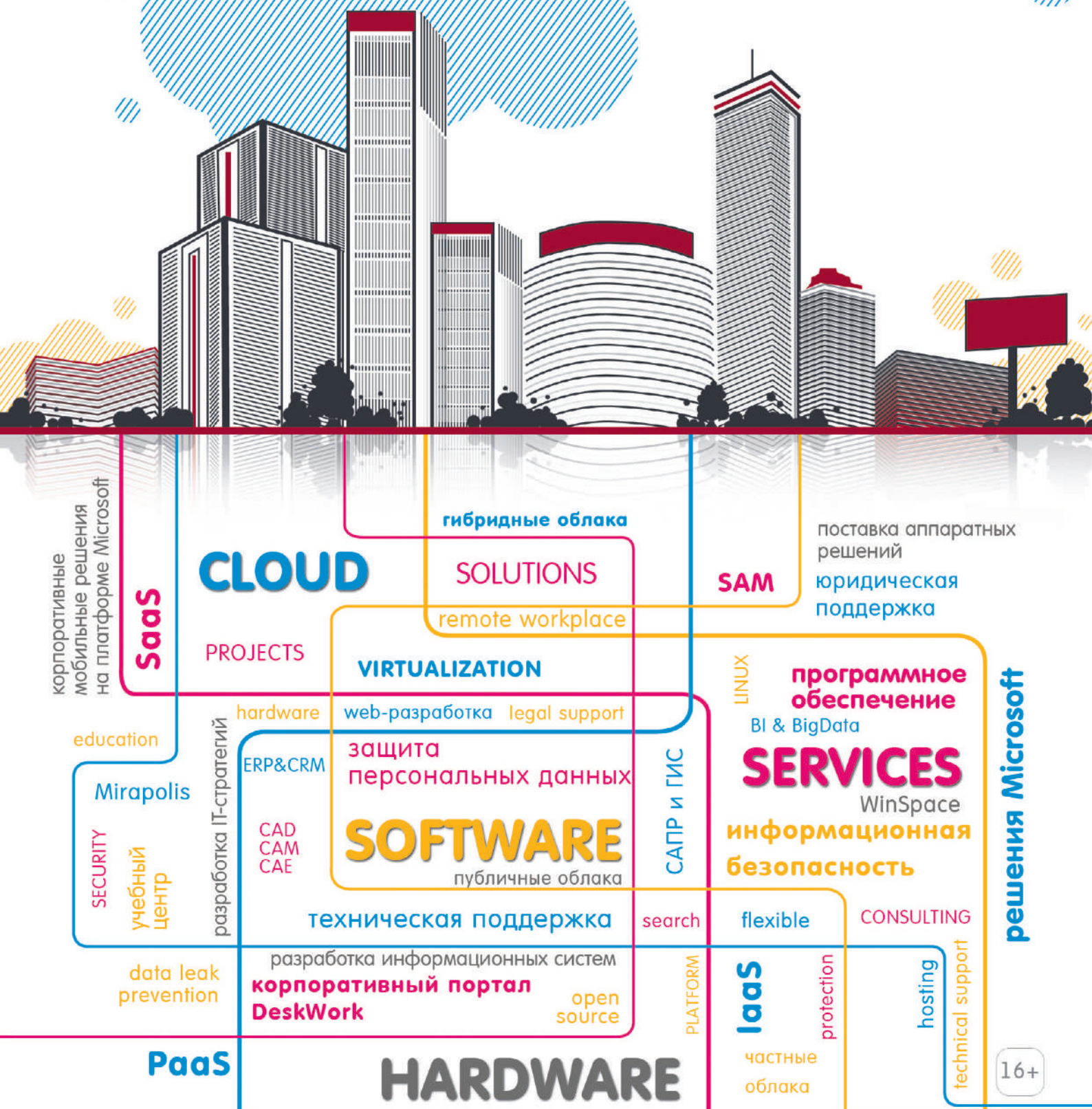
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



Мобильные новости

ВСЕЛЕННАЯ 5G

Этапы большого пути

Huawei продемонстрировала возможности домашней широкополосной сети 5G.

В ходе Форума мобильной широкополосной связи Huawei 2018 компании Huawei и Three UK продемонстрировали работу домашней широкополосной сети по технологии 5G с использованием С-диапазона 100 МГц. Презентация показала возможности 5G-сервисов, радикально меняющие привычные услуги и сервисы. Задействовались новейшие маршрутизаторы домашней широкополосной связи Huawei — первые в мире коммерческие терминалы стандарта 5G — что позволило гостям форума воспользоваться услугами сверхвысокоскоростной широкополосной связи 5G, в том числе облачными игровыми сервисами и потоковой передачей видео 4K. Демонстрация стала следующим этапом после успешного докоммерческого испытания сети, которое Huawei и Three провели в начале

года. Предполагается, что максимальная скорость загрузки достигнет 2 Гб/с со средней скоростью 1 Гб/с для одного пользователя. Huawei продолжит сотрудничать с оператором Three в ходе следующих испытаний услуг 5G в Соединенном Королевстве, которые будут проводиться в густонаселенных городских районах и на железнодорожных станциях, что заложит фундамент для полномасштабного коммерческого запуска сетей 5G в 2019 г. «Проведенные испытания технологии 5G демонстрируют возможности, которые она создает в сегменте домашней широкополосной связи. Huawei продолжит сотрудничать с Three UK, чтобы открыть клиентам полный доступ к преимуществам, которые дает коммерческое использование технологии 5G», говорит Янь Чаобин [Yang Chaobin], президент направления Huawei 5G.



Первые в мире коммерческие терминалы стандарта 5G: абонентские устройства 5G (CPE).

ПРОДУКТЫ SAMSUNG

Технологии автономного вождения

Samsung и KOTSA построят полигон 5G — V2X.

Samsung подписала Меморандум о взаимопонимании (MoU) с Корейским агентством по безопасности перевозок (KOTSA) с целью сотрудничества в области телекоммуникаций следующего поколения и создания инновационных технологий для автономного вождения. Партнеры оборудуют сетями связи 4G LTE, 5G и Tele-To-Everything (V2X), а также соответствующей IT-инфраструктурой испытательный полигон для тестирования технологий автономного вождения K-City в Корею. K-City расположен на территории площадью 360 тыс. кв. м. и моделирует 5 разных условий вождения: город, сельская местность, автомагистрали и стоянки. Благодаря новым технологиям связи между транспортными средствами, на полигоне можно будет

воссоздать типичные реальные дорожные ситуации, а также перекрестки и тоннели. Samsung создаст ключевую IT-инфраструктуру, обеспечив в K-City минимальное время ожидания и наибольшую информационную безопасность. Мобильные вычисления (Mobile Edge Computing, MEC) возле базовых станций критически важны для обеспечения оперативной связи между быстро движущимися автомобилями и позволяют машинам реагировать на неожиданные ситуации. Сетевое решение, реализованное в K-City при поддержке Samsung и KOTSA, будет доступно специалистам и компаниям разного уровня. Потом полигон станет центром исследований автономного вождения и подключенных автомобилей, авангарда развивающихся отраслей, связанных с 5G.

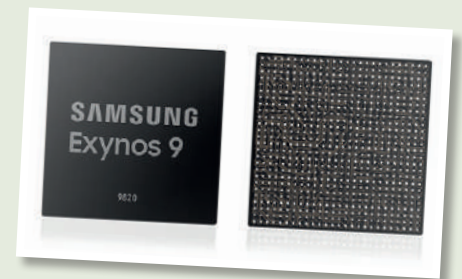
МОБИЛЬНЫЕ ПРОЦЕССОРЫ

Взаимодействие с искусственным интеллектом

Samsung представила процессор для флагманских устройств.

Samsung представила новейший флагманский процессор Exynos 9 Series 9820, с CPU 4-го поколения, 2-гигабитным модемом LTE Advanced Pro и новым модулем нейронных вычислений (NPU), обеспечивающими новый уровень при обработке задач. Благодаря улучшенной архитектуре, модифицированные ядра 4-го поколения Exynos 9820 обеспечивают 20% прироста производительности на ядро и на 40% повышают энергоэффективность по сравнению с предшественником; примерно на 15% повысилась и многоядерная производительность. Новый Exynos получил встроенный графический модуль Mali-G76, на 40% повышающий производительность при снижении энергопотребления на 35%. Встроенный в CPU модем LTE-Advanced Pro обеспечивает быстрый беспроводной доступ к Интернету со скоростью скачивания до 2,0 Гбит/с, агрегацию 8 несущих частот и скорость по каналу «вверх» до 316 Мбит/с.

При скачивании на скоростях до 2,0 Гбит/с загрузка кинофильма в разрешении FHD с весом файла 3,7 ГБ идет около 15 с, а ресурсоемкие многопользовательские онлайн-игры обрабатываются без сбоев и торможений. Объединив несколько несущих частот с антенной 4×4 MIMO (Multiple-Input, Multiple-Output) и схемой 256-QAM (квадратурной амплитудной модуляции) и улучшив использование лицензированных и нелицензированных частотных диапазонов Licensed-Assisted Access (eLAA), модем держит невероятно быструю и стабильную связь даже при движении. Exynos 9820 получил мультимедийный кодек (MFC) с кодированием и декодированием видео в формате 4K UHD на 150 кадр/с. Кодек умеет обрабатывать 10-битные цвета. Массовое производство процессоров Exynos начнут в конце 2018 г.



Exynos 9820: модифицированные ядра 4-го поколения и встроенный модем LTE Advanced Pro.

5G ВСЕ БЛИЖЕ

Демонстрируют прогресс

Обеспечена передача данных 5G в полосе 400 МГц мм-спектра.

Samsung Electronics America, Qualcomm Technologies и оператор связи Verizon объявили об успешной установке канала передачи данных 5G New Radio (NR) на полосе шириной 400 МГц и частоте 28 ГГц, в результате чего пропускная способность сети превысила 1,7 Гбит/с. Соединение позволило передавать данные на тестовый смартфон на скоростях, превышающих 1 Гбит/с. Передача данных на базе стандарта 3GPP 5G NR и двойного подключения (EN-DC) оставалась непрерывной. Коммерческое оборудование 5G NR и 4G LTE от Samsung использовалось для передачи данных в сети 5G NR на частоте 28 ГГц оператора Verizon. Тест продемонстрировал быстрый, стабильный и эффективный путь к запуску коммерческой 5G связи с использованием неавтономного режима работы и двойного подключения на базе технологий 5G и LTE одновременно. Поток данных со скоростью 1,7 Гбит/с передавался на смартфон Qualcomm Technologies с тестового устройства, оснащенного модемом Qualcomm Snapdragon X50 5G и антенными модулями с интегрированными RF-передатчиком и антенной. Тестирование проходило в лаборатории Qualcomm Technologies

в Сан-Диего. Ранее Verizon объявила о первой успешной передаче данных в сети компании на модифицированном для поддержки 5G смартфоне в коммерческой 5G-сети в городе Провиденс, штат Род-Айленд. При тестировании использовались решения Samsung 5G NR и 5G-модем Snapdragon X50, а также инновационные антенные модули Qualcomm QTM052 для мм-спектра. «Успешное сочетание нескольких сетевых технологий еще на шаг приближает нас к коммерческому запуску мобильной связи 5G, — отметил Вучжун Ким [Wooyune Kim], старший вице-президент бизнес-подразделения Network Business Samsung Electronics в Северной Америке. — Использование обширного миллиметрового спектра и подключения EN-DC демонстрирует, как бесшовная связь 5G/LTE позволяет успешно осуществлять скоростную передачу больших объемов данных в сетях следующего поколения». **LXF**

СВЯЗЬ 5G/LTE ПОЗВОЛЯЕТ СКОРОСТНУЮ ПЕРЕДАЧУ БОЛЬШИХ ОБЪЕМОВ ДАННЫХ В СЕТЯХ.

Сравнение

Calligra Suite » Google Docs
» LibreOffice » OnlyOffice » WPS Office



Шашанк Шарма

днем — благово-
спитанный адвокат
в суде Нью-Дели,
а ночью — каратель
открытого кода!

Офисные комплекты

Сперва — настольные комплекты и отдельные пакеты; а потом офисные программы проникли и в Интернет. **Шашанк Шарма** оценил лучшие из них.

ПРО НАШ ТЕСТ...

Мы запускаем последнюю версию всех приложений поверх рабочего стола Budgie. *WPS Office* и *OnlyOffice* предоставляют Snap-пакеты, так что установка происходит с легкостью необычайной. Вы также найдете свежие версии *LibreOffice* и *Calligra* в репозиториях многих дистрибутивов. Вот почему мы не проверяли их на простоту установки. Однако если в репозиториях вашего дистрибутива отсутствует последняя версия *Calligra*, мы всячески предостерегаем вас от попытки установить ее вручную. Компиляция занимает слишком много времени, а приводит в результате только к непреодолимым ошибкам.

Мы протестировали три общих компонента офисных комплектов, но не менее заинтересованы в других предлагаемых компонентах, если таковые имеются. Другим важным фактором является поддержка различных форматов, включая ODF и даже проприетарный формат *Microsoft Office*, а также способ работы с файлами.

Расширяемость функциональных возможностей плагинами и наличие документации, помогающей новым пользователям освоиться с предлагаемым инструментарием, также служат значимыми критериями нашей оценки.



С тех пор, как мы в последний раз рассматривали офисные комплекты, еще в **LXF167**, главные производственные приложения, предлагаемые пользователям Linux, претерпели серьезные изменения. Например, теперь приложения могут похвастаться гораздо лучшей совместимостью с проприетарными форматами. Другим важным событием является введение нескольких комплектов онлайн-офисов, подобных Google Docs. Фактически, многие крупные игроки, такие как *Microsoft Office* и даже *LibreOffice*, теперь предлагают онлайн-решения.

Настольный офисный комплект обычно включает несколько различных приложений: текстовый процессор, обработчик электронных таблиц и средство

проведения презентаций. В некоторых комплектах есть еще база данных или приложение для заметок.

Большинство дистрибутивов Linux, даже облегченных версий, таких как Damn Small Linux, поставляются с офисным комплектом или программами, предоставляющими те же инструменты. Вам не обязательно прибегать к гигантскому проекту, если всё, что вас интересует — это текстовый процессор. В подобных ситуациях вы также можете установить автономные приложения, такие как *Abiword* для обработки текста и *Gnumeric* для электронных таблиц.

Отличает эти приложения друг от друга набор функций по умолчанию. Это станет особенно ясным, если вы сравните онлайн-решение с автономным настольным приложением.

Обработка слов

Сияй, ваш внутренний Шекспир!

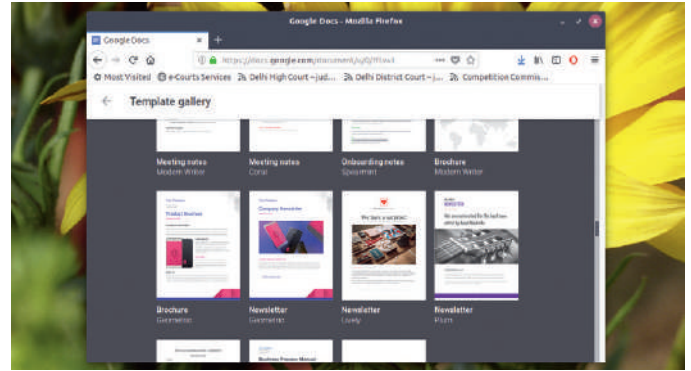
Текстовый процессор в *Calligra Suite* называется *Words* [Слова]. Однако, в отличие от других текстовых процессоров, он имеет боковую панель справа, используемую для настройки различных параметров форматирования документа: вставка комментариев, верхних и нижних колонтитулов и т.д.

Помимо этой неубиваемой боковой панели по умолчанию, приложение также предоставляет кучу различных элементов, называемых докерами. Они включают статистику по документу, добавляют фигуры, и т.д., и занимают немало места на экране, почти не оставляя площади для самого вашего документа. Несмотря на вынужденное и постоянное изменение размеров различных докеров и закрытие лишних, у вас практически не остается места даже для доступа к полю комментариев! Из-за своего нетипичного дизайна этот инструмент не самый простой в работе, особенно если вы когда-либо работали с текстовым процессором.

Google Docs позволяют выполнять поиск по темам в Интернете и перетаскивать результаты в документ, автоматически оформляя цитаты; в этом его фишка [USP], кроме того, что он находится в Сети.

И *LibreOffice Writer*, и *Calligra Words* могут похвастаться встроенной поддержкой различных форматов, включая открытый формат документов и проприетарные форматы *Microsoft Office*. К сожалению, при работе с файлами DOC или DOCX оба этих приложения имеют тенденцию несколько исказить внешний вид документа. *LibreOffice Writer* достаточно часто и плачевно не может определить форматирование текста и отображает документ совершенно иначе, чем оригинал; но *Calligra Words* работает намного лучше. Тем не менее, даже и в *Calligra Words* иногда возникают проблемы с отображением нумерованных списков или портятся поля.

Из всех приложений, представленных в нашем Сравнении, *OnlyOffice* обеспечивает наилучшую поддержку форматов *Microsoft Office* и безупречно



Все проекты, представленные в этом месяце, предусматривают для всех приложений, составляющих офисный комплект, несколько шаблонов на выбор.

открывал все файлы, которые мы тестировали. В отличие от других инструментов, комментарии, добавленные в документ, отображаются слева, но в остальном интерфейс очень похож на стиль ленты в *Microsoft Office*.

Самым большим недостатком *WPS Writer* является отсутствие поддержки Open Document Format (ODF). В самом FAQ объясняется, что разработчики могут рассмотреть возможность добавления функциональности ODF, увеличив свою команду или накопив достаточно пользователей, чтобы оправдать затраты времени и усилий, необходимых для реализации поддержки ODF. Но пока ODF находится у *WPS Writer* в лотке ожидания.

ВЕРДИКТ

CALLIGRA SUITE	6/10	ONLYOFFICE	10/10
GOOGLE DOCS	7/10	WPS OFFICE	5/10
LIBREOFFICE	10/10		

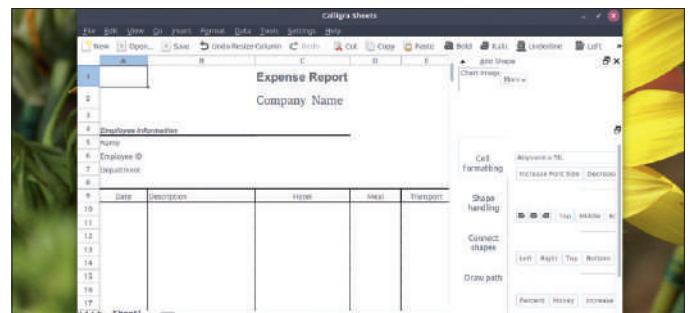
Calligra отстает из-за своего загроможденного интерфейса, а *WPS Office* попал на последнее место, поскольку не поддерживает ODF.

Электронные таблицы

Щелкаем числа, как орехи.

Приложения для работы с электронными таблицами в разных офисных комплектах предлагают практически одинаковую функциональность. Как минимум, каждый из них предоставляет ряд функций для создания формул и выполнения сложных вычислений. Помимо выполнения основных математических функций, все приложения для работы с электронными таблицами имеют встроенные функции для общепринятых финансовых и статистических операций. Одной из лучших функций этих современных замен для карандаша и таблиц является т.н. анализ «что если», когда вы меняете одно поле данных и видите, как инструменты автоматически меняют значения связанных полей.

LibreOffice Calc и *WPS Spreadsheets* превосходят другие по простоте использования. Кроме того, *Calc* может похвастаться множеством мастеров, которые помогут вам освоить его расширенные функции. Легко будет извлекать данные из баз данных и сотрудничать с другими пользователями, что идеально для большинства офисных установок. *Google Spreadsheets* не так уж и далеко позади, и, как и все другие приложения *Google Docs*, предоставляет массу различных шаблонов на выбор. Документация присутствует на месте и предлагает очень точную помощь. Все программы поддерживают немало форматов и позволяют экспортировать данные в виде файла PDF



Интерфейс *WPS Spreadsheets* более загроможден, чем у *OnlyOffice*, но из всех тестируемых продуктов труднее всего работать с *Calligra*.

или CSV, а по умолчанию (кроме *Calc* и *Sheets*) — в формат XLSX. Приложения также выглядят довольно схожими, с единственным исключением — *Calligra Sheets*. И опять-таки, употребление докеров для предоставления различных функций делает работу с *Calligra* далекой от интуитивной, несмотря на наличие тех же функций, что и в других приложениях, например, *LibreOffice Calc*.

ВЕРДИКТ

CALLIGRA SUITE	7/10	ONLYOFFICE	9/10
GOOGLE DOCS	10/10	WPS OFFICE	10/10
LIBREOFFICE	10/10		

Снова страдает *Calligra* — из-за своего интерфейса, а не скудости предлагаемых функций.

Взаимодействие с пользователем

Они похожи друг на друга; но ведут ли они себя одинаково?

Производственные приложения — термин, который в наши дни встречается довольно часто. До 1980-х гг. мир всё ещё вынужден был обходиться печатными машинками или ручкой и бумагой, но с появлением офисных комплектов даже самые старинные приложения такого рода изменили наш способ использования компьютера дома и на работе. Поэтому неудивительно, что в мире ныне осталось очень мало рабочих мест, не требующих базовых навыков работы с офисным программным обеспечением.

Хотя все комплекты предоставляют схожую функциональность в одном-двух приложениях — своих компонентах, они достаточно различны по своим рабочим процессам, и переключение между ними может потребовать некоторого обучения, прежде чем вам станет удобно использовать их в повседневной работе.

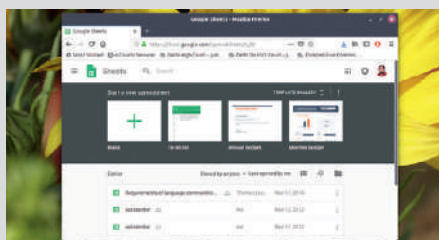
Если вас интересует только один компонент офиса — скажем, электронная таблица или текстовый процессор — то обратитесь к разделу «Рассмотрите также» в конце нашего Сравнения, чтобы получить кое-какие рекомендации.

Google Docs

8/10

Данное предложение отличается тем, что в нашем Сравнении это единственный офисный комплект онлайн. (Но будем справедливы: *OnlyOffice* тоже предоставляет облачное решение.) Первоначально этот комплект лучше всего работал в браузере *Chrome*, но, используя его на какой-либо альтернативе, например, в *Firefox*, вы не упреетесь в проблемы с производительностью или отсутствием функций.

Главное преимущество *Google Docs* — огромное количество шаблонов для каждого из приложений-компонентов. Простая интеграция с *Google* для выполнения поиска и интеграции результатов или обмена файлами с другими пользователями одним щелчком мыши также является бонусом. Если вы часто с кем-то сотрудничаете или, отправляясь с дороги, не можете позволить себе отложить срочные файлы, применение Диска *Google* в качестве хранилища обеспечивает немалую роскошь — ведь вы можете получить доступ к своей работе из любого места, было бы подключение к Интернету.



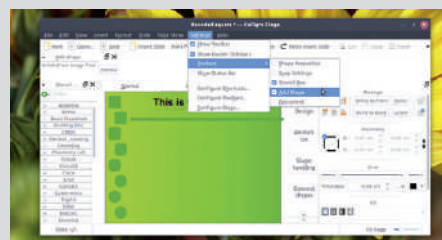
Calligra Suite

7/10

С самой *Calligra* в общем-то всё в порядке, но с ее эзотерическим рабочим процессом придется побороться. Например, нельзя вставить в презентацию или документ какую-либо из стандартных геометрических фигур без доступа к докеру *Add Shapes*. На самом деле, весь процесс переключения между докерами для доступа к элементам и функциям, в альтернативных комплектах легко доступным из самой панели инструментов, необъясним и не интуитивен.

Как правило, мы не осуждаем стиль работы только потому, что он не похож на наш, но для подобных приложений очень важным становится наличие документации. Однако, несмотря на обилие документации по некоторым приложениям, таким как *Sheets* и *Stage*, из самого приложения она не доступна.

Тем не менее, если вы согласны потратить время на освоение данного стиля работы или у вас старая машина, вы по достоинству оцените способность *Calligra* работать на маломощных системах.



Помощь и поддержка

К кому обратиться, когда не отличаешь вход от выхода?

Судя по характеру работы, выполняемой этими программами, нетрудно решить, что пользоваться ими довольно легко. Хотя это в основном справедливо, особенно для прямолинейных предложений *Google Docs*, в других приложениях может потребоваться некоторое содействие — например, когда вы пытаетесь выяснить, как установить плагины на *WPS Writer*, или есть ли они вообще!

Хотя все приложения кроссплатформенные, версия Linux для *WPS Writer* не привлекает такого же внимания, как ее коллега из *Microsoft*. Вот почему в официальной документации вы можете найти решения проблем, не имеющих отношения к версии Linux. Например, официальная справка-Help расскажет вам, как использовать функцию *Mail Merge* [слияние с почтой], но в Linux такой нет вовсе.

На сайте *OnlyOffice* не так много документации сверх инструкций по установке и базовому использованию, но в самой программе вы найдете гораздо больше полезной информации. Нажмите *File > Help* [Файл > Справка], и отобразится категоризированный список, зависящий от используемого приложения; имеется также панель поиска, если вы разыскиваете нечто конкретное.

В отличие от прочих, документации у *Calligra* нет вовсе. При запуске справочника из программы вы попадаете на страницу справочника KDE, которой тоже не существует. Когда вам, наконец, удастся отследить некоторую документацию по базе пользователей KDE, всё, что там перечислено — несколько отдельных функций. К счастью, вы можете постучаться в сообщество пользователей и получить советы и помощь на официальных форумах.

LibreOffice может похвастаться подробными руководствами пользователя, доступными из каждого компонента приложения. Помимо подробных руководств по документации, у проекта также есть сайт вопросов и ответов, под названием *Ask LibreOffice*.

ВЕРДИКТ

CALLIGRA SUITE	4/10	ONLYOFFICE	8/10
GOOGLE DOCS	8/10	WPS OFFICE	4/10
LIBREOFFICE	10/10		

Офисные комплекты вряд ли привлекут новых пользователей, если нет файлов справки, способных помочь с акклиматизацией.

LibreOffice

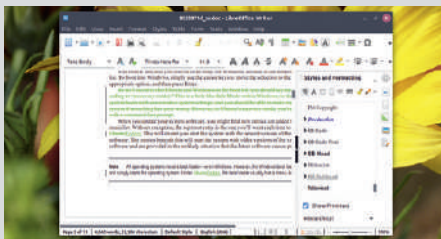
10/10

Это один из наиболее полных комплектов, и каждое из приложений-компонентов содержит подробную документацию, доступную из самого приложения.

В версии 6.0 проект также представил ленточный интерфейс, очень похожий на *Microsoft Office*, хотя по умолчанию он не включен. Если вам нравится ленточный интерфейс, то довольно приятна реализация в *WPS Office*; но, к сожалению, про *LibreOffice* так не скажешь.

При всей простоте его использования, наша единственная жалоба на *LibreOffice* — как плохо он справляется с большинством файлов DOCX. Другая заметная проблема заключается в том, что это пожиратель ресурсов, и если вы вынуждены работать на маломощной машине с ОЗУ только около 2 ГБ, вам, вероятно, больше подойдет одна из альтернатив.

Но *LibreOffice* восполняет эти недостатки, предлагая сотни расширений, обогащающих его функциональные возможности, и в этом отношении не имеет себе равных.



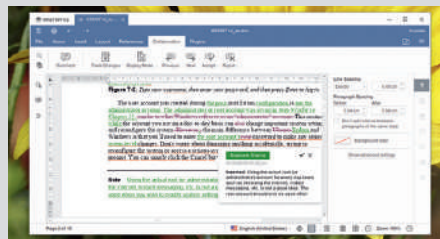
OnlyOffice

8/10

Мы здесь протестировали *OnlyOffice Desktop Editors* [настольные редакторы], автономный вариант офисного комплекта. Эта версия не предоставляет многих полезных функций, например, совместную работу. Желая испытать весь набор функций, вы можете попробовать различные облачные установки на бесплатной 30-дневной пробной версии. Хотя эти варианты могут подойти предприятиям или даже установке домашнего офиса [SoHo], для домашних пользователей они слишком дороги и обременительны.

В отличие от других комплектов, объединяющих отдельные приложения, *OnlyOffice* предоставляет единый интерфейс. Можно в любой момент решить, хотите ли вы работать с презентацией, таблицей или документом, либо со всеми тремя, которые открываются на отдельных вкладках в интерфейсе.

Впрочем, обычным домашним пользователям *OnlyOffice* предоставляет большинство стандартных функций, ожидаемых от трех наиболее распространенных производственных приложений.



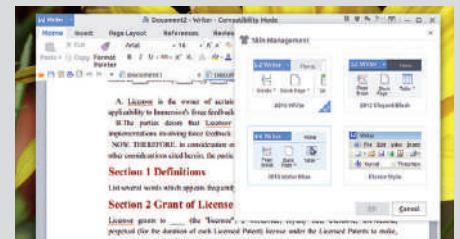
WPS Office

7/10

И *OnlyOffice*, и *WPS Office* предлагают только электронные таблицы, текстовые редакторы и программы для презентаций, тогда как другие комплекты предоставляют также приложение для рисования или инструмент управления базой данных. Из этих двух нам больше нравится *OnlyOffice*, потому что по сравнению с *WPS Office* проект более зрелый, ведь ему повезло выпустить релиз Linux.

При малосильной команде, работающей над редакцией Linux, мы также не уверены, будет ли проект по-прежнему доступен в будущем. Также неизвестно, представит ли он когда-либо некоторые из своих лучших функций, таких как совместная работа, которые пока ограничиваются версией для Windows.

Отсутствие поддержки ODF также разочаровывает и может оттолкнуть многих пуританов открытого исходного кода. Но его реализация ленты впечатляет, и мы бы даже предположили, что этому у *WPS Office* может поучиться даже сам *LibreOffice*.



Другие инструменты

Что еще можно сделать с помощью этих программ?

Текстовый процессор, электронная таблица и приложение для презентаций — три наиболее часто используемых инструмента в офисном комплекте. Наша подборка комплектов для этого *Сравнения* включает и другие приложения.

Forms [Формы] из *Google Docs* можно использовать для быстрого создания наиболее распространенных бланков, таких как контакт, RSVP и приглашение на вечеринку. Шаблоны предоставляют широкий выбор наиболее часто используемых в разных категориях, таких как Personal [Личные], Word (заявление о приеме на работу, отзывы о событиях) и Education [Образование] (оценка, рабочая таблица). Можно также использовать *Google Drawings* для создания блок-схем и карт памяти, но не для художественных работ.

Calligra предоставляет вам *Karbon*, приложение для рисования векторной графики, но инструмент уже не поддерживается. Если вы заинтересованы в графическом приложении для создания баз данных, можете попробовать свои силы в *Xexi*. *Calligra Plan* — это инструмент управления проектами, пригодный даже для управления умеренно крупными проектами.

Как и *Xexi* от *Calligra*, *LibreOffice Base* — графическое приложение для управления базами данных, похожее на *Microsoft Access*. *Base* может

похвалиться различными мастерами, помогающими и новым, и умелым пользователям создавать таблицы, запросы, формы и отчеты. Также предлагаются готовые таблицы для отслеживания активов, клиентов, заказов на продажу и счетов, чтобы упростить быстрое развертывание. Обеспечена встроенная поддержка для большинства широко используемых механизмов баз данных, таких как *MySQL*, *MS Access*, *PostgreSQL* и др. Для сравнения, *Xexi* от *Calligra* не поддерживает *MS Access* и менее развит, чем *Base*.

В отличие от других инструментов, в *WPS Office* и *OnlyOffice* есть только три основных компонента, и нет ни слова о том, будут ли они когда-либо предоставлять приложение для рисования или базы данных.

ВЕРДИКТ

CALLIGRA SUITE	9/10	ONLYOFFICE	5/10
GOOGLE DOCS	7/10	WPS OFFICE	5/10
LIBREOFFICE	9/10		

Оценки здесь отражают только количество инструментов, предлагаемых в каждом наборе, а не удобство их использования.

Сотрудничество

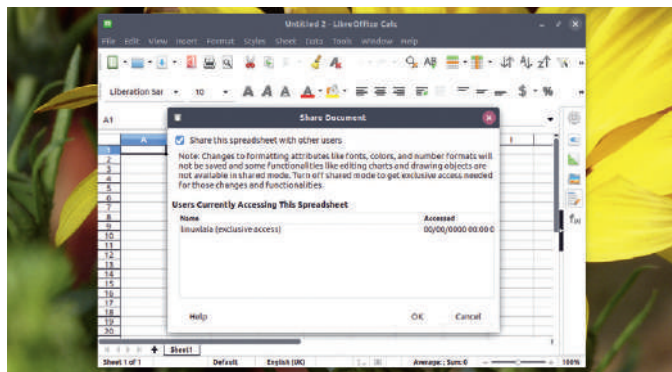
Каковы ваши варианты, когда надо разделить рабочую нагрузку?

Основным фактором популярности офисных онлайн-комплектов является то, что они позволяют нескольким пользователям сотрудничать и работать над одним и тем же документом одновременно. И хотя автономные офисные комплекты продолжают набирать обороты, в этой сфере, естественно, царят сетевые офисные комплекты наподобие вездесущих *Google Docs*.

Google Docs позволяют в режиме реального времени осуществлять по-символьное взаимодействие в инструментах *Google Docs*, *Google Sheets*, *Google Slides* и *Google Drawings*. Когда несколько человек работают над одним и тем же документом в одно и то же время, они все могут видеть изменения, внесенные другими. Эти параметры совместной работы работают в сочетании с параметрами общего доступа, позволяющими устанавливать уровни доступа к файлам, и вы сумеете контролировать, кто видит и редактирует ваши файлы. Как и в случае с другими функциями, можно использовать другой сервис Google, *Google Groups*, чтобы делиться документом с группой людей одним щелчком мыши.

LibreOffice Calc позволяет обмениваться документами с одновременным доступом на запись для нескольких пользователей. Если каждый из пользователей добавил свое имя на странице *User Data* [Данные пользователя] *LibreOffice* в разделе *Tools > Options* [Инструменты > Параметры], довольно просто отследить вклад, внесенный разными пользователями.

В *Calligra Suite* подобная функция до сих пор не реализована, и нет никаких упоминаний о ее внедрении в будущих версиях. На сайте проекта отсутствует



Совместная работа в *LibreOffice* ограничена *Calc*, поскольку другие приложения не допускают одновременного доступа на запись.

дорожная карта для будущих выпусков, поэтому нельзя предсказать, когда эта функция может быть введена — и произойдет ли такое вообще.

С *OnlyOffice* вы получаете визуальную подсказку, информирующую вас об отрывках, над которыми работают ваши сотрудники, и вы можете поговорить с ними, чтобы обсудить идеи. Но эта функция недоступна в оффлайн-редакторе, и для доступа к ней надо обратиться к облачной функции.

ВЕРДИКТ

CALLIGRA SUITE	N/A	ONLYOFFICE	N/A
GOOGLE DOCS	10/10	WPS OFFICE	N/A
LIBREOFFICE	8/10		

Совместная работа – еще одна функция, пока не реализованная в Linux-версии *WPS Office*.

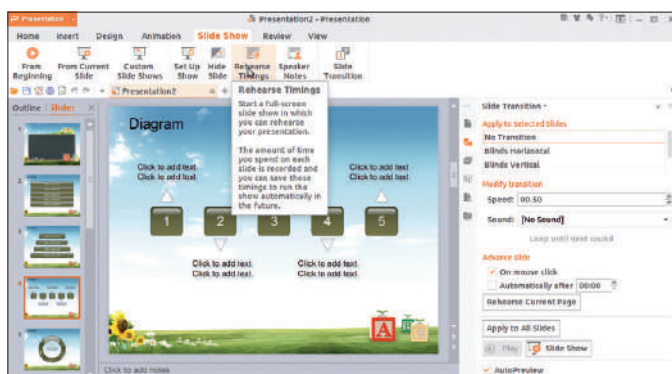
Презентация

Хорошее слайд-шоу требует большего, чем просто выбор маркеров списка.

Как и в случае с текстовыми процессорами, программа для презентаций *Calligra Stage* требует некоторого привыкания. Различные параметры форматирования, скажем, определение анимации и переходов, переданы в отдельный докер в правой части экрана. Добавление фигур аналогично ограничивается докером *Add Shapes*.

Google Slides предлагает выбор из нескольких шаблонов, которые помогут вам начать работу. К каждому слайду можно добавить изображения, фигуры и диаграммы и настроить эффекты перехода. Также можно выбирать различные цветовые схемы и темы для создания динамических слайдов. Предлагается большое количество дополнений, которые означают, что вы можете добавить на слайд еще больше контента. Например, вы можете вставлять в свои слайды бесплатные фотографии HD, используя дополнение *Unsplash*. Дополнения разделены на категории — *Education* [Образование], *Business Tools* [Бизнес-инструменты] и *Productivity* [Производительность].

OnlyOffice позволяет выбирать дизайн фона. Можно вводить фигуры, графики, текстовые изображения и рисунки, а также добавлять к слайдам переходы. Уникальной особенностью является встроенный редактор фотографий. Он может применяться для управления яркостью, контрастностью и резкостью любого изображения, вставленного в слайд. Вы также можете добавить к изображению рамку, обрезать его, изменить размер или ориентацию и сделать много других правок. Фоторедактор работает медленнее, чем специальные инструменты для редактирования изображений вроде *GIMP*. Но он свое дело делает, и удобство редактирования изображений из самого инструмента невозможно переоценить. Вы также можете вставить в слайд видео с YouTube, щелкнув по вкладке *YouTube* в разделе *Plugins* [Плагин]



Функция репетиции тайминга в *WPS Presentation* позволяет сделать в презентации автоматический переход для каждого слайда.

и указав URL-адрес видео. Тем не менее, при запуске слайд-шоу видео не работает, и всё, что вы видите, это его изображение, без кнопки или какой-либо опции для запуска воспроизведения.

WPS Presentation и *LibreOffice* предлагают все функциональные возможности, полагающиеся инструменту презентации, такие как добавление анимации, переходов, дизайнов и т. д. И *WP Presentation*, и *Calligra Stage* позволяют выбирать размер слайда (либо стандартный экран 4:3, либо широкоэкранный вариант 16:9).

ВЕРДИКТ

CALLIGRA SUITE	5/10	ONLYOFFICE	10/10
GOOGLE DOCS	9/10	WPS OFFICE	10/10
LIBREOFFICE	10/10		

Самым большим недостатком *Calligra* является его нетрадиционный интерфейс.

Офисные комплекты

Вердикт

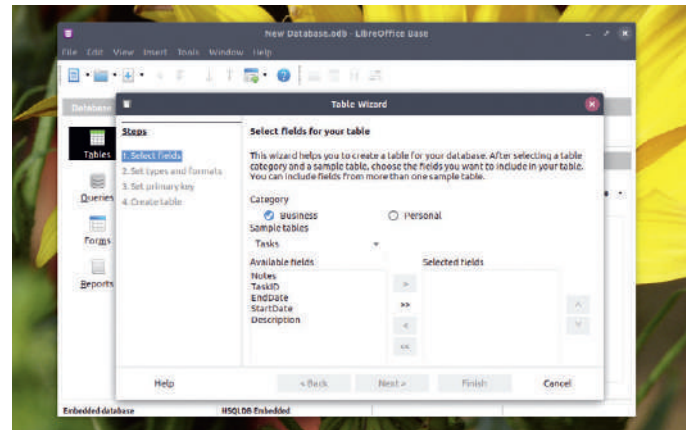
Выбор объектов *Сравнения* за последние несколько выпусков привел нас к выводу, что, возможно, все ваши потребности одним приложением удовлетворить не получится. Это было справедливо для графических и видеоредакторов, и офисные комплекты не стали исключением. При наличии достаточных ресурсов вы можете в одном приложении готовить презентации, а для своих электронных таблиц полагаться на другие программы.

Исключая попавшие на пьедестал, оставшиеся офисные комплекты нас в одинаковой мере и очаровали, и разочаровали. *WPS Office* — высокопроизводительный экземпляр, и мы бы оптимистично посоветовали пользователям дать ему шанс, но отсутствие поддержки ODF, не говоря уже о неизвестности, когда она будет доступна, решило его судьбу в данном *Сравнении*. Не обольщайтесь тем, что он быстр и отзывчив и с ним весело работать. Отсутствие многих полезных функций в редакции Linux весьма серьезно его порочит.

Calligra Office, несмотря на высокую производительность и впечатляющий набор функций, тоже вызвал разочарование, но главным образом из-за его интерфейса. Если вы способны освоить его эзотерический рабочий процесс и не возражаете против поисков своего собственного пути методом проб и ошибок, в нем нет ничего плохого. Но если вы планируете запускать *Calligra* на маленьком экране, скажем, на 14-дюймовом ноутбуке, у вас едва хватит места для работы над документом или слайдами, так что ему тоже нечего делать на пьедестале.

Хотя *OnlyOffice* и *LibreOffice* поддерживают проприетарные форматы *Microsoft Office*, *OnlyOffice* справляется с отображением документов гораздо лучше. Однако, в отличие от других комплектов, он не располагает ни базой данных, ни приложением для рисования, и способен предоставить только электронную таблицу, текстовый процессор и приложение для презентаций. Кроме того, оффлайн-версия для настольных компьютеров не имеет всех тех функций, которые предлагаются в облачных вариантах, и это тоже огорчает. Тем не менее, в ограниченном наборе предлагаемых приложений *OnlyOffice* работает довольно хорошо и менее ресурсоемок, чем *LibreOffice*. Если вам незначительно заботится о сотрудничестве с коллегами, советуем вам обязательно попробовать его.

Google Docs и *LibreOffice* отличаются очень мало, за исключением того факта, что для работы первого требуется подключение к Интернету, а второй — решение для настольных компьютеров. **LXF**

**1^е место****LibreOffice****10/10**

Версия: 6.1.2 **Сайт:** www.libreoffice.org **Лицензия:** MPL 2.0 и др. свободные
Этот зрелый офисный комплект превосходит практически во всем.

2^е место**Google Docs****9/10**

Версия: — **Сайт:** <https://docs.google.com> **Лицензия:** Проприетарная
Делает сотрудничество плевым делом.

3^е место**OnlyOffice****8/10**

Версия: 5.1.29 **Сайт:** www.onlyoffice.com **Лицензия:** AGPLv3
Впечатляющая поддержка проприетарных форматов, и работает как обещано.

4^е место**Calligra Suite****6/10**

Версия: 3.1.0 **Сайт:** www.calligra.org **Лицензия:** GPL
Столь же многофункционален, как LibreOffice, но его интерфейс очень громоздок.

5^е место**WPS Office****5/10**

Версия: 10.1.0 **Сайт:** <http://wps-community.org> **Лицензия:** Проприетарная
Пуристам не понравится проприетарная лицензия и отсутствие поддержки ODF.

» РАССМОТРИТЕ ТАКЖЕ

Других комплектов с открытым исходным кодом, кроме участников нашего *Сравнения*, на свете нет. Если ни один из них вам не подходит, ознакомьтесь с *Apache OpenOffice*. На нем основан *LibreOffice*, но тот продолжает вводить новшества, а вот *OpenOffice* в последние годы стагнирует. Еще один бесплатный комплект — *FreeOffice* на базе *Softmaker Office 2018*, но, как и *WPS Office* и *OnlyOffice*, он содержит только электронную таблицу, текстовый процессор и инструмент для презентаций.

Вам нужна облачная альтернатива? Кроме *Google Docs*, их несколько, в том числе проприетарный *Zoho Office Suite* или *Tiki* с открытым исходным кодом для самостоятельного развертывания. Другой приемлемый вариант — *Feng Office Community Edition*, под лицензией AGPLv3. А если ни один из них вам не понравится, обратитесь к отдельным приложениям: *Gnumeric* или *Pyspread* — для электронных таблиц, *Abiword* и *Sozi* — для обработки текста и создания презентаций соответственно.

ВИРТУАЛИЗАЦИЯ

Джонни Бидвелл учится работать с контейнерами и VirtualBox, которые позволяют ему запускать Linux в любое время, в любом месте, где угодно...



Виртуализация существует со времен мейнфреймов 1960-х, где работа одной программы была отделена от другой. Позднее IBM SP-40 ввел понятие гипервизора и возможность одновременного функционирования нескольких операционных систем.

Всерьез виртуализация стала применяться в середине 2000-х, когда появились 64-битные процессоры с явными функциями для более эффективного запуска гостевых ОС. Возможность виртуализации компьютеров (в теории) сделала жизнь сисадминов легче. Можно было делать моментальные снимки целых систем, припасать их резервные копии и восстанавливать не сложнее, чем файлы. Можно было тестировать критические обновления в виртуальной песочнице, солидно сокращая вероятность проблем при их развертывании в физических системах.

Несколько виртуальных машин (VM, VM) могли сосуществовать в одной системе и при этом быть практически полностью изолированными друг от друга, что повышало надежность и безопасность. Домашние пользователи могли использовать преимущества этого "Лев-лих", не рискуя повредить свою основную ОС.

С тех пор оборудование продвинулось еще дальше, и теперь вы можете передавать на VM целые устройства. Это дает возможность, помимо всего прочего, запускать VM Windows с помощью собственной специальной разогнанной видеокарты.

Вслед за VM появились контейнеры, которые вместо реализации целой операционной системы используют ядро хоста и содержат только то, что необходимо для запуска определенного сервиса или набора сервисов. Это позволяет им поставляться с конкретными библиотеками, избегая

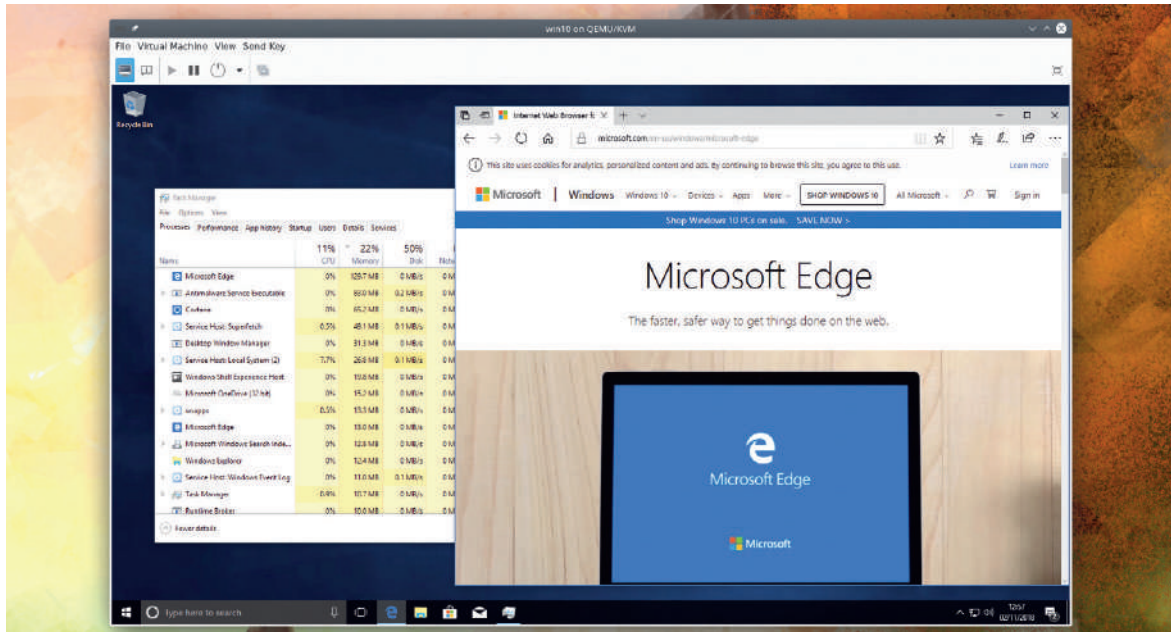
проблемы конфликта версий при установке программы на другой компьютер. Благодаря этому они более переносимы, чем виртуальные машины, и до некоторой степени предлагают тот же уровень изоляции.

Самым ярким примером здесь является Docker, однако контейнеры не ограничиваются одной компанией. Snaps и Flatpaks (см. наш учебник на стр. 64) используют технологию контейнеров и уже повсюду применяются вместо традиционных пакетов. Это означает, что разработчикам становится легко создавать и быстро поставлять не зависящие от дистрибутива пакеты, чтобы включить свои программы в следующий релиз, не дожидаясь составителей пакетов.

Поэтому если вы хотите попробовать супер-пупер Linux в безопасной среде или реализовать целый стек LAMP на своем сервере с помощью одной команды, читайте дальше!

О виртуализации — на пальцах

Вкратце — о технологии, делающей виртуализацию возможной и интересной.



Возможность запуска Microsoft Edge — одна из причин использовать виртуальную машину, только не самую обычную.

Виртуальная машина может работать на любом старом компьютере. В конце концов, люди работают с эмуляторами уже не один год. Но имитация посторонней архитектуры — дело непростое, так что те эмуляции старались сосредоточиться на куда менее мощных компьютерах, чем хост.

Однако, эмулируя компьютер, архитектурно близкий нашему хосту, мы можем сэкономить время и силы. Вместо того, чтобы эмулировать CPU и прочие устройства, мы можем передавать этим устройствам инструкции. И чем больше мы это делаем, тем больше смещаемся от эмуляции к виртуализации.

Для правильной виртуализации нужен гипервизор, который размещается поверх VM и сортирует вызовы между гостем и хостом. Нам не надо, чтобы наш гипервизор бездельничал, иначе он будет бессмысленным и позволит гостю вытворять с хостом нежелательные вещи, но и не надо, чтобы он чересчур надрылся.

Примерно с 2006 г. новые функции CPU (Intel VT-x и AMD-V) позволили разрабатывать элегантные гипервизоры, идеально отвечающие требованиям. В Linux есть KVM, в Windows есть Hyper-V, потом еще есть гипервизор Xen, который работает на привилегированном виртуализированном домене ОС (dom0, где может работать любая ОС). Менее привилегированные (domU) VM используют dom0 для доступа ко всему своему оборудованию, а гипервизор поверх всего обеспечивает полную изоляцию всего. Сосредоточенная на безопасности Qubes OS использует виртуализацию Xen, чтобы обеспечить разграничение приложений. Дальнейшие инновации CPU (VT-d от Intel и AMD-Vi) дают VM прямой доступ к периферии. Именно эта магия вместе с Open Virtual Machine Firmware (OVMF) и чудесами драйвера VFIO позволяет нам передать целую видеокарту VM Windows 10 и заставить ее работать практически с родной скоростью, запуская все эти игры, которые пока что не работают нормально в Steam Play.

Виртуализация — это также отличный способ резервного копирования физического сервера. Обзаведясь виртуальным

зеркалом своего сервера, вы можете сделать его моментальный снимок и экспериментировать с разными настройками или обновлениями, которые было бы неразумно применять в промышленной эксплуатации. Если они не сработают, вы легко вернетесь к прежнему состоянию и сможете попробовать снова. Если откажет ваш физический сервер, то будет легко (в принципе) физически резервную копию на новом оборудовании. В порядке альтернативы, вы просто развернете копию этой VM — в облаке полно виртуальных машин.

» КОНТЕЙНЕРЫ ПРОТИВ VM

Точно так же, как виртуализация захватила мир штурмом в конце 2000-х, примерно пять лет спустя вперед вышли контейнеры. У них есть нечто общее по части изоляции систем и независимости от хоста. Но на самом деле контейнеры очень сильно отличаются от VM.

Для начала, контейнеры не стремятся имитировать весь компьютер целиком. Они не включают полную ОС и используют ядро хоста, поэтому более мобильны. Кроме того, с ними сложнее разобраться. Внешне они похожи на chroot (либо jail в BSD былых времен), но если погрузиться глубже, то вас ждет целая сеть из жаргона и сложностей.

В современном смысле «контейнеры» обозначают целый набор технологий — несколько неявно конкурирующих сред исполнения и разнообразных форматов хранения. Было бы просто ассоциировать их с одной технологией — скажем, с Docker — но это подход лишь одной компании. LXC, LXD, CRI и CoreOS rkt — другие инструменты, применяемые вместо или вместе со стекком Docker.

Контейнеры отлично подходят для распространения приложений, которые в ином случае было бы довольно сложно настроить. Обо всём, что связано с получением непонятных пакетов Python или определенных версий библиотек, позаботятся те, кто занимается поддержкой и использованием образа пакетов.

VirtualBox

Плавнo вступаем в мир виртуальных машин через получившую говорящее имя *VirtualBox*.



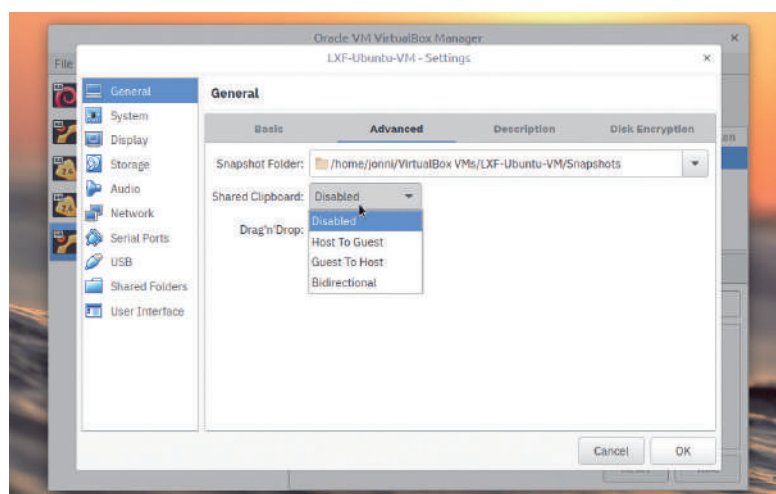
Создать свою первую виртуальную машину легко: труднее всего принять решение о выборе платформы. VMWare и *VirtualBox* предлагают бесплатные инструменты для всех операционных систем (включая Linux). В Windows можно использовать Hyper-V, который, со своей новой функцией Quick Create, может развернуть экземпляр Ubuntu быстрее, чем вы скажете: «Патентная практика Microsoft развивается в полном соответствии с взглядами компании на Linux и открытый код

в целом». Мы будем использовать *VirtualBox*, поскольку она имеет лицензию GPL версии 2 (за исключением Extension Pack, который предлагает такие функции, как транзитная передача USB [USB passthrough] и устройства NVMe — не путайте их с Guest Additions [добавления для гостевой системы], которые теперь тоже выходят под GPL) и одинаково выглядит на всех ОС. Следуйте пошаговому руководству внизу, чтобы приступить к работе, или, если вы уже в курсе дела, читайте дальше и познакомьтесь с некоторыми из ее менее известных функций.

Виртуальные настройки

Предположим, что вы уже ознакомились с нашим руководством, загрузили средство установки и установили Ubuntu на свою *VirtualBox*. Если нет, вам, вероятно, уже стоит это сделать — как и в реальном мире, live-ОС намного медленнее, чем установленные в виртуальном мире. Когда запустится ВМ, вы увидите сообщение об интеграции курсора мыши. Это жутко полезная функция, которая обеспечивает плавное движение мыши на хосте и на госте. Если вы используете гостевую ОС без поддержки этой функции, воспользуйтесь клавишей «правый Ctrl», чтобы вызвать курсор мыши из заточения в гостевом окне.

Настройки *Virtualbox* по умолчанию отлично работают для установки большинства гостевых ОС Linux, но всегда есть место для улучшений. Первое, что вы, вероятно, заметили, это что у ВМ низкое разрешение и что окна перемещаются и изменяют размеры нестерпимо долго. Причина в том, что у нашей виртуальной видекарты мизерные 16 МБ памяти и никаких функций разгона. Чтобы это исправить, надо выключить ВМ. Затем выберите ВМ из списка слева и нажмите на кнопку Settings [Настройки]



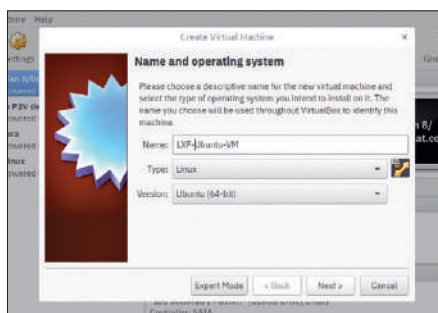
Опция общего буфера предоставляет роскошь копирования и вставки и в хосте и в госте. Оговорка: без обобществления буфера средней кнопки X, только буфера рабочего стола.

УСТАНОВКА И НАСТРОЙКА VIRTUALBOX



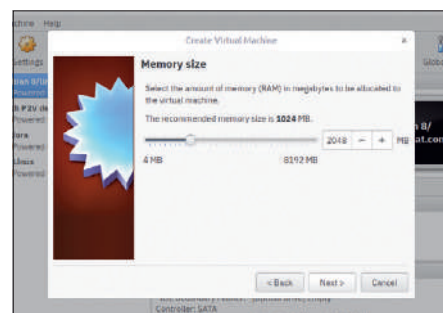
1 Скачайте VirtualBox

Если вы работаете в macOS или Windows, перейдите на www.virtualbox.org и следуйте по ссылке для скачивания. Затем запустите скачанный бинарник для установки. Если вы работаете в Linux, то *VirtualBox* почти наверняка есть в репозиториях вашего дистрибутива. В Ubuntu для установки используйте `sudo apt install virtualbox`. Запустите *VirtualBox* из меню Applications [Приложения] (или как там ваша операционка его называет).



2 Создайте новую ВМ

Нажмите на кнопку New в верхней левой части интерфейса *VirtualBox*. Мы создадим виртуальную машину Ubuntu 18.10; мы назвали свою LXFUbuntu-VM, но вы можете дать ей любое имя, какое вам захочется. Установите Machine Type в Linux, а окно Version — в Ubuntu (64-бит), если только вам не требуется 32-битная ВМ. Если вы уверены в своих познаниях, нажмите на кнопку Expert Mode [Режим Эксперта], а в противном случае — сразу Next [Далее].



3 Выберите объем памяти

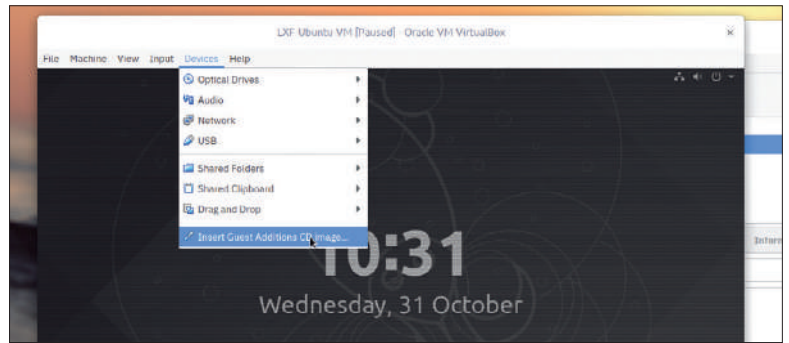
Сколько оперативной памяти вам следует отвести своей виртуальной машине, зависит от того, каким образом вы хотите ее использовать, и от количества памяти на машине хоста. Современные рабочие столы и приложения (особенно браузеры) охотно слопают 2 Гб, тогда как загрузка в консоль требует очень мало. Если вы отведете своей виртуальной машине слишком много памяти, то, естественно, пострадает производительность хоста.

на панели инструментов и перейдите в раздел Display [Отображение]. Здесь вы можете определить спецификации виртуальной видеокарты. Для более высокого разрешения экрана при большей глубине цвета нужно больше видеопамати. С настройками по умолчанию вместо видеопамати используется ОЗУ системы, так что вы, вероятно, сможете выделить здесь как минимум 64 МБ. На самом деле вполне возможно настроить ее побольше, чем позволяет ползунок, употребив инструмент командной строки *VboxManage*. Но об этом — позднее.

Все современные рабочие столы, несмотря на свое сидение на двумерной поверхности, используют для перемещения окон и создания под ними симпатичных теней какой-либо вид 3D-графики (OpenGL, OpenGL ES или недавно появившийся Vulkan). Нажав на окно Enable 3D acceleration [Включить 3D-графику] мы разрешаем нашей ВМ передавать эти базовые элементы более или менее напрямую на видеокарту хоста, а также напрямую получать доступ к его видеопамати. Так что если вы используете встроенную графику (или очень старую видеокарту с небольшой VRAM), убедитесь, что вы не отвели слишком много. Есть соблазн нажать также на окно 2D-графики, но это скажется только для графики DirectDraw в гостевых ОС Windows.

Мы также можем разогнать CPU. Настройки по умолчанию отводят на работу с гостевой ОС только один поток; это довольно мучительно для мира, где многозадачность принимается как должное. Поэтому выберите раздел System [Система] слева и вкладку Processor [Процессор]. Гостю можно отвести столько виртуальных CPU, сколько потоков способен запустить хост (поэтому удвойте число ядер, если ваш процессор поддерживает HyperThreading [гиперпоточность] или как там это сейчас называется у AMD). И опять, если вы отведете гостю слишком много CPU, пострадает хост, что в свою очередь вызовет проблемы у гостя. Универсальным правилом будет не отводить более половины наличных ресурсов CPU. Можно также задать планку выполнения, чтобы ВМ не могла исчерпать все ресурсы ядра на хосте, это бывает полезно для некоторых разнужданных задач.

Некоторым операционным системам требуется выбрать Enable PAE/NX, что даст возможность 32-битным ВМ получать доступ к более чем 4 ГБ памяти. Если вы настраиваете Ubuntu Server



Многие дистрибутивы Linux сейчас включают гостевые добавления, в т.ч. полезные функции, описанные здесь. Но иногда вам придется устанавливать их на госте вручную.

guest (что и указали в окнах Machine Type и Version), это происходит автоматически. Перейдя на вкладку Acceleration [Ускорение], вы должны увидеть, что окна VT-x и Nested Paging отмечены. Для нашего виртуального жесткого диска официально не рекомендуется отмечать Use Host I/O Cache [Использовать кэш ввода-вы-

ИСПОЛЬЗУЙТЕ РЕСУРСЫ МУДРО

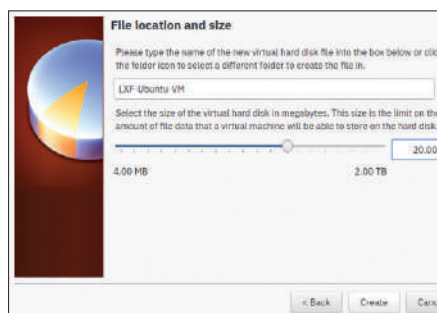
«Если вы отведете гостю слишком много CPU, пострадает хост, что в свою очередь вызовет проблемы у гостя.»

вода хоста], отключая специальный кэш *VirtualBox*, чтобы использовать кэш ОС. Однако есть сообщения, что тогда ускорится выполнение задач с активным использованием ввода/вывода, в частности, установка ОС и обновление пакетов. Мы не будем рассказывать здесь о настройке VM Windows, но если вам это нужно, то вам, вероятно, придется немного поэкспериментировать, чтобы всё работало хорошо.



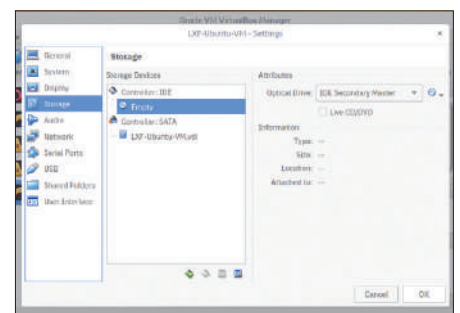
4 Виртуальные жесткие диски

Нашей ВМ нужно некое постоянное хранилище. Современные дистрибутивы Linux склонны капризничать, требуя установки не менее чем на 10 ГБ, и если место позволяет, мудро было бы выделить больше. Чтобы начать процесс, выберите Create a virtual hard disk now [Создать виртуальный жесткий диск сейчас]. Если у вас объемистый раздел данных, стоит хранить ВМ там, чтобы не переполнять вашу домашнюю директорию.



5 Настройка виртуального хранилища

Выберите хранилище VDI по умолчанию. Эти файлы можно исправить или установить размер динамически (до заданного максимума). Динамическое хранилище, разумеется, намного более гибкое, и в наши дни вы вряд ли заметите повышение издержек ввода/вывода, так что используйте его. Вы можете указать местонахождение файла с помощью крошечного значка справа, в ином случае он отправится в `~/VirtualBox VMs/`.



6 Загрузите свою ВМ

Чтобы загрузить свою машину, вам нужен ISO-образ. Либо скачайте вариант Ubuntu на свой выбор, либо скопируйте его с нашего DVD. Перейдите в Settings > Storage [Настройки > Хранилище], укажите оптический диск, нажмите на выпадающее окошечко для диска справа и выберите Choose Virtual Optical Disk File [Выбор файла виртуального оптического диска](sic). Укажите свой ISO и нажмите OK, а затем на зеленую стрелку Start. Процесс пошел!

QEMU и Libvirt

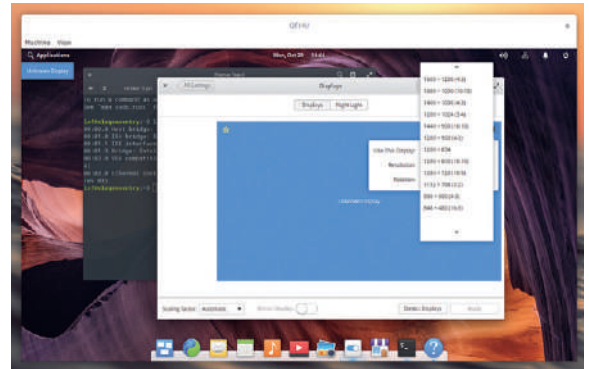


Узнайте о более продвинутых способах и средствах управлять своими VM...

V *VirtualBox* — отличный выбор для начала работы с виртуализацией: вы получаете броское название, дружелюбный интерфейс и кросс-платформенность. Однако в Linux есть и другие опции.

Многие активно избегают *VirtualBox* из-за ассоциаций с Oracle (эту компанию многие считают склочным корпоративным монстром), купившей эту программу вместе с компанией Sun в 2010 г. *QEmu* — инструмент, куда более отвечающий стандартам Linux (ну вы поняли: нескончаемые опции командной строки, заумные файлы настроек и неумолимые сообщения об ошибках). Сам по себе *QEmu* — это эмулятор, но в сочетании с мощью KVM он превращается в первоклассный инструмент виртуализации.

Кроме того, *Libvirt* предоставляет платформу, упрощающую управление VM *QEmu* (и другими). Она берет на себя проблемы определения виртуальных сетей и хранилища и позволяет управлять и получать доступ к VM удаленно. Более подробное объяснение этих трех уровней вы найдете в комментариях гуру виртуализации Берто Гарсии [Alberto Garcia] в **LXF234**. *Libvirt*, в свою очередь, может управляться превосходным *virt-manager*, который дает доступ к большинству опций *QEmu* через понятный и дружелюбный графический интерфейс. Но давайте сначала



Устройство QXL делает настольную VM гораздо приятнее, хотя мы можем пойти еще дальше.

обойдемся старым способом. Чтобы установить *QEmu* (а также элементы *Libvirt*, которыми мы воспользуемся позднее) в Ubuntu 18.04 (и его производных), сделайте следующее:

```
$ sudo apt install sudo apt install qemu-kvm libvirt-clients libvirt-daemon-system bridge-utils virt-manager
```

» ЗНАКОМЬТЕСЬ: VIRGL

Virtual Machine Manager GUI позволяет нам выбрать видеоустройство QXL одним нажатием кнопки. Однако он также позволяет нам пойти еще дальше и соответствовать (или даже превзойти) возможности 3D-графики *VirtualBox*.

Гуру графики Collabora Дэниел Стоун [Daniel Stone] упомянул в нашем интервью (см. **LXF243**) о проекте VirGL (произносится "Virgil") и его использовании в Crosvm, гипервизоре Chrome OS. VirGL умеет передавать вызовы OpenGL прямо в GPU хоста, давая гостю возможность 3D-ускорения. Включить VirGL из *virt-manager* (см. стр. 37) на удивление просто: в разделе Video просто выберите Virtio как модель и отметьте окно 3D-графики. Затем в разделе Display Spice измените Listen type на None (VirGL не работает по TCP) и отметьте окно OpenGL. Возможно, вам также понадобится вручную выбрать ваше видеоустройство в окне, которое появится ниже, даже если у вас оно уже есть.

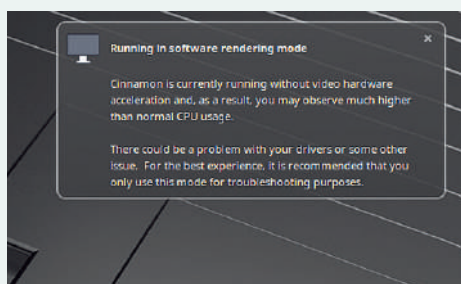
В VM вы можете проверить, активен ли VirGL, запустив

```
$ dmesg | grep drm
```

и поискав

```
[drm] virgl 3d acceleration enabled
```

Если вам нужна еще большая производительность графики, загляните в наше руководство по передаче GPU на стр. 40.



Вам уже не придется снова видеть это предупреждение при запуске Mint в VM.

Моя первая VM QEmu

Прежде чем добыть виртуальную машину, надо обзавестись виртуальным хранилищем. Простейший способ сделать это — использовать *qemu-img*, вот так:

```
$ qemu-img create -f qcow2 lxf.qcow2 20G
```

Создается динамический образ QCOW2, так что в данном случае он не поглотит сразу 20 ГБ на диске. Позаботьтесь создать его в таком месте, где он сможет спокойно расти в объеме, поскольку если на вашем разделе root закончится место, могут произойти всяческие ужасы. Можно также использовать «сырой [raw]» образ со статическим объемом. В некоторых ситуациях это обеспечит преимущества в производительности и не создаст ситуации растраты ресурсов хранилища при добавлении к ним данных, но зато будет не так стабильно при сбоях вашей VM.

Мы определили особенности нашей VM посредством опций командной строки. Как вы понимаете, эти опции способны заполнить собой целый выпуск *Linux Format*, так что мы сосредоточимся сначала на основных. В нашем примере мы воспользуемся ISO Elementary OS (который вы найдете на нашем DVD, но лучше скопировать его на более быстрый диск), хотя ничто не мешает вам загрузить ОС по своему выбору. Столь же спокойно вы можете указать */dev/cdrom* вместо файла ISO, если хотите использовать реальный носитель. Вот как запустить нашу VM и отвести ей 3 ГБ памяти (разумное количество для настольного Linux):

```
$ qemu-system-x86_64 -cdrom elementaryos-5.0-stable.20181016.iso -drive file=lxf.qcow2 -enable-kvm -m 3G
```

Обратите внимание на опцию *-enable-kvm*. *QEmu* — это всё-таки эмулятор, и если мы не разрешим использовать KVM, всё будет эмулироваться, т.е. работать очень медленно. Ну очень медленно. Исследовав среду live (с помощью опции Try Elementary), вы должны обнаружить, что работать в Интернете можно, а вот пропинговать хосты нельзя.

QEmu по умолчанию устанавливает основное сетевое соединение NAT, что прекрасно подходит в большинстве случаев,

но не позволит вам, например, зайти в VM через SSH. Загляните во врезку вверху справа, чтобы узнать более расширенные сетевые опции. Мы также можем пойти дальше и установить Elementary OS, как если бы мы делали это в реальности. По завершении установки выберите Restart [Перезапустить]. Виртуальный оптический носитель извлекается, и вы сможете зайти в систему с данными аутентификации, которые использовали для установки. Вы также должны заметить, что с носителем live всё малость ускорилось, но, вероятно, всё равно далеко от установки на чистое железо.

По умолчанию *QEmu* устанавливает базовую видеокарту VGA, которая в нашей установке Elementary по каким-то причинам справилась только с 800×600. Она должна быть в состоянии обрабатывать более высокие разрешения, но есть опции получше. Одна из них — виртуальное видеоустройство паравиртуализации QXL, которое предлагает 2D-ускорение и может (при соответствующей программе просмотра) использовать протокол SPICE, для еще более гладкой работы. Отключите VM и попробуйте запустить ее снова с устройством QXL:

```
$ qemu-system-x86_64 -drive file=/mnt/stor/VMs/lxf.qcow2
-enable-kvm -m 3G -vga qxl
```

Теперь, запустив *lspci* внутри гостя, вы должны увидеть примерно такую строку:

```
... Red Hat, Inc. QXL paravirtual graphic card
```

а заодно получить доступ к более высоким разрешениям через System Settings > Displays [Системные настройки > Мониторы]. Вы также можете добавить устройству QXL видеопамяти (по умолчанию это 16 МБ, чего достаточно для 1080p при 32-битной глубине цвета, но для 4K понадобится больше), приписав в конце предыдущей команды опцию `-global qxl-vga.vram_size=64MB`. Есть и другие настройки для работы с памятью QXL — `vgamem` и `ram` — но мы предоставим вам самим разбираться в тонкостях их отличий. Помните, что всё это не предоставляет реальной видеопамяти и не поможет вам с 3D-ускорением (не волнуйтесь, про него мы тоже расскажем).

С настройками по умолчанию наша VM эмулирует чипсет Intel 440FX с контроллером диска PIIX3 IDE и установленным эмулированным CPU. По нынешним стандартам это очень простое оборудование, что и неплохо: оно проще для эмуляции. Помимо видеоустройства QXL или драйвера VGA по умолчанию, можно также симитировать старинный Cirrus Logic GD5446 (который в *QEmu* был по умолчанию). Однако вместо эмуляции старых устройств гипервизор может сэкономить силы, создав облегченное виртуальное оборудование и перегоняя данные на соответствующее оборудование реального компьютера. Эта технология известна как паравиртуализация, а магия, позволяющая KVM применять

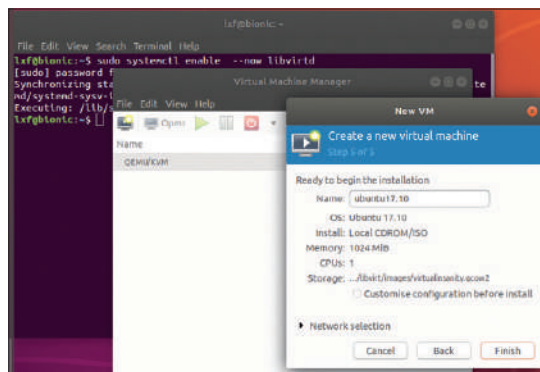
» VIRTUAL MACHINE MANAGER

Чтобы использовать *virt-manager*, нам надо, чтобы работал сервис *libvirt*, и поскольку эта функция, возможно, подсадит вас на VM, мы также можем настроить сервис на запуск при каждой загрузке:

```
$ sudo systemctl enable --now libvirt
```

Теперь вы должны иметь возможность запускать *Virtual Machine Manager* из меню приложений. Нажмите на кнопку New Virtual Machine, и мастер проведет вас по процессу создания VM. Выберите Local install media [Локальное средство установки], если не планируете чего-то другого. Сообщив *Libvirt*, какую ОС вы устанавливаете, вы можете настроить CPU и память. Затем вам понадобится настроить том виртуального хранилища. По умолчанию хранилища обитают в `/var/lib/libvirt/images`. Если вы хотите разместить их в другом месте, можете добавить директорию, нажав на зеленую кнопку + в диалоговом окне настройки хранения, и затем создать хранилище там.

И, наконец, остается выбор сетевых опций. Сеть NAT по умолчанию отлично подходит, но, возможно, вы захотите исследовать устройства vtap, если у вас есть требования по компиляции. Большинство этих настроек можно изменить позднее, но если вам надо настроить загрузку UEFI (например, если вы делаете транзит GPU), это делается до установки ОС, отметкой окна Customise [Настроить] перед нажатием Finish [Готово].



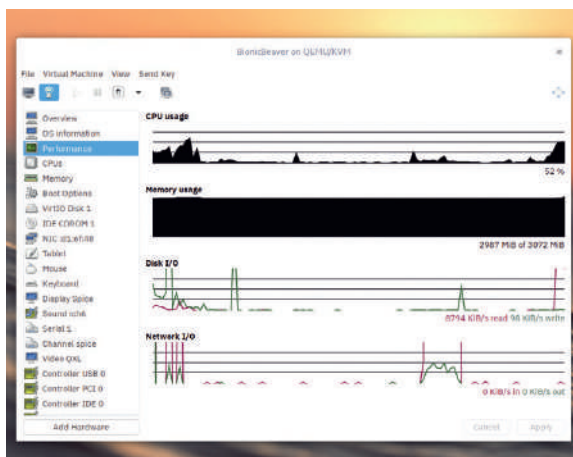
Это окно-флажок настройки Customise легко пропустить, но оно нам понадобится позднее.

ее, именуется VirtIO. Устройство QXL паравиртуализовано в том смысле, что оно обеспечивает доступ к рендерингу программ на машине хоста, так что оно вряд ли побьет рекорды скорости. Но жесткий диск VirtIO может предложить колоссальные преимуще-

ДВОЙНОЕ ВИРТУАЛЬНОЕ УДОВОЛЬСТВИЕ

«Сам по себе *QEmu* — это эмулятор, но в сочетании с мощью KVM превращается в инструмент виртуализации.»

щества по скорости; сетевой адаптер VirtIO может снизить время ожидания и ускорить пропускание по сети. Драйверы VirtIO включены во все дистрибутивы Linux, но для Windows драйверы вам придется устанавливать; найти их можно на <https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/>. Устройства VirtIO можно настроить из командной строки, но это быстро приведет к громоздким командам. Куда проще использовать *virt-manager* с графическим интерфейсом, работающий на *Libvirt*. Который мы весьма кстати успели установить.



По графикам на вкладке Performance [Производительность] удобно следить, чем занимаются ваши VM.



Передача GPU

Наслаждайтесь почти родной скоростью графики в ВМ благодаря магии транзитной передачи PCI. Это поэффектнее таскания кроликов из шляпы!

Не так давно Steam Play привлек усиленное внимание геймеров Linux, и не без причины. Он принес на платформу вагон игр, у большинства из которых шансы на официальное портирование были равны нулю. Однако многие игры, получившие официальное одобрение, работают в Linux хуже, чем в Windows, а у многих других есть куча самых разнообразных ошибок, которые вообще не дают в них нормально играть. Нет, не путайте, с ответвлением Proton Wine от Valve и спонсорством DXVK всё отлично, однако они не побудят закоренелых геймеров со всех ног помчаться на Linux. Даже родные порты Linux от Feral и Aspyr редко способны конкурировать по производительности со своими собратьями в Windows.

Virtual Function IO (VFIO) позволяет ядру Linux напрямую передавать оборудование PCI в ВМ с пренебрежимыми издержками. Сюда входит такое сложное оборудование, как дорогие видеокарты, так что вы можете сообразить, куда это клонится. Помимо возможности играть в ВМ Windows, это полезно, если у вас в ВМ имеется среда для машинного обучения или майнинга. Многие из тех, кто это применил, сообщают о производительности, близкой к 99% реальной. Правда, чтобы всё это заработало, придется проскочить через несколько обручей.

Во-первых, дабы избежать иллюзий, давайте проясним: вам понадобятся две видеокарты. ОС хоста вручает ВМ полный контроль над передаваемым GPU, так что сам хост может о нем забыть. Есть также ряд других аппаратных требований, которые мы попытались охватить во врезке на стр. 39. Если вы используете две отдельные карты, проверьте, достаточно ли им питания. Теоретически, та, что на хосте, не должна потреблять много энергии, пока работает ВМ, но никогда нельзя знать заранее. Современные GPU стали не только мощнее, но и эффективнее, однако в иной трудный день более жадные модели запросто могут оттянуть на себя 250 Вт.

Мне нужно IOMMU

Помимо активации в BIOS, надо приказать ядру использовать IOMMU, через опции загрузчика [bootloader], которые мы либо

можем вставить в строку `linux` в `/boot/grub/grub.cfg`, либо провести рекомендуемую процедуру их добавления в строку `GRUB_CMDLINE_LINUX_DEFAULT` в `/etc/default/grub` и затем запустить `grub-mkconfig`. В системах Intel требуемая опция — `intel_iommu=on`, а на AMD — `amd_iommu=on`. Перезагрузитесь и затем запустите следующий скрипт (взятый с полезной и достойной прочтения страницы wiki Arch по VFIO, https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF):

```
#!/bin/bash
shopt -s nullglob
for d in /sys/kernel/iommu_groups/*/devices/*; do
  n=${d#/iommu_groups/*}; n=${n%/*}
  printf 'IOMMU Group %s ' "$n"
  lspci -nns "${d##*/}"
done;
```

Оборудование PCI может быть передано только тем способом, который признает такое группирование. Если ваша видеокарта (та, которую вы хотите передать) находится в той же группе, что и важные или кажущиеся сложными устройства, простейшим решением будет переместить ее в другой слот и попробовать заново. В нашем случае скрипт выдал

```
...
IOMMU Group 16 01:00.0 VGA ... Hawaii XT / Grenada
XT [Radeon R9 290X/390X] [1002:67b0]
IOMMU Group 16 01:00.1 Audio device ... Hawaii
HDMI Audio [Radeon R9 290/290X / 390/390X]
[1002:aac8]
...
```

Случай практически идеальный. Аудиоустройство HDMI на нашем GPU находится в группе IOMMU, поэтому нам надо передать его тоже, обеспечивая нашей ВМ как видео-, так и аудиооборудование.

Если к нашему передаваемому GPU привязать драйверы ядра, те ни за какие коврижки его не покинут, так что нам надо это заранее пресечь. Это делается с помощью драйвера `vfio-pci`, который следует проинформировать о производителе передаваемого



Инь и Ян: для хоста мы использовали Nvidia GTX 680, а для гостя — Radeon 290 OCX, чуть поновее и побыстрее.



устройства и ID устройства (числа в конце показанного выше вывода). Наличие идентичных GPU у гостя и хоста здесь создает проблемы, потому что их ID одинаковы (загляните в wiki Arch, там приводится обходной путь). Чтобы передать наши GPU и аудиоустройство, мы создаем файл `/etc/modprobe.d/vfio.conf`, содержащий

```
options vfio-pci ids=1002:67b0,1002:aac8
```

Чтобы эта магия заработала, надо загрузить модули VFIO с `initramfs`. В Ubuntu это делается через файл `/etc/initramfs-tools/modules`, а в Arch (который мы, кстати, и используем) мы редактируем строку `MODULES=` в `/etc/mkinitcpio.conf`. В любом случае добавьте четыре модуля `vfio`, `vfio_iommu_type1`, `vfio_pci` и `vfio_virqfd` так, чтобы они появились перед всем, что касается видео. Вероятнее всего, там вообще не будет модулей, если только вы до того не настраивали KMS или нечто подобное. Теперь мы снова создаем `initramfs` с помощью `sudo update-initramfs -u` в Ubuntu или `mkinitcpio -k` в Arch. Перезагрузитесь и проверьте результат:

```
$ dmesg | grep -i vfio
```

Надо надеяться, что вы видите обнадеживающие руны:

```
[ 1.488517] VFIO — User Level meta-driver version: 0.3
[ 1.489469] vfio-pci 0000:01:00.0: vgaarb: changed VGA decodes:
olddecodes=io+mem,decodes=io+me m:owns=none
[ 1.504260] vfio_pci: add [1002:67b0{ffff:ffff}] class
0x000000/00000000
[ 1.520925] vfio_pci: add [1002:aac8{ffff:ffff}] class
0x000000/00000000
[ 5.429259] vfio-pci 0000:01:00.0: vgaarb: changed VGA decodes:
olddecodes=io+mem,decodes=io+me m:owns=none
```

Возможно, вы увидите другое, но это, вероятно, не конец света. Проверьте вывод

```
$ lspci -nnk -d 1002:67b0
```

изменив должным образом пару производитель—устройство. Это покажет вам, был ли драйвер VFIO правильно привязан к устройству или нет.

Не фальшивка, а прошивка

Чтобы всё это сработало, передаваемый GPU должен поддерживать UEFI, а BM надо настроить на загрузку с ним. Open Virtual Machine Firmware (OVMF) — это порт Intel Tianocore (он же — EFI Development Kit II, или `edk2`) прошивки UEFI, которая позволяет BM пользоваться благами UEFI. Установите ее с помощью `sudo apt install ovmf`. Мы предполагаем, что у вас уже есть пакеты QEMU, *Libvirt* и *virtmanager* из предыдущего раздела. Нам нужно сообщить *Libvirt*, где взять эту прошивку, что требует добавления строки в `/etc/libvirt/qemu.conf`. Ищите в этом файле `nvr`, где вы найдете кое-какие объяснения, и добавьте под этими комментариями строки

```
nvr = [
"/usr/share/OVMF/OVMF_CODE.fd:/usr/share/OVMF/OVMF_VARS.
fd"
]
```

Это для Ubuntu. В Arch измените путь к `/usr/share/ovmf/x64/`. Наконец, мы готовы настроить BM в *virt-manager*. Это делается по той же процедуре, что и раньше, только используется ISO Windows (который можно скачать у Microsoft). Дойдя до финальной стадии мастера, обязательно отметьте окно *Customize*, чтобы вы могли выбрать загрузку UEFI (см. экранный снимок внизу справа). У нас тут мало места, так что мы не будем рассказывать об установке Windows — уверены, что вы с этим разберетесь — но обязательно настройте VirtIO SCSI диск и поэкспериментируйте с отключением кэша на случай, если ввод/вывод диска будет всё тормозить. Как только Windows установится, мы можем выключить его и настроить волшебство передачи.

В диалоговом окне *BM Details* [Подробности] нажмите на *Add Hardware* [Добавить оборудование], выберите *PCI Host Devices*

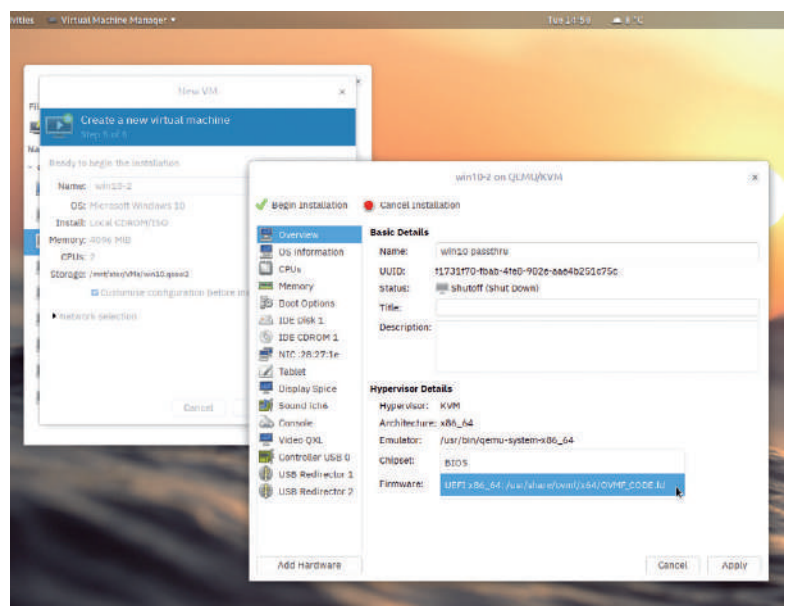
» ТРЕБОВАНИЯ К ОБОРУДОВАНИЮ

Чтобы все эти хитрости сработали, нужно довольно современное оборудование, и даже с ним всё может пойти не так. Видеокарта, используемая хостом, может быть какой угодно, даже встроенной. Если у вас два отдельных GPU, вам следует познакомиться с комбинацией клавиш для входа в BIOS/UEFI, потому что ваш компьютер почти наверняка попытается загрузить не ту видеокарту. Видеокарта гостя должна поддерживать UEFI (материнской плате этого не надо), поскольку мы загружаемся с помощью OVMF. Можно попросить у производителя обновленный BIOS GPU (нам сообщали, что люди так делали, и им шли навстречу) и передать его QEMU вместо заливки в свою видеокарту. В целом всё, произведенное после 2012 г., отлично подойдет. Ваша материнская плата должна поддерживать IOMMU (оно же — *Directed I/O*), что, возможно, понадобится включить в BIOS, и ваш CPU тоже должен его поддерживать (Intel называет это VT-d, а AMD — AMD-Vi; учтите, что это в дополнение к более общим VT-x и AMD-V, упомянутым ранее).

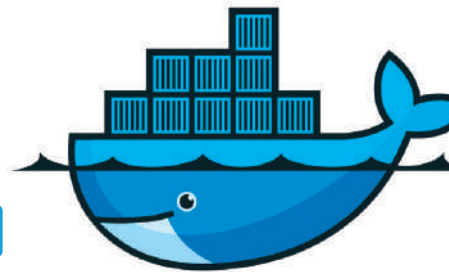
Помимо этих требований вы, возможно, решите потратиться на переключатель KVM (ничего общего с ядром) или дополнительную клавиатуру/мышь/тачпад, поскольку (за пределами соединения VNC или SPICE) гостевая машина не будет иметь доступа к устройствам ввода хоста. Вы можете передать устройства USB, но если на госте что-то пойдет не так, их, возможно, не удастся освободить, и вы лишитесь возможности управлять хостом. Если вам не хочется переставлять кабели, то вам также понадобится монитор с несколькими входами (и еще один видеокабель) либо второй монитор. Если у вас карта Nvidia, вы можете столкнуться с ужасной "error 43", которую можно обойти, поднастроив QEMU.

[PCI-устройства хоста] и выберите всё из группы IOMMU, которую мы подробно разобрали ранее. Хорошей идеей будет также передать клавиатуру и мышь USB, но их нужно отделить от используемых для управления хостом. Нам больше не нужны виртуальные устройства (канал SPICE, видеоадаптер QXL и эмулированные периферийные устройства), и их можно удалить.

А теперь момент истины: загрузите BM, вставьте (или переключите ввод монитора) передаваемое GPU и клавиатуру, нажмите на кое-какие клавиши, и, надо надеяться, вы загрузитесь в Windows. Вот уж не подумали бы, что мы когда-нибудь такое скажем!



Настройки Firmware нельзя изменить после загрузки средства установки, поэтому выберите UEFI.



Нырните в контейнеры

Контейнеризуйте свои приложения, запустите почтовый сервер и гуляйте!

Те из вас, кто носит гордое имя ветерана-линуксоида или когда-либо восстанавливал поврежденную установку Linux с диска live, знакомы с командой `chroot`. Она изменяет директорию `root` текущей оболочки, а будучи вызвана в директории, содержащей сам Linux, эффективно прикрепляется к ней. Если вы предварительно выполните все привязки и монтирование, она позволит вам использовать все программы и библиотеки в директории `/chroot` с ядром и оборудованием из рабочей установки. Именно так работает большинство установщиков Linux.

Если вспомнить о довольно сложных технологиях ядра (в частности, об управлении ресурсами с помощью `cgroups` и изоляции пространств имен), то современные контейнеры являются просто естественным развитием этой идеи. С другой стороны, современные контейнеры — это то, что вы получите, пожертвовав некой частью изоляции ВМ и избавившись от требования наличия полной операционной системы. Поэтому контейнеры более мобильны и просты в развертывании, чем ВМ. С утилитарной точки зрения контейнеры позволяют поместить приложения

в пакеты один раз и затем устанавливать их где угодно без всяких дополнительных зависимостей (кроме должной среды исполнения контейнера).

Установка Docker

Пакет `docker` имеется в репозиториях Ubuntu, но он не имеет никакого отношения к контейнерам, это апплет дока для почтенного менеджера окон WindowMaker. Чтобы установить `Docker` в Ubuntu 18.04, можно использовать Snap (доступный прямо из *Software Centre*) или добавить репозиторий `Docker`. Последнее требует чуть больше работы, однако всё будет немного новее, чем в Snap, и мы поступим именно так. Сначала обеспечьте, чтобы всё обновилось:

```
$ sudo apt update && sudo apt upgrade
```

Затем установите дополнения и помощники для `Apt`, и добавьте ключ `Docker` GPG:

```
$ sudo apt install git apt-transport-https ca-certificates curl software-properties-common
```

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Теперь добавьте репозиторий `Docker` для Ubuntu и обновите список пакетов:

```
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
```

```
$ sudo apt update
```

И, наконец, мы можем установить `Docker` Community Edition, с помощью

```
$ sudo apt install docker-ce
```

Почта как мычание

В **LXF239/240** мы научили вас создавать собственный почтовый сервер. Это заняло добрых три страницы и было довольно примитивно. Мудрый читатель написал нам и рассказал о проекте `mailcow:dockerized` [*mailcow* — *англ.* почтовая корова], посредством которого можно создать более богатый функциональный почтовый сервер (включая интерфейс `webmail`, блокировку спама, поддержку двухфакторной аутентификации [Two Factor Authentication] и прочие прелести) чуть более чем одним нажатием кнопки. Естественно, мы были заинтригованы.

» КУ-КУ, KUBERNETES...

Если вы из всех сил стараетесь не отставать от всяких трендов и идей, то, возможно, будете в шоке, узнав, что инновации не останавливаются никогда. С появлением приложений Cloud Native будет простительно считать, что никто уже не запускает ничего на чистом железе. Kubernetes (созданный Google, но ныне разрабатываемый под эгидой Cloud Native Computing Foundation) описывают как «POSIX облака» — в том смысле, что он очень быстро становится стандартом работы и управления контейнерами и ВМ независимо от инфраструктуры. Микросервисы — это то, что вы получаете, применяя принципы UNIX к сервисам, работающим в облаке: множество контейнеров, каждый из которых выполняет небольшие задачи (но выполняет их хорошо), объединяются для выполнения сложных задач.

Надежность и безопасность контейнеров (особенно по сравнению с ВМ) является предметом бурных обсуждений. Кто-то может счесть, что контейнеры не могут достичь уровня изоляции, предлагаемого ВМ, не говоря уж о магии пространства имен, поскольку в среде с несколькими обитателями они должны общаться через ядро хоста. Контейнеры можно заблокировать с помощью `AppArmor` или `SELinux` или профилей `seccomp`, но это может привести к отказу сложных приложений. Используйте `Kata Containers`, которые запускают контейне-

ры с собственным ядром, и функции CPU для обеспечения изоляции, как в ВМ. Они эффективно превращают контейнеры `Docker`, они же — «поды [pods]», k8s в легковесные ВМ. Что, запутались? Мы — да, и поэтому постарались узнать больше на <https://katacontainers.io>.

Контейнеры `Kata`: размытые границы, повышение изоляции и упрощение безопасности.



Этот твит (перевод см. внизу) доходчиво иллюстрирует, почему `Docker` так полезен.

* «Ваш код не работает!» «На "моей" машине работает.» «Хорошо, доставьте вашу машину!» Вот так начинался `Docker` :)

Конечно же, не всё так просто, и, как и раньше, мы не будем пересказывать всю сагу о настройке вашего доменного имени и записей MX. Вам также надо будет позаботиться, чтобы ваш роутер/брандмауэр перенаправлял/разрешал соответствующие порты. Для более подробной информации загляните на <https://mailcow.github.io/mailcow-dockerizeddocs/prerequisite-system/> и в остальную превосходную документацию.

Комплект приложений mailcow:dockerized — это на самом деле дюжины две разных контейнеров, соединенных между собой волшебством *Docker Compose*. *Docker Compose* использует файл YAML (YAML Ain't Markup Language), который определяет все сервисы и тома, используемые приложением, и позволяет всем связанным контейнерам общаться между собой. Тома — это механизм долговременного хранения данных, созданных контейнерами. Если бы мы не использовали их (или какую-либо альтернативу), то при каждой остановке контейнера все созданные им данные исчезали бы. И при следующем запуске контейнер выглядел бы как при самом первом. Наш пакет mailcow:dockerized определяет шесть томов.

Станем докерами

Docker Compose отсутствует в репозиториях, которые мы добавили ранее, поэтому возьмите его прямо с GitHub:

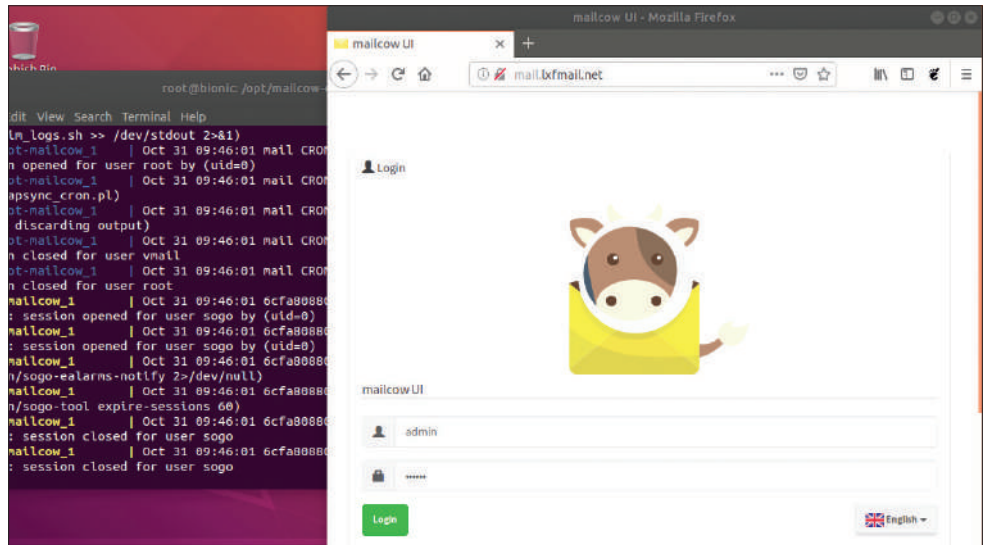
```
$ curl -L https://github.com/docker/compose/releases/download/$curl -Ls https://www.servercow.de/docker-compose/latest.php)/docker-compose
$ echo $(uname -s)-$(uname -m) > docker-compose
$ sudo mv docker-compose /usr/local/bin/
$ sudo chmod +x /usr/local/bin/docker-compose
```

В порядке альтернативы, если вы не любите путаться со всякими там значками доллара, просто загляните на <https://github.com/docker/compose/releases> и скачайте самый свежий релиз. Затем переместите бинарник в `/usr/local/`. Теперь мы можем почти клонировать репозиторий Git для mailcow:dockerized, который мы поместим в директорию `/opt`.

Это надо будет проделать с правами root (перед командой, которую мы запускаем от имени root, будет приглашение `#`), поэтому запустите

```
$ sudo -i
# cd /opt
```

Затем нам следует проверить вывод `umask` — он должен быть `0022` (чтобы наши файлы могли читать обычные пользователи). Если он оказался другим, тогда возьмите это дело на себя,



с помощью `umask 0022`. Теперь мы можем клонировать репозиторий с помощью:

```
# git clone https://github.com/mailcow/mailcow-dockerized
И начинаем настройку:
# cd mailcow-dockerized
# ./generate_config.sh
```

У вас спросят про доменное имя (FQDN) вашего почтового сервера и ваш часовой пояс. Желая поэкспериментировать, можете создать домен и добавить его в свой файл хостов. Например, мы использовали `mail.lxfmail.net` и изменили строку с `127.0.0.1` в `/etc/hosts` на

```
127.0.0.1 localhost mail.lxfmail.net
```

Теперь пошерстим файл настройки, с помощью

```
# nano mailcow.conf
```

Пресвятой mailcow Бэтмен, это работает! Зайдите с аутентификационной информацией по умолчанию и затем как можно скорее измените ее.

УСТАНОВКА DOCKER

«Чтобы установить Docker в Ubuntu 18.04, можно использовать Snap из Software Centre или добавить репозиторий Docker.»

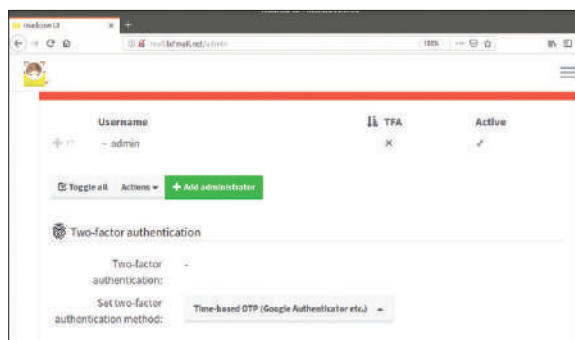
Если, как и мы, вы просто играете с выдуманным именем хоста, вам надо будет задать

```
SKIP_LETS_ENCRYPT=y
```

иначе процесс установки попытается получить сертификат для несуществующего домена, что закончится позорной неудачей. Затем скачаем сложные образы, которые составляют mailcow:dockerized (это займет не одну минуту даже на быстром соединении) и загрузим их:

```
# docker-compose pull
# docker-compose up -d
```

Вы должны увидеть длинный и радующий глаз отладочный вывод. Если (подобно нам) у вас уже работал web-сервер (или любой другой сервис, использующий необходимые для Mailcow порты — загляните в вышеупомянутую документацию), то эта команда не сработает. Web-сервис Mailcow может прекрасно работать через обратный прокси, или просто отключите свой web-сервер. Будем надеяться, что всё работает, тогда вы можете зайти на свою установку Mailcow по указанному адресу и настроить всё по своему желанию. Закончив, просто запустите `docker-compose stop` из директории Mailcow. LXF



Сделайте свою почту безопасной с помощью одноразовой аутентификации Time-based One Time Pad или чего-то подобного.

ОХОТНИК ЗА БОТАМИ

Крис Торнетт после встречи с **Паскалем Геененсом**
отключил свою IP-камеру и зарыл ее на задворках...

Любой, кто работает в сфере кибер-безопасности, скажет, что у злоумышленников есть несправедливое преимущество. Вот почему использование машинного обучения и более продвинутых систем глубокого обучения некоторые считают в этой области священным Граалем. Еще в сентябре 2018 г. Паскаль Геененс [Pascal Geenens] присутствовал на Европейском саммите Linux Foundation Open Networking Summit Europe в Амстердаме, чтобы обсудить автоматизацию обнаружения и смягчения атак в качестве евангелиста EMEA Cyber Security для Radware. А еще, как мы выяснили, он заядлый охотник за ботнетами и владелец ловушки для хакеров.

LXF: В своем выступлении вы упомянули применение киберпреступниками машинного обучения. Есть ли ощущение, что автоматизация безопасности может восстановить баланс?

Паскаль Геененс: Если смотреть на автоматизацию в общих чертах, я говорю также о скриптах, то киберпреступники давно ее используют.

Это всегда было проблемой. Взглянем на нашу область — защиты от DDoS, например: когда у вас есть DDoS-атака, идея состоит в том, чтобы обнаружить ее, затем охарактеризовать, а затем создать сигнатуру, приставить ее к делу и заблокировать атаку. Теперь, если атака останется прежней, без изменений в течение следующих 2–3 часов, всё будет хорошо, но с помощью автоматизации хакеры могут изменить вектор атаки, поэтому они напишут скрипт и поменяют порт. Они переключаются с UDP на TCP, используют другие механизмы атаки и переходят к усилению, отражению и другим способам, т. е. непрерывно меняют вектор своей атаки.

Их можно обнаруживать чем-нибудь вроде, например, NetFlow (технология мониторинга профиля трафика), и по этой причине у меня есть центр безопасности, который разместит сигнатуру, чтобы заблокировать эту атаку; но к тому времени, когда там всё исследуют — посмотрят в рсар, в сам захваченный пакет, охарактеризуют сигнатуру и пристроят ее на место — к этому моменту уже начинается другая атака.

В последнее время, из-за всех этих ботнетов IoT вокруг, мы видим гораздо больше краткосрочных атак. Это очень маленькие атаки, но с большой амплитудой и на короткий период: они длятся 6 секунд. Они создают пиковую нагрузку, но ненадолго. Вся инфраструктура находится на грани падения, а к моменту, когда она восстанавливается, создают еще один пик.

Они так поступают, как правило, потому, что некоторые клиенты используют для мониторинга, например, SNMP [простой протокол управления сетью]. А если вы используете SNMP, то проводите опрос каждые 5 минут, и в таком случае, если у вас небольшой всплеск на 5 секунд, за 5 минут он усреднится, и вы не увидите, что загрузка вашей полосы пропускания превысила порог в 80%, тогда как пакетов в секунду достаточно много, чтобы вызвать отказ в обслуживании. Получается, что вам этого не обнаружить, а если не обнаружить, то вы и не можете перенаправить их в облако, чтобы фактически блокировать атаки.

Это только одна причина, но она также вынудила нас изменить нашу защиту, потому что,

хотя она полностью автоматизирована, мы делаем характеристику атаки. У нас уходит 15 секунд на обнаружение атаки, последующее создание автоматической сигнатуры и ее установку. Из этих 15 секунд атака составляет всего 6 секунд, и она запускает весь цикл, а затем прекращается, и, поскольку мы под DDoS, внутреннего состояния нет. Это означает, что мы не хотим сохранять это состояние, потому что оно плохое, раз уж вы находитесь под DDoS-атакой, т.к. излишнее потребление ресурсов является одной из типичных атак, поэтому мы не хотели сохранять состояние.

Накатывает следующая волна, мы начинаем характеризовать ее, и прежде чем мы доберемся туда... бум! И вот опять. Значит, теперь мы должны изменить такой ход событий и установить защиту: всякий раз, обнаружив такое, мы начинаем оценивать ограничение, а затем будем характеризовать атаку в течение 6 секунд. Состояние сохраним, и затем в следующий раз, когда начнется атака, продолжим с того места, на котором остановились, до тех пор, пока не наберем полных 15 секунд. Это делается с помощью автоматизации, поэтому у них для своей бот-сети есть свой скрипт, есть центр управления, они пишут скрипт «Окей, сделайте атаку на 6 секунд» и отправляют его ботам. Все боты атакуют одновременно одну и ту же цель в течение 6 секунд, а затем отпадают.

LXF: Автоматизация восстанавливает баланс?

ПГ: Да, потому что без автоматизации мы бы никогда не смогли пережить такого рода атаки, даже со стороны скриптописцев-дилетантов. Так что не обязательно быть опытным злоумышленником, потому что можно отправить любую атаку куда угодно. Они просто помещают ее в скрипт и проводят атаки, особенно DDoS. Это очень просто. Поэтому автоматизация в этой области чрезвычайно важна.

LXF: Одна из вещей, о которых вы говорили, это хорошие обучающие данные...

ПГ: Существует несколько проектов с открытым исходным кодом, в которых есть данные, но они не относятся к вашей сети. У вас может быть журнал событий, и это также зависит от того, какое машинное обучение или какой ИИ вы хотите построить. Если вы хотите основать его на событиях из Windows или на журналах сетевого трафика — например, rscars или NetFlow, то вся эта информация выглядит по-разному. И это одна из самых болезненных вещей в ИИ и всей науке о данных. Вам необходимо нормализовать свои данные.

Это не самая веселая часть моделирования. Именно нормализация занимает большую часть работы — данные идентифицируются, затем им придается одинаковый формат, чтобы вы могли протолкнуть их через одну модель и выполнять фактическую работу с ней. Это сложная часть, а потом данные необходимо пометить.

О двух вещах я в своей презентации не говорил: это контролируемое и неконтролируемое

Паскаль на конференции подчеркивает важность обучающих данных.



обучение. Так, при контролируемом обучении нужны помеченные данные. У вас есть изображения, чтобы сказать: это кошка, это собака. Вы подаете их в сеть для обучения на примере, и она будет обобщать на основе того, хорошее ли совпадение. Вы можете показать сети новое изображение, и она скажет вам, кошка это или собака. При неконтролируемом обучении данные не помечены. Здесь вы просите машинное обучение найти в данных структуру, что может быть очень интересно — особенно для сетей.

Как правило, это не глубокое обучение; это более традиционное машинное обучение, которое вы используете: например, метод k ближайших соседей. Это очень простой алгоритм. Он берет

вы сможете пометить данные. Используя этот кластер, вы можете, например, сказать, что все эти точки надо пометить как хорошие, а эта метка, например, Office 365. Затем можно пометить данные, и такие данные уже употребить в контролируемом обучении.

LXF: А если читатели хотят попробовать сами, есть ли какие-то конкретные системы, которые вы бы предложили?

ПГ: Tensorflow, пожалуй, самая известная библиотека, а Azure [ML] довольно продумана, но интересной частью является игровая площадка в Tensorflow. Если вы зайдете на <https://playground.tensorflow.org>, вы увидите визуальное пред-

ПАСКАЛЬ ОБ УГРОЗЕ БОТНЕТОВ

«В последнее время, из-за всех этих ботнетов IoT вокруг, мы видим намного больше краткосрочных атак. Это очень маленькие атаки, но с большой амплитудой и длительностью 6 секунд.»

все точки и находит k-ближайшие точки прямо в пространстве. Делая это на каждом примере, вы находите ближайшие точки. Вы начнете кластеризовать данные, а затем создаете несколько кластеров. У вас нет метки для каждого кластера, но вы знаете, что когда появляется точка, она приближается к одному из существующих кластеров, и вы также можете сказать, к какому кластеру она ближе.

А значит, это поможет вам понять данные; это поможет вам найти аномалии; и это поможет вам построить более новую модель, где

ставление и сможете вводить данные, а затем вы увидите, как модель, обучаясь, проходит разные эпохи, а обучение состоит в том, что вы даете образчики.

Давайте опять обратимся к примеру с использованием изображений. Вы даете модели изображение, она его обсчитывает и выдает некоторый замер. В первый раз, когда вы вставляете изображение в нетронутую модель, этот замер, конечно, будет неправильным. Так что вам нужно изменить и настроить различные веса между персептроном и связями между персептроном.



Эти веса... вам следует настроить их так, чтобы результат был правильным.

LXF: Будь у вас небольшая сеть на базе Linux, какие данные вам быгодились? Вы, конечно, скажете, что смотря когда, правда?

ПГ: [смеется] Ну да, но если и вы хотите поиграть с этим, то у себя дома я начал с IoT, бот-сетей, а теперь больше машинного обучения и ИИ, потому что это алгоритмы, которые мне надо анализировать. Я начал писать свою собственную ловушку, а затем, в октябре 2016 г., произошла атака Dup. Кто-то сказал мне, что каждые две минуты ботнет стучится по каждому IP-адресу в мире. Я сказал, что если это правда, я должен увидеть это дома. Вот я и начал слушать. Сначала слушал простой telnet — и увидел, что это правда. Это было каждые 3 минуты, плюс-минус.

Я стал более тщательно разбираться с ловушкой, вводить больше протоколов, и теперь я сижу на больших объемах данных, которые по-прежнему ограничены, поскольку это мой дом, но мы взяли эту же идею, вложили ее в компанию и создали сети ловушек, а теперь у нас более 300 ловушек по всему миру.

LXF: Такая же штука, только в гораздо большем масштабе.

ПГ: Да, там мы получаем около 10 миллионов событий в день, и эти события должны быть проанализированы. Вы можете провести простое обучение, но вы имеете дело с большими данными. С этих пор мы занялись машинным обучением

LXF: Звучит как отличная заявка на новость дня.

ПГ: Mirai так много, потому что его исходный код был открыт. Поэтому за последние 2 года у нас было сотни вариантов. И последний вышел с 16-ю встроенными эксплоитами. Так что это одна из областей, где очень интересно взглянуть на данные и использовать кластеризацию, но я обычно использую стандартное машинное обучение, я не вдаюсь в глубокое обучение для ИИ, потому что есть много сюрпризов, которые вы можете получить, просто запустив это, не зная, что ищете. Существует масса подводных камней: переобучение, потому что это проблема номер один, которую вы заметите. Вот я хочу тренировать свою модель. Беру мои данные примеров; помещая их в модель, вы видите, как снижается уровень ошибок.

Ваша ошибка 0,001 % — отличная модель! Но в первый же раз, когда вы даете ей реальный образец, ошибка взрывается! Почему? Из-за переобучения. Ваша модель отлично проходит все пункты. А когда вы берете точку, которой нет в вашем наборе данных, вы получаете большую ошибку.

В этом проблема переобучения, но как вы узнаете, что переучиваете? Об этом я не говорил подробно.

Я показал вам [в докладе] одну модель глубокого обучения, многоступенчатую; один этап связан с другим, но у вас есть график со всеми различными моделями. Одни являются циклическими моделями, другие имеют петли обратной связи. Например, у вас есть LSTM [long-term short-term memory], долгосрочная память коротких сообщений. Как правило, она используется для всего, что является предложением — если у вас есть

все точки на изображении — каждая точка имеет три значения, RGB [красный, зеленый, синий] — и делаете каждую точку свойством, и каждая точка связана с персептроном, и оттуда вы просто выкручиваете из своей сети нечто на выход.

Конечно, можно делать и так, но тогда у вас будет очень дорогая сеть с точки зрения вычислений, потому что если у вас есть по три значения в картинке 1000 на 1000 [смеется], тогда это много узлов, так что всё вычисляется в матрице. Кроме того, в этих сетях есть еще забавная вещь. Вам кажется, что в уме это переходит от персептрона к персептрону, но на самом деле это рассчитывается всего за один шаг. Сети просто делают из этого матрицу.

На входе имеется вектор. Вы велите матрице умножить эти матрицы, и получается результат. Перемножение матриц — сложная математическая задача. Чтобы сделать это для миллиона точек... ну, это займет кое-какое время, если вы захотите сделать это дома на своем Raspberry Pi [смеется].

А по сути, это делается десятью строками кода, вот о чем я говорю. Это так доступно, но в то же время это черная магия, когда ты не знаешь, что происходит.

Для идентификации на Facebook кошки либо собаки или распознавания лица — кого волнует, если она напортачит, на самом деле никому нет дела. А если речь идет о распознавании лица, которое использовалось в Китае, чтобы посадить людей в тюрьму? Арестовывая людей на улице как преступников, они получали 80 % ложных срабатываний.

LXF: Считаете ли вы, что производители, в частности, усвоили урок, когда дело доходит до повышения безопасности IoT?

ПГ: Я думаю, что производители осознали это — по крайней мере, некоторые из них попытались ввести защиту в свои устройства IoT; но проблема в том, что мы решаем эту проблему сегодня. А ведь есть еще все эти старые устройства: всякие модемы со старой прошивкой, которые не обновляются; IP-камеры, которые не обновляются.

LXF: Есть устойчивое ощущение в том, что это связано с устаревшими устройствами?

ПГ: Если [новые устройства] поставляются с новой прошивкой и пользователи обновляют прошивку на своих устройствах, то большинство уязвимостей, эксплуатируемых сегодня, будут исправлены. Когда исследователи безопасности обнаруживают уязвимость, они обычно общаются с производством и ожидают 90 дней до выхода заплатки. А когда заплатка выпускается, это самое подходящее окно для прихода хакеров. Они просто копируют уязвимость, помещают ее в свои боты, поскольку знают, что большинство пользователей обновляться не будут — уж явно не в тот же день. Так что если появляется новая большая уязвимость, то не пройдет и 24 часа до того, как

ПАСКАЛЬ РАСКРЫЛ СВОЮ МОТИВАЦИЮ...
«Кто-то сказал мне, что каждые две минуты ботнет стучится по каждому IP-адресу в мире. Я сказал, что если это правда, я должен увидеть это дома. Вот я и начал слушать.»

и ИИ. Я уже делал это раньше, но сейчас я более активно занимаюсь этим и дома. Я использую данные своей ловушки, которые являются интересными, потому что они все плохие. Тем не менее, это не то, что вы хотите использовать для вашей безопасности как таковой, поскольку это больше о поиске структуры в данных. Итак, выясняете, что такое ботнет Mirai, что такое ботнеты Hajime или какие другие атаки они используют, или новые URL-адреса, которые они используют с новыми уязвимостями. Так что это скорее насчет классификации, верно? А вы хотите найти уже классифицированные ботнеты, чтобы, например, отследить, насколько распространен сейчас Mirai? Или сколько вариантов Mirai мы видели за последние 2 года?

несколько слов. Т.е. вы должны учесть небольшую историю предыдущих слов, чтобы быть в состоянии интерпретировать следующее слово. Которое приходит в команде или предложении. Вы должны иметь этот цикл обратной связи, который передается в персептрон короткие отрезки истории, чтобы тот мог принимать решение не только на основе фактических данных, которые вы предоставили сейчас, но также и предыдущих данных из базы. Это типично для речи, и во всех временных рядах будет более или менее использоваться LSTM.

Ну вот, эти мелочи вам надо знать; если у вас нет опыта, вы принимаетесь за свою стандартную нейронную сеть, а она не работает. Почему не работает? Или как вам перевести свои данные в свойства? С изображениями проще: вы берете

новый ботнет злоупотребит ею. Мы увидим это в наших ловушках с данной уязвимостью, попытаюсь использовать ее.

Мы видим, что всё еще используются уязвимости маршрутизаторов D-Link четырехлетней давности — всё еще используются, и небезуспешно. Таким образом, уязвимость, публикации которой всего один день — это золотая жила в терминах IoT, потому что они, вероятно, будут активны в течение 2 или 3 лет. До тех пор, пока люди не выбросят свой модем, или их модем не сломается, или по какой-то причине они не сделают обновление прошивки, потому что их новый iPhone больше не работает. Им нужен стимул для обновления, верно?

LXF: Разве не хороший стимул, что могут украсть твою идентификацию?

ПГ: Если ваша идентификация украдена, вы что, первым делом броситесь обновлять свои IP-камеры или модем?

LXF: В прошлом году вы рассказали об использовании эксплойта telnet и о том, как старые технологии часто служат проникновению.

ПГ: Это использовалось в исходном Mirai. Вероятно, это было связано с BrickerBot, и BrickerBot может людей озаботить, потому что это был ботнет, а не типичный червь. Mirai больше похож на червя: он заражает устройство. Итак, у вас есть IP-камера, IP-камера открывается Интернету. Как она становится открытой? Ну, или кто-то устанавливает ее с публичной точкой доступа, или люди дома устанавливают ее, но есть удобный протокол, называемый UPnP, который автоматически открывает дырку в брандмауэре, чтобы вы могли использовать свой телефон извне и получить доступ к своей IP-камере, вот таким образом это общедоступно. Mirai просто подключился к общедоступным устройствам, в основном IP-камерам, видеорегистраторам. В Mirai они всего лишь использовали 61 пароль по умолчанию.

Всего с 61 паролем Mirai смог заразить сотни тысяч устройств в Интернете. Это был просто telnet. При этом, когда Mirai заражает устройство, оно начинает поиск новых жертв и пытается использовать их через telnet. Найдя, что он может получить доступ к кому-то конкретно с помощью telnet, он отправляет IP-адрес на сервер загрузчика, и тот затем загружает Mirai прямо на это устройство. Вот почему каждые две минуты каждый IP-адрес в Интернете сканируется. Все зараженные устройства начинают поиск новых жертв в Интернете. Это означает, что число зараженных устройств растет экспоненциально.

Теперь BrickerBot — а по сути, действующее лицо за BrickerBot, называющий себя «Janitor [англ. дворник]», которого я засек, используя мои ловушки, когда он вышел к одному из журналистов — он занимается не тем, что активно атакует устройства, но он слушает. Он хотел очистить Интернет от зараженных устройств, поэтому



Паскаль не одобряет методы борьбы Janitor с ботами.

и слушал, и тут почувствовал, что кто-то сканирует его, пытаюсь получить доступ к нему через telnet с одним из этих паролей. Он знал, что это было IoT-устройство, зараженное Mirai, верно? Вот он его и контратаковал.

Он использовал список эксплойтов, пытался внедрить на устройство эксплойт, а затем запускал набор команд для повреждения флеш-памяти. Большая часть флеш-памяти в IoT доступна только на чтение. Тем не менее, существует небольшой раздел, который должен быть доступен на запись, чтобы сохранить настройки. Если он будет поврежден, большинство камер не сможет на угрозу отреагировать.

LXF: Что вы думаете о таких, как Janitor, и что они сделали?

ПГ: Hajime — один из тех ботов, которые существовали до атаки Dyn. За два дня до нападения Dyn вышел первый отчет о Hajime. Hajime — ботнет, подобный Mirai, но Mirai очень неискусственный. Командование и управление — это просто TCP, фиксированная IP-адресация, всего две строки в бинарнике Mirai, вы видите IP-адрес и порт сервера управления и контроля, вы соединяетесь с командным сервером.

Hajime гораздо сложнее, потому что он использовал сеть BitTorrent с динамическими инфорезами [info_hashes]. Каждый день он менял инфорез, а по инфорезу и находится файл. У вас были динамические инфорезы, создающие оверлей, который каждый день перемещался поверх BitTorrent и имел шифрование с закрытым/открытым ключом и по алгоритму RC4. Таким образом, он распространяет команды и элементы управления, которые могут обновлять сам ботнет и обмениваться конфигурацией.

Типичный ботнет может быть удален путем устранения его командования и контроля,

но только не Hajime. Он распространяется так, что вы не можете просто заблокировать его трафик. Он всё еще там, и в какой-то момент он контролировал 300 000 модемов. Но Mirai также защищает себя и от других ботов. Почему? Потому что для них это вроде войны.

У вас есть сотни хакеров, пытающихся создать ботнет для своих служб загрузчика и стрессера для DDoS-атак или майнинга. К счастью для нас, существует большая конкуренция, потому что это очень фрагментирует «рынок», означая, что размеры ботов равны 100 000 или 200 000. Теперь вернемся к BrickerBot и Hajime.

LXF: Я собирался спросить про Janitor.

ПГ: Я понимаю, зачем он хочет это делать, но не согласен с его методами, потому что он ломает вещи и влияет на бизнесы, и его это не волнует. BrickerBot навредил небольшому калифорнийскому интернет-провайдеру, потому что его модемы заразились Mirai, а затем начали искать других жертв. BrickerBot это обнаружил и принял атаковать такие модемы.

Клиенты звонили, говоря: «Эй, я не могу получить доступ к Интернету». Провайдер выясняет: «О, ваша прошивка повреждена, приносите ваш модем. Мы выдадим вам новый», пока не пришлось столько пользователей, что их запасы иссякли.

BrickerBot также навредил Аргентине — попортил миллионы мобильных устройств. В Индии были повреждены 60 000 модемов. Поэтому я не совсем согласен с его деятельностью. Кроме того, у него есть несколько блогов, в которых он называет свой проект «интернет-химиотерапией», [...] очищающей Интернет от всех плохих IoT-устройств, и это... м-мм, такая проблема действительно существует, но, по моему мнению, его решение этой проблемы не является правильным. **LXF**

Лазерное сканирование в Ogof Ffynnon Ddu в Уэльсе революционизировало геодезические съемки в больших пещерах.

ПОГРУЗИМСЯ В ОСМОТР ПЕЩЕР

Майк Бедфорд открывает, как высокотехнологичное оборудование и программы Linux помогают спелеологам в исследовании удивительного мира у нас под ногами.

Исследование пещер и подземных галерей, спелеология или как вам угодно это назвать, часто рассматривается как упражнение, в котором исследователи противопоставляют себя природе. И действительно, посещение некоторых наиболее сложных пещер требует высокого уровня физической подготовки, выносливости и ловкости, а также опыта в спуске и подъеме по веревкам в вертикальных шахтах; но важны и технические навыки работы с электроникой и программным обеспечением.

Хотя на поверхности Земли сейчас мало мест, оставшихся неисследованными, этого нельзя сказать о мире у нас под ногами. Действительно,

много там всё еще ждет своего открытия. Традиционно, при исследовании отдаленных частей земной поверхности, одним из первых задач при документировании нового открытия было его нанесение на карту. Это справедливо и по отношению к подземному царству, и не без уважительной причины.

Точные карты — или съемки, как обычно называют их спелеологи — жизненно важны для многих аспектов спелеологии. Они бесценны, поскольку помогают другим исследовать те же пещеры, они необходимы для дальнейших открытий, и они находят применение в разных областях исследований, включая гидрологию, вулканологию, спелеоархеологию и спелеобиологию. Здесь

мы рассмотрим процесс исследования пещер, сосредоточившись на новейших электронных инструментах съемки и программном обеспечении — большая часть которого работает на Linux — используемом для создания карт, трехмерных навигационных экранных изображений или пролетных изображений и даже печатных 3D-моделей пещер.

Мы рассчитываем, что это введение в изучение пещер даст представление о необычном, но впечатляющем применении технологий. Однако если вы почувствуете вдохновение, чтобы принять участие, то это определенно та область, в которую вы могли бы внести свой вклад, благодаря открытой природе многих пакетов исследований пещер.

» РАДИОЛОКАЦИЯ

В последние годы картографирование претерпело значительные изменения: теперь наряду с более традиционными методами важную роль играют GPS и спутниковые изображения. Тем не менее, в мире съемки пещер всё совсем по-другому. Сигналы GPS не проникают под землю, спутниковые изображения не могут видеть пещеры, и даже традиционные геодезические приборы, такие как теодолиты и тахеометры, слишком громоздки и хрупки, чтобы их можно было протаскать в тесные проходы пещер для повседневной съемки. Мы начнем с рассмотрения опробованных и проверенных методов съемки пещер, чтобы мы могли использовать это в качестве основы для обсуждения современных электронных методов.

Традиционно, исследование пещер — это итеративный процесс получения показаний от одной т.н. стоянки до другой по мере прохождения пещеры. Стоянки, как правило, располагаются на границе видимости с предыдущей стоянкой, и это расстояние обычно ограничивается кривизной подземных проходов. На каждой стоянке снимаются три показания, а именно расстояние по рулетке, азимут по компасу и вертикальный уклон по измерительному инструменту, известному как инклинометр или клино. Нет необходимости говорить, что если местоположение входа в пещеру известно, эти три показания для каждой стоянки позволяют рассчитать местоположение стоянок и, следовательно, построить среднюю линию проходов. На каждой стоянке также обычно измеряют ширину и высоту прохода и рисуют поперечное сечение.

Подземные стоянки

Хотя подход «от стоянки к стоянке» по-прежнему составляет основу исследования пещер, рулетка, компас и клино сейчас



Этот радиолокационный приемник может с высокой точностью определить подземный передатчик для обеспечения фиксированной точки на поверхности.

Пошаговая природа традиционного процесса исследования пещер влечет постоянное накопление ошибок. Таким образом, расчет положения конца длинного пещерного прохода бывает подвержен значительным ошибкам. Одним из способов преодоления этой проблемы является применение радиолокации, чтобы определить абсолютное местоположение крайних точек пещеры и тем самым удвоить число фиксированных точек.

Обычные радиоволны, проходя через скальную породу и почву, сильно ослабляются. Но если используется достаточно низкая частота, сигналы могут проникать сквозь почву на глубину нескольких сотен метров.

Данная технология применяется командами спасателей в пещерах для координации своих операций. Это и ключевой фактор радиолокации. Процесс включает настройку подземного передатчика и связанной с ним

рамочной антенны, которая генерирует четко определенные линии магнитного поля. Затем, используя приемник и рамочную антенну на поверхности, легко определить «нулевой уровень» — иными словами, точку непосредственно над подземным передатчиком.

Приемник GPS позволяет определить абсолютное местоположение. Выполнение измерений угла линий поля на некотором расстоянии от эпицентра также позволяет рассчитать глубину передатчика.



постепенно вытесняются электронными инструментами «всё в одном», большинство из которых было разработано спелеологами. DistoX2, от швейцарца Хибя Бита [Heeb Beat], иллюстрирует принцип работы этих устройств. Он основан на Leica Disto X310 — это недорогой, умеренно прочный и водонепроницаемый портативный геодезический прибор с лазерным дальномером и электронным компасом. Усовершенствование от Бита, которое включает замену одной из схем и изменение прошивки, добавляет функциональность инклинометра, необходимую для съемки пещер.

ЭВОЛЮЦИЯ ИССЛЕДОВАНИЯ ПЕЩЕР

«Рулетка, компас и клино сейчас постепенно вытесняются электронными инструментами „всё в одном“.»

Он связывается с портативным устройством, таким как надежный смартфон Android, так что съемку можно просматривать в процессе создания. По сравнению с подходом рулетка/компас/клино этот процесс намного быстрее, потенциально неразборчивые цифры в блокноте исключаются, а ошибочные показания часто сразу обнаруживаются на соответствующем дисплее. Это также делает практичным проведение нескольких измерений от точки в центре прохода до точек на стене, чтобы получить более точные поперечные сечения. А на поверхности — чреватый ошибками процесс ввода рукописных данных из замызанного блокнота в программу для исследования пещер тоже остался в прошлом.

Основной лазер

Съемка пещер была революционизирована несколько лет назад с появлением лазерных сканеров, хотя эти инструменты



» ПЕЩЕРНЫЕ УДОБСТВА

Печатная карта, безусловно, является наиболее распространенным способом представления данных съемки пещер, хотя просмотр на экране 3D-модели, которую можно вращать, масштабировать и панорамировать, весьма впечатляет. Тем не менее, особое чувство вызывает физическая 3D-модель пещеры у вас в руках, что стало возможным с появлением 3D-печати.

Это отнюдь не мейнстрим, но спелеологи продемонстрировали производство 3D-моделей сплошных пещер с впечатляющими результатами. Это упражнение продемонстрировало разнообразные методы 3D-исследования пещер. Одна модель была получена в результате трехмерного лазерного сканирования, проведенного в исторических пещерах Ноттингема, которые, строго говоря, представляют собой археологические рукотворные полости в мягком песчанике, в отличие от естественных пещер, встречающихся в известняке. Вторая — это модель сложного массива проходов в Кузва-де-Рианьо, Матиенсо, Испания, и в нем использовались результаты обычной съемки. Третий — это модель кальцинированного скелета медведя, также найденного в пещере Матиенсо; модель была создана по методу, называемому фотограмметрией, который использует серию обычных фотографий для создания трехмерной геометрии.

изначально создавались для других целей. Первые лазерные сканеры, и до сих пор наиболее часто используемые, представляют собой большие инструменты, которые устанавливаются на штатив и стоят очень дорого. Эти приборы испускают инфракрасный лазерный луч и измеряют время, необходимое для возврата импульса света после отражения от твердой поверхности. Поскольку скорость света известна, это время легко преобразуется в расстояние.

Лазер приводится в действие двигателем, поэтому он может быть запущен под любым горизонтальным углом вокруг сканера и через ряд вертикальных углов. Это позволяет собирать тысячи точек каждую секунду, и за считанные минуты можно создать полное трехмерное облако точек (координаты x, y, z точек в пространстве) ближайшего окружения. Этот тип инструмента не подходит для съемки пещерных проходов, потому что он недостаточно маневренный, но его обычно используют для создания 3D-моделей пещерных камер или комнат. Например, спелеолог Ру Уолтерс [Roo Walters] поставил перед собой задачу осмотреть крупнейшие в мире подземные полости с помощью лазерного сканера, и сделал несколько поразительных открытий. Например, Саравакская пещера в Малайзии — самая большая в мире по площади, и достаточно велика, чтобы на ее полу можно было припарковать 47 реактивных самолетов Boeing 747. Однако подобная работа бывает очень трудоемкой. Большие полости редко можно

обследовать из одной позиции, потому что некоторые их части всегда будут скрыты, и их надо сканировать с нескольких позиций, а затем сшивать данные. Саравакская пещера является хорошим примером. Ру изучил ее со 101 позиции, что заняло четыре дня и сгенерировало 137 ГБ данных, а затем на объединение данных ушло 20 часов обработки.

Обычный лазерный сканер, установленный на треноге, не способен заменить комбинацию рулетки/компаса/клино или даже электронных инструментов вроде DistoX2, а вот недавно появившиеся портативные лазерные сканеры идеальны. На ручном сканере-оригинале, ZEB1 от GeoSlam, был установлен вращающийся лазер, на большой пружине, которая беспорядочно подскакивала при ходьбе пользователя. Более поздний ZEB-REVO имеет более контролируемое движение, поскольку снабжен двумя механизмами с приводом от двигателя, вращающимися под прямым углом.

В любом случае, создается сложное трехмерное облако точек, которое нуждается в корректировке, чтобы учесть движение пользователя и кивающее и/или вращательное движение головки сканера. Это достигается с помощью данных о местоположении от встроенных инерционных датчиков и сложных методов сопоставления с образцом, которые способны распознавать особенности между последовательными потоками данных. Этот процесс сильно нагружает компьютер. GeoSlam предлагает два варианта обработки: оплата за использование на своих облачных серверах Linux или локально, с более традиционным соглашением о лицензировании, с программным обеспечением, доступным в настоящее время только под Windows.

Ручной лазерный сканер, хотя и намного более дорогой, чем его альтернативы, предлагает значительные преимущества по сравнению с ручными методами съемки. Первое — это скорость. Одна из первых демонстраций ZEB1 под землей прошла в пещере Скирвит [Skirwith Cave] в национальном парке Йоркшир-Дейлз [Yorkshire Dales] в Великобритании. Первые 168 м прохода — до подъема, который стал непреодолимым из-за скопления воды после сильного дождя — были обследованы в обоих направлениях за 36 минут. Сканирование в обратном направлении обеспечило «замыкание», иными словами, внешнее сканирование началось в известном фиксированном положении входа в пещеру, тогда как обратное сканирование завершилось в этом же положении.

Наличие двух известных точек позволяет программному обеспечению минимизировать накапливаемые ошибки. В отличие от этого, обычная съемка заняла бы около 4,5 часов только в одном направлении и, возможно, вдвое дольше при той же длине низких проходов, где надо пробираться ползком. Электронные геодезические инструменты типа DistoX2 улучшили бы эти показатели, но всё же не приблизились бы к лазерному сканированию. Второе основное преимущество заключается в том, что вместо обычных редких поперечных сечений, изредка фиксируемых при традиционной съемке, проход записывается в виде полностью трехмерной модели.

ПО съемки

Для большинства целей данные исследований пещер должны быть представлены в формате, напоминающем карту, а ПО для исследований пещер в последние годы достигло огромных успехов. Большая часть программного обеспечения была произведена спелеологами для спелеологов, и по этой причине имеет тенденцию быть свободно доступным. Как правило, ПО выпускается для работы с различными операционными системами, включая Linux, и весьма обычны проекты с открытым кодом.

Первоначально процесс выполнялся вручную; наиболее фундаментальным требованием к программному обеспечению для исследования пещер является расчет местоположения каждой

» КАК ПРИНЯТЬ УЧАСТИЕ

Если наше введение в изучение пещер вдохновило вас на участие в этой необычной, но полезной комбинации исследования, физической активности, технических штучек и кодирования, мы должны подчеркнуть, что у вас не должно возникнуть соблазна прочувствовать пещеры, заглянув в них самовольно. Единственным разумным вариантом будет обратиться в спелеологический клуб, способный предоставить вам инструкции по безопасности. В основных спелеологических зонах таких клубов десятки, и вы можете найти подробную информацию о клубах спелеологии на сайте newtocaving.com/contact-us.php в Великобритании или www.nssio.org/Find_Grotto.cfm в США.

Тем не менее, технофилы, вероятно, будут заинтересованы в технических аспектах спелеологии, и здесь мы укажем вам на Британскую ассоциацию исследователей пещер (BCRA) и ее группу Cave Surveying, представляющую особый интерес (cavesurveying.org.uk). Также особый интерес представляет Cave Radio & Electronics Group (CREG) по адресу bcra.org.uk/creg, которая публикует ежеквартальный журнал и организует полевые встречи два раза в год.

Несмотря на название, CREG занимается продвижением технологий в различных областях, таких как связь, фотография, регистрация данных, освещение и геодезия. Если ваш интерес к спелеологии более технический, чем спортивный, участие в выездном совещании CREG предоставит вам краткое введение в спелеологию и одновременно даст возможность познакомиться с некоторыми последними разработками в области технологий исследования пещер.

С ручным лазерным сканером вы получите полное 3D-представление пещерных проходов, просто проходя (проползая) по ним.



стоянки по необработанным данным съемки. Эти положения позволили бы нарисовать центральную линию каждого из проходов, но обычно, используя измеренное поперечное сечение прохода на каждой стоянке, вместо этого можно показать стенки проходов, создавая таким образом впечатление об изменении ширины прохода. Рисованные вручную поперечные сечения прохода также обычно добавляются в выбранных точках вдоль каждого прохода. Обзоры сложных пещер, в которых проходы пересекают другие проходы на разных глубинах, как это общеизвестно, трудны для прочтения. Относительно недавняя тенденция в выводе данных обследований заключается в нанесении на график различных частей пещеры или, возможно, разных глубин разными цветами, что значительно улучшает интерпретацию.

То, что мы рассмотрели до сих пор, является лишь азами, и вы, вероятно, могли бы написать какие-то программы и довольно легко справиться с этим. Тем не менее, предположение, что это предел, окажет плохую услугу многим пакетам исследований пещер. Мы всего лишь прошли по поверхности, дав вам представление о том, что предлагается; дополнительные функции, которые вы, вероятно, найдете, включают обнаружение ошибочных данных, возможность вносить данные в проект несколькими геодезистам,

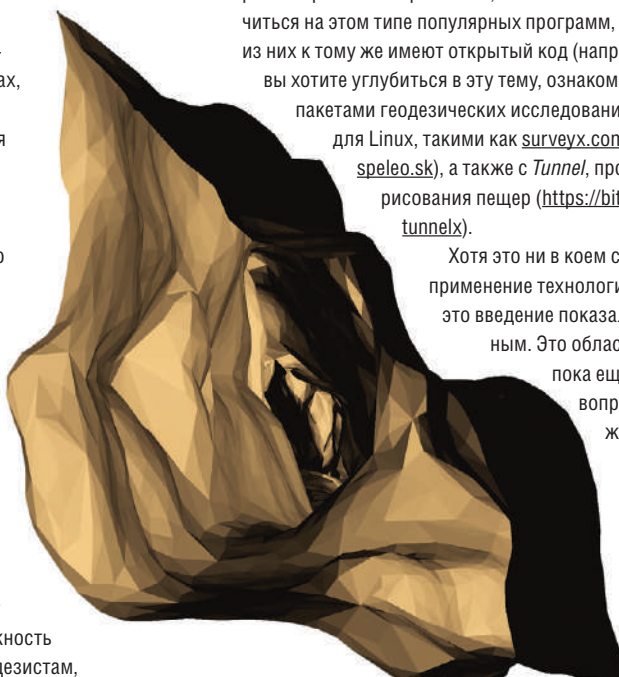
наложение проходов в пещерах на обычные карты поверхности или спутниковые снимки, создание трехмерных моделей в WRL или аналогичных — для экранной манипуляции... и этот список можно продолжать.

Тем не менее, благодаря достижениям в области графических информационных систем (ГИС), некоторые исследователи пещер рассматривают вопрос о том, не больше ли смысла сосредоточиться на этом типе популярных программ, причем некоторые из них к тому же имеют открытый код (например, qgis.org). Если вы хотите углубиться в эту тему, ознакомьтесь с основными пакетами геодезических исследований с открытым кодом для Linux, такими как surveyx.com и Therion (therion.speleo.sk), а также с *Tunnel*, программой на Java для рисования пещер (<https://bitbucket.org/goatchurch/tunnelx>).

Хотя это ни в коем случае не заурядное применение технологий, мы надеемся, что это введение показалось вам увлекательным. Это область, где у спелеологов

пока еще нет ответов на все вопросы, а значит, вы можете принять участие и внести реальный вклад. Если это необычное применение вам нравится, не забудьте заглянуть во врезку Как принять участие (вверху). **LXF**

Данные ручного лазерного сканирования могут быть представлены во внутреннем 3D-виде пещерных проходов, как этот вид пещеры Скирвит в Йоркшир-Дейлз.



Мониторинг систем в Linux стал лучше

В глубине души **Валентин Сеницын** опасался, что мониторинг в Linux уступает *DTrace*, и был счастлив выяснить, что теперь это уже не так!

Мониторинг систем, eBPF и т.п. — модные темы в сообществе Linux. Причина в *bpfftrace* (<https://github.com/iovisor/bpfftrace>), новом наборе инструментов, который был обнародован в начале октября 2018 г. У Linux нет недостатка в подобных инструментах. *SystemTap* (<https://sourceware.org/systemtap/>) не основан на eBPF и пока не полностью вошел в основное ядро. *BCC* (<https://github.com/iovisor/bcc>) популярен, но не очень удобен для скриптов-однострочников. *Ply* (<https://github.com/iovisor/ply>) — противоположность *BCC*: высокоуровневый инструмент пока не добился должного внимания. И это

далеко не полный список. В свете всего этого, что же выделяет *bpfftrace* настолько, чтобы Брендан Грегг [Brendan Gregg] назвал его “*DTrace 2.0*” в своем блоге?

Брендан является одним из двух авторов *bpfftrace*; другой — Аластер Робертсон [Alastair Robertson], создатель проекта. Очевидно, это не единственная причина. *bpfftrace* обеспечивает высокоуровневый синтаксис, похожий на *awk*. Не ново, но уже делает *bpfftrace* хорошим выбором для однострочников. Однажды у нас была проблема с производительностью в подготовительном кластере. Пока я боролся с С и необработанным eBPF, мой коллега собрал в *SystemTap* однострочник и обнаружил, что ядро CPU было занято миганием курсора. *BCC* — это С со вкраплениями Python (**LXF231**); *bpfftrace* во многом похож на *DTrace*. Они не совместимы по синтаксису: *DTrace* использует D (но не тот D, который является преемником C++). Однако неважно, набираете ли вы `this->var` или `$var`. Важнее то, что *bpfftools* умеет то же, что *DTrace*, и даже больше. Незначительные функции (`sizeof()`) отсутствуют, но при необходимости их можно добавить. Зато *bpfftools* может собирать стеки вызовов, что требует последующей обработки в *DTrace*. Все эти факторы заработали *bpfftrace* значок “*DTrace 2.0*”.

Впрочем, еще посмотрим: ведь *bpfftrace* — относительно молодой проект. Он использует LLVM для преобразования своего высокоуровневого синтаксиса в код eBPF, и это может оказаться проблематичным. Основа (eBPF) — это многолетняя и проверенная в работе технология. Возможно, однажды *bpfftrace* будет установлен во всех уважающих себя дистрибутивах Linux, как это было с *DTrace* в Solaris.

» ПОДХОД «СДЕЛАЙ САМ»

У Linux очень энергичное сообщество. Мы не просто пользователи — мы пропагандисты. И мы считаем, что Linux лучше конкурентов, по крайней мере, в некоторых аспектах.

Если вы давний наблюдатель, тут вы можете подумать, что это утверждение не всегда соответствовало действительности. Разумеется, всегда были (и, вероятно, будут) области, в которых Linux сияет, и области, где необходимы улучшения. Однако 10 или 15 лет назад сравнение с конкурентами не всегда было в пользу Linux, особенно на настольных ПК.

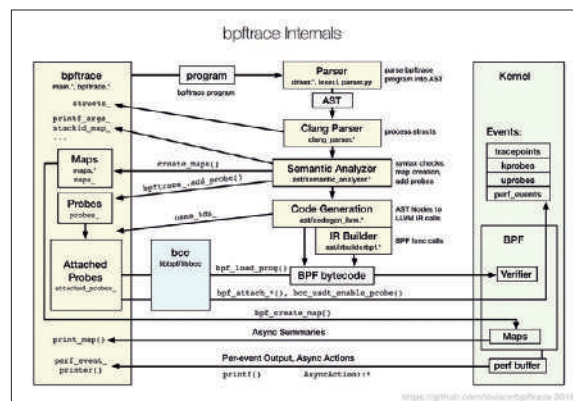
Конечно, некоторые из отсутствующих в Linux функций были незначительными или нишевого применения, но суть в том, что было трудно превзойти коммерческого конкурента исключительно набором функций. В те времена мы употребляли другие аргументы: «Возможно, в Linux нет всего того, что имеется в вашем любимом коммерческом пакете, но зато он свободный и бесплатный. Так что можете добавить нужную функцию — или обойтись без нее и сэкономить деньги». Иногда это работало, иногда не очень; оглядываясь назад — это было неважно. Суть в том, что время доказало: этот аргумент предельно верен.

Возьмите *DTrace*. Когда он впервые появился в Solaris, я помню, коллеги-инженеры в Sun сказали мне, что Linux отстает, так как ему не хватает этого важного инструмента. Позднее, когда Solaris стал свободным, специально позаботились о том, чтобы для *DTrace* стало непросто появиться в Linux. Это было трудное время, поскольку даже наш второй аргумент уже не работал. Но как сообщество мы восприняли свой же собственный совет и разработали недостающую функцию. Пять лет спустя мы победили. И в этом красота свободного ПО.



**НАШ
ЭКСПЕРТ**

Д-р Валентин Сеницын — разработчик облачной инфраструктуры в Яндекс днем, активный участник сообщества открытого кода ночью, интересующийся всем, от АН до X509.



bpfftrace объединяет flex, Bison, части Clang, солидную порцию LLVM и eBPF для создания полнофункционального высокоуровневого мониторинга. Источник: https://github.com/iovisor/bpfftrace/blob/master/images/bpfftrace_internals_2018.png

tmux: Современная работа в командной строке

Старую собаку новым трюкам не научишь, а вот с терминалами можно развлечься — нужно всего лишь запустить дополнительную команду.

Обычный пользователь Linux не слишком часто открывает свой терминал. Для нас, администраторов, всё иначе. Мы верим в автоматизацию, а графические инструменты в этом не очень сильны. Мы часто осуществляем удаленный доступ к нашим компьютерам, когда использование командной строки экономит нам пропускную способность. Более того, при нестабильных GPRS-соединениях это может оказаться единственной опцией.

Независимо от нашей роли, все мы заслуживаем удобства в работе. Необходимость перезапуска из-за прерванного SSH-соединения приносит неудобства. Иногда надо запустить команды и просмотреть системный журнал по мере работы. И уж всяко мы хотим иметь возможность остановить администрирование сервера, закрыть крышку ноутбука, прийти домой и продолжить с того момента, когда мы остановились.

Уставившись в экран

Тип программы, который облегчает реализацию вышеописанного сценария, часто именуется «мультиплексор терминала». Обычно он дает вам возможность общего использования одного (псевдо) терминала в нескольких окнах, которые вы можете назвать и «вкладками». Мультиплексоры терминала обычно сохраняются после выхода, и это означает, что вы сможете возобновить сессию позднее.

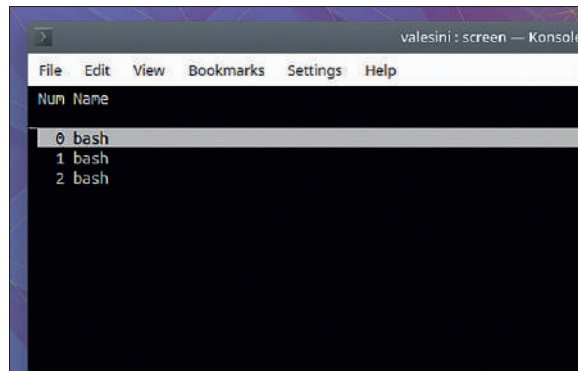
Классический пример этого семейства — *GNU screen*. Он на четыре года старше Linux и почти наверняка доступен в репозиториях пакетов вашего дистрибутива. *GNU screen* не слишком привлекателен внешне [Ред.: — На его фоне и я красавчик], но со своей задачей справляется хорошо. Он довольно широко используется и может стать единственным способом зайти на чужой компьютер. Поэтому вам может пригодиться изучение основных комбинаций клавиш, таких как Ctrl+A C (создать окно), Ctrl+A N/P (перейти к следующему/предыдущему окну) и Ctrl+A D (отключить сеанс). Я работал с *GNU screen* много лет, но в итоге перешел на *tmux*.

Имя последнего означает — вы уже догадались — «terminal multiplexer [размножитель терминала]», и *tmux* тоже должен быть в репозиториях вашего дистрибутива. *tmux* не является частью GNU, но больше поддается настройке и предлагает ряд расширенных функций, например, плагинов. Если вам показалось, что на него стоит взглянуть более пристально, читайте дальше.

Сессии и окна

Обычно вы создаете новую сессию *tmux* с помощью *tmux*. Это то же самое, что и *tmux new*. Желая дать вашей сессии имя, назовите ее *tmux new -s mysession*.

Давать имена сессиям бывает удобно, если вы запускаете несколько таких параллельно. *tmux attach* (или, для краткости, *tmux at*) прикрепляется к последней сессии. Чтобы прикрепить вместо этого определенную сессию, запустите *tmux at -t mysession*. Список имеющихся сессий можно вывести командой *tmux ls*. Этот список включает номера сессий, которые можно использовать для прикрепления к определенной, но безымянной сессии.



GNU screen хорошо справляется со своей работой, но смтреть там особо не на что.

Внутри *tmux* состоит из клиента и сервера, причем оба являются одиночным бинарником. Сервер обслуживает сессии и процессы, которые они содержат. Вызывая *tmux attach*, вы запускаете клиент и прикрепляете его к одной из сессий, размещенных на конкретном сервере. Серверы обычно выделяются на каждого пользователя каждого (псевдо)терминала, поэтому другие вкладки или окна в вашем эмуляторе терминала (например, *Konsole*) другие сессии видеть не будут. Однако можно переключаться между сессиями на одном сервере с помощью Ctrl+B (и). Ctrl+B — это так называемая «клавиша префикса», настройка по умолчанию, которую вы можете изменить. Кому-то может больше понравиться клавиша обратного апострофа — “’”.

» ПУСТЬ ПРОГРАММЫ ДЛЯТСЯ

Поддержание работы программ по выходу из сессии, возможно, является наиболее распространенным вариантом использования *tmux* и *GNU screen*. Однако есть и другие опции для достижения этой цели.

Когда вы выходите из сессии, Linux закрывает терминал или, что более вероятно в наши дни, псевдо-терминальное устройство. Процессы, которые и использовали его в качестве управляющего терминала, то есть интерактивные программы, которые вы запустили во время сессии, затем получают сигнал SIGHUP. По умолчанию этот сигнал вызывает завершение процесса. Есть способы обойти настройку по умолчанию: игнорируйте сигнал, если вы автор программы; или просто добавьте к вызову префикс *nohup*. Этот простой инструмент перенаправляет стандартные потоки ввода/вывода с терминала (вывод обычно идет на *nohup.out*), эффективно предотвращая отправку SIGHUP. Запустите программу с &, чтобы она работала в фоновом режиме, и она выживет после прерывания сессии SSH.

А если вы вызвали удаленную программу, которую не собирались запускать надолго, но затем обстоятельства изменились? Конечно, вы можете остановить ее и перезапустить, но есть способ получше. Сначала введите Ctrl+Z, чтобы приостановить запущенную программу и вернуться в оболочку. Затем запустите *disown -h N*, где N — номер процедуры, который оболочка отобразила в квадратных скобках, когда вы остановили команду. Когда оболочка получает SIGHUP, она обычно передает этот сигнал всем процедурам, как работающим, так и остановленным; *disown -h* дает инструкцию не игнорировать процедуру. И, наконец, запустите *bg N*, чтобы возобновить процедуру и перевести ее в фоновый режим.

Независимо от того, как вы попадете в сессию *tmux*, вы, скорее всего, увидите зеленую полосу внизу экрана. Точный цвет и внешний вид могут отличаться в зависимости от имеющихся в вашей установке плагинов *tmux*, но суть всегда одна и та же. Нижняя строка перечисляет «окна» в сессии *tmux*. Окна запускают интерактивные процессы — как правило, оболочки.

Большинство нажатий на клавиши идут прямо в эти интерактивные процессы. Чтобы перейти в *tmux*, введите **Ctrl+B** или иную определенную вами клавишу. Для этого **Ctrl+B** с создает новое окно, а **Ctrl+B 0–9** переключается на окно под заданным номером. Если у вас более 10 окон, то нажатие на **Ctrl+B** позволит вам ввести желаемое число, а нажатие **Ctrl+B w** выведет меню. Чтобы переименовать окно, используйте **Ctrl+B**. Можно также переименовать сессию из *tmux* с помощью **Ctrl+B \$**.

Простейший способ закрыть окно — остановить работающий внутри него процесс. Если это оболочка, вполне подойдет **exit**. А если процесс обнаглел и отказывается выключаться, можете принудительно закрыть окно с помощью **Ctrl+B &**. В этом случае *tmux* просит подтверждения. По закрытию последнего или единственного окна в сессии *tmux* сам останавливает свою работу. И всегда можно отсоединить сессию, оставив работать свои драгоценные процессы, с помощью **Ctrl+B d**.

Беспроблемные панели

Пока что *tmux*, вероятно, кажется вам довольно примитивным. Однако всё станет куда интереснее, когда вы узнаете, что можно и дальше разделить каждое окно на прямоугольные области, именуемые «панелями [pane]».

Панели облегчают многие варианты использования. Например, вы можете писать код и одновременно держать открытой map-страницу для функций, которые вы используете. Иногда полезно просматривать журналы сервисов, выполняя настройки конфигурации. Некоторые из нас, в том числе и я, предпочитают группировать связанные сессии SSH в одном окне, хотя это по большей части дело вкуса.

Чтобы создать панель, вы разделяете текущее окно либо по вертикали с помощью **Ctrl+B |**, либо по горизонтали с помощью **Ctrl+B %**. Это не идеальные мнемоники, и рекомендуется их изменить на **Ctrl+B l** и **Ctrl+B -** (т.е. вертикальные и горизонтальные полосы). Стрелки **Ctrl+B** перемещают вас на панель слева, справа, сверху или снизу от текущей, и вы также можете набрать **Ctrl+B o** и **Ctrl+B ;** для перемещения назад и вперед между панелями. Если вы хотите развернуть панель в полное окно, нажмите **Ctrl+B !** сделает именно это. В порядке альтернативы вы можете временно «увеличить масштаб» текущей панели с помощью **Ctrl+B z**. Первое нажатие клавиши расширяет панель

до размера окна, а второе возвращает его в нормальное состояние. Я часто использую это, когда в одной из моих SSH-консольей имеется длинный вывод.

Когда вы разделяете окно, полученные в результате панели будут одного размера, но это не обязательно. **Ctrl+B Ctrl+стрелка** увеличивает или уменьшает область в соответствующем направлении символ за символом, тогда как **Ctrl+B Meta+стрелка** делает то же самое, но большими шагами (пять символов). Клавиша **Meta** не встречается на большинстве современных клавиатур, но обычно она соответствует одной из клавиш **Alt**.

Помимо изменения размера, вам может показаться полезным изменить расположение панелей в окне. **Ctrl+B Ctrl+o** разворачивает их вперед, **Ctrl+B Meta+o** делает то же самое, но в противоположном направлении. **Ctrl+B {** и **Ctrl+B }** меняет местами текущую панель с предыдущей или следующей, соответственно.

Панель закрывается так же, как и окно. Остановите либо процесс, работающий в ней, либо саму панель — с помощью клавиш **Ctrl+B x**.

Копирование и вставка

На самом деле, *tmux* — это не просто интересный менеджер окон. В нем есть скрытые клады, и для начала давайте рассмотрим буфер истории. Это правда, что большинство графических эмуляторов терминала поддерживают прокрутку, но она может не сработать, если вы запустите *tmux* внутри них. Поэтому в *tmux* реализован собственный механизм прокрутки, а также средства копирования и вставки. Последние являются автономными и не полагаются на X или любой другой буфер обмена в Linux.

Попробуйте сами: создайте команду, у которой большой вывод, например, **ls -lR**, затем нажмите **Ctrl+B [**. Строки перестанут мелькать, и вы сможете прокручивать вывод, используя клавиши стрелок, **PgUp** и **PgDown**. На самом деле, *tmux* может эмулировать привязки клавиш *Vi* или *Emacs*, поэтому соответствующие навигационные клавиши (скажем, «**h**» и «**j**») также доступны, если эта функция включена. Конечно, всё это работает и без программы, которая выплевывает строки текста на заднем плане. Нам просто надо было сгенерировать вывод для прокрутки.

Включив привязки клавиш *Vi* (см. ниже), переключитесь в режим копирования с помощью **Ctrl+B [** и затем введите **?**. Вы увидите всплывающее окно, которое позволит вам осуществлять поиск по истории «up [вверх]» (назад). «**/**» делает то же самое, но осуществляет поиск «down [вниз]». «**n**» переходит к следующему событию, «**N**» вернет вас к предыдущему.

Чтобы скопировать текст, начните выделять его пробелом. Перемещайтесь (помните, большинство клавиш *Vi* работают!), затем нажмите **Enter**. Если вас интересует, можно ли вместо этого использовать «**v**» и «**y**», то ответ — да, но это должно подождать до следующего раздела. Копирование выделения завершает режим копирования; чтобы выйти без выделения, просто нажмите **q**.

По вашей команде

Теперь, когда вы видите, что *tmux* похож на *Vim*, вас не должно удивлять наличие у него собственной командной строки. Вы вызываете ее с помощью **Ctrl+B :**. Поддерживается история команд и автодополнение по **Tab**. В *tmux* нет встроенной справки (кроме **Ctrl+B ?**, которая показывает текущие привязки клавиш), но если вы неправильно введете команду, *tmux* за доли секунды покажет краткую сводку. Если вы не можете прочитать его так быстро, **Ctrl+B ~** покажет самое последнее сообщение.

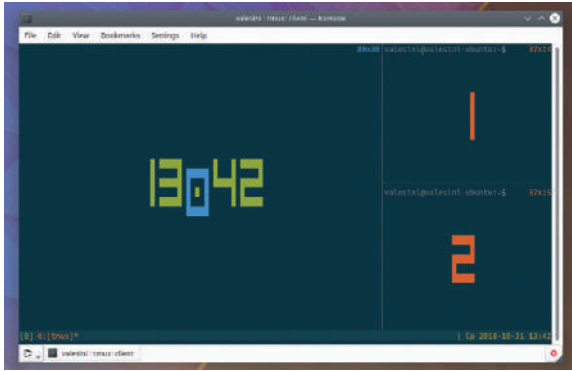
Многие команды добываются из файла **~/tmux.conf**, поставляемого при запуске. Вот что может там присутствовать у обычного пользователя *Vim*:

```
set-window-option -g mode-keys vi
bind-key -T copy-mode-vi 'v' send -X begin-selection
bind-key -T copy-mode-vi 'y' send -X copy-selection
```

На первый взгляд *tmux* не очень отличается от *GNU screen*. Но он, по крайней мере, признает, что ваш монитор может работать с цветом!

```
File Edit View Bookmarks Settings Help
(0) - 0: 3 windows (attached)
(1) |> 0: bash (1 panes) "valesini-ubuntu"
(2) |> 1: bash- (1 panes) "valesini-ubuntu"
(3) |> 2: bash* (1 panes) "valesini-ubuntu"

2 (sort: index)
valesini@valesini-ubuntu:~$
```

Эта схема панелей именуется «главная вертикаль [main-vertical]». Цифры — это индексы панелей. Обратите внимание на часы на фоне: я вывел их с помощью `Ctrl+B t`.

В этом фрагменте кода мы включили привязки клавиш в режиме *Vi* и заставили клавиши “v” и “y” отправлять в *tmux* команды `begin-selection` и `copy-selection`, находясь в режиме копирования. Пока что большинство привязок клавиш были на самом деле ярлыками для соответствующих команд *tmux*.

У некоторых команд нет соответствующих привязок клавиш по умолчанию. Хорошо известный пример — команда, заставляющая поменяться местами два окна:

```
swap-window -s 1 -t 0
```

Она превращает первое окно во второе и наоборот (*tmux* ведет отсчет, начиная с 0). Здесь вы обращаетесь к окнам по их индексам. Если вместо этого вам надо поменять местами панели, вы будете использовать команду `swap-pane`. `Ctrl+B q` (кратко) показывает нужные вам индексы панелей.

Однако это не единственный режим обращения. К окнам можно обращаться, используя имена или даже подстановки оболочки (`bash*`) для соответствия этим именам. Для панелей можно использовать относительное положение, например, `{left-of}` [слева], `{previous}` [предыдущая] или `{next}` [следующая]. Два последних работают и для окон, и их можно далее сократить до `+` и `-`, соответственно, или поставить перед ними смещение `(+2)`. Предположим, активно было окно 2; значит, приведенная выше команда может быть переписана как

```
swap-window -t -2
```

Следующая команда меняет местами текущую панель с ее соседкой слева:

```
swap-pane -t {left-of}
```

» РАСШИРЯЕМ TMUX

У *tmux* много функций, но всегда есть место для большего количества. Поэтому *tmux* поддерживает плагины, которые добавляют функции. Они доступны на <https://github.com/tmux-plugins>.

Плагин *tmux-sensible* предлагает функции по умолчанию, которые нужны каждому пользователю *tmux*. Если какая-то из них вас беспокоит, можете поднять этот вопрос на Github. *Tmux-sensible* не отменяет настроек в `~/.tmux.conf` и позволяет не загромождать этот файл.

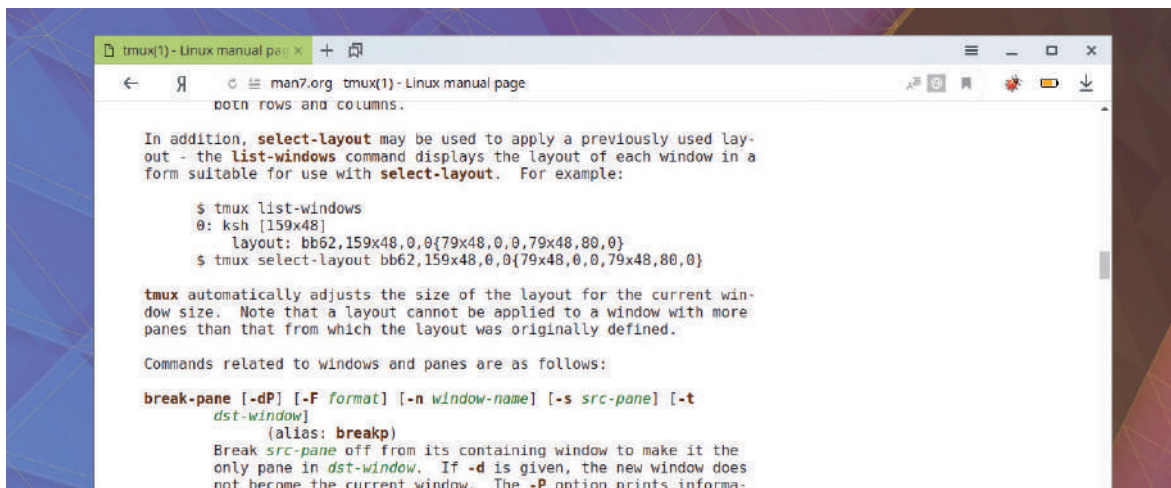
И у *tmux*, и у *X* имеются буферы обмена, и синхронизировать их вполне естественно. Именно это делает *tmux-yank*. Настраивать ничего не надо: вы просто устанавливаете его, и всё.

Tmux не может сохранять и восстанавливать сессии. Это становится проблемой для тех, кто не перезагружает свой компьютер неделями, а потом вдруг внезапно решает это сделать. Плагин *tmux-resurrect* добавляет две новые привязки клавиш: `Ctrl+B Ctrl+S` для сохранения текущей сессии и `Ctrl+B Ctrl+O` для ее восстановления. Опционально *tmux-resurrect* может также восстановить панели, включая их точную схему и даже запущенные программы. Родственный плагин, *tmux-continuum*, делает сохранение и восстановление сессий автоматическим.

Хотя установить плагины вручную несложно, *Tmux Plugin Manager (TPM)* делает эту задачу легкой донельзю. Во-первых, вы клонируете репозиторий и настраиваете `~/.tmux.conf` в соответствии с Readme. После этого добавление нового плагина сводится к его перечислению в `@plugin`, а нажатие `Ctrl+B I`. `Ctrl+B U` обновляет все выделенные плагины.

Кстати о панелях: иногда удобно размещать их в заранее определенных схемах, например, равномерно по горизонтали [`even-horizontal`], по главной горизонтали [`main-horizontal`], точно так же по вертикали или мозаикой [`tiled`]. Для этого есть привязки клавиш (`Ctrl+B Meta+Цифра`), хотя, возможно, вам придется немало потрудиться, чтобы заставить их работать в графическом терминале: `Alt+Digit` несколько превышает лимиты. Поэтому, например, `select-layout main-horizontal` поможет вам избежать изменения размеров вручную. Попробуйте эти схемы сами, чтобы лучше понять, как они работают.

Если вы проводите достаточно времени в терминале, графическом или текстовом, то вам определенно стоит познакомиться с *tmux*. Он поможет не загромождать рабочую область, и у него много плагинов, повышающих производительность. Просто найдите те, которые будут «вашими», и очень скоро начнете дивиться: как это вы без него обходились? **LXF**



tmux понимает пару десятков команд; ман-страница `tmux(1)` перечисляет их все, вместе с аргументами и привязками клавиш.

Red Hat Enterprise Linux

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

Самая популярная в мире Linux платформа для бизнеса

Обеспечивает высокую производительность, надежность, масштабируемость и безопасность

Сертифицирована ведущими производителями оборудования и разработчиками ПО



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

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

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



ГНУ/Линуксцентр — Linux-эксперт для вашего бизнеса

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

Red Hat — ведущий серверный дистрибутив Linux

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

Специальное предложение для читателей Linux Format!

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



Москва
+7 (499)

283-86-06

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Учебники



Евгений Балдин
Подтвердивший
свою квалификацию
физик.

» СВОБОДНАЯ ФИЗИКА

Фу: Если поделить это на три части, то сколько каждому достанется?
Мугэн: Без вас двоих мне достанется все.
«Самурай Чамплу»

В декабре компания Nvidia добавила к уже открытым на GitHub исходникам физического движка PhysX свободную “3-clause BSD” лицензию. PhysX наряду со все еще проприетарным движком Havok — самые популярные, хотя их и поджигает снизу еще один свободный движок, Bullet.

Игровые физические движки нужны для более-менее разумного в плане потребления ресурсов моделирования разного рода физических явлений. То, что в реальном мире нам кажется естественным, например, что твердые тела при столкновении не проникают друг в друга, а лишь деформируются, в случае компьютерного моделирования является не такой уж и тривиальной визуализацией. Более того, до сих пор та же игровая физика деформации твердого тела в лучшем случае зиждется на линейных приближениях, будто бы тела состоят из шариков, соединенных пружинками. Это, очевидно, весьма сильно отличается от реального состояния дел, так как данный процесс сильно нелинеен и зависит от структуры самого тела. Подобные «неестественности» легко заметны на игровых экранах. Однако надо же с чего-то начинать, и как минимум для визуализации (не для расчетов!) подобные движки будут полезны весьма и весьма долгое время.

Очень хорошо, что Nvidia решила освоить свой движок, не дожидаясь, пока свободный Bullet победит его по всем фронтам. Всегда интересно посмотреть на конкурентное свободных идей и реализаций.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...

ТЕРМИНАЛ

Скрывать данные 56
Для **Шашанка Шармы** одним из простых способов обрести душевное спокойствие является применение шифрования.

LIBRE OFFICE

Редактировать PDFы 58
А вы и не подозревали, что такое возможно... но **Нейт Дрейк** показывает, что наш любимый офисный пакет способен на чудеса.

СЕРВЕР

Запускать сервер 60
Скромняга **Матс-Таге Аксельссон** привык не занимать много дискового пространства, поэтому запускает мини-сервер из командной строки.

FLATPAK

Устанавливать пакеты 64
Flatpak убедил **Аарона Питерса** отказаться от привычных Deb-пакетов. Это и формат пакета, и инструмент установки и управления пакетами.

GNUPLT

Интерпретировать данные 68
Джон Лэйн напоминает, что картинка стоит тысячи слов. Глядя, как унылые столбцы цифр превращаются в красивый график, нельзя с ним не согласиться.

SCRIBUS

Верстать журналы 72
В *Scribus* вы просто щелкаете по чему-то, перетаскиваете это, и всё готово, утверждает **Джон Найт**. Так что дерзайте.

АКАДЕМИЯ КОДИНГА

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

HTTPS

Переделаем сайты 76
Небезопасные сайты на WordPress и Drupal **Михалис Цукалос** конвертирует в защищенные, пользуясь TLS 1.3.

WEB

Боты не пройдут 80
Кент Ельчук создает web-формы, застрахованные от ненавистного всем спама — по крайней мере, частично.

GO

Судуку на языке Go 84
Забавы обычных смертных не чужды **Михалису Цукалосу**, но и их он умеет подчинить законам программирования.

ТОМБ

Шифрование по-быстрому

Соккрытие данных трудновато для простых смертных в нашем взаимосвязанном мире, но это не проблема для мастера ниндзя *Bash*, подобного **Шашанку Шарме**.



**НАШ
ЭКСПЕРТ**

Шашанк Шарма — судебный адвокат в Дели и рыный пользователь Arch Linux. Он всегда в поисках техно-сувениров.

Было время, когда для большинства людей самым большим страхом современного мира была простуда в общественном транспорте. Но времена изменились, и теперь наш общий страх — потеря незаменимых данных: либо случайно, либо через их кражу.

Поскольку всё больше людей выбирают цифровые банковские операции с комфортом и удобством мобильного телефона, производители вынуждены прибегать к таким мерам, как датчики отпечатков пальцев, чтобы помочь вам защитить свои секреты на мобильном устройстве. Но как насчет всех других данных на ваших ноутбуках и настольных ПК? Одним из простых способов обрести душевное спокойствие является применение шифрования, чтобы навсегда закрыть все ваши драгоценные секреты, будь то файлы, банковские выписки или семейные фотографии.

Используя *Tomb*, вы можете легко хранить все важные данные в зашифрованных, защищенных паролем каталогах.

```
18:03:22 ~ [linuxlala@playground-budgie:~/tomb-temp] 216$
18:02:49 ~ [linuxlala@playground-budgie:~/tomb-temp] $ tomb lock hidey-hole.tomb
~k hidey-hole.tomb.key
tomb . Commanded to lock tomb hidey-hole.tomb

[sudo] Enter password for user linuxlala to gain superuser privileges

tomb . Checking if the tomb is empty (we never step on somebody else's bones).
tomb . Fine, this tomb seems empty.
tomb . Key is valid.
tomb . Locking using cipher: aes-xts-plain64:sha256
tomb . A password is required to use key hidey-hole.tomb.key
tomb . Password OK.
tomb (*) Locking hidey-hole.tomb with hidey-hole.tomb.key
tomb . Formatting luks mapped device.
tomb . Formatting your Tomb with Ext3/Ext4 filesystem.
tomb . Done locking hidey-hole using luks dn-crypt aes-xts-plain64:sha256
tomb (*) Your tomb is ready in hidey-hole.tomb and secured with key hidey-hole.tomb.key
18:03:22 ~ [linuxlala@playground-budgie:~/tomb-temp] 216$
```

» ПРЯЧЕМ ФАЙЛ КЛЮЧЕЙ В КАРТИНКЕ

Стеганография — это процесс сокрытия информации, такой как сообщения, файлы и изображения, в других файлах или изображениях. Преимущество применения стеганографии в том, что практически невозможно сказать, содержит ли какое-то изображение или файл секретную информацию. Сравните это с обычной криптографией, например с той, которая установлена *Tomb* для создания файла ключей. Из-за самого своего имени, с расширением **.key**, файлы ключей легко идентифицировать — и подвергнуть атакам, хотя бы перебором, чтобы сломать парольную защиту.

Разве не было бы замечательно маскировать свои файлы ключей внутри безобидных с виду файлов? К счастью, *Tomb* позволяет довольно легко упрятать файлы ключей в изображения JPEG, используя утилиту *steghide*, которую вы можете установить из репозитория своего дистрибутива.

После установки вы можете командовать `tomb bury -k hidey-hole.tomb.key imagename.jpg`.

Команда не уничтожает исходный файл ключа, но теперь для блокировки или открытия скрона вместо него можно выбрать изображение: `tomb open -k ~/Pictures/Wallpapers/P1040866.JPG hidey-hole.tomb`.

Извлечь файл ключа из изображения можно с помощью команды `tomb exhume`. Команда ожидает два аргумента. Вы должны задать имя файла ключа, который будет извлечен из изображения — имя исходного файла ключа или иное. Вы также должны указать имя файла изображения, использованного для сокрытия файла ключа: `tomb exhume -k keyfile.key imagename.jpg`. Внедренный файл ключа будет извлечен, только если вы правильно введете пароль, связанный с файлом ключа.

Чтобы получить полный список параметров и понять, почему *Tomb* не любит пространство подкачки, обратитесь к man-странице проекта.

Такие каталоги на языке инструмента называются `tomb [схрон]`, и вы можете создать столько схронов, сколько захотите, и даже увеличить размер имеющегося скрона, если вам нужно больше места для защиты ваших данных.

Установка скрона

Утилита командной строки устанавливается довольно просто, но сначала вы должны получить все ее зависимости. Помимо *zsh*, инструмент также требует *gnupg*, *cryptsetup* и *pinentry-curses*. Последнее из них, *pinentry-curses* — это утилита, используемая для безопасного предоставления паролей и пин-кодов.

За исключением *pinentry-curses* и *cryptsetup*, прочие зависимости, вероятно, уже установлены на вашем настольном Linux. В любом случае, все они имеются в репозиториях программного обеспечения наиболее популярных дистрибутивов и могут быть установлены с помощью `sudo apt install zsh pinentry-curses cryptsetup` или `sudo dnf install zsh pinentry-curses cryptsetup` в Ubuntu и Debian или Fedora, соответственно.

Теперь зайдите на <https://files.dyne.org/tomb> и загрузите последний tar-архив исходника:

```
$ wget https://files.dyne.org/tomb/Tomb-2.5.tar.gz
```

```
$ tar zxvf Tomb-2.5.tar.gz
```

```
$ cd Tomb-2.5
```

```
$ sudo make install
```

При условии, что все зависимости уже установлены, последняя команда приведенного блока кода установит утилиту.

Причина, по которой при установке *Tomb* не требуется запускать команды `.configure` и `make`, как при установке других пакетов из исходного кода, заключается в том, что *Tomb* — всего-навсего скрипт. Вы можете поместить этот скрипт, найденный в распакованном каталоге **Tomb-2.5**, в любое место в вашем `$PATH`, чтобы иметь возможность его выполнить. Запуск `sudo make install` просто копирует этот скрипт в каталог `/usr/local/bin` и позволяет вам получить доступ к справочной man-странице.

Соккрытие секретов

После всех предварительных процедур вы можете начать создавать схрон, где держать свои секреты. К каждому созданному схрону вам также надо создать ключ безопасности для блокировки его содержимого. Каждый файл ключа, в свою очередь, защищен паролем — для дополнительной безопасности. По понятным причинам никогда не храните свои схроны и связанные с ними файлы ключа в одном каталоге.

Первый шаг — выделить место для схрона, и это делается с помощью команды `tomb dig`:

```
$ sudo tomb dig -s 100 hidey-hole.tomb
$ tomb dig -s 150 hidey-hole.tomb
tomb . Commanded to dig tomb hidey-hole.tomb
tomb (*) Creating a new tomb in hidey-hole.tomb
tomb . Generating hidey-hole.tomb of 150MiB
150+0 records in
150+0 records out
157286400 bytes (157 MB, 150 MiB) copied, 1.4048 s, 112 MB/s
-rw----- 1 linuxlala linuxlala 150M Oct 25 17:26 hidey-hole.tomb
tomb (*) Done digging hidey-hole
tomb . Your tomb is not yet ready, you need to forge a key and lock it:
tomb . tomb forge hidey-hole.tomb.key
tomb . tomb lock hidey-hole.tomb -k hidey-hole.tomb.key
```

Как вы можете видеть из блока кода, каждый созданный схрон должен иметь расширение `.tomb`. Параметр команды `-s` вслед за собой ожидает числовое значение, которое обозначает размер схрона в МБ. Помимо сообщения об успешном создании схрона и его прав доступа по умолчанию, проект также информирует вас о командах, которые вы должны выполнить дальше. Ищите их в двух нижних строках блока.

Для создания ключа к указанному схрону используется команда `tomb forge`. Файл ключа имеет расширение `.key`, а в остальном сохраняет то же имя, что и схрон, которые вы создаете командой `tomb dig`. Команда `forge` создаст ключ, используя алгоритм шифрования AES256. Процесс может занять некоторое время, в зависимости от размера схрона. Для генерирования достаточной энтропии также помогает, если вы продолжаете использовать машину. Через некоторое время вам предложат ввести пароль. В зависимости от инструмента *pinentry*, установленного в вашем дистрибутиве, это может быть либо через графическое диалоговое окно, либо через интерфейс, управляемый *curses*.

При создании файла ключа схрон пожалуется и прекратит работу, если обнаружит на вашем жестком диске пространство подкачки. Вы должны либо использовать в команде параметр `-f`, чтобы заставить инструмент генерировать файл ключа, либо запустить команду `sudo swaponoff -a`, отключив подкачку. Затем вы можете снова скомандовать `tomb forge`, чтобы сгенерировать файл ключа.

Но подождите, мы еще не закончили. Теперь, когда у вас есть ключ, вы должны заблокировать свой схрон с помощью `tomb lock hidey-hole.tomb -k hidey-hole.tomb.key`. Эта команда форматирует ваш зашифрованный схрон с файловой системой Ext4.

По умолчанию все схроны смонтированы в каталоге `/media`. Чтобы открыть схрон, надо запустить `sudo tomb open -k hidey-hole.tomb.key hidey-hole.tomb`.

Теперь вы можете добавить все файлы, которые хотите сохранить в секрете, в смонтированный каталог схрона. Вы можете использовать обычные файловые операции, такие как копирование или перемещение, чтобы поместить файлы в ваш смонтированный схрон. Когда закончите, не забудьте закрыть схрон командой `tomb close`.

Работа с tomb

Вы можете запустить команду `tomb list`, чтобы вывести список всех открытых схронов. В дополнение к предоставлению списка, эта команда также сообщит вам, как долго схрон был открыт, его общий размер и доступное свободное место.

Со временем ваши схроны будут расти, чтобы вместить разные файлы. Пока схроны открыты, выполните команду `tomb index`, которая создаст или обновит поисковый индекс, если он уже существует. Создав индекс, вы сможете использовать команду `tomb search <шаблон>` для поиска файлов в открытых схронах. Шаблон может быть любой строкой, и инструмент попытается сопоставить ее с именами файлов в индексе для всех открытых схронов.

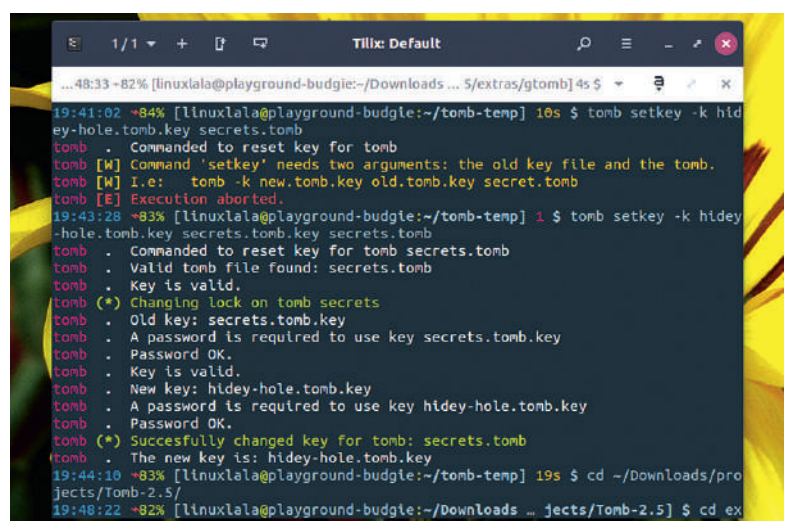
Если вы не хотите индексировать какой-либо конкретный схрон, вы должны сначала открыть его, а затем запустить команду `touch .noindex` из каталога схронов. *Tomb* будет автоматически пропускать индексирование всех схронов, где обнаружит файл `.noindex`:

```
$ tomb index
tomb (*) Creating and updating search indexes.
tomb . Indexing [hidey-hole] filenames...
tomb . Search index updated.
tomb . Skipping [hole] (.noindex found).
tomb . Indexing [secrets] filenames...
tomb . Search index updated.
```

Хотя проект довольно прост, он также поставляется с различными дополнительными инструментами, такими как значок *GTK* в лотке и *gtomb*, который представляет собой скрипт запуска графического интерфейса инструмента. Вы найдете их и многое другое, войдя в каталоге **extras**, который находится в распакованном каталоге **Tomb-2.5**. **LXF**

СКОРАЯ ПОМОЩЬ

Если вы решите заменить файл ключа, связанный с любым схроном, используйте команду `tomb setkey -k newkeyfile.key oldkeyfile.key tombname.tomb`. Поскольку файл ключа защищен паролем, вы можете использовать один и тот же файл ключа для нескольких схронов.



Пароль, используемый для любого файла ключа, можно изменить командой `tomb passwd -k keyfile.key`.

LIBREOFFICE

Редактируем PDFы

Закатав рукава своей моднучей рубашки, **Нейт Дрейк** затевает правку, разбиение и переделку PDF-файлов в *LibreOffice*. Ценнейший сотрудник!



**НАШ
ЭКСПЕРТ**

Нейт Дрейк — независимый журналист в области технологий, специализирующийся на кибербезопасности и ретротехнике.

СКОРАЯ ПОМОЩЬ

Чтобы указать для печати отдельные страницы, пользуйтесь запятой: 2,4,6. Используйте дефис, чтобы указать диапазон: 7-10. Чтобы открыть изображение PDF в средстве просмотра по умолчанию из *LibreOffice Draw*, щелкните правой кнопкой мыши и выберите Edit with External Tool [Редактировать внешним инструментом].

Ubuntu поддерживает экспорт отдельных страниц документа в формат PDF.

Пользователям компьютеров начала 1990-х такие термины, как Farallon Replica и Common Ground Digital Paper, были хорошо знакомы, поскольку различные компании вели ожесточенную войну форматов, чтобы установить общий стандарт для просмотра электронных документов.

Поворотный момент наступил в 1993 г., когда Adobe Systems сделала PDF (Portable Document Format) бесплатным для цифровой общественности. Стандарт теперь поддерживается независимо, и в настоящее время большинство платформ оснащено встроенным просмотрщиком PDF, что по сути гарантирует: документ, созданный на одном компьютере, будет выглядеть точно так же на другом.

Однако, создав PDF, его уже непросто редактировать. Если вы первоначальный разработчик рассматриваемого документа, это не проблема, ведь вам можно просто внести все необходимые изменения в исходный документ, например, в текстовом процессоре, а затем создать новый PDF.

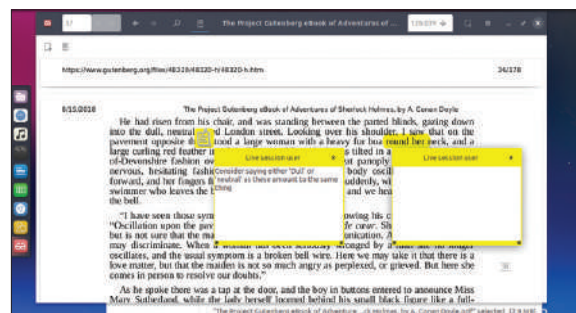
А вот если вы потеряли исходный файл или вам надо внести мелкие изменения, такие как заполнение в электронном бланке ваших данных, вы можете столкнуться с трудностями.

К счастью, для пользователей Linux есть целый ряд программ и интегрированных инструментов, которые помогут вам внести необходимые изменения.

Если вы пользователь Ubuntu, вы, возможно, заметили, что в некоторых приложениях ОС поддерживает создание PDF через окно Print [Печать]. Это полезно, если вы хотите создать PDF только из части страниц документа, например, только из первой страницы.

Чтобы сделать это, дважды щелкните, чтобы открыть PDF в просмотрщике документов по умолчанию в Ubuntu [Document Viewer]. Выберите Print с помощью кнопки Options [Опции] в правом верхнем углу, а затем нажмите Print to File [Печать в файл]. Теперь вы можете указать выбранные вами страницы в разделе Range [Диапазон]. Наконец, выберите Print в правом верхнем углу.

Если вы ранее использовали *Adobe Acrobat Reader* на другой платформе, вы, возможно, заметили довольно удобную функцию, которая позволяет вставлять заметки в различные места



Просмотрщик PDF по умолчанию в Ubuntu поддерживает вставку PDF-аннотаций. Для вставки выберите Add Text Annotation или Add Highlight Annotation.

в PDF. Это очень полезно, если вы корректируете документ или помечаете эссе.

К сожалению, *Acrobat Reader* не работает в Linux; однако собственный просмотрщик документов в Ubuntu поддерживает эту функцию.

Чтобы начать, откройте свой PDF и выберите кнопку Annotate the Document [Примечания к документу] в верхней части окна. Здесь вы можете выбрать Add text annotation [Добавить примечание] или Add highlight annotation [Добавить примечание к выделенному фрагменту] для вставки заметки.

Обильное редактирование

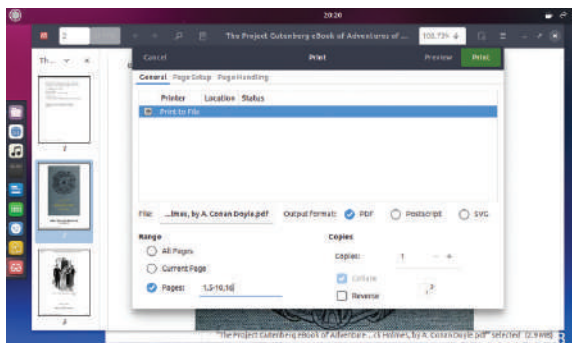
Если вам нужно внести более обширные изменения, помимо извлечения нескольких страниц и добавления заметок, вам потребуется доступ к редактору PDF.

Для этого существуют различные онлайн-сервисы, но, потратив время на эксперименты с различными инструментами, мы обнаружили, что *LibreOffice Draw* лучше всего подходит для ваших потребностей в редактировании.

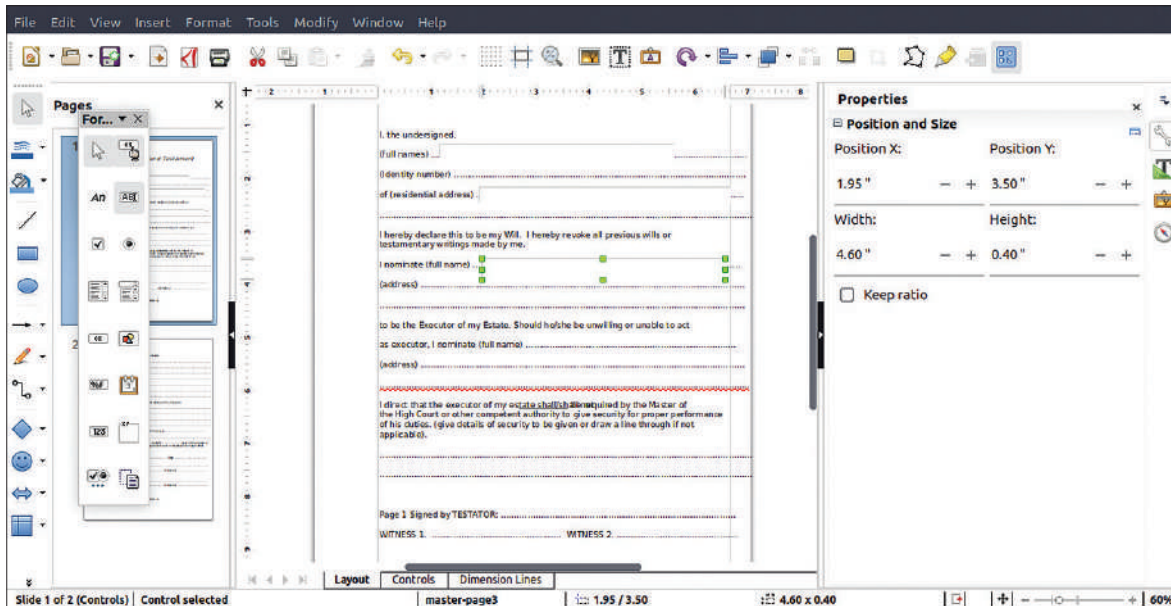
Как следует из названия, *Draw* [Рисование] в первую очередь предназначен для создания графики, но также может открывать и изменять файлы PDF. Приложение является частью пакета *LibreOffice*, который в Ubuntu установлен по умолчанию. Если у вас другой вариант Linux, вы тоже можете легко установить пакет через сайт *LibreOffice* или из командной строки. Для этого выполните действия руководства на следующей странице.

Draw работает аналогично текстовому редактору: вы можете изменять, добавлять и удалять текст в PDF. Приложение будет автоматически соответствовать вашему стилю и форматированию текста. Вы можете посмотреть эти и другие параметры в панели Properties [Свойства] в правой части окна.

Можно также вырезать, копировать и вставлять изображения, щелкнув правой кнопкой мыши. После внесения изменений вы можете экспортировать файл в новый PDF или в более удобный для редактирования формат, например, HTML.



LibreOffice поддерживает вставку в PDF интерактивного содержимого, такого как текстовые поля.



Если вы работаете с бланком в формате PDF для внесения информации, таким как форма завещания, традиционный способ его заполнения — это распечатать, заполнить, отсканировать и отправить по электронной почте.

Но вы можете сэкономить усилия, вставив в соответствующих областях вашего PDF интерактивные текстовые поля. Затем можно будет ввести необходимую информацию и распечатать или сохранить соответственно. Для начала откройте PDF-форму в LibreOffice Draw. Затем включите Form Controls [Элементы управления формой], выбрав меню View [Просмотр] и прокрутив вниз до Toolbars > Form Controls [Панели инструментов > Элементы управления формой]. Слева появится панель инструментов. Отсюда вы можете вставить интерактивные текстовые поля, а также переключатели и флажки.

Чтобы еще больше упростить заполнение форм, рассмотрите возможность создания цифровой копии вашей подписи (см. врезку «Подпись» справа).

Остерегайтесь подводных камней PDF

Основная проблема при редактировании PDF заключается в том, что наша отредактированная версия может не вполне соответствовать форматированию оригинала. Может отличаться положение таблиц и изображений или перестанет работать гиперссылка в документе. По этой причине обычно лучше создать новый PDF в LibreOffice с помощью редактора, такого как LibreOffice Writer. Во всех приложениях LibreOffice вы можете нажать Export as PDF в верхней части окна открытого документа, чтобы создать новый файл.

По состоянию на июль 2017 г. PDF не содержит проприетарного кода, и это отличная новость для пользователей Linux. Однако некоторые файлы могут быть зашифрованы и/или содержать функции DRM, т.е. изменить их нельзя.

В настоящее время в Linux нет встроенной поддержки печати PDF с аннотациями, что может быть неприятно, если вы предпочитаете экрану чтение с бумаги. Если это необходимо, отправьте свой PDF доброму другу, имеющему доступ к Adobe Reader, и попросите его выполнить следующие действия: <https://helpx.adobe.com/acrobat/kb/print-comments-acrobat-reader.html>.

Если вас беспокоит безопасность, рассмотрите вариант применения GPG для шифрования ваших PDF-документов. Преимущество таких действий заключается в том, что вы можете безопасно отправлять защищенные паролем PDF по электронной почте. Программное обеспечение также поддерживает цифровую

подпись файлов, поэтому ваш получатель знает, что они пришли от вас. Узнать больше о GPG можно на <https://gnupg.org/documentation/howtos.html>.

Воссоздание PDF может привести к значительному увеличению его размера, затруднив его распространение по Интернету. Подумайте об использовании сайта вроде [TinyPDF.com](https://tinypdf.com), чтобы урезать свой файл на несколько мегабайт. **LXF**

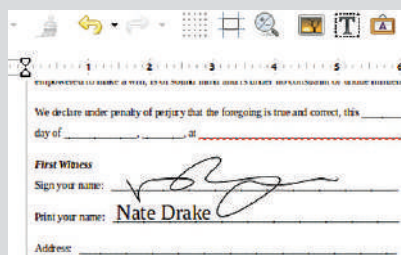
» ПОДПИСЬ

Одна из распространенных причин редактировать PDF — необходимость заполнить и подписать электронный бланк-форму. Это одна из немногих областей, где Linux не справляется, поскольку пользователи Adobe Reader на других платформах могут использовать для создания цифровой подписи его инструменты.

К счастью, не так уж сложно подписать PDF старомодным способом с помощью LibreOffice Draw. Имейте в виду, что вам придется создать свою подпись с помощью мыши, если только у вас не настроены стилус и планшет. Если вы этого еще не сделали, установите LibreOffice, а затем откройте форму PDF.

Затем нажмите кнопку Show Draw Functions [Показать функции рисования] в правом верхнем углу. Функции появятся на левой панели. Щелкните правой кнопкой мыши в любом месте панели и выберите Freeform Line [Рисованная линия], затем поставьте свою подпись на пунктирной линии, как если бы вы делали это ручкой. Вы также можете выбрать Insert Text Box [Вставить текстовое поле], если вам надо напечатать свое имя или нечто подобное.

Если вам регулярно нужно подписывать формы, подумайте о создании факсимиле вашей подписи для вставки при необходимости. В LibreOffice Draw выберите File > New > Drawing [Файл > Создать > Рисование]. Измените размер холста, выбрав Dimension Lines [Размерные линии] под холстом. Повторите эти шаги, чтобы использовать инструмент Freeform Line для создания вашей подписи, затем выберите Export [Экспорт] в верхней части окна. Сохраните свою подпись в формате PNG. В будущем, чтобы добавлять свою подпись, пользуйтесь кнопкой Insert Image [Вставить изображение] в LibreOffice Draw.



Используйте функции рисования в LibreOffice для цифровой подписи и печати своего имени в формах PDF.

HTTP-СЕРВЕР

Запуск небольшого web-сервера

Незачем загромождать квартиру стойками — на запуск web-сервера для тестирования можно настроить свою систему, говорит **Матс-Таге Аксельссон**.



НАШ
ЭКСПЕРТ

Матс-Таге Аксельссон начал свои приключения в Linux, используя IBM Thinkpad и установочную дискету. Он всё еще пытается выжать максимум из устаревшего оборудования, сквалыга.

СКОРАЯ ПОМОЩЬ

Для других языков тестируйте короткие фрагменты с помощью специальных языковых инструментов. Для Python используйте `python -m SimpleHTTPServer`. Он обслуживает только статический контент, созданный Python, но это самое простое решение для Python.

Web-сервер — это увесистый набор программного обеспечения, работающего на стойном сервере, который — что и не удивительно — мало кто из нас имеет дома. Поэтому, желая создать новую web-страницу, вы должны использовать онлайн-сервис для тестирования своего кода CSS, такой как Codepen или JsFiddle.

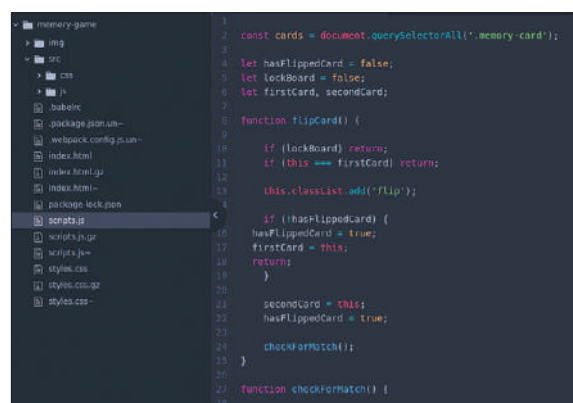
Однако эти службы обычно ограничиваются обработкой небольших фрагментов кода, а разработка с помощью этих средств кода сложнее становится затруднительной. С большей сложностью возникают другие риски, такие как совместимость. Если вы надеетесь стать профессиональным программистом, вам надо уметь использовать несколько инструментов, чтобы убедиться, что вы используете подходящую платформу, на которой вы в конечном итоге развертываете. На нашем уроке вы узнаете несколько приемов для этой цели.

В свое время большая часть JavaScript была встроена в код HTML. Это было удобно, потому что на каждой странице было всего несколько строк. Однако теперь, когда многие пользователи создали фреймворки из своего исходного кода, стали необходимы отдельные программные файлы. Принимаясь программировать на JavaScript, имеет смысл начать с отдельных файлов JavaScript и создать четкую структуру для всего вашего контента, включая HTML, CSS и изображения. Это гарантирует, что вы будете знать, как включить фреймворки, необходимые для выхода за рамки известного уровня программирования “Hello World”.

Для начала создайте типичный “Hello World” и поместите его в файл JavaScript с именем `app.js`:

```
const http = require('http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});
server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

В начале код запускает web-сервер и устанавливает минимальные параметры для его локального запуска. Это самый быстрый способ запустить всё как локальный сервер. Далее вы можете увидеть код, который запускает скрипт “Hello World”. Чтобы запустить этот JavaScript, просто скомандуйте `node app.js`. Это подходит, если вы просто хотите посмотреть, как работает



Вот пример хорошо структурированного файлового дерева для ваших проектов. Для поддержания кода больших проектов это становится жизненно важным.

определенный фрагмент кода. Тем не менее, когда вы участвуете в более крупных проектах, есть другие решения, которые проще в обращении, хотя они требуют модуля. Эти модули запускают сервер, чтобы все файлы в вашем дереве читались так же, как если бы они были на реальном сервере.

Маленький и крепко сбитый

А зачем вообще столько модулей? Ответ такой: вам надо, чтобы модуль был небольшим, но при этом поддерживал все функции, которые вам могут понадобиться. Вы можете найти пример в модуле `react-native-httpd`. Не каждый хочет использовать *React*, но вам нужна поддержка правильных фреймворков, иначе код просто не будет работать. С другой стороны, если у вас есть неиспользуемые элементы, то при развертывании результаты могут отличаться. Основная причина в том, что мы следуем классическому практическому правилу программирования *nix: заставьте каждый программный инструмент делать одну вещь и делать это хорошо.

Одной из наиболее распространенных сред для кода JavaScript является Node.js. Для установки этих пакетов обычно используется упакованная версия. Это может быть не совсем удачным, потому что там обычно не самая новая версия. Вам также может потребоваться несколько версий из-за уровней программного обеспечения сайта. Лучше зайти на <https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions> и найти пакет там.

Скрипт установки изменяет список репозиториях вашей системы, поэтому вы можете продолжать использовать новейшую версию. Вы можете зарегистрироваться на www.npmjs.com, но это件но только тогда, когда вы начнете делиться кодом.

В реестре *npm* вы найдете несколько пакетов для запуска небольшого локального web-сервера. Одни предназначены только для статического содержимого, а другие — только для работы с файлами. Пакет под названием *httpd-server* может делать то же, что и большинство обычных серверов, и, что еще лучше, он относительно небольшой и настраиваемый.

```
$ sudo npm install http-server -g
```

Запустите ваш сервер командой *http-server*. Без параметров сервер запустится с текущим каталогом в качестве основного. Это позволяет легко перейти в репозиторий исходников и запустить всё, что вы создали.

Можно также запустить команду с параметрами, указывающими на правильный каталог. Ваш исходный код должен быть организован так, чтобы вы могли легко выполнить развертывание с тем же набором файлов на реальном сервере. В случае с *http-server* у вас теперь есть готовое решение. Помните, что любые файлы вне каталога, из которого вы начинаете, невидимы для сервера.

Настройка HTTP-сервера

Чтобы всё работало как можно ближе к реальности, некоторые параметры можно подправить. Порт по умолчанию — 8080, но вы можете выбрать собственный порт, если у вас есть другие локальные серверы. Параметр *-g* выберет GZIP-версию любого файла, который вы упаковали и который имел то же имя перед расширением *gz*. Вы также можете изменить расширение HTML по умолчанию, что бывает件но, если вы создаете приложение PHP.

Когда вам нужны безопасные web-страницы, вы также можете употребить SSL и указать сертификат и файл секретного ключа. Без этих двух файлов сайт не запустится, поскольку шифрование件возможно — людям, работающим с защищенными сайтами, это уже件но. Чтобы убедиться, что код всегда обновляется в браузере, вы также можете отключить кэширование, установив время кэширования в *-1*.

Чего вы не можете делать? Ну, поскольку сервер должен быть маленьким, у вас не будет полной поддержки каких-либо сред, кроме Node.js. Пакет также может использовать функцию прокси. Она отправит некоторые запросы на web-страницу, даже если ваш код и данные являются локальными. Имея это в виду, вы можете проверить, как собирать информацию с других сайтов, например, встраивать удаленный контент. Вы можете исправить это, добавив его позже, но прежде чем сделать это, посмотрите другие доступные опции:

```
$ http-server -p 8080 --gzip -c -1
```

Если при использовании этого крошечного сервера вы хотите применять сжатие, следует сжать все файлы до того, как они понадобятся серверу. На более продвинутом web-сервере, например, *Nginx*, вы можете выбрать сжатие по требованию. В любом случае件но иметь возможность измерять прирост скорости от сжатия файлов.

Естественно, это решение является только одним из многих. В этом разделе вы узнаете о возможностях и ограничениях именно его.

Другим вариантом являются web-серверы, имеющие большую гибкость и/или большее количество функций. Пакет *local-web-server* включает более широкий набор функций и возможностей настройки. Чтобы установить его, используйте менеджер *npm*:

```
$ npm install -g local-web-server
```

Сделав это, проще всего будет разместить статический сайт без аргументов — так же, как и с предыдущим пакетом,



Mozilla Foundation имеет фантастический набор учебных пособий на своей web-странице документации. Это один из лучших источников информации о JavaScript в Интернете.

командой *ws*. Порт по умолчанию — 8000, но порт вы можете выбрать сами. Желая обслуживать одностороннее приложение, используйте аргумент *--spa*. В этом пакете вы также можете использовать прокси.

Чтобы обслуживать HTTPS и HTTP2, начните с аргументов *--https* или *--http2* соответственно. Есть встроенные ключи и сертификаты, поэтому вы можете работать без собственных ключей, хотя можно и предоставить свои собственные.

В случае, если вы взаимодействуете с другим сайтом, будь то в Сети или с другой частью вашего сайта, вы можете настроить имитацию его реакции и отправлять отклик на запрос с существующего сайта, используя предварительно подготовленный ответ. Тогда вы сможете проводить тестирование, даже если у вас нет контакта с конкретным сайтом, даже если вы находитесь оффлайн.

Чтобы назначить фиктивные ответы, используйте файл JavaScript, подобный приведенному ниже:

```
module.exports = MockBase => class MockPaths
```

СКОРАЯ ПОМОЩЬ

У Electronic Frontier Foundation есть хорошая статья о шифровании файлов с помощью PGP (<http://bit.ly/2KP3g00>). Только не забудьте придумать длинный пароль к своему частному ключу.

» ONLINE-ИНСТРУМЕНТЫ ТЕСТИРОВАНИЯ

Прежде чем приступить к настройке всего этого на домашней системе, а также если вы хотите поиграть с небольшими фрагментами кода, попробуйте онлайн-решения, такие как Codepen и JsFiddle.

Codepen имеет приятный дизайн, и его легко перемещать по рабочему столу. Для полноты картины у вас есть окно для HTML, окно для CSS и, наконец, окно для JavaScript. В каждом окне также есть параметры для анализа вашего кода различными способами. Есть функции Tidy, Analyse и Show Compiled. Они полезны, если у вас есть плохая привычка путать отступы или вы просто не представляете, в чем ошиблись. Можно выбрать, где показывать ваш код. Три окна с кодом могут быть справа, слева или внизу. Если вы хотите начать собственный проект, у вас есть несколько шаблонов, доступных для начала. Ваш готовый проект можно скачать в виде ZIP-архива.

JsFiddle похож на Codepen, но имеет больше опций просмотра и больше шаблонов. Он также имеет аккуратный раскрывающийся список, в котором вы можете выбрать язык, библиотеку и другие параметры для вашего кода. В JsFiddle у вас также есть функция сотрудничества. Когда вы активируете эту функцию, вы создаете ссылку, которой делитесь с кем-то, кому вы доверяете, и вы можете вместе работать над одним кодом.

Эти инструменты отлично подходят для тестирования новых идей и создания исходного кода. Впрочем, если у вас уже есть код на Github, вы можете напрямую связать его с JsFiddle.



СКОРАЯ ПОМОЩЬ

Если вы не используете Ruby или Python или просто не любите их решения, у вас тем не менее есть варианты. Thttpd был разработан для IoT, но полезен для простых развертываний. Это похоже на наличие демона tftp.

```
extends MockBase{
  mocks () {
    return {
      route: '/paths',
      responses: [
        {
          response: {
            type: 'json',
            body: [
              {name: 'Roslagsleden', goesThrough: 'Roslagen'},
              {name: 'Sörmlandsleden', goesThrough: 'Sörmland'},
              {name: 'Nordkalottleden', goesThrough: 'Norway'},
            ]
          }
        }
      ]
    }
  }
}
```

Сохраните приведенный код, назовите его **mock-example.js** и запустите web-сервер:

```
$ ws --mocks mock-example.js
```

После этого вы можете получить отклик с помощью команды

```
$ curl http://127.0.0.1/paths
```

Теперь вы можете запросить свои данные из локальной системы, для проверки, что ваш код способен обработать ответ.

По мере роста ваших проектов возрастают и ваши требования к тестированию. В помощь проверке функциональности во многих сценариях можно использовать *dev-web-server*, у которого больше функций. Он также поддерживает определение конечных точек API — и с помощью этих функций вы можете вызывать API. У вас к тому же появится возможность определять успех и неудачу конечных точек. Со всей этой функциональностью полную разработку и тестирование можно выполнить локально, делая весь процесс более безопасным и менее подверженным задержкам, которые находятся вне вашего контроля.

Пакет *dev-web-server* также имеет функцию для сценариев при сбоях. Чтобы всё настроить, рассмотрите следующую команду запуска сервера:

```
$ dev-web-server DOMAIN 0.0.0.0 PORT 1234 BASEDIR.\rep\
httpdocs DELAY 2000 ENDPOINTS.\rep\server\my-endpoints.js
```

Как видите, задержка установлена в 2000. Используя этот аргумент, вы можете проверить, как всё будет работать, когда у вашего пользователя большое отставание от web-страницы. Все параметры также могут быть определены файлом **dev-web-server.json** в каталоге запуска. Любые аргументы в командной строке переопределяют файл. Файл примера определяет пути и задержку. Он также определяет, используете ли вы CORS или нет. CORS — это способ, позволяющий программам извлекать данные из других доменов, с которыми они сотрудничают. Это обычно блокируется из-за проблем безопасности.

```
{
  "domain": "0.0.0.0",
  "port": 13002,
  "baseDir": "./test/test-pages",
  "delay": 0,
  "endPointsFilePath": "./api-proxy/api.js",
  "withCORS": "true"
}
```

Другие параметры задают базовый каталог, где вы размещаете файлы web-страницы, и, наконец, файл оконечной точки. В файле оконечной точки вы определяете, как обрабатываются ответы. Эти ответы включают как успехи, так и неудачи. Формат представляет собой файл JavaScript, в котором определены функции. В примере указывается на файл **api.js**, и часть этого файла показана ниже.

```
var Repository = {
  count:0
};
module.exports = {
  '/example': function (req, res, params, sendSuccess, sendError) {
    // Response result is: '{"test":"Cool, success","count":1}'.
    // HTTP code is 200
    sendSuccess(req, res, {
      test: 'Cool, success',
      count: Repository.count++
    });
  },
  '/exampleError': function (req, res, params, sendSuccess, sendError) {
    // Response result is: '{"code":401,"message":"You need to login"}'.
    // HTTP code is 401
    sendError(req, res, 401, 'You need to login');
  }
};
```

На самом деле следует охватить гораздо больше кодов ошибок, а это две ошибки из основных. Кроме того, учтите, что пакет *dev-web-server* имеет две зависимости: *colors* и *object-tools*. Обычно это не проблема, но упомянуть стоит.

Горячая перезагрузка установки

Если вы используете модуль *webpack*, вас заинтересует *webpack-dev-server*. Как следует из названия, пакет предназначен только для разработки. Одной из наиболее интересных функций является функция горячей перезагрузки, которая немедленно загружает все ваши изменения. Часто, когда вы обнаруживаете проблему в своем коде, изменение кода выполняется быстро, но затем надо убедиться, что код действительно используется сервером. Это мелкое, но раздражающее свойство приносит много неприятностей. Кэш браузера и сервера способствуют экономии пропускной способности для пользователей, но когда вы кодируете, это не всегда очевидно, а какой уровень использует ваш браузер, не всегда понятно.

Для непосвященных, модуль *webpack* предназначен для того, чтобы взять ваш проект и его зависимости и упаковать их в один

» СИСТЕМА REACT

React создан для работы с интерфейсами пользователя, поэтому вы должны рассматривать его как часть своего набора инструментов, а не как целое решение.

Чтобы использовать *React*, вы можете рассмотреть, какую логику вам нужно объединить с ним. Объединение *React* с CMS — это один из способов, и для него даже есть web-страница (<https://reactfordrupal.com>). Однако Drupal — не единственный выбор. WordPress имеет API, который упрощает разработку интерфейса.

После установки *NPM* можно искать пакеты, используя встроенный поиск. В результате получается длинный список пакетов, поэтому обязательно выберите *react-dom* для web-страниц и *react-native* — для собственных сред. Список пакетов, использующих *React*, еще длиннее. Фактически, одна только эта длина должна побудить вас изучить *React* и подумать об его применении в вашем следующем проекте. Мысль о том, как часто он используется в реальных установках, усилит вашу мотивацию. *React* является основной движущей силой перехода к консольной работе, к которой движется большинство CMS, и если вы не хотите использовать CMS, вы всё равно можете использовать его для своего обычного приложения. Он идеально подходит для одностраничных приложений, где страница ведет себя скорее как приложение, чем web-страница.

модуль. Это означает, что любой, кто заинтересуется вашим модулем, может загрузить пакет и использовать его как есть, не беспокоясь о зависимостях. В случае *webpack-dev-server* пакет останется в памяти и подается клиенту.

Чтобы проекты *npm* вообще работали, вы всегда должны инициализировать свой каталог. Создастся каталог **node_modules**, содержащий все модули, которые вы будете использовать.

```
$ npm init -y
```

Без *init* пакеты не будут правильно установлены в каталог; вы можете установить их глобально, однако больший контроль у вас будет при локальной установке. Зависимости обрабатывает менеджер пакетов, но ему надо сообщить, что вы ищете. Вы должны убедиться, что в вашей системе доступен модуль *webpack*. Если его нет, установите его с помощью

```
$ npm install webpack --save-dev
```

```
$ npm install webpack-cli --save-dev
```

Аргумент *save-dev* означает, что пакет будет применяться только для разработки. Это важно, потому что одни пакеты вам могут потребоваться для разработки, а некоторые другие критически важны для развернутого сервера. Вы получите несколько предупреждений, но предупреждение о зависимости *fs-events* можете спокойно игнорировать, поскольку оно не поддерживается в Linux.

Затем надо адаптировать файл конфигурации к вашим личным нуждам. В этом файле вашей главной заботой является точка входа. Это файл JavaScript, где *webpack* начинает просматривать ваше приложение. В случае простого приложения файл JavaScript у вас может быть только один, но при увеличении его размера лучше всего иметь много. Отличный пример доступен на GitHub (<https://github.com/mschwarzmueller/yt-webpack2-basics>). Он сфокусирован на *webpack*, но содержит некоторые подробности о сервере web-разработки. Сначала вам надо создать файл **package.json** — в нем вы определите, как будет запускаться пакет:

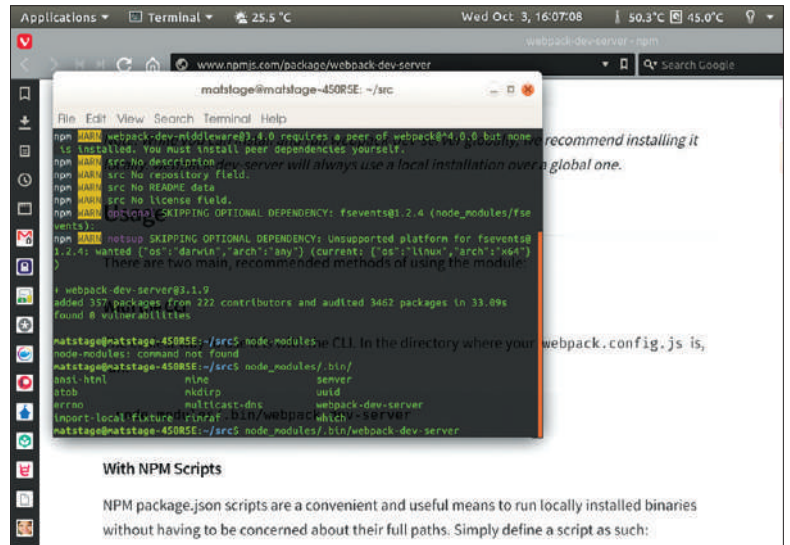
```
{
  "name": "webpack2-basics",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "run-server": "webpack-dev-server --entry ./src/js/app.js --output-filename ./dist/bundle.js",
    "build:prod": "webpack src/js/app.js dist/bundle.js -p"
  },
  "author": "",
  "license": "MIT",
  "devDependencies": {
    "webpack": "^2.2.1",
    "webpack-dev-server": "^2.4.1"
  }
}
```

В каталоге файла **package.json** запустите *npm*. Аргумент *run-server* берется из настроек **package.json**. В этом случае вы можете использовать *build:prod* для создания файла пакета:

```
$ npm run run-server
```

Теперь вы сможете увидеть сайт в своем web-браузере. Порт появится в окне вашего терминала.

Для более расширенного использования *webpack* используйте файл **webpack.config.js**. Вначале его создание может быть отчасти неудобно. Вы всегда можете попробовать этот инструмент: <https://webpack.jakoblinde.no>, который создает файлы конфигурации за вас. Результат будет хорошей отправной точкой.



Webpack-dev-server может использоваться для тестирования и упаковки вашего JavaScript в один пакет. Это значительно ускоряет загрузку готовой web-страницы.

Параллельное выполнение

Сначала ваши web-страницы будут тривиальны, но всё же полезно знать, где искать пакеты, которые могут заставить вашу систему работать параллельно. Имеется несколько решений для этого; два из наиболее известных — ‘forky’ и ‘throng’. Из них ‘forky’ является более продвинутым, но в функции JavaScript вам следует активировать оба. ‘Forky’ имеет один главный файл и несколько файлов процессов, а ‘throng’ — только рабочие файлы и процессы.

Какой бы вариант вы ни выбрали, рекомендуется рассмотреть эти вопросы задолго до того, как вы подумаете, что они могут вам понадобиться.

При более профессиональном подходе вы должны научиться создавать контейнерную среду, чтобы вам была доступна правильная система, независимо от того, где вы в конечном итоге развернете свой код. Чтобы быть абсолютно уверенным, вам надо создать образ сайта, на который выполняется развертывание. Сначала может показаться, что вы получили сервер-гигант, которого на нашем уроке мы стремились избежать, но это не совсем так!

Если вы используете систему изоляции, такую как Docker (см. стр. 40), то вы можете запустить всё это, не затрагивая свою систему, и созданный вами «web-сервер» будет действовать как настоящий. Хорошей отправной точкой для web-серверов является использование минимального образа — меньше нескольких сотен килобайт. Даже Ubuntu One с *Nginx* сверху довольно мал, но при использовании Docker управление пакетами — это еще одна проблема. Это история на другой раз...

Что мы узнали?

На этом уроке вы узнали о базовых методах запуска web-сервера на локальном компьютере для тестирования. Крошечный специализированный сервер не будет перегружать ваше локальное оборудование — для работы будет достаточно обычного ноутбука. *Npm* предлагает множество отличных решений для этого вида работ.

Для более крупных работ вам, скорее всего, понадобится использовать контейнерную технологию, такую как Docker, чтобы сделать то же самое. Тем не менее, для раннего обучения и экспериментов *npm* вполне вам подойдет. **LXF**

СКОРАЯ ПОМОЩЬ

Если ваш любимый язык — Ruby, вы можете запустить простой web-сервер с помощью *ruby -run -ehttpd.p 8000*. Стандартный web-сервер, *Webrick*, является более продвинутым и умеет обрабатывать SSL, CGI и многие другие web-функции.

FLATPAK

Получайте новейшее ПО с Flatpak

Аарон Питерс много лет назад пошел ва-банк с форматом Deb-пакета Ubuntu — и уже не оглядывался назад. Тем не менее, всегда есть шансы передумать...



**НАШ
ЭКСПЕРТ**

Аарон Питерс хотел бы быть программистом, но, к сожалению, вынужден довольствоваться указаниями программистов, что делать, в форме документации к ПО.

СКОРАЯ ПОМОЩЬ

Flatpak идентифицируются по схеме своего рода «URL наоборот», такой же, которую вы можете использовать при именовании пакетов кода. Сначала идет домен верхнего уровня, за ним домен, затем поддомен, и т.д.

Linux добился больших успехов в области совместимости программного обеспечения. Или, точнее, установка программного обеспечения стала более единообразной в различных дистрибутивах и их собственных форматах пакетов. Но всё еще требуется сложная работа создателей пакетов. Им надо взять приложение и настроить его под конкретные требования разных дистрибутивов и даже разных их версий. Однако появилось несколько новых универсальных форматов пакетов, которые обещают: «упаковал однажды — работает везде» (во всяком случае, в Linux).

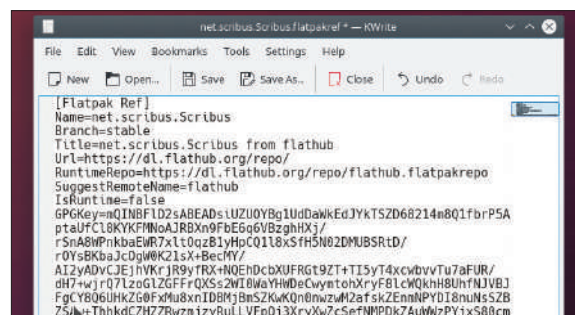
Flatpak (<https://flatpak.org>) входит в число фигурантов Маянка Шармы, рассмотренных в **LXF234**, и это говорит о многом. На нашем уроке мы рассмотрим *Flatpak*, узнаем, как его использовать, и рассмотрим две важных разработки.

Режим работы

Flatpak является и форматом пакета, и инструментом для установки и управления этими пакетами. Многие форматы пакетов Linux работают с концепцией зависимостей, означающей, что разработчик может определить, что пакет A зависит от пакета B. Когда пользователь пытается установить мой пакет A, система распознает зависимость и устанавливает пакет B вместе с ним. Так, *apt install* устанавливает одно приложение и получает сотни пакетов для загрузки и установки. *Flatpak*, напротив, предоставляет вам и приложение, и всё необходимое для работы в независимом от дистрибутива формате. У этого подхода есть свои плюсы и минусы:

Плюсы	Минусы
Независимость от дистрибутива	Использует дополнительное пространство и дублирует файлы.
Простота установки	Установка как пользователем root, так и обычным пользователем.
Нет конфликтов с локальной системой	Невозможно установить серверные приложения.

Но основная идея *Flatpak* привлекательна. Найдя программу, вы легко установите ее и запустите, не беспокоясь, что нечто иное в вашей системе использует несовместимую версию той же библиотеки. Кроме того, установка приложений в виде Flatpak позволяет одновременно использовать и сопоставлять программы так, как это невозможно с обычной системой пакетов, например:



Файл flatpakref содержит всю информацию, необходимую для поиска приложения, определения его зависимостей и установки всего.

- Использовать новейшую версию приложения, сохраняя при этом и стандартную версию. Пример: *LibreOffice*, показанный на скриншоте на стр. 66, работает одновременно в версиях 5 и 6.
- У вас есть проблемы с безопасностью, и вы чувствуете себя лучше с программой, работающей в песочнице. Пример: *Flash Player* от Adobe.
- Использовать более свежую, более стабильную и/или более многофункциональную версию приложения, чем предоставляет ваш дистрибутив. Пример: *Calibre*, которая сильно отстает даже на Ubuntu 16.04.

Тем не менее, как мы видели в **LXF234**, существуют другие форматы, которые предоставляют аналогичные преимущества. К ним относятся формат Snap в Ubuntu (www.snapcraft.io)

и AppImage (<https://appimage.org>). Хотя все они выполняют схожие функции, в сравнении *Flatpak* имеет фору по нескольким пунктам:

- *Snapcraft* — это проект Canonical, и хотя он доступен в других дистрибутивах, но в первую очередь предназначен для Ubuntu. Если вы используете другой дистрибутив или просто предпочитаете программную систему, не контролируемую одной коммерческой организацией, *Flatpak* — лучший выбор, чем Snap.
- *AppImage* использует способ просто загрузки и запуска одного файла. Это просто, но не обязательно легко, особенно для новых пользователей. Для начала, вам надо установить загруженный файл AppImage как исполняемый, прежде чем вы сможете запустить приложение. Кроме того, нет встроенного механизма для обновления ваших AppImage. Проект предлагает внешний инструмент для этого, но вам еще необходимо загрузить и настроить его.

Несмотря на все эти разговоры о плюсах и минусах, вы можете одновременно использовать любой (или все!)

из вышеуказанных форматов на своей машине. Это противоречит логике, не так ли? Как разработчики добились этого? Давайте рассмотрим формат Flatpak немного подробнее, чтобы выяснить это.

Распаковка Flatpak

Как и большинство проектов с открытым исходным кодом, *Flatpak* для достижения своих целей становится на плечи гигантов. Две ключевые технологии, лежащие в основе *Flatpak* — это *ostree* и *bubblewrap*.

Устанавливая Flatpak, вы устанавливаете «комплект» *ostree*, содержащий приложение. Сайт проекта описывает его как нечто вроде «Git для бинарников операционной системы». В дополнение к другим преимуществам, данный подход обеспечивает атомарные обновления — это такой витиеватый способ сказать, что обновляется только то, что нужно обновлять. Он также обеспечивает механизм отката.

Предположим, у вас есть программа с файлом 1 и файлом 2. Обновление предоставляет новую версию файла 2, который по факту обозначен как файл 3. Программа будет работать довольно успешно, используя файл 1 и файл 3, пока вы не захотите откатить версию. Но вам не надо делать всю операцию удаления или переустановки. Вам просто нужно изменить приложение, чтобы оно использовало файл 2 вместо более нового файла 3 (который всё ещё существует на тот случай, если вы хотите выполнить «откат»).

Другой вспомогательной технологией является *bubblewrap*, программный уровень, который, в свою очередь, использует функцию пользовательского пространства имен ядра Linux. Это позволяет учетным записям обычных пользователей выполнять роль учетных записей *root* в контейнерах, включив песочницу *Flatpak*. При запуске приложение *Flatpak* вызывает подсистему, состоящую из приложения *Flatpak* и определенного количества системных ресурсов для использования в песочнице. В вики *Flatpak* на GitHub (<https://github.com/flatpak/flatpak/wiki/Sandbox>) есть страница, которая подробно описывает содержимое песочницы — вот некоторые ключевые свойства:

- Среда(-ы) выполнения приложения монтируются в песочнице в **/usr** (структура которой более подробно описана во врезке справа).
- Само приложение монтируется в **/app**.
- Конфигурации из каталога хоста **/etc** монтируются в **/usr/etc** песочницы.
- Из песочницы видны только процессы в песочнице, например, каталог **/proc**.
- Такие ресурсы, как значки, шрифты и т.д., доступны через “bind-mounts” или путем монтирования одного файла/каталога (вне песочницы) в другое место в файловой системе (внутри песочницы).
- В целом, песочница имеет доступ только к ресурсам, строго необходимым ей для работы, причем только с минимально допустимыми привилегиями.

Теперь, если вы еще не слишком запутались, мы добавим еще один уровень сложности. Для установки большинства программ в системе Linux требуются права *root*, но Flatpak могут устанавливать и обычные пользователи. В следующем разделе мы рассмотрим разницу между этими «пользовательскими установками» и более традиционной «системной установкой».

Система против пользователя

При установке программного обеспечения бывалые пользователи Linux привыкли предоставлять пароль *root*. Это программное обеспечение попадает в каталог за пределами вашего **\$HOME**, такой как **/usr/bin**, **/opt** или, возможно, даже **/usr/local**. Но, как упоминалось в нашем списке плюсов, Flatpak поддерживает

```
aaron@acp-lubuntu-vm: ~$ flatpak search calibre
Application ID      Version Branch Remotes Description
io.gtklab.jstest_gtk 62f6e2d stable flathub jstest-gtk - A graphical joys
tick tester
com.calibre_ebook.calibre 3.32.0 stable flathub calibre e-book management
com.hamrick.VueScan 9.5.94 stable flathub Scanning software for 5000+ s
canners
aaron@acp-lubuntu-vm: ~$
```

Результаты команды поиска flatpak дают идентификатор обратного домена из трех частей, который вам понадобится для установки.

» АНАТОМИЯ FLATPAK

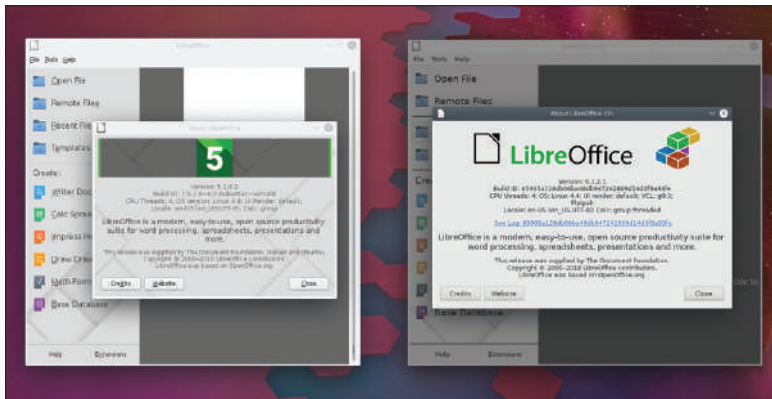
Установка Flatpak создаст файловую структуру для программы и ее сред выполнения, а также некоторые метаданные. Базовая структура состоит из четырех каталогов верхнего уровня в **\$HOME/local/share/flatpak** или **/var/lib/flatpak**, в зависимости от того, была ли это системная или пользовательская установка.

Первым из них является **repo**, содержащий информацию об установленных и внешних приложениях и другие метаданные. Каталог **app** — это место, где находятся сами приложения. У каждой программы есть подкаталог с именем, состоящим из трех частей (например, **org.gnome.GEdit**). Существуют дополнительные подкаталоги, выделенные для архитектуры (например, **x86_64**), номеров версий и связанных с ними контрольных сумм.

Среда выполнения управляется таким же образом, но хранится отдельно в каталоге **runtime**. Это связано с тем, что они могут использоваться несколькими приложениями. Последний каталог — **exports**, содержащий элементы, связанные с программами Flatpak, но находящийся вне песочницы. К ним относятся такие вещи, как значок приложения и файл **.desktop** (который появится в меню приложений на рабочем столе), а также системные ссылки, такие как подключение к оконной системе. Каталог **exports** содержит ссылки на вещи внутри Flatpak и является одной из тщательно контролируемых точек соприкосновения между хост-системой и песочницей Flatpak.

Ниже показан экспорт команды **tree** после установки *LibreOffice* Flatpak:

```
$ aaron@acp-demon:/var/lib/flatpak$ tree -L 2
app
├── org.libreoffice.LibreOffice
├── appstream
├── flathub
├── exports
├── bin
├── share
├── repo
├── config
├── extensions
├── flathub.trustedkeys.gpg
├── objects
├── refs
├── state
├── tmp
├── runtime
├── org.freedesktop.Platform
├── org.freedesktop.Platform.ffmpeg
├── org.freedesktop.Platform.Locale
└── org.libreoffice.LibreOffice.Locale
```



Две различных версии *LibreOffice* работают одновременно на одной машине. Волшебство? Нет, просто Flatpak.

СКОРАЯ ПОМОЩЬ

Если вы только что установили Flatpak и свою первую программу, вы можете заметить, что в системном меню нет ее значка. Решение состоит в том, чтобы выйти, а затем снова войти в систему. Это перезагрузит переменную среды `$XDG_DATA_DIRS`, к которой Flatpak добавил свои пути.

возможность установки программы для конкретного пользователя, а не для всей системы. Это работает, потому что обычный пользователь имеет доступ к необходимым ресурсам для запуска программы, и песочницы, созданные пользователем, получают такой же доступ.

Когда вы запустите любую из команд установки, которые мы увидим через несколько минут, вам предложат ввести пароль. Дело в том, что по умолчанию *Flatpak* будет выполнять системную установку. Это означает, что программа будет доступна всем пользователям, а ее файлы будут храниться в `/var/lib/flatpak`.

С другой стороны, если вы используете с командой флаг `--user` — он установится только для вашей учетной записи (пароль у вас запрашивать не будут). Файлы в этом случае будут находиться в `$HOME/.local/share/flatpak`. Это полезно, если у вас нет учетной записи администратора на компьютере, или, возможно, если вы хотите попробовать новейшую версию программы, которая установлена в другом месте системы.

Пользовательская установка возможна благодаря устройству песочницы. Думайте об этом так: как пользователь, вы в состоянии запускать приложение. Это приложение имеет права (через вашу учетную запись) на создание экранного окна, запись файлов на диск и доступ к сетевым подключениям. Когда вы устанавливаете Flatpak как пользователь, любые ресурсы хост-системы

» СТАНОВЛЕНИЕ FLATPAK

Эта статья увидела свет, когда Flatpak порадовал двумя важными событиями. Первым является релиз v1.0 проекта, который вышел 20 августа 2018 г. Как и подобает этому этапу, пресс-релиз (<https://flatpak.org/press/2018-08-20-flatpak-1.0>) назвал новую версию «полнофункциональной», со всеми необходимыми ресурсами как для разработчиков, так и для пользователей. Поэтому в дополнение к командам *Flatpak*, о которых упоминается в этой статье, и поддержке большинством основных менеджеров пакетов с графическим интерфейсом, есть также инструменты, помогающие разработчикам упаковать свои приложения в виде Flatpaks. Если вы программист, обратитесь к разделу “Build applications” справки *Flatpak* (`flatpak --help`), чтобы узнать, какие утилиты есть в вашем распоряжении.

Но запуск другой инициативы, возможно, более интересен. Проект WinePak (www.winepak.org) использует *Flatpak*, чтобы облегчить установку программ Windows. Это достигается путем предоставления приложения и оптимизированной установки *WINE* в формате, который легко установить. Он обещает покончить со всей возней с настройкой, обычно сопровождающей установку в *WINE*. Хотя выбор на данный момент довольно мал, в нем уже есть несколько топовых игр, включая *Overwatch*, *World of Warcraft* и *League of Legends*. Конечно, установив их, вы можете в конечном итоге установить *WINE* в своей системе десятки раз. Но они в песочнице, а место на диске дешево, так что безумствуйте!

ограничиваются теми, что вы сами можете получить с помощью «обычной» программы.

Установка Flatpak

Первый компонент, который вам нужен для установки Flatpaks (формат пакета) — это *Flatpak* (приложение). Большинство дистрибутивов предоставляют его в своих репозиториях. В Ubuntu вы можете выполнить следующее, чтобы получить и запустить:

```
$ sudo apt install flatpak
```

Вы, конечно, получите версию, поставляемую с текущей версией вашего дистрибутива. Если ваша версия — с долгосрочной поддержкой, например, Ubuntu 16.04 (Xenial), это может означать, что новых функций у вас не будет. Установка с официального PPA *Flatpak* — рекомендуемый способ обеспечить, чтобы у вас всегда была самая свежая версия:

```
$ sudo add-apt-repository ppa:alexlarsson/flatpak
```

```
$ sudo apt update
```

```
$ sudo apt install flatpak
```

После установки вы сможете устанавливать программы Flatpak из командной строки. Но некоторые менеджеры пакетов с графическим интерфейсом также поддерживают этот формат. Например, *Discover* в KDE может управлять вашими файлами Flatpak, если вы установите соответствующий движок.

Установив правильные инструменты, надо сделать еще один шаг, прежде чем вы сможете начать поиск новых приложений.

Хабы Flatpak

Похоже, что для описания коллекций программного обеспечения, таких как RPM- или Deb-репозитории, *Flatpak* принял термин «хабы». Сам сайт *Flatpak* держит первый хаб, который вы должны добавить — FlatHub. Зарегистрируйте его в своей системе, используя менеджер пакетов, или в терминале по команде

```
$ flatpak remote-add --if-not-exists FlatHub https://flathub.org/repo/flathub.flatpakrepo
```

Вы можете увидеть список подключенных вами хабов, выполнив

```
$ flatpak remotes
```

В меню

Существует несколько различных типов приложений, которые можно упаковывать как Flatpaks. Первый из них — это коммерческие приложения. Если разработчики коммерческих продуктов не позволяют сообществу разбирать свои приложения, то они сами должны всё упаковывать.

Flatpak дает единую цель для пакетов — они будут работать в разных дистрибутивах, а песочница приложений поможет сократить расходы на поддержку. Некоторые примеры коммерческих (хотя и бесплатных) программ на FlatHub включают *Steam*, *Slack* и *Sublime Text*.

Другой тип приложений с хорошими Flatpak — большие, сложные приложения. Они выигрывают от атомарных обновлений *Flatpak*, делая процесс обновления чуть менее болезненным, а процесс отката — еще менее. *LibreOffice*, который обычно обновляет 200 МБ, и *Android Studio*, который еще больше, получают несколько удобнее в виде Flatpaks.

Наконец, кроссплатформенные приложения уже имеют две совершенно разные целевые ОС (по крайней мере, для настольных компьютеров, и даже больше — для мобильных). Разумеется, разработчикам и упаковщикам не нужно беспокоиться об особенностях десятков дистрибутивов Linux. Просто упакуйте ваше приложение как Flatpak, и всё готово.

Установка программ Flatpak

Установка приложений из Flatpak будет очень знакома, если вы привыкли использовать в своей системе *apt*, *yum* или *dnf*.

Команда `search` покажет вам совпадения по любому из настроенных вами ключевых слов:

```
$ flatpak search calibre
```

Вернется результат для отличного менеджера электронных книг *Calibre*, который в репозиториях Ubuntu часто устаревший. Обратите внимание на первый столбец: в нем указан идентификатор приложения в своего рода обратном доменном формате (сначала `com`, затем `domain` и т.д.). Вам нужно передать это команде `install` для Flatpak, чтобы установить *Calibre*:

```
$ flatpak install com.calibre_ebook.calibre
```

На этом этапе Flatpak начнет загружать и устанавливать приложение и все необходимые ему среды выполнения (`org.freedesktop` — см. экранный снимок внизу справа). Среда выполнения — это набор библиотек и другого программного обеспечения для приложения. Это способ разрешения зависимостей во Flatpak.

«Подождите, — скажете вы, — разве не то же делают *apt* и *yum*?» Это справедливый вопрос. Есть несколько примечательных факторов, которые отличают среду выполнения от обычного управления зависимостями Linux.

- Среда песочницы Flatpak гарантирует, что всё вами установленное не будет конфликтовать с вашей системой.
- Среда выполнения — это фиксированный набор программного обеспечения, поэтому вы знаете, что получите именно то, что вам нужно для запуска приложения: ни больше, ни меньше.
- Flatpaks созданы с учетом определенной среды выполнения, поэтому вам не надо беспокоиться о том, что обновления внезапно повредят вашу программу.

Компромисс сводится к уверенности в том, что программа всегда будет иметь всё, что ей нужно для правильного запуска, но учтите, что для этого потребуются больше места на диске.

Есть несколько других методов, которые вы можете использовать для установки приложений. Если вы столкнулись с инородным Flatpak и скачали его, установите его из командной строки, используя приведенную команду:

```
$ flatpak install somefile.flatpak
```

Наконец, Flatpak предоставляет метод для установки приложения путем запуска установочного файла. Этот файл, называемый **flatpakref**, содержит некоторые метаданные о программе, в частности:

- Полное имя программы
- Хаб, с которого ее нужно скачать.
- Версия программы
- Есть ли среда выполнения
- Подпись GPG

Установка из файла **flatpakref** аналогична установке из локально загруженного файла Flatpak:

```
$ flatpak install ./somerefile.flatpakref
```

Вы также можете использовать менеджер пакетов с графическим интерфейсом, который должен отображать и устанавливать Flatpak так же, как и другие пакеты (при условии, что вы установили движок).

Запуск программ Flatpak

Установив программу, вы можете запустить ее из командной строки следующим образом:

```
$ flatpak run com.calibre_ebook.calibre
```

После запуска программы вы заметите пару вещей. Во-первых, у вас будет доступ ко всем обычным файлам каталога **\$HOME**. «Но, — спросите вы, — я думал, что песочница Flatpak изолирована от остальной системы?» Это очень проницательно с вашей стороны. Однако от запуска программы, неспособной

взаимодействовать с вашими файлами, пользы мало. Разработчики Flatpak обошли это.

Вы также должны увидеть пункты приложения в меню вашего рабочего стола. Если вы их не видите, врезка «Скорая помощь» на стр. 66, подскажет, как это исправить. На самом деле, в большинстве случаев вы не сможете отличить эти программы от предоставляемых вашим дистрибутивом. Вы можете управлять своими файлами, воспроизводить файлы мультимедиа, и вообще использовать их как «обычную» программу.

Управление установками Flatpak

После одной или двух установок Flatpak, вы, конечно, можете использовать менеджер пакетов с графическим интерфейсом для их администрирования. Но если вы предпочитаете терминал, пригодятся некоторые из приведенных ниже команд. Вы можете получить список установленных Flatpak (как приложений, так и сред выполнения) с помощью следующей команды:

```
$ flatpak list
```

Результат отображает не только установленные модули, но и тип их установки (**системный** или **пользовательский**, **system/** **user**). Вы можете обновить все свои Flatpak одной командой, так же, как и собственные менеджеры пакетов:

```
$ flatpak update
```

Если вам надо удалить Flatpak, введите в командной строке следующее:

```
$ flatpak remove [full-qualified app name]
```

Если вам надо удалить один из настроенных вами хабов, это делается следующим образом:

```
$ flatpak remove-remote [name of remote]
```

Заключение

Преимущества формата Flatpak многочисленны. Он предоставляет разработчикам возможность ориентироваться на несколько дистрибутивов, предоставляя пользователям несколько инструментов для установки полученных приложений. Он позволяет программам получать доступ к нужным им ресурсам, одновременно ограждая их от того, что им не полагается. Он решает некоторые проблемы с зависимостями традиционных форматов упаковки за счет небольшого дискового пространства. Наконец, он открывает перед разработчиками возможности упаковать приложения, которые традиционно трудоемки в установке, в простой формат.

Поэтому, если вы хотите попробовать новейшую версию *LibreOffice* или запустить в *WINE* игру вроде *World of Warcraft*, попробуйте Flatpak. И не ошибетесь! **LXF**

Как и традиционные менеджеры пакетов, Flatpak загрузит и установит *Calibre* и среду выполнения, от которого он зависит.

СКОРАЯ ПОМОЩЬ

Если во время работы Flatpak вы командуете `ls /` и не видите никакой разницы, запустите следующее: `$ flatpak run --command=sh org.libreoffice.LibreOffice`. В песочнице вашего приложения запустится оболочка. Теперь, введя `ls /`, вы должны увидеть такие каталоги, как `/app`.

GNU PLOT

Графики из ваших данных, для ясности

Джон Лэйн умеет заставить данные блеснуть, не касаясь мыши. Это лишь один из многих навыков эксперта по Linux, делающий его званые обеды такими памятными.



НАШ
ЭКСПЕРТ

Джон Лэйн консультирует и пишет о Linux, обычно на ThinkPad, в саду, с холодным пивом...

СКОРАЯ ПОМОЩЬ

Gnuplot умеет также создавать графики непрерывных функций (например, $\sin(x)$) в двух или трех измерениях, мы ранее рассматривали это в LXF210.

Говорят, что картинка стоит тысячи слов, и вы это оцените, если нужно сравнить наборы числовых данных. Мы делали это в LXF243, когда тестировали хранилища и использовали для этого *gnuplot*. На этом уроке мы покажем, как использовать *gnuplot* для построения графиков и диаграмм по собственным данным.

gnuplot — это инструмент для представления данных. Он не имеет ничего общего с проектом GNU, его название — это компромисс, с такой же легкостью он мог бы быть *llamaplot* (см. gnuplot.info/faq, пункт 1.2). Он не использует GNU GPL, а предпочитает собственную лицензию для открытого исходного кода.

Мы будем использовать *gnuplot* для представления некоторых входных данных в виде «графика», показывающего их тенденции и изменения, и в качестве «диаграммы», которая иллюстрирует различия и сходство.

Вы должны иметь возможность установить *gnuplot* из репозитория пакетов вашего дистрибутива:

```
sudo apt install gnuplot #Debian, Ubuntu, etc
sudo dnf install gnuplot #Fedora, CentOS, etc
sudo pacman -S gnuplot #Arch Linux
```

Построение графика данных

Для иллюстрации наших примеров нам понадобятся образчики данных; *gnuplot* ожидает, что данные будут предоставлены в текстовом файле, где каждая выборка представляет собой строку, а столбец — каждый выборочный параметр.

Мы используем пример, который вы видите на скриншоте внизу. Он иллюстрирует первые несколько выборок (по одной в строке) по времени (в секундах) алгоритма, написанного на C, Ruby, Python и Perl, аналогичным образом вычисляющего математическую константу π до пятнадцатого десятичного знака,

используя бесконечный ряд Нилаканты [Nilakantha], математика XV века. Однако данные важнее для изучения *gnuplot*, а не алгоритмов, которые его предоставили.

Первые строки, начинающиеся с `#`, являются комментариями, которые *gnuplot* игнорирует. Значения данных каждой выборки разделяются табуляцией и/или пробелами, а первая строка без комментариев является необязательным заголовком; его значения могут использоваться для обозначений в графике.

Это показывает, каким *gnuplot* ожидает структурирование входных данных. Формат хорошо подходит для любых типов выборочных данных — возможно, периодические замеры с различных датчиков, подключенных к вашему последнему проекту Raspberry Pi?

С помощью *gnuplot* построить график просто. Откройте терминал и введите *gnuplot*, чтобы запустить его интерактивный интерфейс командной строки. Вы можете вводить команды непосредственно в приглашении *gnuplot>*. Вроде этой:

```
plot for [i=1:4] 'pi-nilakantha-15.dat' using 0:i with lines title
columnheader
```

Команда *plot* — одна из основных команд *gnuplot*. Она строит график по данным, в данном случае содержащимся в файле *pi-nilakantha-15.dat*, используя данные из столбцов 0 для координат X и *i* для координат Y, для *with lines* — построения линий, соединяющих точки данных и заголовок *title* из столбца заголовков *columnheader*. Это повторяется — *for* — для значений *i* от 1 до 4 для каждого из четырех столбцов данных в файле. Столбец 0 — это номер строки, начинающийся с 0 и увеличивающийся для каждой строки.

На графике показаны четыре линии, по одной на столбец, с обозначением строки по оси X и значением данных по оси Y. Каждая линия окрашена, а легенда отображается с использованием имен столбцов, взятых из строки заголовка. Это значения по умолчанию, но *gnuplot* гибок, и мы продемонстрируем, как его можно настроить.

Мы могли бы определить каждую линию отдельно вместо использования *for*:

```
plot 'pi-nilakantha-15.dat' using 0:1 with lines title columnheader, \
    " using 0:2 with lines title columnheader, \
    " using 0:3 with lines title columnheader, \
    " using 0:4 with lines title columnheader
```

Здесь показано, как *plot* принимает ряд разделенных запятыми инструкций. Показано, что вы можете опустить имя файла при последовательном использовании. *Gnuplot* заменяет пустую строку (""), ранее использованным именем файла (в нашем случае *pi-nilakantha-15.dat*).

```
# pi-nilakantha-15.dat
# Nilakantha computations to 15 decimal places
# real 2m0.927s user 1m58.854s sys 0m9.817s
C Ruby Python Perl
0.000319 0.0049285888671875 0.02756023406982422 0.00995494293212891
0.000319 0.00495363639831543 0.027103900909423828 0.00965499877929688
0.000319 0.005598783493041992 0.027437686920166016 0.00961995124816895
0.000319 0.005292415618896484 0.026427268981933594 0.0094001293182373
0.000319 0.005041360855102539 0.027116775512695312 0.00946387182312012
0.000319 0.004895687103271484 0.026533842086791992 0.00952816009521484
0.000319 0.004894733428955878 0.02673029899597168 0.00987887382507324
0.000319 0.004968404769897461 0.027356624603271484 0.00955915451049805
0.000319 0.005297422409057617 0.026798725128173828 0.0096588134765625
```

Gnuplot предпочитает выборки данных, упорядоченные по строкам и столбцам. Комментарии и заголовки не обязательны.

И показано, как вы можете использовать символ продолжения строки — обратную косую черту (`\`), чтобы разбить оператор на несколько строк.

Многие команды *gnuplot* допускают сокращения. Пример вверху можно сократить до

```
p for [i=1:4] 'pi-nilakantha-15.dat' u :i w l t columnh
```

что значительно хуже читается и может сбивать с толку при обучении, поэтому наши примеры написаны с использованием их полной формы. К сожалению, многие примеры и учебные пособия, включая документацию по *gnuplot*, употребляют сокращенную форму. Можно поискать сокращения *gnuplot* (<https://superuser.com/q/508644>), но это не так просто, как могло бы быть.

При желании вы можете пояснить график, вставив заголовки и метки осей:

```
set title "Pi Computation comparison"
set xlabel "Execution time (seconds)"
set ylabel "Sample number"
```

Если они заданы перед выполнением построения, они появятся на графике. Заголовок появится над графиком.

Переменные, функции и макросы

Если у вас несколько наборов данных (мы взяли несколько алгоритмов вычисления π), вы можете построить их все. Имена наших наборов данных можно сохранить в переменной:

```
datasets = "nilakantha simpson viete"
```

Это просто строковая переменная, но мы можем использовать ее, чтобы выполнить команды `do` для каждого из этих наборов данных — `for`:

```
do for [dataset in datasets] {
  plot for [i=1:4] 'pi-'.dataset.'-15.dat' using 0:i with lines title
  columnheader
  pause 5 "Displaying ".dataset
}
```

Команды, перечисленные в фигурных скобках, выполняются с переменной `dataset`, установленной для каждого (разделенного пробелами) слова в `datasets`. Мы используем это для создания имени файла объединением подстрок оператором `.`, который, очевидно, полагается на наши наборы данных, хранящихся в файлах с соответствующими именами. Мы также используем `pause`, давая немного времени между графиками, чтобы можно было разглядеть каждый из них. Это также показывает, как можно вводить многострочные блоки (ограниченные фигурными скобками): приглашение командной строки внутри блока меняется на `more>`.

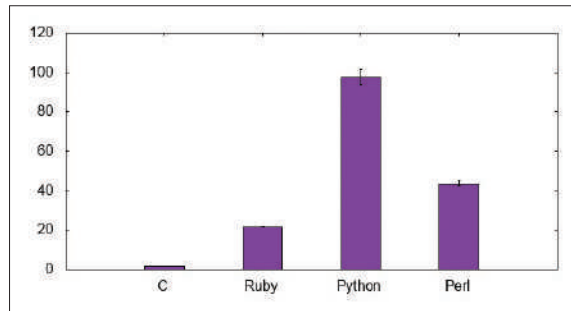
Как и переменные, мы можем определить функции. Они позволяют создавать выражения, дающие строковый или числовой результат. Можно использовать функцию для создания имени файла набора данных:

```
file(dataset,extn) = "pi-".dataset."-15.".extn
```

Здесь, учитывая имя набора данных и расширение файла, мы возвращаем отформатированное имя файла. Функции в *gnuplot* должны иметь такую форму: имя и несколько аргументов, а затем выражение, которое возвращает строковое или числовое значение. Вы не можете программировать функцию, состоящую из нескольких операторов (нельзя использовать блок `do`) или функцию, которая изменяет параметры, использует итераторы или другое, что обычно имеет смысл в функции. Поэтому функции *gnuplot* несколько ограничены (<https://stackoverflow.com/a/27835753>), но всё же могут быть полезными.

Макросы предлагают альтернативу функциям, которые не принимают параметры. Они заменяют макрос его значением. Мы можем использовать макрос во избежание повторного ввода команды `plot`:

```
draw_plot = "plot for [i=1:4] file(dataset,'dat') using 0:i with lines
title columnheader"
```



«Усы» погрешностей предлагают простой способ обогатить график.

Здесь мы имеем простое присвоение строке значения. Это похоже на переменную, и это действительно так. Но если перед именем поставлено `@`, то перед выполнением оно заменяется. Вроде этого:

```
do for [dataset in datasets] { @draw_plot; pause 5 }
```

Макросы также можно вложить:

```
draw_plots = "do for [dataset in datasets] { @draw_plot; pause 5 }"
```

Это позволяет рисовать графики с помощью простого `@draw_plots`. Обратите внимание, как *gnuplot* позволяет использовать точку с запятой для размещения нескольких команд в одной строке, что полезно для макросов, поскольку они не могут содержать перевод строки. Мы будем использовать функции и макросы на протяжении всего урока, чтобы избежать повторного ввода.

Просто stats

Следующее, что мы хотим сделать, это создать диаграмму данных, чтобы выделить различия между наборами данных. Мы можем использовать команду `stats` из *gnuplot* для получения средних значений, минимумов и максимумов для них, а затем отобразить их на гистограмме.

Команда `stats` суммирует один или два столбца и сообщает в терминал. Вы можете увидеть это с помощью `stats 'pi-nilakantha-15.dat'`. Она также устанавливает переменные, которые вы можете использовать на графиках — встроенная справка объясняет это: `help stats`. Для демонстрации мы могли бы использовать `stats`, чтобы получить число столбцов в качестве переменной для использования в команде `plot` вместо жесткого кодирования его как 4. Одна из множества переменных, которые она устанавливает, называется `STAT_columns`, поэтому мы получим `stats 'pi-nilakantha-15.dat' nooutput; plot for [i=1:STAT_columns] { ... }`

СКОРАЯ ПОМОЩЬ

Вы можете сконструировать из команд *gnuplot* скрипт с помощью `#!/usr/bin/gnuplot` и выполнять их.

» ЦВЕТОВАЯ ПАЛИТРА

Цвета, использованные в примерах, были выбраны *gnuplot*. Определено восемь типов линий, представляющих стили, задающие цвет линии (или прямоугольника, или чего-то еще). Эти определения настраиваются с помощью `set linetype`, и можно добавлять другие.

Если график содержит больше линий, чем типов линий, он будет циклически повторять определенные типы линий (так что девятая линия будет того же цвета, что и первая, и т.д.). Можно определить и более восьми типов линий, причем любые определенные типы линий будут иметь приоритет над циклическими: если вы не определили тип линии девять, то для нее будет использоваться тип линии один; если вы определили тип линии 10, то для 10-й линии будет использоваться именно он, а не тип линии два.

Чтобы уменьшить количество цветов, вам нужно сделать две вещи: установить период цикла и не определять типы линий по этому числу (в противном случае они будут иметь приоритет):

```
set linetype cycle n
unset for [l=n+1:99] linetype l
```


СКОРАЯ ПОМОЩЬ

Чтобы объединить заголовки столбца с другой строкой, используйте `some_string.columnhead(i)`. Обратите внимание, что в данном контексте это скорее «шапка», а не «заголовок».

Использование `nooutput` подавляет отчет по умолчанию, который команда выдала бы в противном случае, поэтому она только устанавливает свои переменные.

Если вы попытаете вышеприведенное с нашим примером файла, в строке 4 вы можете заметить предупреждение о неверных данных. Это происходит потому, что заголовки столбцов не являются значениями данных, которые может обрабатывать `stats`. Предупреждение можно проигнорировать, или вы можете сообщить `gnuplot`, что в файле есть данные заголовка:

```
set key autotitle columnhead
```

Ограничение `stats` первыми двумя столбцами необходимо преодолеть, когда, как и в наших данных, требуется рассмотреть более двух столбцов. Еще раз мы применяем `for` и `using` для отчета по одному столбцу за раз, «печатая» нужные нам данные, по одной строке на столбец —

```
do for [i=1:4] { stats 'pi-nilakantha-15.dat' using i nooutput; print i, STATS_mean, STATS_min, STATS_max }
```

и получаем

```
1 1.8755842 1.873858 1.879206
2 21.7233386516571 21.4641873836517 22.1631166934967
3 97.6620037794113 93.8680853843689 101.46163392067
4 43.4605897188187 42.7861561775208 45.4818699359894
```

Мы хотим пометить строки исходными столбцами данных, из которых они были получены, чтобы четко маркировать диаграммы. Но `stats` не записывает переменную, содержащую заголовок столбца (`STATS_columnheader` не существует). Тем не менее, мы можем написать функцию:

```
columnheading(f,c) = system("awk '/^#/ {next}; {print $\".c.\";exit}' \"f\")
```

Она использует команду `system` из `gnuplot` для выполнения команды оболочки. В этом случае мы используем `awk` для извлечения необходимого заголовка столбца с из файла `f`. Жаль, что `gnuplot` не может этого сделать, потому что выход из оболочки вовлекает зависимости, которые могут быть не переносимы (если

вам интересно, `gnuplot` также работает на MacOS и Windows). Команда `awk` пропускает все строки комментариев и затем печатает требуемое слово из следующей строки, а затем завершает работу, возвращая свой вывод в сеанс `gnuplot`.

Нам также надо как-то захватить вывод `stats`. Можно использовать `set print` для записи `gnuplot` в файл вместо терминала, и весь последующий вывод `print` будет идти в указанный файл. Мы можем написать строку заголовка и использовать `stats`, как описано выше, для записи отчета в файл. Имя набора данных записывается как пространство заголовка для нулевого столбца, чтобы оно было доступно позже при создании диаграммы. Мы можем завершить всё это и выполнить `do` для всех наших наборов данных — `for`:

```
do for [dataset in datasets] {
    set print file(dataset,'stats')
    print dataset.' mean min max'
    do for [i=1:4] {
        f = file(dataset, 'dat')
        stats f using i nooutput
        print columnheading(f,i) ' ', STATS_mean, STATS_min,
        STATS_max
    }
}
unset print
```

Последний оператор восстанавливает вывод `print` в терминал. Мы можем сжать всё это в макрос, если свернуть его в одну строку и завершать каждый оператор точкой с запятой:

```
build_stats = "do for [dataset in datasets] { set print
file(dataset,'stats'); print dataset.' mean min max'; do for
[i=1:4] { f = file(dataset, 'dat'); stats f using i nooutput; print
columnheading(f,i) ' ', STATS_mean, STATS_min, STATS_max } };
unset print"
```

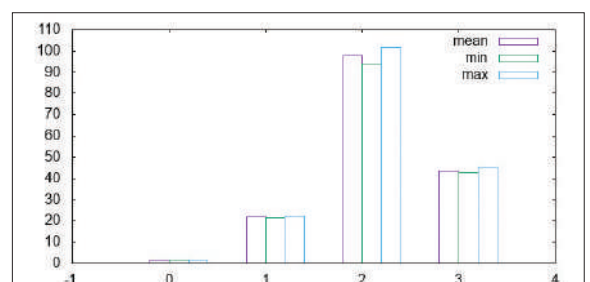
Столбчатые диаграммы

Теперь мы можем создать гистограмму из данных `stats`, которая по умолчанию будет сгруппированной. Это означает, что каждая строка данных представлена группой столбиков с одним столбиком на столбец, каждая группа отделена пробелом, эквивалентным ширине двух столбцов.

```
plot for [i=2:4] file('nilakantha','stats') using i with histogram title
columnheader
```

Заядлый статистик может указать, что это столбчатая диаграмма, поскольку она отображает категоризованные данные, тогда как гистограмма отображает исторические количественные данные, такие как возраст людей, и часто сгруппирована (например, по возрастам). Гистограмма — это составное слово, означающее «историческая диаграмма»; однако тип отображения `gnuplot`, который выдает нам нужный результат, называется гистограммой, поэтому его мы здесь и используем.

Наша диаграмма выглядит довольно пресно, но ее можно улучшить с помощью стилей. Мы сделаем несколько изменений,



Сгруппированная гистограмма отображает несколько показателей для каждой выборки, но рендеринг по умолчанию довольно простейший.

» ТЕРМИНАЛЫ ВЫВОДА

Помимо терминала вывода интерактивного окна по умолчанию, `gnuplot` поддерживает множество других опций вывода, включая PNG, PDF, EPS (Postscript) и LaTeX. Вы можете узнать доступные с помощью `set terminal` или `print GPVAL_TERMINALS`. Обратите внимание, что `terminal` в этом контексте относится к графическому устройству вывода, а не к текстовому терминалу, который вы используете для взаимодействия с `gnuplot`.

Изменение терминала вывода является двухэтапным процессом, когда выбираются терминал и выходной файл:

```
set terminal pdf size 8,10 inch
set output "report.pdf"
```

Поддерживаемые опции зависят от выбранного терминала вывода. Чтобы увидеть эти опции, можно обратиться к встроенной справке, например:

```
help terminal pdf
```

Вы можете установить параметры терминала по отдельности, используя `set termoption`. Например, некоторые терминалы имеют «расширенный» текстовый режим, в котором текстовые строки интерпретируются так, что можно указывать такие вещи, как нижние и верхние индексы. Отключение этого режима делается так:

```
set termoption noenhanced
```

Расширенный текстовый режим прерывает отображение простого текста, содержащего управляющие символы режима, такие как подчеркивание, что затрудняет отображение таких символов.

Чтобы вернуться к интерактивному терминалу по умолчанию:

```
unset output; unset terminal
```

Или к предыдущему терминалу:

```
unset output; set terminal pop
```

чтобы проиллюстрировать возможности, но также посмотрите документацию.

```
set style data histogram
set style histogram gap 1
set style fill solid border -1
set boxwidth 0.8
```

Мы начнем с изменения стиля данных `data style` по умолчанию, т. е. того, как `plot` отображает данные, когда стиль не указан. Установка `histogram` по умолчанию означает, что мы можем опустить `with histogram` в командной строке `plot`. Далее мы устанавливаем промежуток между группами гистограммы, изменяя его со значения по умолчанию 2 на 1. Затем мы устанавливаем сплошную заливку со сплошной границей, используя тип линии по умолчанию — сплошная черная (-1). Наконец, мы устанавливаем ширину прямоугольника равной 0,8, чтобы оставить небольшой зазор между столбиками (ширина варьируется от 0 до 1).

Вы можете просмотреть доступные типы линий и другие атрибуты отображения с помощью встроенного теста терминала. Введите `test` в командной строке. Вывод основан на терминале, поэтому, если вы планируете экспортировать графики в PNG или PDF, сначала настройте их.

Другой способ представления данных — «усы» погрешности:

```
set style histogram errorbars
set key off autotitle columnhead
plot 'pi-8.stats' using 2:3:4 title columnheader, \
" using (0):xticlabels(1) with lines
```

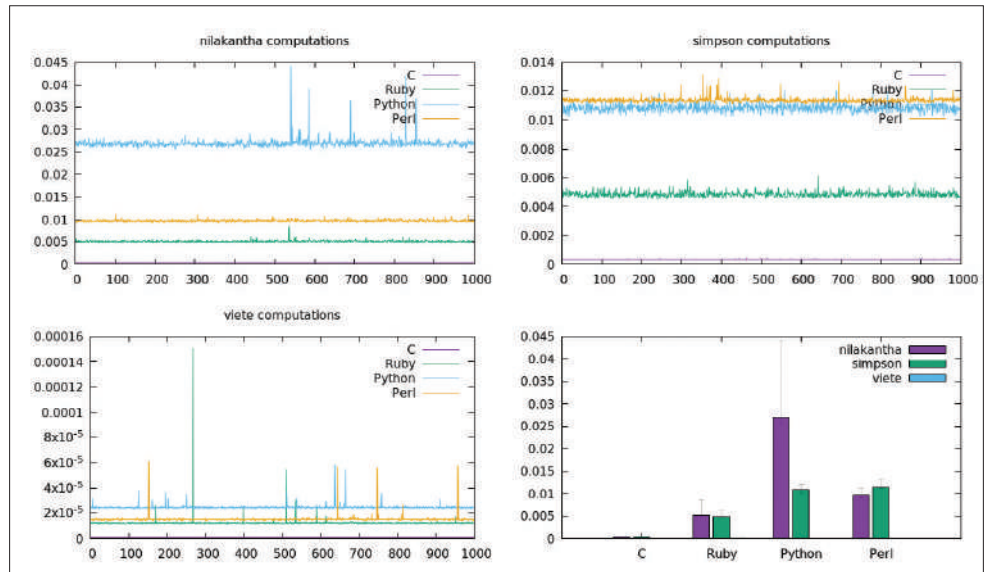
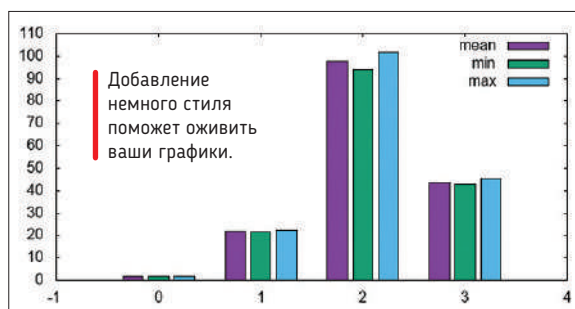
Это дает один столбик на категорию, представляющий его среднее значение, но и отмечает минимальные и максимальные значения сверху с помощью «усов» погрешности [error bars]. Значение имеет только одну запись (среднее значение), поэтому мы отключаем его — `set key off`. Нам всё еще нужно сообщить `gnuplot`, что есть строка заголовка, следовательно, оставляем `autotitle columnhead`.

В порядке последнего украшения, обозначим ось X, но это требует некоторого пояснения. Эта дополнительная инструкция для `plot` задает (0) для оси Y, а для оси X — `xticlabels(1)` — функцию, которая устанавливает метки-засечки на оси X из первого столбца в файле. Мы определяем `with lines`, потому что вдоль заданной `y=0` рисуется нежелательная линия, но заслоняется осью X и поэтому невидима (рисуи мы точками, вы бы их увидели!).

Мы можем собрать всё это вместе, чтобы отобразить все наборы данных в сгруппированной гистограмме с «усами» погрешностей:

```
plot for [dataset in datasets] file(dataset,'stats') \
using 2:3:4 title columnheader(1), \
" using (0):xticlabels(1) with lines
```

Оформим это как макрос — и сможем использовать позже:



```
plot_histogram = "plot for [dataset in datasets] file(dataset,'stats') \
using 2:3:4 title columnheader(1), \" using (0):xticlabels(1) with \
lines"
```

Самое время немного подробнее объяснить, как работает `using`: это зависит от стиля отображения данных (в `set style` или определения `with`). Двумерные графики ожидают, как и наш первый, пару координат `x` и `y`, но значение `y` может быть опущено: `using 1` — это то же самое, что и `using 0:1`, и получится график с номером строки (из столбца 0) по оси X со значением из столбца 1 по оси Y.

Стиль гистограммы предполагает `using` с одним значением для указания столбца, которое будет представлено в виде вертикальных полос, идущих вверх по оси Y. Стиль `errorbars` ожидает дополнительных значений — мы даем в общей сложности 3, чтобы указать столбцы, содержащие значение столбика, а также минимальные и максимальные значения для его погрешности.

Объединение графиков

Мы можем взять все графики, которые мы успели создать, и использовать функцию `multiplot` из `gnuplot`, чтобы объединить их. Сначала надо определить произвольную сетку для макета графика:

```
set multiplot layout 2,2
```

Здесь определено 2×2, что может вместить четыре графика; этого достаточно для трех графиков набора данных и сгруппированной гистограммы. Нам просто надо построить их по очереди:

```
do for [dataset in datasets] { @draw_plot }
```

```
@draw_chart
```

Наконец, закрываем `multiplot`:

```
unset multiplot
```

Это наиболее распространенное использование `multiplot`, когда графики выводятся слева направо и сверху вниз. `help multiplot` покажет, как можно управлять раскладкой макета или использовать абсолютное позиционирование, даже накладывая один график на другой.

Для графического отображения дискретных данных мы ввели множество функций `gnuplot`, но узнать можно еще много чего. Вы можете многому научиться только по интерактивной справке, но руководство пользователя, доступное на www.gnuplot.info, является очень полным, хотя больше справочным, чем учебным пособием.

Есть также книга *Gnuplot in Action*, которая во втором издании обновлена до пятой версии. Мы нашли эту книгу более удобной для начинающего пользователя `gnuplot`. **LXF**

multiplot использует сетку для выравнивания и размещения графиков.

СКОРАЯ ПОМОЩЬ

Пользовательские функции хранятся в виде переменных с префиксом `GPFUN`, и вы можете использовать `print`, чтобы увидеть определение функции: `print GPFUN_file`.

Верстка журнала видом как у профи

Джон Найт пытается разобраться в пугающем редакторе журнала *Scribus* и находит, что все страхи — это много шума из ничего.



**НАШ
ЭКСПЕРТ**

Когда **Джон Найт** не пишет электронные книги о том, как играть на барабанах, он любит поиграть с эмулятором Commodore 64.

СКОРАЯ ПОМОЩЬ

Выбрав шрифт, его и придерживайтесь. Многие журналы используют только один-два шрифта. Ваш выбор может создать впечатление от вашего журнала, так что выбирайте с умом.

Связанные текстовые фреймы — убойная функция, которая может сманить вас от пресловутого текстового процессора.

Мы брались за *Scribus* с трепетом. Хотя он намного мощнее какого-нибудь *Microsoft Publisher*, интерфейс полностью отличается от привычного для большинства людей, что создает ему устрашающую репутацию. Но эта репутация незаслуженна, и отличный способ освоить *Scribus* — самостоятельно что-нибудь издать.

Обсуждение всего *Scribus* способно занять журнал целиком, и вместо этого мы проведем вас по основным элементам интерфейса, дав достаточно для начала работы. Мы сосредоточимся на создании журнала, но те же принципы применимы к чему угодно, например, к деревенским новостям или школьной газете.

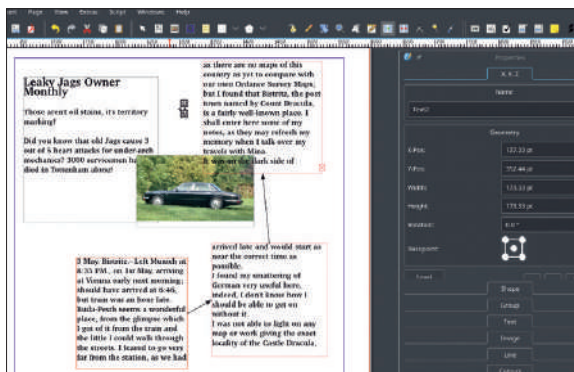
Простейший способ начать в *Scribus* — это работать с шаблоном, и для многих проектов бывает достаточно найти понравившийся шаблон и просто заменить текст и изображения. Итак, мы начнем с шаблона, а попозже — с нуля, чтобы изучить остальную часть графического интерфейса.

Ничего нового

При запуске *Scribus* автоматически открывается окно New Document [Новый документ] со вкладкой New Document, готовой начать с нуля. Но мы-то к этому пока еще не совсем готовы, так что посмотрите вправо: следующая вкладка называется New from Template [Создать из шаблона]. Здесь вы найдете достаточно большой выбор шаблонов с описанием справа.

Для простоты мы выбрали 'Newsletter 1 [Информационный бюллетень]'. Его макет довольно прост, все изображения бесплатны, и автор разработал его так, чтобы вы могли просто заменить любой текст или изображение своим собственным.

Хотя иные элементы управления в *Scribus* могут показаться довольно технарскими, в своей основе они довольно просты и вращаются вокруг двух концепций: всё размещается в блоках-фреймах [frame — англ. рамка], и всё уложено стопками по слоям.



Scribus разработан в основном вокруг текстовых и графических фреймов. Работаете ли вы над текстом или изображением, у любого типа фрейма можно изменить по вкусу размер или форму, и вы, как правило, можете заменить любое имеющееся содержимое тем, чем вам угодно. Давайте начнем с текста.

Добавление текста

Выберите любой текстовый блок и щелкните правой кнопкой мыши. Появится меню с немалым количеством опций, но сейчас мы рассмотрим две: Get Text [Получить текст] и Edit Text [Редактировать текст]. Get Text берет содержимое из внешнего файла и помещает его в блок. Очень круто, но вы, скорее всего, просто скопируете и вставите в Edit Text.

Edit Text открывает окно Story Editor [Редактор раздела], похожее на простой текстовый процессор. Большинство элементов управления вы сумеете угадать, хотя есть и несколько странностей. Закончив редактирование, вы можете просто нажать на большую зеленую кнопку с галочкой. Однако в Story Editor нет функции предпросмотра, и каждый раз, когда вы захотите внести изменение, вам надо будет нажимать кнопку Edit Text.

Лучше сдвинуть Story Editor в сторону, чтобы вы могли видеть исходный текстовый блок, который хотите редактировать. Затем, когда вы внесли некоторые изменения, нажмите Update Text Frame [Обновить текстовый фрейм] на главной панели инструментов Story Editor, и вы увидите, что исходный текстовый фрейм в главном окне меняется.

Управление изображениями

Чтобы заменить изображение, щелкните правой кнопкой мыши в блоке и выберите Get Image [Получить изображение]. Откроется файловый браузер. Выбрав свое изображение, нажмите OK. Теперь во фрейме изображения окажется ваша картинка, но если их пропорции не совпадают, размеры, вероятно, будут неправильными: часть фрейма будет пустой или появится только кусок изображения.

Чтобы изменить размеры изображения, щелкните по нему правой кнопкой мыши и выберите Properties [Свойства]. В окне Properties перейдите на вкладку Image [Изображение]. Могут быть включены опции Scale To Frame Size [Масштабировать к размеру фрейма] и Proportional [Пропорционально]. Если это так, то одним из вариантов исправления для недозаполненного фрейма является снятие флажка Proportional, что растянет изображение до размера фрейма.

Однако это может выглядеть ужасно, поэтому вы можете настроить изображение, выбрав параметр Free Scaling [Произвольный масштаб], с которым вы можете перемещать и растягивать

изображение как пожелаете. Или измените размер фрейма изображения соответственно.

В шаблоне Newsletter 1 фреймы изображения и текста зафиксированы, т.е. вы не можете изменить их размер. Чтобы разблокировать фреймы, щелкните правой кнопкой мыши любой блок изображения или текста и снимите флажок **Is Locked** [Заблокировано]; теперь вы можете менять размер блока как угодно.

При минимальном везении, для создания требуемого документа будет достаточно работы с шаблоном. Если это так, сохраните свою работу и экспортируйте ее. Родным форматом *Scribus* является **.sla**, но если вы отправляете свою работу кому-либо, не пользователю *Scribus*, вам нужно использовать нечто более универсальное, например, PDF. Это легко сделать, нажав **File > Export > Save as PDF** [Файл > Экспорт > Сохранить в формате PDF].

Заметьте, что и при экспорте файлов, и в режиме **Print Preview** [Предпросмотр печати] вы можете встретить предупреждающие сообщения. Это вполне нормально — *Scribus* весьма педантичен! Не обращайте на них внимания, всё будет в порядке.

С нуля

Если у вас в планах нечто более экзотическое, подходящего шаблона может не найтись, а значит, вам нужно будет создать свой документ с нуля. Но не волнуйтесь, мы здесь, чтобы познакомить вас с основами.

Для начала, если у вас есть, например, журнал, стиль которого вы хотите имитировать, стоит взять линейку и измерить размеры каждой рамки, строки, изображения и колонки. Макет *Scribus* можно настроить на работу в физических единицах; следовательно, вы должны иметь возможность копировать элементы форматирования вашего любимого журнала.

Ваш автор и его брат попытались здесь воспроизвести стиль и вид *Commodore Format* из начала 1990-х гг., так что мы можем работать над собственной книгой и предоставить классный шаблон для других авторов о ретроиграх. Помимо копирования физических размеров и языка стилей, журналы обычно держались макета из трех столбцов, так что это хорошее место для начала. Привнесите свои идеи по форматированию, и мы начнем.

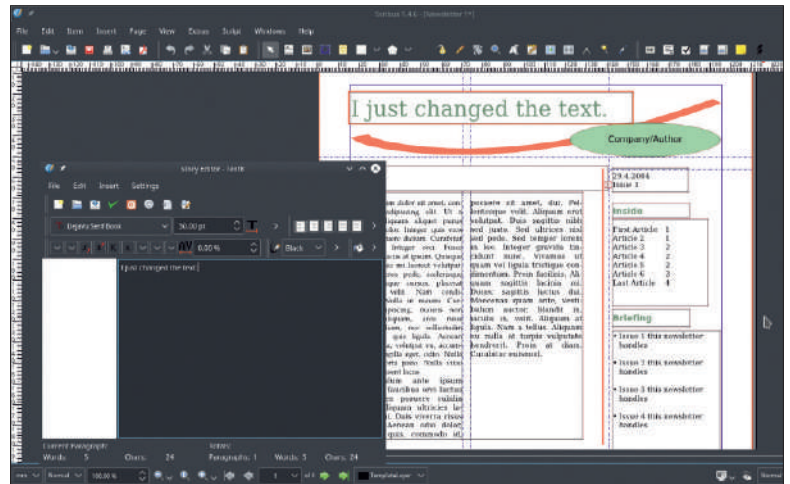
Нажмите **File > New** [Файл > Создать], чтобы начать заново, и появится окно **New Document**, где вы определяете свой макет. Для простоты мы будем придерживаться макета **Single Page** [Одиночная страница]. В остальном всё довольно условно, кроме выбора **Default Unit** [Единица измерения по умолчанию], которая называется **Points (pt)** [пункты]. Бог знает, что это значит, так что переключите ее на что-нибудь полезное, например, миллиметры!

Теперь вы можете легко определить поля страницы слева, и если вы установите флажок **Automatic Text Frames** [Автоматические фреймы текста], вы сможете установить количество колонок и расстояние между ними. Если вы хотите сделать обычную страницу со сплошным текстом, оставьте значение **Columns** [Колонки] как 1, но большинство журналов используют две или три колонки, а в газетах их обычно четыре.

Текстовые блоки

Когда вы нажмете **OK**, у вас будет пустая страница с тремя столбцами, если вы так выбрали. Визуальными направляющими будут маркеры колонок, и если вы проверите свою работу в режиме предпросмотра, вы увидите, что эти ужасные штуки, к счастью, не включены! Очевидно, что начать надо бы с текста, поэтому давайте создадим наш первый текстовый блок.

Посередине главной панели инструментов находится кнопка **Insert Text Frame** [Вставить текстовый фрейм]. На вашей пустой странице нажмите и разместите свой первый фрейм, где захотите. При желании можно даже перекрывать направляющие колонок и границы страниц — впрочем, это можно изменить; для получения дополнительной информации см. врезку



Наш совет любому новичку в *Scribus* таков: найдите подходящий шаблон и смело экспериментируйте.

о направляющих (стр. 75). Учтите, что когда вы вставите фрейм, *Scribus* снова переключится в режим выбора элемента, так что вам надо будет нажимать кнопку каждый раз, когда вы хотите создать новый текстовый блок.

Чтобы добавить текст, используйте тот же метод, что и раньше: щелкните правой кнопкой мыши и выберите **Edit Text**. Если вам не хватает места в первом фрейме, не волнуйтесь. Создайте еще несколько пустых фреймов, и мы сможем изучить одну из самых крутых возможностей *Scribus*: связанные текстовые фреймы, в которых текст плавно перетекает из одного фрейма в другой.

Начните с выбора фрейма, с которым будут связаны все остальные фреймы. Если вы нажмете **Link Text Frames** [Связать текстовые фреймы] на главной панели инструментов, указатель мыши изменится на изображение двух текстовых фреймов со стрелкой связи. При щелчке на следующем фрейме они станут связанными.

Отныне незначает беспокоиться о том, что у вас слишком много текста для одного фрейма. Просто пишите всё, что душе угодно, и всё, что не помещается в первый фрейм, перейдет в следующий. Вы можете проделать это несколько раз, чтобы создать

СКОРАЯ ПОМОЩЬ

Очень удобная функция **Sample Text** [Шаблонный текст], особенно опция **разных языков**. Если в текстовых фреймах фигурируют разные языки, вы можете обнаружить ошибки связывания, которые иначе могли бы и не увидеть.

» ПРИМЕНЕНИЕ СЕТКИ

Если вы нажмете **View > Show Grid** [Вид > Показать сетку], появится наложенное изображение в стиле миллиметровки с большими квадратами (основная сетка), содержащими меньшие квадраты (вспомогательная сетка). Вы можете использовать эту сетку как простые визуальные направляющие, позволяющие вам рисовать от руки и что-нибудь размещать там, где вам нравится, или жестко «привязывать» объекты к ней. Если вы делаете броский журнал со свободным размещением объектов, оставьте ее как есть. Но если вы делаете что-то достаточно консервативное со множеством страниц, понадобится функция привязки.

По умолчанию сетка будет просто служить визуальным ориентиром, но если вы хотите включить функцию привязки, в главном меню выберите **Page > Snap to Grid** [Страница > Привязать к сетке]. Правда, размеры сетки по умолчанию немного странные, и поля страницы не вписываются в них должным образом.

Чтобы сделать сетку более полезной, измените ее настройки, нажав **File > Document Setup** [Файл > Структура документа], и в новом окне нажмите **Guides** [Направляющие]. Включение опции **Show Page Grid** [Показать сетку страниц] позволит вам изменить настройки сетки. Интервал по умолчанию составляет 35,278 мм для главной сетки и 7,056 мм для вспомогательной. Если изменить их на 35 мм и 7 мм соответственно, поля страницы теперь будут соответствовать сетке.



СКОРАЯ ПОМОЩЬ

Print Preview обычно дает низкое разрешение, отчего выглядит довольно грубо. Чтобы это сгладить, нажмите Enable Antialiasing [Активировать сглаживание] в поле Display Settings [Настройки отображения] в верхнем правом углу.

сложные конструкции. Фреймы можно связывать вверх или вниз, в любом направлении. Вы можете даже пропустить другие фреймы, оставив место посередине.

Всякий раз, когда вы связываете несколько фреймов, *Scribus* будет использовать стрелки, чтобы показать, какие фреймы связаны и в каком направлении. Вы узнаете, что фрейм связанный, если при щелчке по нему вам предложат использовать кнопку Unlink Text Frames [Открепить текстовые фреймы]. Если вы делаете это, *Scribus* тогда укажет поблизости фрейм, который будет отключен, просто вам в помощь

Фреймы изображений

Фрейм изображения вставляется так же, как текстовый. Нажмите кнопку Insert Image Frame [Вставить фрейм изображения] на главной панели инструментов, затем щелкните и перетащите границы фрейма в требуемый размер. Щелкните правой кнопкой мыши внутри фрейма и выберите Get Image, чтобы открыть браузер файлов. Как и с текстовыми фреймами, каждый раз, когда вы хотите создать новый фрейм, надо нажимать кнопку Insert Image Frame.

Как и в разделе про шаблоны, ваше изображение для только что созданного фрейма, вероятно, будет иметь неправильный размер и разрешение. Чтобы исправить это, щелкните правой кнопкой мыши по изображению и выберите Properties.

В Properties перейдите на вкладку Image и выберите Scale To Frame Size.

Если установлен флажок Proportional, изображение сохранит исходное соотношение сторон. Тогда, скорее всего, во фрейме останется какое-то место, и вам придется сдвинуть границы в соответствии с изображением. Если вы не установите флажок Proportional, изображение растянется, заполнив фрейм.

Для таких элементов, как клипарт и мелкие логотипы, вы, вероятно, захотите иметь прозрачный фон. Прозрачные фоны позволяют назначить им цвет, а размещение нового изображения поверх всего не оставляет конфликтующих границ или цветов. Это также поможет в следующих двух разделах, с широкими контурами, которые вы хотите наложить поверх чего-то другого, или когда текст обтекает фрейм.

Управление слоями

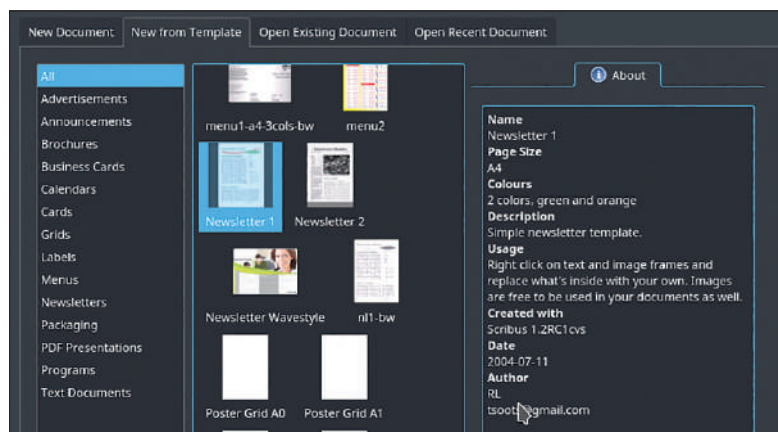
Как только вы начинаете изучать изображения, используемые в слоях, *Scribus* действительно начинает становиться интересным. Со слоями можно создавать сложные изображения и работать над частями сцены независимо, не нарушая ничего другого.

Например, вы можете иметь статическое фоновое изображение, а затем размещать собственные изображения поверх — что, кстати, является первичным методом анимации. На самом деле, представьте себе аниматора с кусочками целлулоида, складывающего отдельные рисунки поверх фонов — это та же самая концепция. Всё, что вы делаете в одном слое, не влияет на то, что находится выше или ниже его.

Действительно, если вы попытаетесь подойти к *Scribus* с точки зрения чего-то вроде *Microsoft Word*, всё покажется чуждым и кошмарно сложным. Но если вы мыслите в терминах реальных листов бумаги, целлулоида и ножниц, всё это становится гораздо легче понять.

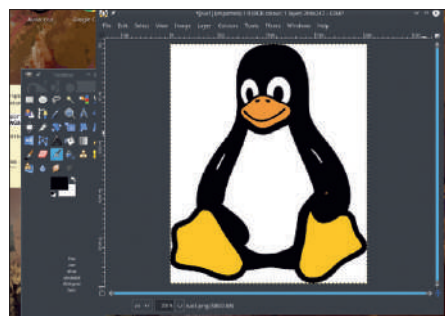
В качестве хорошего примера слоев мы подумали обратиться к журналам о старинных играх, которые мы имитируем, где берут яркие оконтуренные фигуры и используют их для привлекательных фреймов. Сначала мы поместим цветной многоугольник в фрейм изображения, а сверху поместим текстовый фрейм. Наконец, позади мы создадим красивый фон.

Сначала мы создали фрейм изображения, чтобы поместить в нее как изображение, так и отдельный текстовый фрейм, что означало намеренное перекрытие границ столбца (типично для 1990-х гг.).



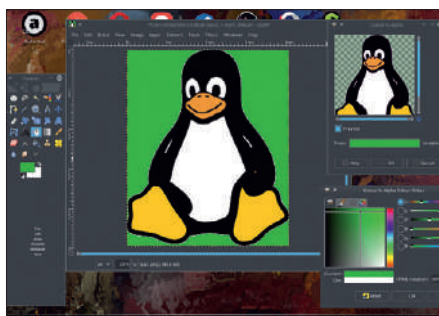
Шаблоны можно получить с таких сайтов, как opendesktop.org, и они могут быть доступны в вашем менеджере пакетов. Некоторые дистрибутивы даже включают их в состав самого *Scribus*.

ПРИДАНИЕ ПРОЗРАЧНОСТИ ИЗОБРАЖЕНИЯМ В GIMP



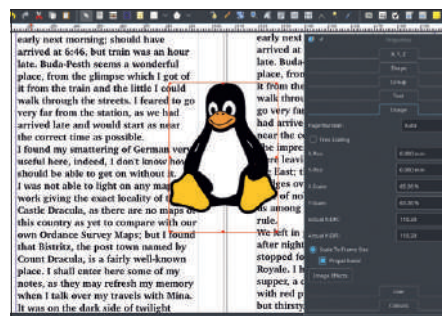
1 Сплошной фон

GIMP — редактор изображений, используемый *Scribus*. Сначала убедитесь, что ваше изображение имеет сплошной цвет фона, который компьютер легко отличит от остального изображения. Хорош белый — или зеленый, для пользователей цифровых эффектов зеленого экрана.



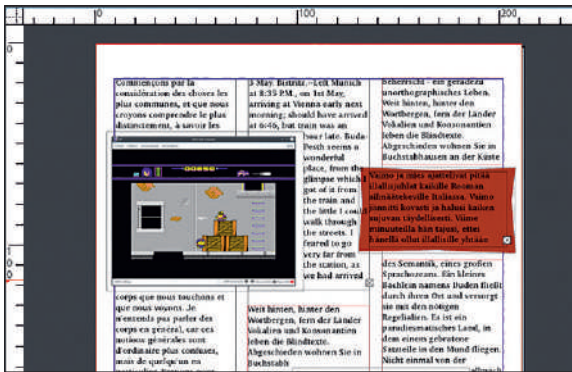
2 Colour to Alpha

Открыв изображение, в главном меню выберите Colours > Colour to Alpha [Цвета > Цвет к альфа-каналу] и нажмите OK. Если всё хорошо, он должен обнаружить всё автоматически и выделить объект от фона. Если нет, вам может понадобиться инструмент Color Picker [Выбор цвета].



3 Экспорт изображения

Если у вас нет возможности выбрать Color to Alpha, измените режим изображения на RGB, щелкнув Image > Mode > RGB [Изображение > Режим > RGB]. Учтите, что вам надо сохранить изображение в формате PNG или GIF, поскольку файлы JPEG не имеют фоновой прозрачности.



Используя фреймы, слои и симпатичные фоны, мы сможем помаленьку создать визуально привлекательный журнал об играх – без особого напряжения.

Чтобы добиться нужного размера, мы открыли Properties фрейма и вкладку Image. Здесь мы включили Scale To Frame Size, но не включили Proportional, а значит, мы можем изменить размеры и форму фрейма, чтобы идеально вписать абзац вовнутрь. Затем мы нарисовали текстовый фрейм поверх фигуры и вставили информативный и интересный текст.

Теперь хороший момент, чтобы изучить элементы управления слоями Scribus. Сейчас текст располагается над изображением, как он и должен; а вдруг мы что-то сделали неправильно, и как это исправить? Итак, сейчас мы намеренно скроем текстовый слой под слоем изображения, а затем вернем его обратно.

Щелкните правой кнопкой мыши по текстовому фрейму и выберите Level > Lower [Уровень > Ниже]. Текст теперь должен быть скрыт, раз изображение находится сверху. Чтобы вернуть текст, надо понизить уровень изображения, поэтому щелкните правой кнопкой мыши и снова выберите Level > Lower. Изображение вернется обратно за текст и восстановит текстовый слой сверху.

Теперь зададим общий фон. Мы уже подготовили в GIMP простой фон с красивым переходом, хотя должно подойти любое изображение портретной ориентации. Затем мы просто нарисовали фрейм изображения по всей странице и вставили изображение. Первоначально он просто скрывал всё, но нажатие Level > Lower to Bottom [Уровень > Понизить до дна] вернуло всё остальное на передний план, в то же время получив красивый фон.

Текст должен плавно течь

В последней части головоломки Scribus позволит разместить графику в любом месте внутри текста и заставит окружающий текст обтекать ее. Если вы создаете фрейм изображения, перекрывающий текстовые фреймы, вы можете опробовать эту функцию, щелкнув правой кнопкой мыши на изображении, выбрав Properties и открыв вкладку Shape [Фигура]. В поле Text Flow Around Frame [Обтекание фрейма текстом] мы хотим сосредоточиться на четырех опциях: Disabled [Отключено], Use Frame Shape [Использовать форму фрейма], Use Bounding Box [Использовать границы поля] и Use Contour Line [Использовать линию контура].

Disabled полностью отключает эту функцию, и текст будет просто проходить над или под изображением, полностью игнорируя его форму. Use Frame Shape заставит ваш текст строго придерживаться любой формы, которую вы придали фрейму. Use Bounding Box заставит текст всегда укладываться в идеально прямоугольное поле вокруг изображения. При этом могут оставаться большие пустые промежутки, но некоторым людям именно это и требуется.

Для продвинутых пользователей есть опция Use Contour Line. Ее трудно объяснить без живой демонстрации, но она добавляет вторую форму для обтекания текста, независимо от того, как изображение может выглядеть прямо сейчас. Контур будет основан

» ПРИМЕНЕНИЕ НАПРАВЛЯЮЩИХ

Направляющие в Scribus — штука сложная. Если вы откроете главное меню и нажмете View > Show Guides [Вид > Показать направляющие], появятся крупные уродливые линии, показывающие, где всё происходит. При необходимости их можно включить или выключить по умолчанию, в зависимости от особенностей документа. В совершенно новом документе их включение или выключение, вероятно, ничего не даст, поскольку обычно это вы должны определить сами.

Для новых документов Page > Snap to Guides [Страница > Привязать к направляющим] менее полезна, чем Snap to Grid, поскольку она будет привязываться только к границам страницы, а не к колонкам — можно было бы предположить, что Snap to Guides будет работать с колонками с самого начала. К сожалению, если вы хотите, чтобы ваши фреймы привязывались к колонкам, придется ввести их вручную в Guide Manager (в главном меню выберите Page > Manage Guides [Страница > Управлять направляющими]). Это правильный подход, но у нас просто нет места, чтобы рассмотреть его на этом уроке.

Большинству людей будет достаточно рисовать фреймы от руки по видимым направляющим, хотя, если вы собираетесь работать в одном и том же макете день за днем, вероятно, стоит сделать привязку. Но мы предоставим вам решить самим, где расположить каждую строку и колонку, ведь писать и писать об этом крайне нудно! Может быть, через месяц. Может быть.

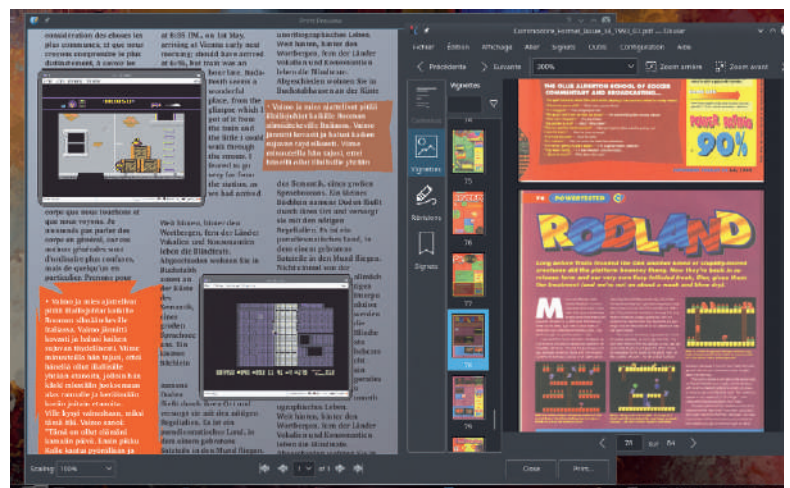
на том, какой была исходная форма фрейма изображения, но нажав Edit, вы можете сделать фигуру практически любой, используя довольно сложные элементы управления.

Готовы к большему?

Закончив форматирование и установив стиль, вы можете создать целый журнал, основанный на вашем собственном стиле и вкусах. Вы также можете использовать это как шаблон, создавая особый стиль, которым смогут наслаждаться ваши читатели.

На этом уроке мы рассмотрели только основы создания макета журнала, больше налегая на работу с текстом. Через месяц мы сосредоточимся на графике, такой как обложки, плакаты и т.д.

Очевидно, что это не среда для замены текстового процессора, но будь это нечто вроде Word или LibreOffice Writer, вы бы без конца мучились, пытаетесь заставить графику удержаться в должном месте и постоянно опасаясь, что разрушите хрупкое форматирование. В Scribus вы просто щелкаете по чему-то, перетаскиваете это, и всё готово. Интерфейс не то что сложный, он просто другой. Придерживайтесь его, и вы будете вознаграждены. **LXF**



Что вы думаете об этой имитации наших предков из Commodore Format 1990-х? Могли ли они использовать те же офисы, что Джонни и Нейл? [Ред.: – Пробыте!]

HTTPS

Обезопасьте HTTP-сайт сертификатом

Михалис Цукалос открывает, как превратить ненадежный HTTP WordPress, Drupal 7 или статическую страницу *Monit* в прочный как скала сайт HTTPS.



**НАШ
ЭКСПЕРТ**

Михалис Цукалос — эксперт UNIX и автор *Go Systems Programming* и *Mastering Go*. Вы можете связаться с ним через www.mtsoukalos.eu и @mactsouk.

И так, вы хотите конвертировать HTTP-сайты в HTTPS? Тогда вы попали по адресу! Для этого урока в качестве основного примера будет использоваться существующий сайт WordPress, но большая часть представленной информации является общей. Процесс переключения HTTP-сайта WordPress на использование HTTPS относительно длительный, но следовать ему нетрудно — надо просто быть аккуратным и терпеливым. Кроме того, вы увидите, как дать инструкцию использовать HTTPS сайту Drupal 7, а также статическому сайту, который в данном случае будет сайтом инструмента мониторинга *Monit*, работающим на собственном web-сервере.

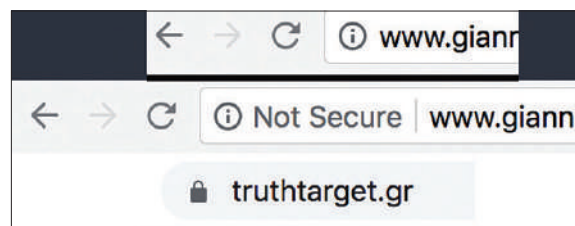
Web-сервис <https://www.whynopadlock.com> поможет вам выяснить, есть ли у вашего сайта HTTPS какие-либо проблемы. Наконец, начиная с *Google Chrome* версии 68, каждый сайт, не использующий HTTPS, будет помечен в *Chrome* как “Not Secure [Незащищенный]” — это можно увидеть на рис. справа. Кроме того, *Google Chrome* 69 больше не отображает слово “Secure” на web-сайтах HTTPS, и это также можно увидеть на рисунке.

Что такое HTTPS?

HTTP означает HyperText Transfer (или Transport) Protocol, тогда как HTTPS означает Hypertext Transfer Protocol Secure. Именно последнее слово и определяет различие в работе сайтов. Причина в основной проблеме с простым старым HTTP: передаваемые данные (простой текст) не шифруются, что позволяет их украсть.

Хотя более старые версии HTTPS использовали для шифрования протокол Secure Sockets Layer (SSL), самые последние реализации HTTPS используют Transport Layer Security (TLS). Это означает, что HTTPS можно также называть или «HTTP через SSL», или «HTTP через TLS», соответственно.

На практике то, что делает HTTPS и чего не делает HTTP — это аутентификация доступа к сайту вместе с его контентом. Это, в свою очередь, защищает конфиденциальность и целостность данных, которыми обмениваются между собой HTTPS-клиент и HTTPS-сервер. Оно, может, и не кажется особо важным, но представьте, насколько это серьезно, когда вы хотите получить доступ к web-серверу своего банка, чтобы получить информацию о своих счетах или перевести денежные средства. Проще говоря, вы хотите быть уверены, что вы подключились к реальному сайту своего банка и что никто больше не видит данные, которыми вы обмениваетесь.



Здесь показано, как *Google Chrome* обращается с сайтами HTTP до и после *Chrome 68*. *Chrome 68* отображает для сайтов HTTP сообщение «Незащищенный». *Chrome 69*, кроме того, отмечает сайты HTTPS замочком.

SEO означает Search Engine Optimisation [Оптимизация поисковой машины] и занимается тем, что пытается получить как можно больше трафика из поисковой машины. Google утверждает, что если ваш сайт не использует HTTPS, он потеряет рейтинг, что плохо! И если ваша компания зависит от SEO, то использование HTTPS обязательно!

О сертификатах

Сертификат — это небольшой цифровой файл, который, помимо всего прочего, содержит информацию об организации. Важно то, что сертификат также включает открытый ключ, необходимый для запуска безопасной сессии HTTP.

Есть компании, предоставляющие сертификаты за плату; есть и такие, которые предоставляют сертификаты бесплатно. На нашем уроке будут использоваться бесплатные сертификаты от Let's Encrypt. Вы можете получить свои сертификаты из любой точки мира — общий процесс будет в основном идентичным. Т.е. вам необходимо получить сертификат открытого ключа для домена web-сервера, подписанный доверенным центром сертификации, которым в этом случае также будет Let's Encrypt. Кроме того, как вы узнаете позже на этом уроке, срок действия сертификатов конечен, и их необходимо обновлять. Это похоже на регулярную смену пароля.

Let's Encrypt!

Прежде чем вы сможете следить за материалом данного раздела, позаботьтесь, чтобы у вас на компьютере Linux был установлен *Git*, поскольку мы будем скачивать утилиты Let's Encrypt.

СКОРАЯ ПОМОЩЬ

Вы можете узнать больше о HTTP и HTTPS из соответствующих RFCs, а именно: tools.ietf.org/html/rfc2616, tools.ietf.org/html/rfc7231, tools.ietf.org/html/rfc2818 и tools.ietf.org/html/rfc2660.

```

apache2 apache-htcacheclean
root@mail:/opt/letsencrypt# /etc/init.d/apache2 stop
[ ok ] Stopping apache2 (via systemctl): apache2.service.
root@mail:/opt/letsencrypt# ./letsencrypt-auto certonly --standalone -d truthtarget.gr -d www.truthtarget.gr
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for truthtarget.gr
http-01 challenge for www.truthtarget.gr
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/truthtarget.gr/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/truthtarget.gr/privkey.pem
   Your cert will expire on 2018-10-07. To obtain a new or tweaked
   version of this certificate in the future, simply run
   letsencrypt-auto again. To non-interactively renew *all* of your
   certificates, run "letsencrypt-auto renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
   Donating to EFF: https://eff.org/donate-le

root@mail:/opt/letsencrypt# /etc/init.d/apache2 stop
[ ok ] Stopping apache2 (via systemctl): apache2.service.
root@mail:/opt/letsencrypt# /etc/init.d/apache2 start
[ ok ] Starting apache2 (via systemctl): apache2.service.
root@mail:/opt/letsencrypt# ls /etc/letsencrypt/live/truthtarget.gr/
cert.pem chain.pem fullchain.pem privkey.pem README
root@mail:/opt/letsencrypt#

```

Вывод команды Let's Encrypt, использованной для создания сертификатов для одного сайта. Для этого шага вам нужно, чтобы TCP-порт 80 не использовался.

Инструменты командной строки для скачивания Let's Encrypt требуют исполнения следующей команды с правами root:

```
# git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt
```

Эта команда помещает всю установку Let's Encrypt вместе с требуемыми утилитами в **/opt/letsencrypt**. Теперь вы готовы создать сертификат для своего сайта. Обратите внимание, что эта команда не имеет ничего общего с WordPress или Drupal, или с тем, как вы создадите свои сайты.

Получите сертификат

Получение сертификата от Let's Encrypt для данного сайта с собственным доменным именем (www.truthtarget.gr) — первый шаг на пути использования им HTTPS. Он требует исполнения команды **letsencrypt-auto** от имени root:

```

# cd /opt/letsencrypt
# ./letsencrypt-auto certonly --standalone -d truthtarget.gr -d www.truthtarget.gr
...
Problem binding to port 80: Could not bind to IPv4 or IPv6.
[Проблема соединения с портом 80: Невозможно соединиться с IPv4 или IPv6]

```

Обратите внимание, что если у вашего сайта несколько имен (алиасов), но один файл настройки Apache 2, вам надо ввести каждое из этих имен в качестве значения в опцию **-d**. В данном случае доступ к www.truthtarget.gr можно также получить через truthtarget.gr, отсюда использование **-d** дважды.

Последнее сообщение об ошибке гласит, что имеется процесс UNIX, занимающий TCP-порт 80, и в большинстве случаев это приложение web-сервера (*Apache* или *Nginx*). Чтобы последняя команда была успешно выполнена, вам нужно остановить *Apache* (**/etc/init.d/apache2 stop** или **service apache2 stop**), выполнить команду и снова запустить *Apache* (**/etc/init.d/apache2 start** или **service apache2 start**).

На рис. вверху справа показан вывод всего процесса, и в нем вы также можете увидеть местоположение, где Let's Encrypt сохраняет сертификаты, а также дату истечения срока действия сертификатов («Срок действия вашего сертификата истекает

2018-10-07»). Созданные сертификаты для сайта www.truthtarget.gr хранятся в **/etc/letsencrypt/archive/truthtarget.gr/** или как символические ссылки UNIX в **/etc/letsencrypt/live/truthtarget.gr**.

В этом пункте вам надо перейти в каталог, где *Apache* хранит свои файлы настройки, чтобы внести необходимые изменения в файл настройки *Apache* 2 для вашего сайта. Если у вас несколько сайтов, вам придется выполнять вышеупомянутые команды несколько раз, по одной для каждого домена. Проще говоря, вам нужен отдельный сертификат для каждого сайта, который вы хотите конвертировать в HTTPS.

Настройка Apache

Следующий шаг заключается в настройке *Apache* на использование сертификатов, которые вы создали в предыдущем разделе. Файл настройки www.truthtarget.gr вызывается его администратором как **/etc/apache2/sites-available/truthtarget.gr**, и это сайт, созданный на WordPress. Рис. внизу стр. 78 показывает финальную версию файла настрой-

ки *Apache* для сайта WordPress.

Здесь есть один хитрый момент. Хотя сайт будет работать только с HTTPS, файл настройки *Apache* состоит из двух разделов: один для старого доброго HTTP и другой для более безопасного HTTPS. Последний будет обслуживать запросы, тогда как первый будет перенаправлять запросы ко второму, потому что вам всё еще нужна поддержка HTTP.

И, наконец, учтите, что обычно HTTP использует TCP-порт номер 80, а HTTPS использует TCP-порт номер 443.

Как насчет WordPress?

Давайте выясним, как настроить WordPress, потому что www.truthtarget.gr использует WordPress для поддержки HTTPS, и прекратим обслуживание страниц с помощью HTTP. Это простой

СКОРАЯ ПОМОЩЬ

Если ваш сайт WordPress всё еще не работает с HTTPS, то вам нужно начать процесс отладки, отключив все плагины WordPress. Обычно плагины WordPress не разрешают перенаправление HTTPS.

» БЕЗОПАСНЫЙ DRUPAL 7 С HTTPS

У процесса переключения Drupal на использование HTTPS вместо HTTP много общего с сайтом WordPress, поэтому для целей нашего урока будет использоваться сайт <http://www.mtsoukalos.eu>. Сначала вам надо создать сертификаты для этого сайта, которые не имеют отношения к способу реализации сайта:

```

# /etc/init.d/apache2 stop
# ./letsencrypt-auto certonly --standalone -d mtsoukalos.eu -d www.mtsoukalos.eu
# /etc/init.d/apache2 start

```

Во-вторых, надо настроить *Apache* или *Nginx*, как вы делали с сайтом WordPress. Но здесь сходство заканчивается, потому что после этого вам придется внести изменения в основной файл конфигурации Drupal 7, расположенный на **./public_html/sites/default/settings.php**. Первое изменение — добавить в конце файла `$conf['https'] = TRUE;`, а второе изменение — заново определить значение `$base_url`, которое в данном случае будет `$base_url = 'https://www.mtsoukalos.eu';`.

Вносить изменения в файл **.htaccess** сайта Drupal 7 незачем, что и радует: меньше мороки — это всегда хорошо.

Рис. внизу стр. 79 показывает результаты <https://www.whynopadlock.com> для <https://www.mtsoukalos.eu>.



СКОРАЯ ПОМОЩЬ

Если ваш сайт WordPress использует плагин iThemes Security, то в конечном итоге вы можете начать видеть странные сообщения об ошибках при попытках доступа к своему HTTPS. В этом случае загляните на <https://darekay.com/blog/solve-redirect-loop-problem-wordpress/>, чтобы найти решение.

двухэтапный процесс. Первым делом вам надо добавить следующую строку в файл `wp-config.php` сайта WordPress:

```
define('FORCE_SSL_LOGIN', true);
```

Данное изменение позволит вам подключиться к панели управления вашего сайта WordPress по протоколу HTTPS — это необходимо для выполнения остальных изменений. Если вы можете подключиться к панели администрирования WordPress по протоколу HTTPS, это хороший признак того, что ваши сертификаты и конфигурация *Apache* работают.

Следующее, что надо сделать — перейти в Settings > General [Настройки > Общие] на панели администрирования WordPress и изменить значения “WordPress Address (URL)” и “Site Address (URL)” на использование HTTPS. На рис. справа показано состояние до и после внесения изменений.

В зависимости от вашего сайта и типа используемых им ресурсов, вам может потребоваться установить дополнительный модуль WordPress для перезаписи существующих URL-адресов HTTP в их HTTPS-эквивалент. Плагином, который будет использоваться на нашем уроке, будет SSL Insecure Content Fixer. Вы можете найти более подробную информацию о нем на <https://ssl.webaware.net.au>. Учтите, что если сайт WordPress, использующий HTTPS, вы создаете с нуля, вполне вероятно, что вам не понадобится такой плагин, потому что все ссылки будут использовать HTTPS. Однако для имеющихся сайтов это может сэкономить вам массу времени!

Рис. сверху стр. 79 показывает настройки плагина SSL Insecure Content Fixer. Вообще говоря, используйте самый простой уровень, который сделает вашу работу — это означает, что вы не должны активировать уровень “Capture All”, когда достигли уровня “Content”. Поиграйте с настройкой, для выяснения, что работает для вас.

Изменение .htaccess

Текущая конфигурация *Apache* требует, чтобы вы не вносили изменений в файл `.htaccess`, который находится в корневом каталоге установки WordPress. Но если вы предпочитаете изменить `.htaccess` вместо того, чтобы работать с файлами конфигурации *Apache*, можете поместить в `.htaccess` следующий текст:

```
<IfModule mod_rewrite.c>
RewriteEngine On
```



Изменения, которые вам надо внести на экране Settings > General в WordPress, чтобы он использовал HTTPS вместо HTTP.

```
RewriteCond %{HTTPS} off
```

```
RewriteRule ^(.*)$ https://%{HTTPHOST}%{REQUEST_URI} [L,R=301]
```

```
</IfModule>
```

Имейте в виду, что внесение изменений в `.htaccess` имеет одно преимущество и один недостаток. Преимущество состоит в том, что изменения вступают в силу немедленно, без перезапуска каких-либо процессов. Однако опечатка или неправильная конфигурация в `.htaccess` могут привести к тому, что ваш сайт станет недоступным. Поэтому любые изменения, внесенные в `.htaccess`, следует тестировать.

Если после изменения `.htaccess` у вас возникли проблемы с подключением к вашему сайту, первое, что надо сделать — это временно отключить все плагины WordPress. Дело в том, что плагины WordPress могут запутать *Apache* и выполнить собственное переписывание динамических страниц вашего сайта. Чем сложнее плагин WordPress, тем более вероятно, что он мешает переписыванию правил *Apache* и файла `.htaccess`!

Обновление сертификатов

Сертификаты, как правило, следует обновлять регулярно. Процесс обновления существующего сертификата Let's Encrypt требует исполнения следующих команд с правами root:

```
# cd /opt/letsencrypt
# ./letsencrypt-auto certonly --standalone --renew-bydefault-d
truthtarget.gr -d www.truthtarget.gr
```

```
...
```

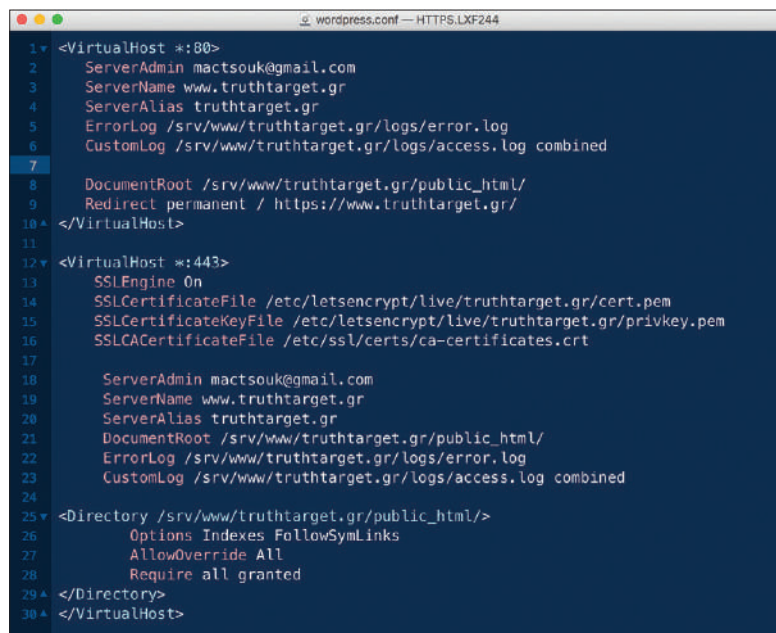
```
Problem binding to port 443: Could not bind to IPv4 or IPv6.
```

Опять же, перед выполнением команды для обновления ваших сертификатов вы должны остановить *Apache 2* (`service apache2 stop`), потому что, если *Apache 2* уже работает, процесс обновления завершится неудачно. Таким образом, после остановки вашего web-сервера команда возобновления будет выполнена успешно и сгенерирует вывод другого типа. Интересная часть этого вывода выглядит следующим образом:

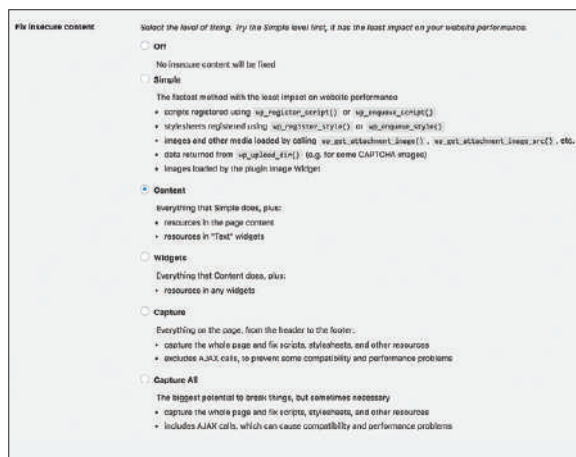
```
Your cert will expire on 2018-12-05. To obtain a new or tweaked
version of this certificate in the future, simply run letsencrypt-
auto again. To non-interactively renew *all* of your certificates,
run "letsencrypt-auto renew" [Срок действия вашего
сертификата истекает 2018-12-05. Чтобы получить новую или
измененную версию этого сертификата в будущем, просто
снова запустите letsencrypt-auto. Чтобы не обновлять *все*
ваши сертификаты в интерактивном режиме, запустите
"letsencrypt-auto renew"]
```

Этот вывод показывает, что срок действия нового сертификата истекает в начале декабря 2018 г., поскольку Let's Encrypt продлевает свои сертификаты на три месяца, а эта команда была выполнена в сентябре 2018 г. Использование новых сертификатов не требует никаких дополнительных действий с вашей стороны.

Наконец, вы можете автоматизировать процесс и выполнять команды для обновления сертификатов Let's Encrypt в качестве процедур `cron`. Тем не менее, в первый раз перед использованием `cron` было бы неплохо выполнить процесс вручную, просто чтобы убедиться, что всё работает, как ожидалось.



Финальная версия файла конфигурации *Apache*, использованного для сайта WordPress. Как видите, здесь два раздела: один для поддержки HTTP и еще один — для HTTPS.



Показана настройка плагина WordPress SSL Insecure Content Fixer, используемого для преобразования ссылок на существующие ресурсы с HTTP в HTTPS.

Monit с использованием HTTPS

В учебнике **LXF242** мы говорили о Munin и Monit. В этом разделе мы узнаем, как конвертировать сайт Monit с HTTP, который мы создавали в этом учебнике, на HTTPS.

Поскольку Monit использует свой собственный web-сервер, процесс будет немного отличаться от предыдущего. Вам надо будет указать Monit, что вы хотите подключиться к его web-интерфейсу по HTTPS, а также создать сертификаты для поддержки этой функции.

Сначала вам надо внести изменения в файл конфигурации Monit, указав, что вы хотите использовать HTTPS вместо HTTP. Неплохо было бы создать в этот момент резервную копию `/etc/monit/monitrc`. Итак, поместите следующие строки в разделе `set httpd` в `/etc/monit/monitrc`:

```
set httpd port 2812 and
  ssl enable
  pemfile /etc/monit/certs/monit.pem
  use address 127.0.0.1
  allow admin:monit
```

Самые важные строки здесь — `ssl enable` и строка, устанавливающая путь к переменной `pemfile`, которая является сертификатом. Затем надо создать этот сертификат с помощью OpenSSL — пожалуйста, установите в своей системе Linux OpenSSL, если его там еще нет. После этого надо будет выполнить следующие команды с правами root:

```
# mkdir /etc/monit/certs/
# cd /etc/monit/certs/
# vi monit.cnf
# openssl req -new -x509 -days 365 -nodes -config ./monit.cnf \
-out ./monit.pem -keyout ./monit.pem
# openssl dhparam 1024 >> /etc/monit/certs/monit.pem
# openssl x509 -subject -dates -fingerprint -noout -in ./monit.pem
# chmod 700 monit.pem
# service monit restart
```

Изучите исходный код, поставляемый с этим учебником, чтобы выяснить содержание `/etc/monit/certs/monit.cnf` — это файл конфигурации, используемый OpenSSL для создания сертификата Monit.

Если процесс пройдет хорошо, вы сможете без проблем подключиться к `https://localhost:1234/`. Однако здесь опять есть сложный момент. Поскольку используемый нами сертификат не сертифицирован ни одной организацией, *Firefox* будет на это

» СРАВНЕНИЕ HTTPS С HTTP

Главное преимущество HTTPS в том, что он быстрее, чем HTTP. Однако есть несколько технических причин, по которым HTTPS лучше, чем HTTP. Злоумышленник может перенаправить HTTP-трафик на поддельные сайты, чтобы украсть имена пользователей и пароли, используя технику, называемую перехват, или перенаправление, DNS. HTTPS может предотвратить перехват DNS, который может происходить на компьютерах web-сервера, а также на других сетевых устройствах, таких как коммутаторы, роутеры и модемы, которые обмениваются данными по протоколу HTTP. Кроме того, в существующую страницу HTTP могут быть вставлены рекламные объявления или другой контент, причем исходная страница ничего не будет об этом знать. Точно так же на HTTP-странице без ведома ее владельца может быть вставлено и выполнено программное обеспечение для майнинга. Все эти гнусные действия подпадают под зонтичные атаки типа «посредник», а HTTPS дает вам довольно прочную уверенность в том, что в вашей двунаправленной связи с web-сервером нет самозванца, выступающего в роли посредника.

Ну и вообще, важно не то, что HTTPS безупречен или совершенен, а то, что он имеет меньше недостатков и недочетов, чем HTTP. По всем этим и многим другим причинам вам стоит начать конвертировать все ваши сайты в протокол HTTPS.

жаловаться, и вам придется добавить в *Firefox* исключение, чтобы доверять IP-адресу `127.0.0.1`.

Ключевым моментом данного раздела является то, что если сайт использует свой собственный web-сервер, процесс преобразования этого сайта в HTTPS может немного отличаться. Кроме того, переделывать сайты со статическими файлами на использование HTTPS намного проще, чем когда вам приходится иметь дело с динамическими сайтами.

Удачи вам с обеспечением безопасности ваших сайтов.

И в заключение автор данного учебника хотел бы выразить благодарность Каллиоппи [Kallioipi] (www.popelix.gr) за ее помощь и оценку. **LXF**



Вывод сайта <https://www.whynopadlock.com> при исследовании <https://www.mtsoukalos.eu>. Неплохо также потратить несколько минут, проверяя свои сайты на предмет ошибок или нестыковок.

JAVASCRIPT

Ботостойкие web-формы

Кент Ельчук обсуждает, как создать и улучшить безопасность web-форм, и объясняет тонкости ботов и спама.



**НАШ
ЭКСПЕРТ**

Кент Ельчук — опытный web-разработчик и энтузиаст Linux, который на досуге программирует и выращивает гидропонные пищевые культуры.

СКОРАЯ ПОМОЩЬ

Дополнительный уровень защиты в формах — проверка слов в теле сообщения электронной почты. Таким образом можно сразу отфильтровать сообщения, содержащие слова типа Viagra, SEO, отпуск и бесплатно.

Формы в Сети вездесущи. Мы используем их, чтобы войти в Facebook, получить доступ к нашей панели администратора WordPress и расспросить компанию о конкретном продукте или услуге. Такими мы видим их в браузере. Но это не объясняет, как они работают. Так что давайте заберемся под капот и посмотрим поближе.

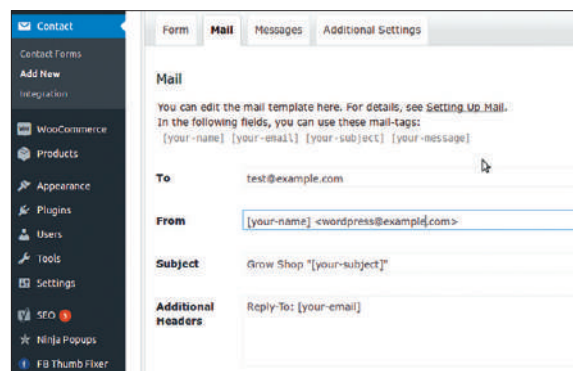
По завершении этого урока мы освоим многие тонкости форм, таких как написание кода, чтобы пользователи могли предоставить информацию, да еще и написать код для обработки этого ввода после его отправки. Помимо форм, мы более подробно рассмотрим безопасность и web-ботов, тем самым раскрыв, что web-формы — это гораздо больше, чем кажется на первый взгляд. Без достаточных знаний о создании, защите и предотвращении спама из форм мы могли бы получать массовые рассылки с сомнительной рекламой фармацевтических фирм и нежелательных услуг SEO, способные загромождать почтовый ящик.

Создание форм

В принципе, web-формы создаются двумя разными способами. Один для интерпретации данных использует код на стороне сервера и «что-то с ними делает», а другой использует для обработки данных JavaScript (или библиотеку JavaScript, такую как JQuery). Во втором случае для обработки данных web-формы обычно применяется метод Ajax. Ajax использует JavaScript (или JQuery), но умеет взаимодействовать и с серверным языком, вроде PHP, для обработки отправленных форм и взаимодействия со страницей. Это также означает, что данные на странице могут обновляться без необходимости обновления страницы.

Форма, опубликованная без Ajax, перезагрузит всю страницу, на которой она размещена. Хотя это и работает, метод, конечно, никоим образом не идеальный. Чтобы проиллюстрировать это, мы начнем с самой простой формы Contact Us [Свяжитесь с нами]. Теги формы начинаются с `<form>` и заканчиваются `</form>`.

Как мы видим ниже в первой строке, первый тег формы также содержит два атрибута. Один атрибут — это `action`, он отправляет данные формы по указанному URL; который в данном случае является файлом с именем `contact_submitted-LF.php`. Другой метод атрибута сообщает, как отправлять данные формы, что в данном случае осуществляется методом POST. Другим распространенным вариантом является GET, который, в отличие от POST, небезопасен. А теперь глянем на приведенный ниже HTML-код, а потом разберемся, что всё это значит.



Пока мы не позаботимся о добавлении Google Recaptcha, формы будут крайне уязвимы к спаму.

```
<form action="contact_submitted-LF.php" method="post">
  <div class="control-group">
    <label>Name</label>
    <div class="controls">
      <input type="text" name="names"/>
    </div>
  </div>
  <div class="control-group">
    <label>Email <span class="color-red">*</span></label>
    <div class="controls">
      <input type="text" name="email">
    </div>
  </div>
  <div class="control-group">
    <label>Message</label>
    <div class="controls">
      <textarea rows="8" name="message"></textarea>
    </div>
  </div>
  <div class="control-group">
    <input type="submit" class="btn theme-btn" name="submit"
    value="Submit"/>
  </div>
</form>
```

Итак, мы уже сообщили, что форма находится внутри тегов `<form>`. Нам также надо знать, что есть несколько полей ввода,

текстовая область и кнопка отправки, которая используется для получения данных и их отправки.

Давайте начнем с полей ввода и текстовых областей. Все они имеют атрибут `name` и специальное значение, прикрепленное к ним. Например, `name = "names"` будет способом для интерпретатора отследить информацию из поля, где люди вводят свое имя. То же относится и к другим входам и к блоку `textarea`. Мы должны отметить, что тип ввода `submit` не является текстом, как другие элементы ввода. Он так и называется — `submit`. Как и в других полях, у него есть атрибут `name`, который здесь имеет значение `submit`. Однако это значение представляет собой просто текст, который вы видите на кнопке при просмотре в браузере.

Теперь, когда мы видим детали формы, наш следующий шаг — разобраться с данными постов, чтобы они могли сделать нечто полезное, например, вставить данные в базу и/или отправить электронное письмо с подробностями. В нашем случае мы отправим нашу форму по электронной почте. Имея это в виду, давайте рассмотрим приведенный ниже код, а затем объясним, что происходит. Серверным языком, который будет использоваться для обработки форм, является PHP:

```
$instructions = array(
    'names' => FILTER_SANITIZE_STRING,
    'email' => array('filter' => FILTER_VALIDATE_EMAIL),
    'message' => array('filter' => FILTER_SANITIZE_STRING));
$filtered = filter_input_array(INPUT_POST, $instructions);
if ($filtered['names'] == "" || $filtered['message'] == "" ||
    empty($filtered['email'])) {
    echo "<h2>Контакт неудачен</h2>";
    echo "Пожалуйста, вернитесь и заполните все поля";
} else {
    if ($filtered['email'] == "") {
        echo "<h2>Контакт неудачен</h2>";
        echo "Пожалуйста, вернитесь и укажите адрес почты.";
    } else {
        if ((substr_count($filtered['message'], 'http') || substr_count(
            $filtered['message'],
            '&lt;a'
        )) || substr_count($filtered['message'], '<a') > 0
        ) {
            echo "<h2>Контакт неудачен</h2>";
            echo "Пожалуйста, вернитесь и удалите ссылки.";
        } else {
            $email = $filtered['email'];
            $email_send_to = "test@example.com";
            $subject = "Enquiry From Website";
            $name = $filtered['names'];
            $message = $filtered['message'];
            $body = "";
            $body .= "name: ";
            $body .= $name;
            $body .= "\n";
            $body .= "message: ";
            $body .= $message;
            $body .= "\n";
            $mail_send = mail($email_send_to, $subject, $body,
                "From: <$email>");
            ?>
            <h2>Контакт удачен</h2>
            <p>Мы ответим с наивозможной скоростью.</p>
            <?php }
        }
    }
}
```

Во-первых, две переменные PHP, с именами `$instructions` и `$filtered`, очищают данные от пользователя и гарантируют, что электронная почта принимает требуемый формат. Причина,

по которой мы очищаем данные, заключается в том, что некоторые люди могут попытаться записывать данные в текстовые поля, что может вызвать, например, SQL-инъекцию и запись в базу данных. Эти меры предосторожности, принятые выше, в наше время необходимы. Затем функция `filter_input_array()` создает массив входных данных из формы. Таким образом, введенный пользователем адрес электронной почты — `$filtered['email']`, а его имя — `$filtered['names']`.

Переходим к следующей строке: у нас есть условия, которые надо проверить и убедиться, что опубликованные данные не пусты. Если есть пустые поля имени, адреса электронной почты или сообщения, пользователю предлагается вернуться назад и исправить положение. В дополнение к проверке данных нам надо убедиться, что у пользователя в электронном письме нет ссылок. Часто ссылки являются признаком спама. И в нашем случае мы не будем проявлять терпение, а отправим их обратно, чтобы избавиться от них.

Функция `substr_count()` выполняет эти проверки ссылок как для фактического начального кода ссылки `<a`, так и для своего объекта HTML `<a`, чтобы охватить обе возможности.

Если всё идет хорошо, у нас есть посетитель, который ввел действующий адрес электронной почты, настоящие имена и достойное сообщение. В этом пункте мы создаем и собираем переменные для получателя электронной почты, темы, сообщения и заголовков. В нашем случае электронное письмо будет отправлено на адрес `test@example.com`. Темой будет запрос с сайта. Тогда мы узнаем, что это пришло из нашей формы. Переменная `body` будет телом электронной почты и будет содержать отправителя и сообщение.

Наконец, мы забираем адрес электронной почты отправителя, чтобы мы могли легко ответить ему. Функция PHP, которая связывает всё это вместе и отправляет электронную почту — `mail()`.

СКОРАЯ ПОМОЩЬ

Никогда не доверяйте вводу пользователей. При создании форм, особенно если данные из них вводятся в базу, можно использовать встроенные функции PHP, например, `htmlspecialchars()` или `htmlentities()`, и очистку фильмами с `https://bit.ly/2QNRDAm`. Помните: данные, передаваемые по SSL (HTTPS), шифруются, а HTTP — простой текст, его легко перехватывает Wire-shark через Wi-Fi.

» УПРАВЛЕНИЕ СЕССИЕЙ

Управление сессией может отслеживать активность пользователей и помогать нам в случае проблем с пользователями. Мы просто добавляем следующую строку кода в начало нашего файла для работы с сессиями:

```
session_start();
```

Элемент управления сессией, который мы собираемся использовать, позволит кому-то заполнить наши формы только один раз, поскольку сессия создается после первой отправки формы, а последующие посещения этой страницы распознают сессию и обходят функцию `mail()`.

Чтобы добавить управление сессией, мы можем окружить следующими скобками кода `else` наш предыдущий проверочный код:

```
if(isset($_SESSION['sent'])) {
} else {
}
```

Код, приведенный выше, переводится на человеческий язык. Это PHP-код, который гласит: «Если переменная сессии с именем 'sent' существует, ничего не делать; в противном случае продолжайте и запустите остальную часть содержимого, заключенную в скобки».

Это только часть проверок сессии. Мы добавляем следующий набор кода в конец файла чуть выше последнего тега `</body>`:

```
if(isset($_SESSION['sent'])) {
} else {
    $_SESSION['sent'] = true;
}
```

Код, который мы добавили, ничего не делает, если отправленная переменная сессии уже существует; а если ее не существует, он создает ее. Таким образом, после того, как письмо отправлено нам, эта переменная существует и останется, даже если пользователь возвращается к нашей форме и отправляет ее заново.

СКОРАЯ ПОМОЩЬ

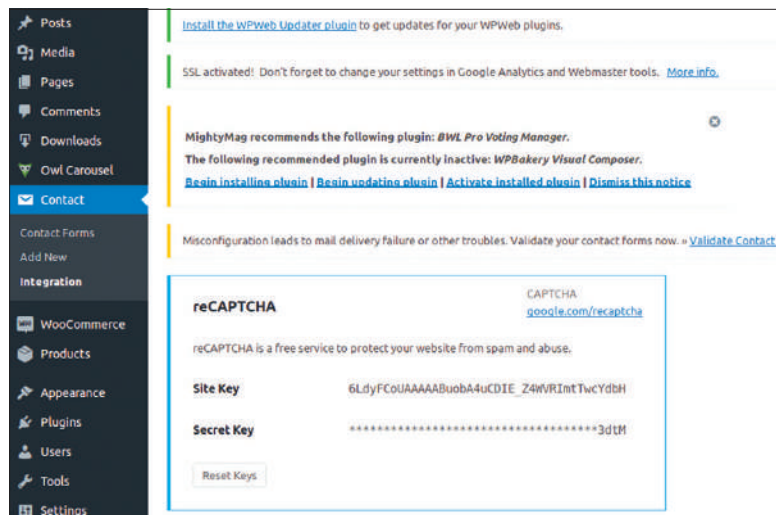
Формы — прекрасный способ налаживания контакта с web-пользователями, не раскрывая нашего электронного адреса, что может помочь нам избежать появления в спам-списках.

Если бы мы хотели сделать дополнительный шаг, мы могли бы получить IP-адрес пользователя и добавить его в тело письма. В худшем случае это дает нам возможность занести в черный список IP-адреса, запретив им посещать наш сайт.

Хотя всё вышеперечисленное кажется в некоторой степени безопасным и защищенным, правда в том, что это не спасет от спама. Не совсем глупый спамер может легко использовать небольшой фрагмент кода, показанный ниже, и запускать автоматизированные задачи для отправки своей электронной почты. Кроме того, хотя это и не показано в примере, посетитель может подделывать такие вещи, как агенты браузера, и создавать массивы имен, сообщений и адресов электронной почты, которые могут выглядеть так, будто другой пользователь заполняет наши формы, хотя это тот же самый базовый скрипт, имеющий губительное действие.

Здесь у нас немного кода, который можно запустить из командной строки, из cron или из браузера. Итак, давайте посмотрим, что он делает. После этого мы расскажем о методах, позволяющих гарантировать, что с этими тактиками можно будет справиться, добавив такие функции, как Google Recaptcha.

```
<?php
$postfields = array();
$postfields["action"] = "submit";
$postfields["names"] = "Kent";
$postfields["email"] = "spammer@example.com";
$postfields["message"] = "Carl, Blah blah";
$url = "https://example.com/contact_submitted-LF.php";
$useragent = "Mozilla/5.0";
$referer = $url;
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, 1);
//We post $postfields data
curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);
//We define an useragent (Mozilla/5.0)
curl_setopt($ch, CURLOPT_USERAGENT, $useragent);
//We define a referer ($url)
curl_setopt($ch, CURLOPT_REFERER, $referer);
//We get the result page in a string
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
$result = curl_exec($ch); //We exit CURL
curl_close($ch);
//Finally, we display the result
echo $result;
echo $url;
```



Нам надо настроить ключи Google Recaptcha с нашей учетки Google на <https://bit.ly/2QNRDAm> и добавить их в нашу установку WordPress.

Мы начнем с создания массива элементов — `$postfields`.

К ним относятся действия, необходимые для отправки формы, переменной `names`, нашего адреса электронной почты и сообщения. После этого мы определяем `$url`, куда мы будем отправлять эти данные; это https://example.com/contact_submitted-LF.php.

Остальное, по сути, работает с функциями `curl`. Мы инициализируем ее с указанным нами `$url`, создаем параметры, добавляем наши `$postfields`, выполняем ее и закрываем. Вот и всё. Как только мы запустим файл, владелец формы получит электронное письмо с данными из этого скрипта, и нам даже не нужно было ничего вводить в онлайн-форму. Чудеса!

Google Recaptcha

Подобно тестированию на вторжение на вашем собственном сайте, атака на наши формы — наш лучший способ гарантировать, что они не выйдут из-под контроля. К счастью, помимо проверки данных, с формами прекрасно работает интеграция Google Recaptcha, поскольку это еще один уровень проверок, чтобы убедиться: форму заполняет реальный пользователь, а не автоматический скрипт.

Проясним последние замечания: причина, по которой Google Recaptcha работает, заключается в отправке формой дополнительной переменной `post`, которая будет проверена, чтобы убедиться, что форму заполняет не бот и не скрипт. Чтобы использовать Recaptcha, нам надо зарегистрироваться и указать наш сайт, чтобы получить ключ на стороне клиента и секретную строку на стороне сервера. Сайт — www.google.com/recaptcha. Зарегистрировавшись, мы сможем добавить его на наш сайт. Почти немедленно мы получим наш «ключ» и «секрет». Потом это очень легко реализовать. Итак, давайте рассмотрим, как мы добавляем его на наш сайт.

Для начала мы открываем нашу страницу с формой. Сделав это, мы добавим код ниже перед закрывающим тегом `</head>`:

```
<script src='https://www.google.com/recaptcha/apijs'></script>
```

После этого мы добавляем в форму пустой тег `<div>`. Пример показан ниже:

```
<div class="g-recaptcha" data-sitekey="6LcV4RAUAAAAAE-2iJS1bejxxxxxxNrxRxFz4fg"></div>
```

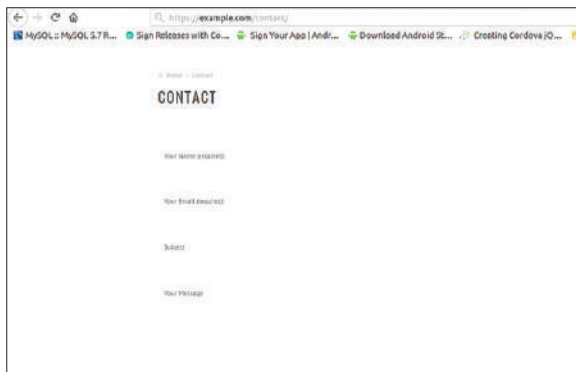
Теперь вернемся к нашему коду, который проверял имена, адреса электронной почты и пустые сообщения. Мы можем добавить код ниже, который отправит ответ из формы и наш ключ, следующим образом:

```
if(isset($_POST['g-recaptcha-response'])){
    $captcha = $_POST['g-recaptcha-response'];
    $secretKey = "6LcV4RAUAAAAEYPv7-qJ4glPpfjCFCqSzZiBU_i";
    $response = file_get_contents("https://www.google.com/recaptcha/api/siteverify?secret=" . Key."&response=" . $captcha);
    $responseKeys = json_decode($response,true);
    if(intval($responseKeys["success"]) != 1) {
        echo '<h2>Вы не отправили эту форму с сайта. Пожалуйста, продолжите.</h2>';
    } else {
        echo '<h2>Спасибо за ваш комментарий.</h2>';
    }
}
```

Поскольку этот код может проверять и фильтровать спам, он очень помогает. Конечно, мы будем использовать его как дополнительную защиту для нашей функции `mail()`, чтобы добавить дополнительный уровень, который мы ранее не учитывали — это скрипт бота, который прошел другие наши фильтры.

К этому моменту мы знаем, как создавать, тестировать и защищать наши web-формы, чтобы наша папка Входящие не заполнялась быстрее, чем McDonalds во время обеденного перерыва.

Теперь давайте продолжим и рассмотрим некоторые случаи, из-за которых мы можем захотеть добавить третий уровень



Основная web-форма, имеющая интеграцию Google Recaptcha. Теперь мы готовы наладить контакт.

защиты: защиту от автоматического заполнения форм в браузере. Поскольку к браузерам вроде *Chrome* можно добавлять расширения, например, Form Filler, они позволяют web-пользователям заполнять формы текстом без необходимости печатать. Да, вы правильно догадались. Посетитель может щелкнуть значок, чтобы заполнить форму, затем коснуться Google Recaptcha и нажать кнопку Send [Отправить].

Мы и охнуть не успеем, как в нашем почтовом ящике появится бессмысленный спам от кого-то, чья идея насчет повеселиться нас раздражала. Что еще хуже, пользователь успеет отправить нам три или четыре письма-спам, пока Google Recaptcha не сообщит пользователю «щелкнуть по всем изображениям с витринами» или «щелкнуть по всем изображениям с уличными знаками», прежде чем действительно остановится и сообразит, что лучше переждать и приняться за спам в следующей зоне бесплатного Wi-Fi.

Нам явно надо что-то сделать во избежание этого. К счастью, такое возможно с помощью управления сессиями.

Формы WordPress

Поскольку множество сайтов создается на WordPress, мы рассмотрим его наиболее популярный плагин для форм, называемый Contact Forms 7. Естественно, для его использования нам понадобится установка WordPress. Плагин можно установить одним щелчком мыши после входа в систему администратора WordPress. Как только он установится, надо будет произвести несколько небольших изменений, и мы сделаемся готовы к свершениям.

Чтобы получить доступ к нашему плагину Contact Forms 7, мы можем выбрать Contact в левом меню или выбрать один из дочерних вариантов, таких как Contact Forms [Контактные формы], Add New [Добавить новую] и Integration [Интеграция]. Опция Integration позволяет нам определить наш ключ сайта Google Recaptcha и секретную строку, чтобы мы могли добавить его в любую форму по нашему желанию. Код ниже показывает, как это выглядит, когда мы создаем новую форму и добавляем в нее Recaptcha:

```
<label> Your Name (required)
[text* your-name] </label>
<label> Your Email (required)
[email* your-email] </label>
<label> Subject
[text your-subject] </label>
<label> Your Message
[textarea your-message] </label>
[recaptcha]
```

» АТАКА СПАМЕРОВ

На нашем уроке мы увидели, как спам можно применять к отдельным URL-адресам и активировать его, просто запустив PHP-файл, который использует *curl* для получения URL-адреса и автоматической отправки формы.

Поскольку данные формы обычно отправляются на адрес электронной почты, пользователь может задавать общие вопросы, такие как цены или запросы, связанные с услугами. С точки зрения владельца сайта, на форму обычно отвечают дружеским письмом.

Это дружеское ответное письмо может оказаться в списке спама, который будет преследовать этот адрес годами. Кроме того, электронное письмо может попасть в списки, которые также продаются другим людям, что приводит к увеличению количества спамеров.

Списки адресов электронной почты могут быть помещены в базы данных MySQL за считанные секунды и проходить через циклы *foreach* с помощью очень простых скриптов. Сами пользователи могут использовать SMTP из учетных записей, таких как Gmail, для отправки их с новых адресов электронной почты.

Несмотря на то, что немало спама в электронной почте отправится в черные списки, эти небольшие скрипты можно запускать с мобильных точек доступа на разных IP-адресах; например, в кофейнях. Как видите, пользователь может просто слоняться по округе и при этом достичь умеренного успеха как спамер.

[submit "Send"]

Получив нашу новую форму, мы можем щелкнуть по вкладке почты и указать адрес электронной почты, на который мы хотим отправить письмо, а затем нажать Save [Сохранить]. Это всё, что имеется в простых формах WordPress. Следует отметить, что мы добавили Recaptcha в форму по умолчанию. Без Google Recaptcha для автоматической отправки форм WordPress мог бы использоваться скрипт *curl*, который мы обсуждали ранее.

Если подумать, существует, вероятно, бесконечное множество сайтов, небезопасно работающих с этим известным плагином. Отличия от кода, о котором мы уже писали, заключаются только в разных полях сообщений. С помощью контактных форм мы по-прежнему можем читать HTML с помощью Inspector браузера и получать URL-адрес сообщения, а также все переменные сообщения.

Пример показан ниже, так как мы могли бы просто изменить несколько строк, которые написали ранее, адаптировав их для контактных форм WordPress. Например:

```
$postfields["_wpcf7"] = "1510";
$postfields["_wpcf7_version"] = "5.0.2";
$postfields["_wpcf7_locale"] = "en_CA";
$postfields["_wpcf7_unit_tag"] = "wpcf7-f1510-p1511-o1";
$postfields["_wpcf7_container_post"] = "1511";
$postfields["your-name"] = "My Name";
$postfields["your-email"] = "spammer_test@example.ca";
$postfields["your-subject"] = "Super Migraine Pills";
$postfields["your-message"] = "Carl, Blah blah";
$url = "https://example.com/test/#wpcf7-f1510-p1511-o1";
```

Не так уж много спама

Теперь у нас есть понимание создания web-форм и их защиты, чтобы мы могли минимизировать нежелательный трафик и автоматических ботов, которые пытаются отправлять общие сообщения как можно большему числу источников. Следующим этапом может быть создание более сложных web-форм, пригодных для создания наших пользовательских web-приложений. Удачного web-формирования! **LXF**

Создаем, решаем и тестируем Судоку

Михалис Цукалос показывает, как писать программы на Go, проверяющие правильность решения головоломки Судоку.



**НАШ
ЭКСПЕРТ**

Михалис Цукалос — автор книг *Go Systems Programming* и *Mastering Go*. Вы можете связаться с ним через @mactsouk.

Сражаетесь с головоломками Судоку? Тогда мы здесь, чтобы облегчить вам жизнь — правда, вам придется заняться программированием на Go, так что... Хотя сложность Судоку зависит от количества чисел, имеющихся у вас при решении, данное руководство не имеет отношения к этому аспекту создания головоломок Судоку.

Мы закодируем две версии программы Go, но обе версии читают квадрат Судоку из текстового файла. Основное различие между ними в том, как они получают доступ к элементам срезов Go, которые хранят головоломку Судоку. Нашим финальным шагом будет сравнительный тест обеих, чтобы посмотреть, какая быстрее. И, наконец, вы увидите, как увеличение размера массива или среза [slice] влияет на производительность программы.

Что такое Судоку?

Судоку — это игра в цифры. И хотя в игре присутствуют числа, она относится не к математике, а к логике. Каждая головоломка Судоку занимает сетку 9×9, которая также делится на сетку 3×3 — каждый элемент этой сетки 3×3 (блок) сам по себе является головоломкой с сеткой 3×3, содержащей 9 чисел.

Допустимые числа в головоломке Судоку с игровым полем 9×9 могут варьироваться от одного до девяти или от нуля до восьми. Все головоломки Судоку, используемые на нашем уроке, будут содержать цифры от одного до девяти, но вообще-то вы можете использовать любые девять чисел, какие захотите. Однако, если вы собираетесь использовать другие числа, вам нужно будет внести изменения в код Go этого урока.

Первая проверка, которая должна быть выполнена, состоит в выяснении, содержит ли каждый из этих девяти блоков все числа от одного до девяти. Это означает, что каждое из этих чисел должно появляться только один раз и что должны присутствовать все девять чисел. Кроме того, необходимо выполнить еще две проверки, чтобы выяснить, содержат ли каждый столбец и каждый ряд квадрата Судоку все доступные числа без дубликатов. Несмотря на то, что большинство оценок Судоку не проходят первый тест, есть вероятность, что они могут пройти первый тест и потерпеть неудачу в двух других.

Поскольку мы пытаемся решить проблему с помощью компьютера, нам придется выполнить все эти тесты и либо остановиться, как только один из тестов не пройден, либо продолжить, пока все тесты не будут успешными. В первом случае

```
13 func importFile(file string) ([]int, error) {
14     var err error
15     var mySlice = make([]int, 0)
16
17     f, err := os.Open(file)
18     if err != nil {
19         return nil, err
20     }
21     defer f.Close()
22
23     r := bufio.NewReader(f)
24     for {
25         line, err := r.ReadString('\n')
26         fields := strings.Fields(line)
27         temp := make([]int, 0)
28         for _, v := range fields {
29             n, err := strconv.Atoi(v)
30             if err != nil {
31                 return nil, err
32             }
33             temp = append(temp, n)
34         }
35         if len(temp) != 0 {
36             mySlice = append(mySlice, temp)
37         }
38     }
39 }
```

Код Go функции `importFile()`, которая построчно читает текстовый файл и импортирует его в срез Go.

головоломка Судоку решена неправильно, тогда как во втором случае головоломка Судоку решена правильно.

Импорт головоломок

Нам нужно найти способ импортировать головоломки Судоку как файл. Вообще говоря, использование файлов для подобных операций очень разнообразно и дает возможность легко передавать головоломки Судоку между разными компьютерами.

Поскольку массив Судоку имеет девять строк и девять столбцов, вы будете считывать 81 цифру от одного до девяти. Однако любая проверка чисел на ошибки будет проводиться на более поздней стадии, потому что функция `importFile()`, которая будет отвечать за чтение простого текстового файла с числами Судоку в будущих программах Go, должна быть как можно более общей.

Единственное, что проверяет функция `importFile()` — это читает ли она правильные целые числа или нет. Проще говоря, `importFile()` будет принимать отрицательные целые или целые числа, большие девяти, но не примет значение "a", которое не является числом или числом с плавающей запятой. Еще один тест, который будет выполнять `importFile()` — проверка того, что все строки в файле ввода имеют одинаковое количество

СКОРАЯ ПОМОЩЬ

Полный исходный код есть на DVD, или скачайте его с <https://linux-format.com/archives?issue=245>.

целых чисел. Первая строка в текстовом файле ввода как раз указывает количество столбцов, которое должно быть в каждом файле ввода.

Простоты ради, файл **loadFile.go** на Go будет содержать только реализацию **importFile()**, чтобы вы могли проверить ее правильность перед использованием в другой программе. Реализацию функции **importFile()** со всеми проверками можно увидеть на рис. стр. 84. Что делает этот код? Во-первых, функция **importFile()** берет один аргумент и возвращает две переменных: срез, который хранит содержимое файла ввода, и переменную ошибки. Если ошибки нет, то переменная ошибки будет **nil** — в ином случае она будет содержать сообщения об ошибке. В последнем случае значение переменной среза будет **nil**. Следующее утверждение **if** используется для проверки того, чтобы все строки среза имели одно и то же количество целых чисел:

```
if len(temp) != len(mySlice[0])
```

Этот код проверяет количество элементов в текущем значении временного среза и сравнивает его с количеством элементов в первой строке среза **mySlice**. Это означает, что в первой строке текстового файла ввода указывается количество элементов, которые вы ожидаете увидеть во всех других строках текстового файла ввода. Если между этими двумя значениями есть разница, функция **importFile()** в качестве возвращаемого значения для среза вернет **nil**, а также переменную ошибки с индивидуальным сообщением об ошибке!

Подтверждение головоломок

Первая версия программы будет сохранена как **sudoku.go**.

Следующий код из **sudoku.go** показывает цикл **for**, используемый для оценки всех элементов среза Go. Учтите, что между этими циклами **for** имеется дополнительный код Go, который здесь не показан:

```
for i := 0; i <= 2; i++ {
    for j := 0; j <= 2; j++ {
        for k := 0; k <= 2; k++ {
            for m := 0; m <= 2; m++ {
```

Как видите, чтобы оценить все элементы головоломки Судоку, **sudoku.go** использует четыре цикла **for**. Хотя использование четырех циклов **for** для массива 9×9, возможно, и не составит

```
33 func validPuzzle(sl []int) bool {
34     for i := 0; i <= 2; i++ {
35         for j := 0; j <= 2; j++ {
36             iEl := i * 3
37             jEl := j * 3
38             mySlice := []int{0, 0, 0, 0, 0, 0, 0, 0, 0}
39             for k := 0; k <= 2; k++ {
40                 for m := 0; m <= 2; m++ {
41                     bigI := iEl + k
42                     bigJ := jEl + m
43                     val := sl[bigI][bigJ]
44                     if val > 0 && val < 10 {
45                         if mySlice[val-1] == 1 {
46                             fmt.Println("Value appeared two times!")
47                             return false
48                         } else {
49                             mySlice[val-1] = 1
50                         }
51                     } else {
52                         fmt.Println("Invalid value:", val)
53                         return false
54                     }
55                 }
56             }
57         }
58     }
59     // Testing columns
60     for i := 0; i <= 8; i++ {
61         sum := 0
62         for j := 0; j <= 8; j++ {
63             sum = sum + sl[i][j]
64         }
65         if sum != 45 {
66             return false
67         }
68     }
69 }
```

А это реализация функции **validPuzzle()**, как она выглядит в **sudoku.go**. Функция использует четыре цикла **for** для оценки всех элементов поля Судоку.

```
func validPuzzle(sl []int) bool {
    for i := 0; i <= 8; i++ {
        mySlice := []int{0, 0, 0, 0, 0, 0, 0, 0, 0}
        x := (i % 3) * 3
        y := (i / 3) * 3
        for j := 0; j <= 8; j++ {
            k := (j % 3)
            m := (j / 3)
            bigI := x + k
            bigJ := y + m
            val := sl[bigI][bigJ]
            if val > 0 && val < 10 {
                if mySlice[val-1] == 1 {
                    fmt.Println("Value appeared two times!")
                    return false
                } else {
                    mySlice[val-1] = 1
                }
            } else {
                fmt.Println("Invalid value!")
                return false
            }
        }
    }
}
```

проблемы с точки зрения производительности, оно определенно стало бы проблемой, если бы нам пришлось работать с намного большим массивом.

Оценки элементов недостаточно — надо выполнить еще две проверки в функции **validPuzzle()**:

```
if val > 0 && val < 10 {
    if mySlice[val-1] == 1 {
        fmt.Println("Value appeared two times!")
        return false
    } else {
        mySlice[val-1] = 1
    }
}
```

Итак, если значение в блоке появляется более одного раза, то головоломка Судоку недействительна, и программа, значит, должна остановиться немедленно. Более того, если значение больше 10, равно 10 или равно 0, то у нас тоже неправильное

Реализация функции **validPuzzle()** в том виде, в каком она имеется в **faster.go**. Функция использует только два цикла **for** для оценки всех элементов поля Судоку.

СКОРАЯ ПОМОЩЬ

Инструменты CI/CD вроде Jenkins (<https://jenkins.io>) и Semaphore (<https://semaphoreci.com>) автоматизируют ваши сборки и покажут вам правильность головоломок Судоку намного быстрее, здорово облегчив вам жизнь.

» ПИШЕМ ТЕСТЫ В GO

Есть несколько правил написания сравнительных тестов в Go. Во-первых, подкоманда **go test** отвечает за тестирование программы. Затем рекомендуется включать функции сравнения в файлы Go, которые заканчиваются на **_test.go**. Все имена функций сравнения должны начинаться с **Benchmark** — функция, имя которой начинается со строчной буквы **b**, не обрабатывается Go как функция сравнения.

Кроме того, у функций сравнения должна быть заданная подпись, которая означает, что функции сравнения должны только принимать параметр ***testing.B** и не возвращать никаких значений. И, наконец, пакет, который содержит функции сравнения должен включать пакет стандартов Go **"testing"**.

Функции сравнения исполняются с помощью команды

```
$ go test -bench=. file.go file_test.go
```

Значение параметра **-bench** задает контрольные функции, которые будут выполняться с применением регулярных выражений. Используемое значение является регулярным выражением, которое соответствует всем действительным определенным функциям сравнения. Если вы пропустите параметр **-bench** или если совпадений не будет, то функция сравнения не будет выполнена.

По умолчанию каждый сравнительный тест выполняется как минимум одну секунду. Если функция сравнительного теста возвращается за время, меньшее одной секунды, значение **b.N** увеличивается, и функция запускается снова. В первый раз значение **b.N** равно 1, затем оно становится 2, 5, 10, 20, 50 и т.д. Это происходит потому, что чем быстрее функция, тем больше раз вам нужно ее запустить, чтобы получить более точные результаты. Однако в сравнительном тестировании труднее всего интерпретировать результаты — конечно, при условии, что ваши функции сравнения вообще работают.

СКОРАЯ ПОМОЩЬ

Чтобы установить Go, вы можете использовать свой любимый менеджер пакетов или перейти на <https://golang.org/dl/> и следовать инструкциям. Узнать, какую версию Go вы используете, можно, выполнив `go version`.

значение, что также немедленно завершит процесс проверки. Эти два требования также делают **sudoku.go** немного быстрее, если мы имеем дело с неверно решенной головоломкой Судоку.

Конечная версия **sudoku.go** будет содержать улучшенную версию `validPuzzle()`, и ее можно увидеть на экранном снимке внизу стр. 85.

Улучшение решения

Хотя предыдущее решение работает, возможно, его можно слегка оптимизировать — данный раздел покажет вам, как это сделать. Общая идея здесь заключается в том, что для большей скорости нам может потребоваться доступ ко всем элементам среза по-другому. Представленное решение будет использовать два цикла `for` для доступа ко всем элементам среза, что сказалось бы сильнее, если бы нам пришлось работать со срезами из более чем 50 столбцов и 50 строк.

Логика **faster.go**, которая позволяет получать доступ ко всем элементам головоломки Судоку, можно обнаружить в следующем коде Go:

```
for i := 0; i <= 8; i++ {
    for j := 0; j <= 8; j++ {
```

Заметьте, что основная логика **faster.go** та же, что и в **sudoku.go** — единственное различие между этими двумя реализациями заключается в том, что каждая программа обращается к элементам среза, который представляет элементы головоломки Судоку. На экранном снимке вверху стр. 85 показан код Go для `validPuzzle()`, который можно найти в **faster.go**. Еще одну реализацию `importFile()` можно увидеть на экранном снимке на стр. 84. Реализация `validPuzzle()` в **faster.go** компактнее и элегантнее, чем в **sudoku.go**.

Следующие два раздела будут сравнивать два решения, чтобы выяснить, есть ли некий выигрыш по производительности при использовании двух циклов `for` вместо четырех для доступа ко всем элементам среза Go.

Сравнение алгоритмов

Мы сравним производительность **sudoku.go** и **faster.go** двумя разными способами. Первый из этих двух — более простой

```
code$ time go run sudoku.go OK.txt
Correct Sudoku puzzle!

real    0m0.275s
user    0m0.216s
sys     0m0.055s
code$ time go run sudoku.go noOK1.txt
Incorrect Sudoku puzzle!

real    0m0.247s
user    0m0.190s
sys     0m0.054s
code$ time go run faster.go noOK1.txt
Incorrect Sudoku puzzle!

real    0m0.249s
user    0m0.178s
sys     0m0.067s
code$ time go run faster.go OK.txt
Correct Sudoku puzzle!

real    0m0.254s
user    0m0.178s
sys     0m0.073s
code$
```

Результаты команд `time go run sudoku.go` и `time go run faster.go`.

```
code$ go test -bench=. sudoku.go sudoku_test.go
testing: warning: no tests to run
BenchmarkValidPuzzle      100000      22945 ns/op
BenchmarkInvalidPuzzles   30000      46914 ns/op
PASS
ok      command-line-arguments  4.425s
code$ go test -bench=. faster.go faster_test.go
testing: warning: no tests to run
BenchmarkValidPuzzle      100000      22692 ns/op      9008 B/op
BenchmarkInvalidPuzzles   30000      44255 ns/op      18016 B/op
PASS
ok      command-line-arguments  4.272s
code$ go test -bench=. faster.go faster_test.go
testing: warning: no tests to run
BenchmarkValidPuzzle      100000      21655 ns/op      9008 B/op
BenchmarkInvalidPuzzles   30000      43830 ns/op      18016 B/op
PASS
ok      command-line-arguments  4.147s
code$ go version
go version go1.7.4 linux/amd64
code$
```

Результаты сравнения для **sudoku.go** и **faster.go**. В данном случае имя **faster.go** [*faster* – *англ.* быстрее] не имеет никакого отношения к производительности программы!

и включает применение утилиты *time*. Следующий раздел использует более сложный метод.

На рис. внизу слева показан вывод **sudoku.go** и **faster.go** при использовании *time*. К сожалению, используя *time*, в таких небольших задачах трудно измерить разницу в производительности между **sudoku.go** и **faster.go**. Вывод *time* был бы более точным, если бы мы несколько раз тестировали **sudoku.go** и **faster.go**, используя больший пример головоломки Судоку — Go может помочь вам в этом, поэтому в следующем разделе вы увидите способ Go написания более интеллектуальных тестов с помощью функций сравнения Go.

Иногда компилятор Go может оптимизировать данный код за кулисами. Поскольку срезы являются популярными структурами данных, метод доступа к ним обычно является одним из первых моментов, которые компилятор пытается улучшить.

Сравнительные тесты в Go

Go позволяет писать сравнительные тесты для сравнения производительности разных реализаций функций, поэтому в данном разделе мы напомним несколько сравнений Go, чтобы сравнить реализации функции `validPuzzle()` в **sudoku.go** и **faster.go**.

Код Go для сравнительных тестов двух функций будет один и тот же. Однако нам надо создать два новых файла: один для **sudoku.go** и другой — для **faster.go**, которые мы назовем **sudoku_test.go** и **faster_test.go**, соответственно.

Важный код Go **sudoku_test.go** и **faster_test.go**, который работает с реализацией функции сравнения, следующий:

```
func BenchmarkValidPuzzle(b *testing.B) {
    benchmark_validPuzzle(b, "/OK.txt")
}

func BenchmarkInvalidPuzzles(b *testing.B) {
    benchmark_validPuzzle(b, "/noOK1.txt")
    benchmark_validPuzzle(b, "/noOK2.txt")
}
```

Функция `BenchmarkValidPuzzle()` проверяет правильную головоломку Судоку, тогда как `BenchmarkInvalidPuzzles()` обрабатывает две неправильные головоломки Судоку. На рис. вверху показаны результаты тестирования как для **sudoku.go**, так и для **faster.go**. Каждый сравнительный тест выполняется двумя способами: во-первых, обычным и, во-вторых, с помощью параметра `-benchmem`, который отображает информацию об использовании памяти каждой функцией сравнения.

Итак, что говорят нам результаты сравнительного теста? Прежде всего, они показывают, что производительность **faster.go** и **sudoku.go** практически одинакова. Это главным образом связано с тем фактом, что мы работаем со срезом 9×9, который

очень мал. Кроме того, поскольку обе программы Go должны иметь доступ к одинаковому количеству элементов, логично, что для этого им требуется одинаковое количество времени.

Тогда зачем нам два решения? Во-первых, решение в **faster**. **go** намного элегантнее, чем реализация **sudoku.go**, главным образом благодаря меньшему числу циклов **for**; и во-вторых, всегда хорошо попытаться заставить всё работать быстрее!

Срезы пирога побольше

В последнем разделе нашего урока будет проведен сравнительный тест программы Go, которая обращается к большим срезам, не выполняя никаких вычислений. Срез будет заполнен случайно сгенерированными числами от одного до девяти. Этот срез может иметь переменное число строк и столбцов — однако, если программе не задан аргумент командной строки, срез будет иметь 1002 строки и 1002 столбца, к которым будет осуществляться доступ в блоках (сетки 3×3).

Имя программы Go, которая будет включать реализацию, будет **big.go**, а функции сравнения будут включены в **big_test.go**. Пожалуйста, учтите, что код Go, позволяющий программе получать доступ ко всем элементам среза, похож на то, как функция **validPuzzle()** в **faster.go** получает доступ ко всем элементам среза.

В **big_test.go** содержится три функции сравнения. Первая использует срез с 1002 строками и 1002 столбцами, вторая — срез 102×102, и последняя — срез 9×9. Код Go последней таков:

```
func Benchmark_9_accessPuzzle(b *testing.B) {
    var r bool
    sl := createSlice(9)
    for i := 0; i < b.N; i++ {
        r = accessPuzzle(sl, 9)
    }
    result = r
}
```

Причина сохранения результата **accessPuzzle()** в переменной с именем **r** и последующего использования другой глобальной переменной с именем **result** довольно непростая. Достаточно сказать, что этот метод используется, чтобы не дать компилятору выполнить какие-либо оптимизации, которые исключат функцию, которую вы хотите измерить, поскольку ее результаты никогда не используются. Это также произошло в **faster_test.go**

» РЕШЕНИЕ ГОЛОВОЛОМКИ СУДОКУ

Хотя в этом руководстве представлен код, который занимается проверкой головоломок Судoku, которые уже были решены, существуют алгоритмы и методы, способные помочь вам решить нерешенную головоломку Судoku или создать новую такую головоломку. Как вы можете себе представить, эти алгоритмы и методы намного сложнее представленных здесь, требуют большей вычислительной мощности и имеют больше правил и условий.

Основная сложность при создании нового квадрата Судoku — убедиться, что головоломка Судoku имеет решение. Отнюдь не стоит рассчитывать, что головоломка решаемая, просто поместив в квадрат Судoku несколько чисел!

Вообще говоря, один из способов решения головоломки Судoku состоит в том, чтобы продолжать попытки решить ее до тех пор, пока вы не зайдете в тупик. Затем вы должны вернуться назад, выбрать другое значение и продолжать, пока не решите головоломку или пока опять не зайдете в тупик. Если вы решите использовать этот подход, вам следует больше узнать о технологии перебора с возвратом.

Наконец, если у вас достаточно вычислительной мощности, можно попробовать все возможные пути и найти те, которые действительно решают данную головоломку Судoku — это метод лобовой атаки. Хотя методы лобовой атаки, как правило, медленные, их легко применять в головоломках Судoku, поскольку используются относительно небольшие квадраты. Тем не менее, применение метода лобовой атаки к квадрату 100×100 не рекомендуется!

и **sudoku_test.go**. Это очень распространенная технология при написании сравнительных функций в Go.

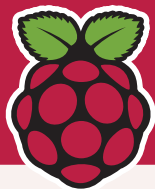
Результаты сравнения **big_test.go** можно увидеть на рис. внизу. Не обращайте внимания на предупреждение “no tests to run [нет тестов для выполнения]” в выводе, поскольку она относится к тестам модулей, которых мы не писали — более новые версии Go этого предупреждения не отображают. Как видите, число элементов в срезе или массиве сильно влияет на производительность программы, которая получает доступ ко всем элементам среза или массива.

Посмотрите на **big.go** — он поможет вам понять всё это более детально. И потом, когда вы уже поймете разные концепции, принимайтесь за решение головоломок Судoku и поупражняйте свой мозг! **LXF**

```
3. mtsouk@mail: ~/docs/article/working/SudokuGo.LXF245/code (ssh)
code$ go test -bench . big.go big_test.go
testing: warning: no tests to run
Benchmark_1002_accessPuzzle      500          2498624 ns/op
Benchmark_102_accessPuzzle       50000         27255 ns/op
Benchmark_9_accessPuzzle         3000000         434 ns/op
PASS
ok      command-line-arguments  5.563s
code$ go test -benchmem -bench . big.go big_test.go
testing: warning: no tests to run
Benchmark_1002_accessPuzzle      500          2505891 ns/op          16465 B/op          2 allocs/op
Benchmark_102_accessPuzzle       50000         27509 ns/op             1 B/op             0 allocs/op
Benchmark_9_accessPuzzle         3000000         434 ns/op              0 B/op              0 allocs/op
PASS
ok      command-line-arguments  5.588s
code$
```

Здесь показаны результаты сравнения из **big.go** и **big_test.go**, чтобы лучше понять, как размер среза или массива влияет на скорость выполнения программы.

» **АРХИВ ЖУРНАЛА LINUX FORMAT** ищите на www.linuxformat.ru!



Лес Паундер работает с организациями типа Raspberry Pi Foundation, помогая развивать навыки умельцев.

» ДУХ ЕДИНЕНИЯ

Не так давно я, как и многие другие, выиграл премию Джона Пиннера [John Pinner] на PyConUK. Эта награда названа в честь создателя PyConUK, который, увы, умер в 2015 г. Джон был образцом для подражания сообщества Python, и бескорыстно посвящал свое время и усилия сообществу. Нам будет очень его не хватать.

В его честь награды Джона Пиннера вручаются тем членам сообщества, которые не получают заслуженного ими признания. Моя награда отражала работу, которую я выполняю в британском сообществе, помогая детям изучать Python и делать свои первые шаги в становлении себя как умельца.

Но награды заставили меня больше задуматься о сообществах. Они являются источником жизненных сил для таких плат, как Raspberry Pi и Arduino. Без сообщества эти устройства — просто куски кремния. Сообщество формируется вокруг продукта и продвигает его вперед, расширяя границы предполагаемого создателем.

Сообществам следует помогать, приветствовать их и вознаграждать за их тяжелый труд, через поддержку и пропаганду организации выбранной ими платформы. Признание нам нужно, чтобы показать сообщество в целом. Сообщество оживляет продукт, иначе срок его службы короток. Мы все когда-то были в начале нашего пути и нуждались в помощи сообщества, чтобы завершить проект или суметь найти работу мечты.

Помощь и уважение друг к другу — это центр каждого сообщества, и именно это мы все должны делать, независимо от платформы или языка.

Включите, настройтесь, Pi-едем

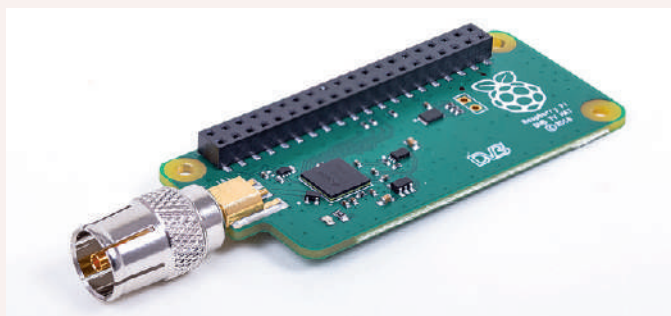
Новейшая официальная приставка Raspberry Pi делает живые ТВ трансляции на Pi. Правда, без пульта...

У Raspberry Pi, развернутого в самых разных областях, одной из возможностей, которой до сих пор очень не хватало, был ТВ-тюнер. Поэтому неудивительно, что последнее официальное расширение от Raspberry Pi Foundation — это именно цифровой тюнер под названием TV HAT.

Сейчас устройство доступно для покупки по всей Европе (регионы потом добавят). Оно стоит всего £20 и основано на телевизионном тюнере Sony CXD2880, который добавляет к вашему Pi

прием DVB-T2. Имейте в виду, что из-за нагрузки на процессор рекомендуется Raspberry Pi 2 или последующие.

TV HAT позволяет просматривать прямые телевизионные трансляции прямо с вашего Pi или в потоковом режиме по локальной сети. Представление TV HAT также дебютирует с новым стандартом половинной ширины HAT, того же размера, что Pi Zero; это чисто механическое изменение. Узнайте больше на www.raspberrypi.org/products/raspberry-pi-tv-hat.



Теперь к возможностям своего Pi можно добавлять потоковое ТВ.

Заядлый вязальщик!

Шерстяное мышление.

Возьмите одного Arduino-одержимого графического дизайнера, Pi и ночное небо, и вы получите взломанную вязальную машину Brother KM950i и обширную вязаную версию небесной сферы. Окончательная версия вязаной карты заняла 100 часов, и это 15 кг чего-то, называемого шерстью. Посмотреть ленту Twitter умельца из Мельбурна можно здесь: <https://twitter.com/HeartOfPluto>.



Вязанный проект размером с галактику.

Робокот 2

Petoi вернулся!

Еще в **LXF236** мы рассказали о том, как Ронгчжонг Ли [Rongzhong Li] разработал свой проект Pi-кошки на основе Paw. В итоге получился успешный проект на Indiegogo, который можно поддержать и купить прямо сейчас! Это полный DIY-набор для кошки, целиком открытый — разве не мило? На www.indiegogo.com/projects/nybble-world-s-cutest-open-source-robotic-kitten# вы найдете дополнительную информацию.



Вернулся с лучшим и более открытым исходным кодом, чем когда-либо!

Raspberry Pi PoE HAT

Лес Паундер полон энергии! И теперь она приходит по Ethernet. Может ли он отключить блок питания микро-USB и жить уже только с Ethernet-кабелем?

ВКРАТЦЕ

НАТ-совместимая плата, обеспечивающая питание по Ethernet для пользователей Pi 3 B+. Устанавливается на четыре новых контакта PoE на Pi 3 B+ и закрывает все выводы GPIO, PoE HAT связывается с Pi через специальный слой конфигурации, которое настраивается пользователем через скрипт. Доступ к GPIO возможен, но требует дополнительных расширений для контактов разъема.

Питание по Ethernet (PoE) не является чем-то новым. На самом деле, ему много лет. Но это проходило мимо Raspberry Pi... до сих пор.

PoE HAT является официальным продуктом Raspberry Pi и предназначен для работы только с новым Pi 3 B+ через контакты PoE, имеющиеся на этой плате. Чтобы использовать PoE HAT, надо прикрепить HAT ко всем 40 контактам GPIO и четырем контактам разъема PoE. Затем мы подаем питание через соединение Ethernet, а не через источник питания микро-USB. Это требует либо PoE-совместимого коммутатора или маршрутизатора, либо PoE-инжектора, который будет передавать 48 В питания и данные по одному кабелю Ethernet. Будьте осторожны, используя это в своей сети, поскольку устройства, не совместимые с PoE, могут быть повреждены.

Питание через эту плату требует от нас установки последней версии Raspbian. Но как только всё это сделано, плата просто работает, и всё, что нам нужно — это один кабель!

В PoE HAT обычно используется 25-миллиметровый 5-вольтовый вентилятор, предназначенный для охлаждения процессора на Raspberry Pi. Когда вентилятор работает на полной скорости, он издает особый звук. Это немного раздражает, но терпеть можно. Однако с PoE HAT есть проблема, и довольно большая.

Она закрывает

Поскольку плата закрывает весь GPIO, мы не можем подключать какие-либо дополнительные платы или непосредственно подключаться к GPIO. Существует обходной путь, который включает 40-контактное расширение разъема и четырехконтактное расширение разъема PoE, но требующиеся компоненты не поставляются. Это позор, потому что умельцы будут использовать PoE HAT для питания проектов, в которых используются датчики. Расположение HAT поверх PoE HAT возможно, но тогда понадобятся латунные стойки, чтобы предотвратить контакт PoE HAT с платой расширения. Хуже того, зазор между вентилятором и платой расширения составляет всего миллиметры и, как правило, плата просто нагревается.

Управление вентилятором осуществляется автоматически по заданному температурному порогу, установленному файлом настройки, который взаимодействует со встроенным чипом Atmel с помощью I2C. При 45°C вентилятор включится и охладит систему. Чем выше температура, тем быстрее будет вращаться вентилятор.

Существует особенность PoE HAT, особенно портов USB. С подключением клавиатур и мышей всё в порядке, но любое



PoE HAT помещается поверх Raspberry Pi, блокируя доступ к GPIO, если не добавить контакты расширения.

устройство, потребляющее более 40 мА, может вызвать отключение системой питания портов USB до тех пор, пока не отключится ток. Эта проблема была решена Эбеном Аптоном [Eben Upton], который сказал, что причина была во «взаимодействии между довольно низкочастотным переключающим регулятором на HAT и одной из двух марок ограничителей тока USB, которые мы используем на главной плате». Кроме того, тестирование сильноточных нагрузок проводилось с помощью токоограничивающего переключателя одной марки, и по несчастливому совпадению были испытаны более низкие нагрузки, испытываемые этим проблемным компонентом. Так что есть вероятность, что мы не сможем подключить внешние USB-устройства, требующие более 40 мА. Это если мы не хотим начать пайку компонентов на PoE HAT.

Хороша ли PoE HAT? Да, если для вашего проекта нужно решение с единственным кабелем. Проблема в том, что PoE HAT ограничивает доступ к GPIO, если мы не добавим дополнительные разъемы. А ведь именно GPIO делает Raspberry Pi таким многофункциональным устройством. Без GPIO это просто маломощный компьютер с Linux. **LXF**

ВЕРДИКТ

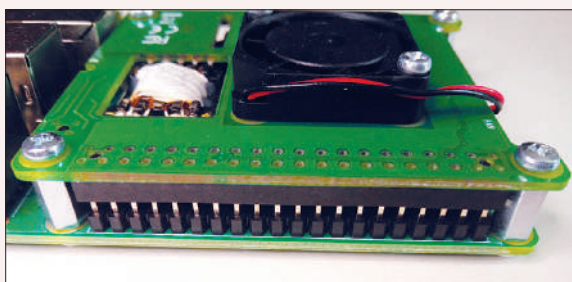
ПРОИЗВОДИТЕЛЬ: Raspberry Pi Trading
САЙТ: www.raspberrypi.org/products/poe-hat
ЦЕНА: £18,35

ФУНКЦИОНАЛЬНОСТЬ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	6/10
УДОБСТВО В РАБОТЕ	8/10
ОПРАВДАНОСТЬ ЦЕНЫ	6/10

Просто в использовании и установке. Но факт блокировки доступа к GPIO означает, что эта плата лишает Pi самого большого преимущества. Это устройство из двух половинок.

» **Рейтинг 7/10**

Перед покупкой спросите себя, сильно ли вы хотите питать свой Pi по кабелю Ethernet.



RASPBERRY Pi

Играем с блоками в стиле Minecraft

Лес Паундер провел выходные, программируя *Minecraft* в классе, полном детей, и вот достижения, которых детям хотелось!



НАШ
ЭКСПЕРТ

Лес Паундер — мастер-фрилансер, который работает с такими организациями, как Raspberry Pi Foundation, продвигая навыки умельцев. Его блог размещен на bigl.es.

Minecraft — потрясающий инструмент обучения. Дети играют — и в то же время учатся создавать блоки из лавы, алмазов и воды. Однако для малышей синтаксис команд сложен. Поэтому для нашего урока в этом выпуске мы создадим простую в использовании серию команд, которую мы дерзко назвали *Minecraft Zero* в честь библиотек *GPIO* и *GUI Zero*, позволяющих детям развлекаться и заодно знакомиться с Python.

На своем Raspberry Pi откройте редактор Python 3, находящийся в меню Programming [Программирование]. Затем нажмите File > New [Файл > Создать] для создания нового пустого документа. Теперь нажмите File > Save [Файл > Сохранить] и назовите файл **mczero.py**. Не забывайте постоянно сохраняться в процессе.

Наш проект состоит из серии функций, блоков кода Python, которые мы можем использовать, вызвав их имя. Первая из них — наиболее простая, и заключается в размещении сообщения в окне чата *Minecraft*. Как и во всех функциях, мы должны задать имя, и мы назовем ее **chat**, но вы увидите в скобках **msg**. Это **msg** — текст, который мы хотим отобразить в окне чата.

```
def chat(msg):
```

Затем внутри функции мы будем использовать библиотеку *Minecraft*, которую позднее импортируем и переименуем в **mc**, а именно — ее функцию **postToChat**. Обратите внимание, что мы также используем вспомогательную функцию **str()**, чтобы обеспечить преобразование всех данных, использованных для сообщений, в строковую переменную, потому что функция **postToChat** работает только со строковыми переменными:

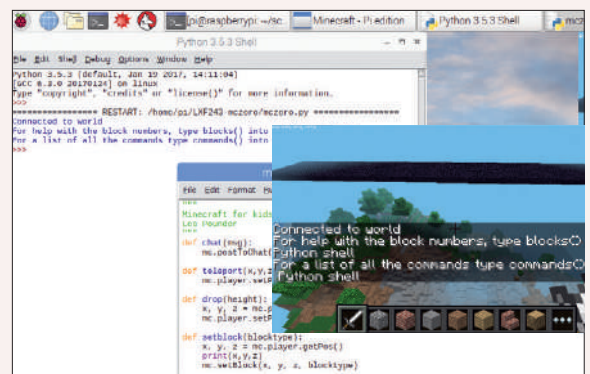
```
mc.postToChat(str(msg))
```

Отправляйтесь в дорогу

Следующая функция называется **teleport** и используется для отправки игрока в любую точку мира. Всё, что ей нужно — это координаты **x**, **y**, **z**, которые находятся в верхней левой части экрана. Функция работает аналогично **chat**, в том смысле, что мы вызываем функцию с параметрами положения — **x**, **y**, **z**. Затем в функции мы задаем положение игрока (**setPos**) в этих координатах:

```
def teleport(x,y,z):
    mc.player.setPos(x, y, z)
```

Дети любят летать в облаках, а потом шлепнуться на землю. Что ж, мы можем сделать это с помощью функции **drop**. Функция принимает параметр высоты **height** (в блоках), с которой падает стив. Затем она используется в функции, когда мы узнаем текущее положение игрока с помощью **getPos** для обновления



Когда впервые используется библиотека *mczero*, она печатает полезные подсказки в оболочке Python и окне чата *Minecraft*.

координат **x**, **y**, **z**. Затем с помощью **setPos** мы изменяем значение **y** так, чтобы оно увеличилось на значение **height**.

```
def drop(height):
    x, y, z = mc.player.getPos()
    mc.player.setPos(x, y+height, z)
```

Следующая важная часть *Minecraft* — изменение блоков. С помощью Python мы можем отредактировать любую часть мира, превратив траву в потоки лавы, алмазы или золото. Для этого нам надо изменить тип блока — **blocktype**, и наша функция **setblock** сделает это, изменив блок, на котором мы стояли, на то, что мы передали в качестве параметра. Если вы не знаете типы блоков (числа), не волнуйтесь, мы расскажем о них позднее в нашей функции **blocks()**.

```
def setblock(blocktype):
    x, y, z = mc.player.getPos()
    mc.setBlock(x, y, z, blocktype)
```

Кубики на заказ!

Наша функция **cube** принимает два параметра: размер куба (в блоках) и тип блоков, из которых он сделан. Поэтому мы можем создавать кубы из алмазов и использовать их, чтобы преградить дорогу врагам! В этой функции мы используем **setBlocks** и даем две позиции **x**, **y**, **z**. Первая — это положение игрока плюс один блок на каждой оси, а вторая — положение игрока плюс размер куба. Затем *Minecraft* заполняет значения между ними, чтобы создать сплошной куб, вот так:

```
def cube(size,blocktype):
```

ВАМ НУЖНЫ

- > Любая модель Pi
- > Самый свежий Raspbian
- > <https://github.com/lesp/LXF243-mczero/archive/master.zip>


```
x, y, z = mc.player.getPos()
mc.setBlocks(x+1, y+1, z+1, x+size, y+size, z+size, blocktype)
```

А если вам нужен большой полый куб? Что ж, наша функция `hollow_cube` сделает это для вас. Она создает сплошной куб из алмазных блоков (тип блока — 57), а затем мы создадим меньший по размеру пустой куб внутри него из блоков воздуха (тип блока — 0):

```
def hollow_cube(size):
    x, y, z = mc.player.getPos()
    mc.setBlocks(x+1, y+1, z+1, x+size, y+size, z+size, 57)
    size = size - 2
    mc.setBlocks(x+2, y+2, z+2, x+size, y+size, z+size, 0)
```

Тик-тик, БУМ!

Новая функция создаст цепь активных блоков TNT [тринитро-толуол], которые отмечают все ваши шаги. Когда вы ударяете по ним мечом, они взрываются! Для этого нам надо выяснить положение игрока, затем задать блок под его ногами как TNT (тип блока — 46), а также передать значение 1, чтобы TNT мог взрываться. Затем мы «спим» 0,1 секунды, чтобы уменьшить количество блоков TNT на экране. Эту функцию лучше всего поместить внутри цикла `while True`, чтобы она работала непрерывно.

```
def tnt_trail():
    x, y, z = mc.player.getPos()
    mc.setBlock(x, y, z, 46, 1)
    sleep(0.1)
```

Наша функция `tnt_cube` создаст куб, набитый TNT! Просто задайте параметр размера, жажните по кубу и бегите! Учтите, что на более старом Pi максимальный размер должен быть $10 \times 10 \times 10$. Для Pi 3 можно рискнуть и выбрать $20 \times 20 \times 20$, но помните, что это способно подвесить Pi:

```
def tnt_cube(size):
    x, y, z = mc.player.getPos()
    tnt = 46
    mc.setBlocks(x+1, y+1, z+1, x+size, y+size, z+size, tnt, 1)
```

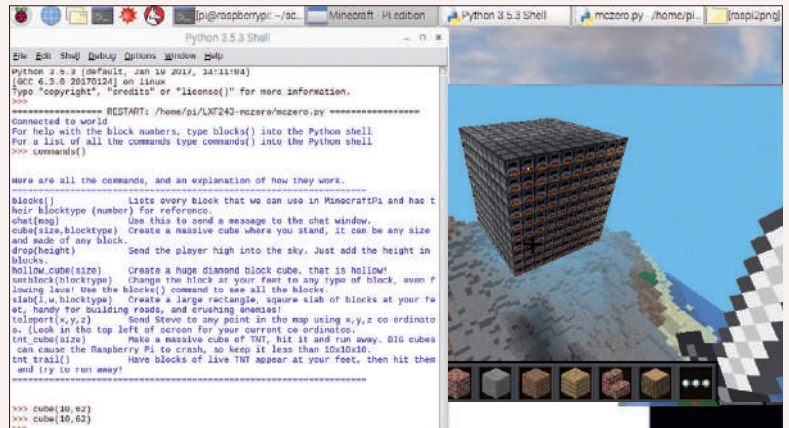
Строить длинные линии из блоков скучно! Но с помощью нашей новой функции `slab` мы можем создавать большие платформы из блоков, задав длину (l), ширину (w) и тип блоков. В целом это то же самое, что и наша функция `cube`, но значение `y` не меняется, так что платформа строится только по двум осям:

```
def slab(l, w, blocktype):
    x, y, z = mc.player.getPos()
    mc.setBlocks(x+1, y+1, z+1, x+w, y+1, z+l, blocktype)
```

Последняя игровая функция называется `fire_feet`, и она помещает у ваших ног горящие факелы (тип блока — 50), куда бы вы ни отправились. Лучше всего использовать ее с циклом `while True`, как функцию `tnt_trail`.

```
def fire_feet():
    x, y, z = mc.player.getPos()
    mc.setBlock(x+1, y, z+1, 50, 1)
    sleep(0.1)
```

Следующие две функции — встроенные инструменты поддержки. Первая называется `blocks()`, и она хранит словарь (под названием `block_dictionary`), тип данных для хранения больших объемов данных, на которые ссылаются через отношение ключ:значение [key:value]. В данном случае ключи — это имена блоков, а значения — номера типов блоков. Поэтому, например, ключ `AIR` [Воздух] возвращает тип блока 0. В словаре имеется множество разных блоков, и чтобы показать их все, мы используем цикл `for`, который итерирует по ним и выводит ключ и значение с паузой в 0,1 секунды между строками. Вы можете найти листинг кода этой функции в загрузке для нашего проекта.



С помощью функции `commands()` мы нашли функцию для создания куба и сделали $10 \times 10 \times 10$ горнов — просто потому, что можем! А для еще большего хаоса попробуйте всё заливать водой!

Последняя наша функция называется `commands()`, она просто распечатывает все команды и их действия. Очень удобно, особенно когда вы находитесь в самом разгаре битвы! Листинг кода этой функции вы тоже можете найти в загрузке для данного проекта.

Для выполнения всего кода мы используем обработчик исключений, который попытается запустить код. В этом случае он импортирует библиотеку *Minecraft*, библиотеку блоков и функцию `sleep` из библиотеки *time*. Затем он создает соединение с запущенной игрой *Minecraft*. Если это работает, то текст отображается в оболочке Python и в окне чата *Minecraft*, информируя игрока о функциях справки (`blocks()` и `commands()`), но если соединение не удается, возникает исключение и в оболочку выводится ошибка.

Сохраните код, запустите игру *Minecraft*, и когда попадете в игровой мир, нажмите Run > Run Module [Запустить > Запустить Модуль] в редакторе Python. Теперь испытайте команды и готовьтесь учинять хаос и свистопляску! **LXF**

СКОРАЯ ПОМОЩЬ

В оболочке Python введите `commands()` и нажмите Enter, чтобы увидеть все команды. Это работает и для `blocks()`, показывая все блоки, создаваемые с ее помощью, и соответствующий им тип (номер).

» ДОБАВЛЕНИЕ КОДА В ПРОЕКТ

Наш код `mczero.py` можно использовать в оболочке, чтобы интерактивно помериться силами с другими игроками. Но его можно употребить и как часть проекта так же, как мы импортировали другие библиотеки кода, например, когда мы импортировали библиотеку *Minecraft* (`mcp`) в наш код `mczero.py`. Для этого надо обеспечить, чтобы весь созданный нами код хранился в той же директории, что и файл `mczero.py`. Сначала импортируем библиотеку `mczero`:

```
import mczero
```

Теперь мы можем использовать любую из функций в нашем коде, вызывая их по именам, перед которыми указано имя библиотеки, в данном случае — `mczero`. Например, давайте выведем сообщение в окне чата:

```
mczero.chat("Hello Linux Format")
```

Затем давайте создадим цикл для нашей функции `fire_feet`:

```
while True:
    fire_feet()
```

Любой код, который должен работать непрерывно, может быть помещен в цикл, поэтому наши функции `tnt_trail()` и `fire_feet()` можно выполнять вместе. Однако большие функции, которые создают кубы, платформы и т.д., в идеале следует запускать один раз или по требованию, используя условный оператор (`if..elif..else`). Например, ту, которая проверяет тип блока, на котором стоит игрок.

OMXPLAYER

Смотрите видео по случаю

Лес Паундер обожает мультики, и теперь может выбрать случайное видео по запросу! Осталось припасти гигантское ведро попкорна!



**НАШ
ЭКСПЕРТ**

Лес Паундер — независимый разработчик, он сотрудничает с такими организациями, как Raspberry Pi Foundation, развивая мастерство умельцев. Читайте его блог на bigl.es.

Н едоумеваете, что смотреть этим вечером с семьей? Не страдайте — наш проект поможет сделать так, чтобы наш Raspberry Pi произвольно выбирал видео для просмотра. Итак, давайте начнем создавать и кодировать собственный видеоплеер для произвольного просмотра! Весь код и подробную схему в высоком разрешении можно скачать с <https://github.com/lesp/LXF244-VideoPlayer/archive/master.zip>.

Сборка

Создать схему для этого проекта просто. Нам нужны только три кнопки, подключенные к контактам GPIO 2, 3 и 4, и каждая из этих кнопок также должна быть подключена к GND [Земля]. Для этого мы можем использовать один вывод GND на Pi и перемычку мама–папа. Подключение к шине макетной платы означает, что у нас есть несколько соединений с GND, которые могут быть подключены двумя перемычками папа–папа. Подробности смотрите в схеме, включенной в загрузку для этого проекта.

Собрав «железо», подключите аксессуары, необходимые для вашего Pi, и загрузитесь в рабочий стол Raspbian.

Настройка

Прежде чем мы сможем написать какой-либо код Python, надо установить две библиотеки, которые помогут нам создать проект. Откройте терминал и введите следующее:

```
$ sudo pip3 install glob
$ sudo pip3 install keyboard
```

Мы обсудим их позже. А пока откройте редактор Python 3, который находится в меню Programming [Программирование], и выберите File > New [Файл > Создать], чтобы создать новый пустой файл. Сразу же нажмите File > Save [Файл > Сохранить] в новом окне. Сохраните код как **VideoPlayer.py** и не забывайте сохранять его почаще!

Итак, теперь мы начинаем писать код, и наш первый шаг — импортировать нужные библиотеки. Первые три библиотеки используются для обнаружения нажатий кнопок (*GPIO Zero*), останки однократного выполнения кода и выхода (*pause*), а также для выбора случайного видео (*choice*).

```
from gpiozero import Button
from signal import pause
from random import choice
```

Последние три импорта — это библиотека, которую мы используем для отображения содержимого каталога (*glob*), запуска команд терминала (*subprocess*), а последняя — библиотека для эмуляции клавиатуры (*keyboard*).

```
import glob
```



Воспроизвести видео на нашем Raspberry Pi очень просто, используя этот скрипт и команду **omxplayer**, которая плавно проигрывает видео 1080p.

```
import subprocess
import keyboard
```

Далее мы создаем три функции — блоки кода, которые мы можем потом использовать, вызывая их имена. Первая функция называется **play_video**, и сначала она создает список (объект хранения данных) с именем **videos**.

```
def play_video():
    videos = []
```

Чтобы заполнить список всеми видео, которые мы можем воспроизвести, мы используем цикл **for**. Он переберет все попавшиеся в каталоге видеофайлы MP4. Мы используем **glob** для доступа к каталогу **/media/pi/Videos**, который на самом деле является USB-накопителем с именем **Videos**, заполненным файлами MP4. Измените это в соответствии с именем выбранного вами каталога, заполненного видеофайлами. Каждый раз, когда обнаруживается файл MP4, он добавляется в список **videos**, который мы только что создали.

```
or file in glob.glob("/media/pi/Videos/*.mp4"):
    videos.append(file)
```

Убедитесь, что список заполнен именами файлов, выведя содержимое списка в оболочке Python:

```
print(videos)
```

Выберите случайное видео из списка, сохраните его в переменной с именем **chosen** и выведите его в оболочке Python:

```
chosen = choice(videos)
```

ВАМ НУЖНЫ

- > Любая модель Raspberry Pi
- > Последняя версия Raspbian
- > Три кнопки
- > Макетная плата
- > Четыре перемычки мама–папа
- > Две перемычки папа–папа

```
print(chosen)
```

Последняя строка этой функции использует класс `Popen` из библиотеки `subprocess` для запуска команды, как если бы мы были в терминале, и следующая команда должна открыть медиа-плеер `omxplayer`, а затем воспроизвести выбранное видео: `subprocess.Popen(['omxplayer', (chosen)])`

Очередная функция называется `stop_video` и, как вы можете догадаться, она остановит воспроизводимое в данный момент видео. Для этого мы используем библиотеку `keyboard`, в частности функцию `press_and_release` для имитации нажатия Q.

```
def stop_video():
    keyboard.press_and_release('q')
```

Последняя функция называется `pause_video`, и она эмулирует нажатие клавиши пробела на клавиатуре, по которому `omxplayer` приостанавливает видео:

```
def pause_video():
    keyboard.press_and_release('space')
```

После создания функций нам нужно сообщить нашему коду, куда подключены наши кнопки. У нас подключены три кнопки: `randomiser` (воспроизведение), `stop` и `pause_button`. Эти кнопки подключены к контактам GPIO 2, 3 и 4, соответственно.

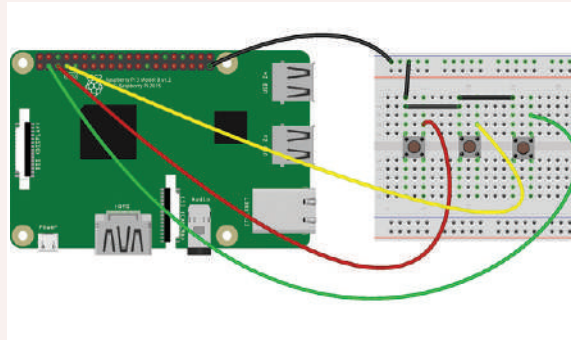
```
randomiser = Button(2)
stop = Button(3)
pause_button = Button(4)
```

Нажмем кнопку

Перейдем к последней части, которая ищет видео при нажатии кнопки и реагирует соответствующим образом. Но сначала мы обернем этот раздел в обработчик исключений. Он попытается запустить код, но если пользователь нажмет Ctrl+C, код завершится. Итак, для раздела `try` наш код сначала выведет три строки в оболочку — это инструкции для пользователя. И между всеми цветами — это сокращение Python для вставки новой строки между инструкциями:

```
try:
    print("Для старта нажмите ЗЕЛЕНУЮ кнопку\nдля паузы\nЖЕЛТУЮ\nдля остановки КРАСНУЮ")
```

Наши три кнопки ожидают нажатия, и с помощью библиотеки `GPIO Zero` мы вызываем класс `when_pressed`, чтобы определить, когда нажата каждая из кнопок. Когда это происходит, запускается соответствующая функция. И когда мы нажимаем зеленую (воспроизведение) кнопку `randomiser`, она случайным



образом выбирает видео с USB-накопителя и воспроизводит его. Вы заметите, что функции не имеют `()` в конце имени функции. Это потому, что в противном случае код будет выполняться автоматически. А сейчас они готовы работать по требованию.

```
randomiser.when_pressed = play_video
stop.when_pressed = stop_video
pause_button.when_pressed = pause_video
```

Последняя строка в этом разделе — это обычная `pause()`, используемая для поддержания работы кода, а не просто для выхода после однократного запуска.

```
pause()
```

А что произойдет, если пользователь нажмет Ctrl+C? Ну, это Keyboard Interrupt [прерывание с клавиатуры], и если это произойдет, мы просто выведем на экран пустую строку, а затем `Exit`.

```
except KeyboardInterrupt:
    print("\nEXIT")
```

Теперь сохраните свой код и откройте терминал в каталоге, где вы сохранили код. Чтобы использовать код, надо запустить его через `sudo` (с правами root), потому что к библиотеке `keyboard`, используемой в проекте, допускается только root.

Для запуска кода наберите

```
$ sudo python3 VideoPlayer.py
```

Все вместе усевшись рядышком на диван, нажмите зеленую кнопку, чтобы воспроизвести случайно выбранное видео; для паузы нажмите желтую, а для остановки нажмите красную. Приятного вечера... да поаккуратнее, не сорите попкорном! **LXF**

Сборка проекта проста; выполняйте ее пошагово, следя за соединениями.

СКОРАЯ ПОМОЩЬ

Вы можете встроить проект в прочную коробку, чтобы шkodливые ручки не поломали электронику. Можно также взять кнопки в стиле аркады, которые гораздо надежнее и ломаются с меньшей вероятностью.

» МНОГОЛИКОСТЬ KEYBOARD

В этом проекте мы использовали для эмуляции клавиатуры `Keyboard`, библиотеку Python, используя кнопки только для управления воспроизведением, приостановкой и остановкой воспроизведения видео. Библиотеку `Keyboard` можно использовать для нажатия любых клавиш, поэтому мы можем автоматизировать одновременное нажатие нескольких клавиш одной кнопкой. Библиотека также пригодится для вывода текста на экран.

Но самой опасной функцией этой библиотеки является то, что она может использоваться для записи всех нажатий клавиши на целевом компьютере. Эти нажатия клавиш могут быть записаны в список, затем сохранены в файл и использованы для гнусных целей. Конечно, запись нажатий клавиш пользователем без его согласия является незаконной и может доставить вам массу неприятностей. Так что не делайте этого! Однако `Keyboard` можно использовать как мощный инструмент при отладке взаимодействия пользователя с вашим кодом, поэтому при тестировании ПО (по согласованию с широкой общественностью и домашними тестировщиками) вы можете увидеть, какие клавиши они нажимали непосредственно перед блокировкой кода и/или почему он стал термоядерным!

Библиотека `Keyboard` работает с Linux и Windows, и с ней очень весело играть. Чтобы прочитать немного больше об этой библиотеке, перейдите к посту в блоге <https://bigl.es/tuesday-tooling-record-replay-keystrokes-with-python> и посмотрите несколько примеров ее применения в повседневной жизни.

```
File Edit Format Run Options Window Help
from gpiozero import Button
from signal import pause
from random import choice
import glob
import subprocess
import keyboard

def play_video():
    videos = []
    for file in glob.glob("/media/pi/Videos/*.mp4"):
        videos.append(file)
    print(videos)
    chosen = choice(videos)
    print(chosen)
    subprocess.Popen(['omxplayer', (chosen)])

def stop_video():
    keyboard.press_and_release('q')

def pause_video():
    keyboard.press_and_release('space')

|
randomiser = Button(2)
stop = Button(3)
pause_button = Button(4)
try:
    print("Press the GREEN button to start\nYELLOW to pause\nRED to stop")
    randomiser.when_pressed = play_video
    stop.when_pressed = stop_video
    pause_button.when_pressed = pause_video
    pause()
except KeyboardInterrupt:
    print("\nEXIT")
```

В этом проекте только 34 строки кода, однако наш простой медиа-плеер способен воспроизводить видео 1080p.

НАВЫКИ УМЕЛЬЦЕВ

Соберите свою электронную плату

Хотите собрать приставки к Raspberry Pi или Arduino, но не знаете, с какого конца браться за паяльник? **Майк Бедфорд** дает совет, который вы ищете.



НАШ ЭКСПЕРТ

Интерес Майка Бедфорда к электронике предшествовал его увлечению компьютерами. Теперь эти два интереса работают рука об руку и дали некоторые удовлетворительные результаты.

СКОРАЯ ПОМОЩЬ

Вам, хотя это и не обязательно, может пригодиться покупка дешевого тестера. Он будет полезен, если и когда вы начнете собирать что-то по схемам, но даже сейчас он пригодится для измерения напряжений батареи и значений резисторов, а также для проверки на обрыв или короткое замыкание.

Когда появились первые персональные компьютеры, многие пользователи умели собирать электронные схемы. Некоторые из ранних моделей даже продавались как комплекты для самостоятельной сборки.

Излишне говорить, что времена изменились, и очень скоро большинство пользователей ПК перестали интересоваться внутренней работой своих компьютеров. Теперь, благодаря популярности одноплатных компьютеров (SBC — Single-Board Computer), таких как Raspberry Pi и Arduino, спираль сделала виток.

Технически мыслящие пользователи компьютеров вновь возымели потребность вникать в электронику, но, в отличие от компьютерных энтузиастов 1980-х, соответствующий опыт в электронике у них отсутствует. Цель данной статьи — сделать первый шаг в предоставлении этих знаний сегодняшнему поколению пользователей SBC.

Это нелегко охватить в одной статье, поэтому мы начинаем с основ и с того, как собрать печатную плату. Не все HAT [Hardware Attached on Top, плата-приставка] для Pi и модули для Arduino доступны в собранном виде, готовыми к работе. Как и старый Sinclair ZX80 1980-х гг., некоторые из этих модулей SBC поставляются только в виде комплектов деталей, включающих печатную плату (PCB — Printed Circuit Board) и сопутствующие ей компоненты, которые вы должны собрать самостоятельно.

Некоторые конструкции даже не настолько продвинутые. Если вы заглянете на сайт инициативы по разработке открытого исходного кода Fritzing (<http://fritzing.org>), вы найдете множество плат, которые пользователи внесли в сообщество. Для этих конструкций вам придется сначала купить все компоненты, прежде чем собирать плату.

Мы собираемся помочь вам справиться с обоими этими подходами: предоставим пошаговые инструкции по сборке печатной платы, обсудим, как закупить подходящие детали для проекта, и расскажем, как выглядит каждый тип детали, чтобы вы могли распознать их, получив пакет с деталями по почте.

Сборка платы

Вместо использования Pi HAT или модуля для Arduino, в качестве примера мы создадим собственное решение. Тогда вам нечего будет бояться, что из-за недостатка опыта вы спалите свой Pi или Arduino.

Чтобы удовлетворить растущий интерес к электронике для хобби, в настоящее время несколько компаний предоставляют комплекты деталей. Мы выбрали один такой комплект от www.kitronik.co.uk. Он классифицируется как средний по сложности, но не пугайтесь: некоторые из «простых» наборов уж вовсе

тривиально просты. Данный набор представляет собой сигнализацию с микропереключателем и стоит всего £6,14. Всё, что вам нужно сверх того — это четыре батарейки AA.

В случае набора Kitronik полная информация о конструкции доступна онлайн, но поскольку мы заботимся и о тех, кому может потребоваться собрать схему на другой плате, предоставляемой без комплекта деталей, мы предоставляем иллюстрированные пошаговые инструкции. О пайке в пошаговом руководстве упоминается лишь мельком, но поскольку это один из самых сложных навыков при монтаже печатной платы, обязательно прочитайте следующие подробные инструкции о выполнении этого важного шага.

Учимся паять

Для начала включите паяльник и подождите несколько минут, чтобы он нагрелся. Пока вы ждете, намочите губку в основании подставки для паяльника. Паяльники работают правильно, только если жало чистое и блестящее. Поэтому несколько раз проведите жалом по влажной губке, чтобы удалить старый потускневший припой или обычную грязь, если вы используете его впервые. Теперь приложите припой к жалу паяльника и удалите его избыток, проведя по губке. Надеемся, что это приведет к лужению жала (покрытию тонким слоем блестящего нового припоя). Возможно, потребуется несколько раз выполнить этап нанесения припоя, чтобы получить правильно луженое жало. Кроме того, обратите внимание, что вам также, возможно, придется снова лудить жало, если есть большие перерывы между пайкой, пока паяльник включен.

После лужения жала паяльника вы готовы сделать свою первую пайку. Печатная плата должна располагаться горизонтально деталями вниз; приложите паяльник к плате так, чтобы он касался и вывода детали, и площадки. Это круглая или прямоугольная область вокруг отверстия для вывода детали. Теперь приложите конец припоя к противоположной стороне площадки. В течение секунды или двух конец припоя должен начать плавиться. Как только это произойдет, отнимите припой и паяльник, чтобы припой затвердел.

Проверьте, что вы не злоупотребили количеством припоя, из-за чего соседние контактные площадки могут слиться в одну. Если так, то надо удалить припой с помощью отсоса для припоя и попробовать снова. Или вы могли недобрать припоя, в результате чего вывод детали не полностью прикрепился к площадке. В этом случае припой можно чуть-чуть добавить. Убедитесь, что спаянное соединение блестит. Если оно тусклое, то соединение не пропаяно, что способно ухудшить электрический контакт,

поэтому удалите его с помощью отсоса для припоя и попробуйте снова. Учтите, что слишком долгое удерживание паяльника в контакте с площадкой является распространенной причиной плохого соединения, поэтому старайтесь делать пайку как можно быстрее. И помните, что некоторые компоненты, особенно интегральные схемы, могут быть повреждены из-за перегрева.

Перечислим детали

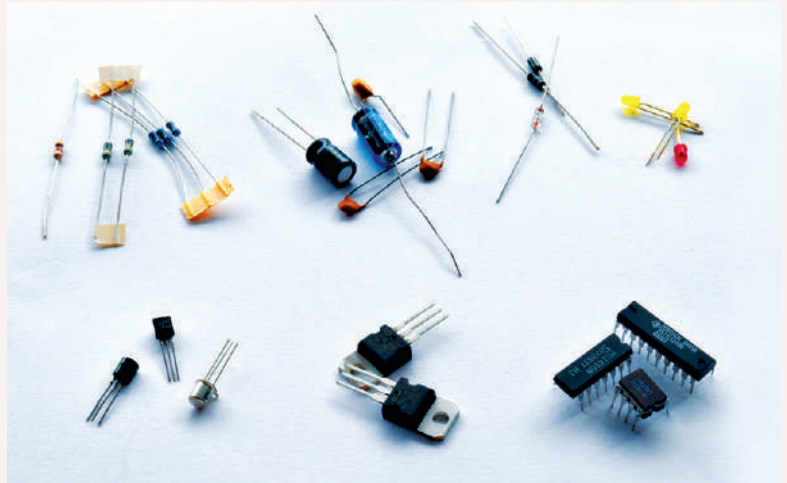
Заказывая набор по почте, важно уметь распознавать детали. Часто вам помогут инструкции из набора, но вы действительно должны знать, как выглядят различные типы деталей. Помимо чтения следующих описаний, обязательно ознакомьтесь с фотографиями различных деталей.

Резисторы определить легко. У них небольшой цилиндрический корпус, возможно, 6 мм длиной и 2 мм в диаметре, и выводы с обоих концов корпуса. Они имеют несколько цветных полос, обозначающих величину их сопротивления — см. врезку внизу о том, как «читать» резисторы.

Конденсаторы распознать не так просто, поскольку у них очень много видов, но фотография справа вверху поможет вам определить наиболее распространенные варианты. Вообще говоря, если нечто имеет два вывода и не является резистором, диодом или светодиодом [Light-Emitting Diode, LED], то это, вероятно, конденсатор. В отличие от резисторов, значение будет напечатано на нем в виде текста. На поляризованных электролитических или танталовых конденсаторах большой емкости ее значение наносится полностью, например, 47 мкФ или 47μ, хотя может быть только 47, в этом случае мкФ предполагается. На конденсаторах меньшей емкости просто наносят значение с префиксом, например .47μ, но, скорее всего, это будут три цифры, часто с буквой после. В этом случае первые две цифры являются значащими цифрами, а третья — множитель, т.е. число нулей, а значение в пФ. Так, например, 103 [10×10³] означает 10000 пФ = 10 нФ.

Диоды имеют небольшие цилиндрические корпуса с выводами на каждом конце корпуса. В отличие от резисторов, у них нет цветных полос. Обычно у них черные или прозрачные корпуса с одной полосой, обозначающей катод — отрицательный вывод. Обозначение детали напечатано на корпусе, например, 1N4148.

Светодиоды обычно имеют небольшие прозрачные куполообразные корпуса и бывают разных цветов — цвет



Начиная сверху справа и до внизу слева мы видим резисторы, конденсаторы, диоды, светодиоды, транзисторы, регуляторы напряжения и интегральные схемы (ИС).

должен указывать на окраску их свечения. Оба вывода выходят из плоского конца корпуса. Плоский конец корпуса срезан с одной стороны, обозначая катод.

Транзисторы, за исключением особо мощных устройств, имеют черные или металлические цилиндрические корпуса, иногда сплюснутые по одной стороне; а порой — черные прямоугольные корпуса. Обозначение детали напечатано на корпусе, например, BC548. У них три вывода. Назначение трех выводов различается для разных транзисторов и может быть определено только по таблице характеристик, хотя дизайн печатной платы обычно иллюстрирует ориентацию устройства.

Регуляторы напряжения похожи на транзисторы, хотя наиболее распространенные типы имеют черные пластиковые прямоугольные корпуса. Их обозначение будет напечатано на них. Как и для транзисторов, вам следует обратиться к описанию, чтобы определить выводы, но при хорошем дизайне печатная плата покажет ориентацию.

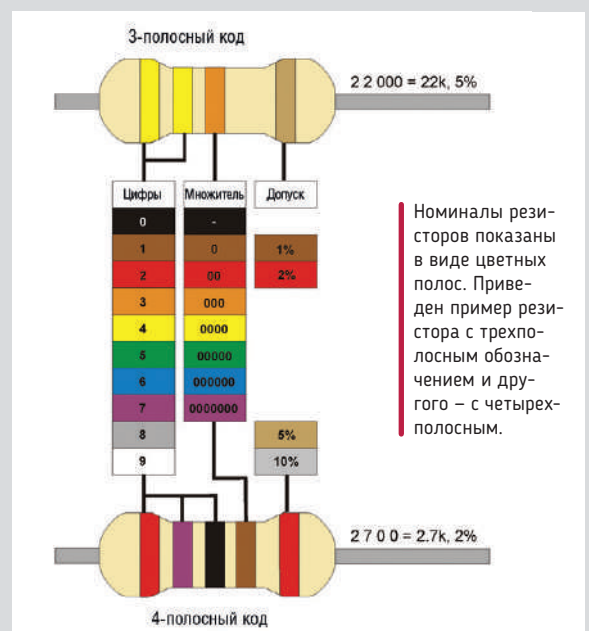
Интегральные схемы (ИС) обычно представляют собой черные пластиковые прямоугольники с выводами по двум краям. Устройства, как правило, имеют восемь, 14 или 16 выводов.

» ЧТЕНИЕ ЗНАЧЕНИЙ РЕЗИСТОРОВ

В отличие от большинства деталей, номинал резистора обозначается цветными полосками. На самом деле, способов обозначений вам может встретиться два: первоначальный цветовой код, имеющий три полоски, и более новый, с четырьмя полосками. В обоих случаях к этим трем или четырем добавляются еще полоски, определяющие допуск и, возможно, температурный коэффициент. Обычно эти дополнительные полоски можно игнорировать, но важно уметь прочитать значение в омах.

Если используются три полоски значения, то первые две представляют первые две цифры, а третья является множителем, т.е. числом приписываемых нулей. Если используются четыре полоски значения, то первые три представляют цифры (третьей цифрой обычно будет ноль), а четвертая — множитель. Если исключить цвета, используемые для допусков, то это черный (0), коричневый (1), красный (2), оранжевый (3), желтый (4), зеленый (5), синий (6), фиолетовый (7), серый (8) и белый (9).

Два примера на рисунке справа помогут вам лучше понять схему, и вскоре это станет второй натурой. Единственная потенциальная загвоздка заключается в том, что иногда бывает трудно определить цвета, поскольку, например, коричневый и фиолетовый могут выглядеть одинаково, как и другие пары цветов. Если сомневаетесь, можете измерить значение с помощью тестера.



Номиналы резисторов показаны в виде цветных полос. Приведен пример резистора с трехполосным обозначением и другого — с четырехполосным.



СКОРАЯ ПОМОЩЬ

Печатные платы представляют собой изолирующие пластины с токопроводящими полосами на одной или обеих сторонах, которые при подсоединении деталей образуют цепь. Выводы деталей проходят через отверстия в печатной плате и припаиваются к контактным площадкам с другой стороны. Сложные платы имеют также слои с дорожками внутри платы, и могут иметь детали на обеих сторонах.

Тип устройства будет напечатан на корпусе. На одном из коротких краев обычно есть выемка. Она указывает расположение вывода 1, который находится слева от нее. Узнав, как выглядит ИС, вы сможете определить связанные с ней сокеты DIL, если они используются.

Покупка деталей

Проще всего, если имеются полные сведения о каждой детали и, возможно, даже обозначения деталей у конкретного производителя. Часто, однако, будут доступны только неполные данные, и в этом случае вам надо будет сделать некоторые предположения. Также хорошо бы сначала разжиться печатной платой, чтобы можно было выполнить любые необходимые измерения.

Мы будем рассматривать только «сквозные» компоненты, то есть такой тип, у которых выводы проходят сквозь отверстия в плате. Более новые детали для поверхностного монтажа (SMD — Surface Mounting Device), обычные на профессионально изготовленных печатных платах, очень мелкие и канительные, и требуют специальных методов монтажа. Поэтому деталей для поверхностного монтажа на первых порах избегайте.

С резисторами обычно всё просто, даже при ограниченных знаниях. По крайней мере, у вас будет значение — в омах, килоомах или мегаомах, которые будут сокращены до Ω или R, k Ω или просто k, или M Ω или просто M соответственно. Иногда аббревиатура заменяет десятичную точку, поэтому 2,2 k Ω можно записать как 2k2. Если это единственная информация, которая у вас есть, можно с уверенностью предположить, что номинальная мощность и допуск не имеют значения, поэтому выбирайте небольшой и дешевый резистор, который, вероятно, будет иметь мощность 250 мВт и имеет допуск в один процент.

Емкость измеряется в фарадах, хотя эта единица слишком велика для реальных конденсаторов. Это будут пикофарады, нанофарады или микрофарады, сокращенно — пФ или просто p, нФ или просто n, или мкФ или просто μ , соответственно. Опять же, сокращение может заменить десятичную точку, поэтому 4,7 нФ

может выглядеть как 4n7. Если это единственная информация, которая у вас есть, можно с уверенностью предположить, что номинальное напряжение и тип диэлектрика (керамика, полиэстер и т.д.) не имеют значения, поэтому выберите низкое напряжение и любой диэлектрик. Конденсаторы большой емкости, 1 мкФ или более, обычно будут поляризованными и, следовательно, электролитическими или танталовыми. Метод изготовления и размер имеют решающее значение, поэтому перед выбором взгляните на печатную плату, измерьте расстояние между выводами для конденсатора и посмотрите, что подойдет. Конденсаторы могут иметь либо осевые выводы, т.е. выводы с разных концов корпуса конденсатора, либо радиальные выводы, т.е. выводы с одной стороны. В случае осевых выводов проверьте длину корпуса, а в случае радиальных выводов проверьте расстояние между выводами.

Большинство других деталей — сюда мы включаем диоды, транзисторы, светодиоды и интегральные схемы (ИС) — будут надлежащим образом определены, поэтому не должно быть никакой двусмысленности. Однако обратите внимание, что полное обозначение микросхем, которые вы видите в продаже, часто намного длиннее обозначения, данного разработчиком, и может отличаться у разных производителей. Так, если разработчик указывает 74HC138, полным обозначением микросхемы может быть нечто вроде SN74HC138N. О микросхемах следует помнить еще одно: хотя вы можете припаять их прямо к плате, но из предосторожности — особенно когда вы только начинаете — лучше вставлять их в припаянные к плате разъемы. Тогда вы вряд ли повредите микросхему перегревом во время пайки, а если она выйдет из строя, ее можно заменить без пайки. Тип разъемов, используемых для большинства микросхем, называется DIL (Dual In Line) и определяется числом контактов, поэтому соответствующий разъем для 16-выводной 74HC138 будет указан как 16-контактный разъем DIL.

А теперь переходите на следующую страницу — стр. 97, к нашему пошаговому руководству! **LXF**

» ПОЛЕЗНЫЕ ИНСТРУМЕНТЫ

Для сборки электроники вам понадобятся инструменты. Покажем, какие покупки вы должны рассмотреть. Вы вряд ли найдете всё это на главной улице, особенно после кончины Marlin, хотя онлайн-магазинов много.

Для всех видов электронных конструкций, кроме макетных плат, необходим паяльник. Естественно, вам нужен паяльник для электроники, а не тот, что для сантехники, но вам не стоит платить больше, чем около £15, если вы собираетесь использовать его только изредка.

Припой, хотя это и не инструмент как таковой, вам тоже понадобится. Избегайте припоя сантехников в пользу тонкой (приблизительно 1 мм) трубки припоя с наполнением из флюса, применяемого в электронике.

Вам понадобятся плоскогубцы и кусачки, но не используйте универсальные, которые продаются в магазинах «сделай сам», — они слишком велики. Вместо этого возьмите пару узких плоскогубцев и пару маленьких кусачек для электроники. £10 должно хватить на оба этих инструмента.

Пара небольших отверток — одна с плоским лезвием, другая с крестообразной головкой — удобны для использования с винтовыми клеммами и винтами, крепящими крышки к коробкам, если вы хотите разместить свой проект в корпусе. Они легко доступны всего за пару фунтов.

С помощью кусачек можно осторожно снять изоляцию с концов проводов, чтобы их можно было припаять на печатную плату. Однако более простым вариантом является использование специального устройства



для зачистки проводов, которых существует несколько типов, значительно различающихся по цене.

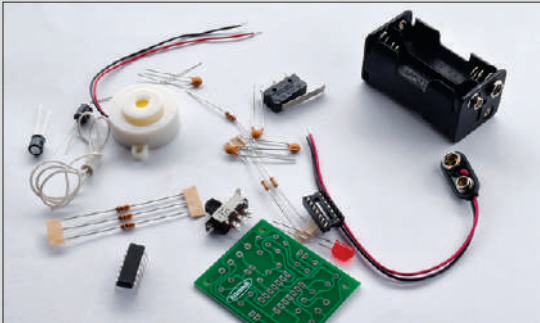
Если всё идет хорошо, то это вам не понадобится, но хорошая мера предосторожности — иметь средства для удаления деталей с печатных плат. Приспособление для удаления припоя, называемое отсосом для припоя, это подпружиненный инструмент, отсасывающий с платы расплавленный паяльник припой.



» МОНТАЖ ДЕТАЛЕЙ НА ПЕЧАТНОЙ ПЛАТЕ

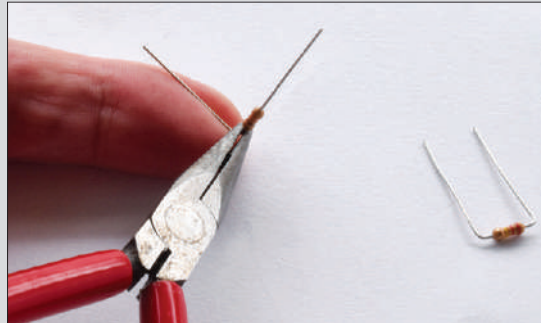
СКОРАЯ ПОМОЩЬ

Помимо деталей, вам понадобятся изолированные провода для подключения внешних компонентов, таких как переключатели в пошаговом руководстве. Многожильный провод лучше одножильного, потому что он более гибкий и менее подвержен обрыву. Однако его труднее протолкнуть через отверстие в печатной плате, поэтому сначала залудите зачищенный конец припоем. 7/0,2 мм – хороший выбор.



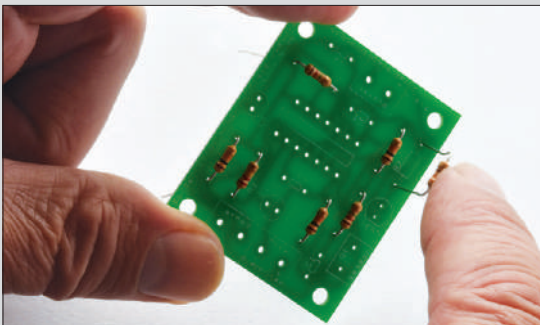
1 Определите детали

Первая задача – убедиться, что у вас налицо все детали и вы распознали каждую из них. Если вы не можете похвастаться знакомством с электронными деталями и не очень хорошо умеете читать цветные коды резисторов, то было бы неплохо разложить детали на листе бумаги и написать рядом номер каждой детали (например, R1, R2, C5, U1 и т.д., как указано на печатной плате).



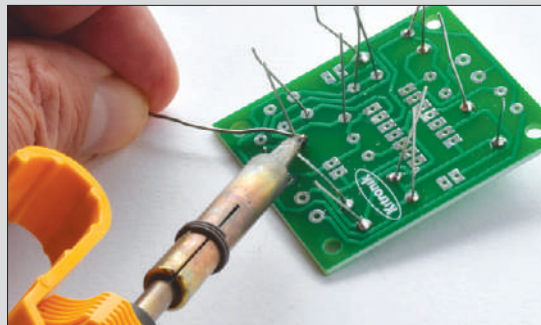
2 Согните выводы деталей

Подготовьте детали с осевыми выводами, например, резисторы, согнув их выводы под прямым углом так, чтобы они входили в отверстия платы. Согните выводы так, чтобы расстояние между кончиками выводов совпадало с расстоянием между отверстиями в плате. Удерживайте вывод между корпусом детали и местом сгиба небольшими остроконечными плоскогубцами, чтобы не расшатать соединение вывода с корпусом.



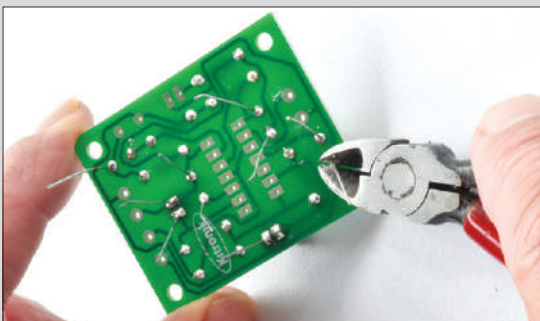
3 Установите детали на печатной плате

Установите детали на плату, пропустив выводы через соответствующие отверстия. Затем плоскогубцами слегка согните выводы, чтобы они не выпали при переворачивании платы. Или, прежде чем перевернуть печатную плату, положите поверх всех деталей кусок поролона, чтобы удержать их на месте. Поскольку выводы уже припаянных компонентов могут препятствовать пайке новых, рекомендуется устанавливать по несколько штук за раз, повторяя шаги 3, 4 и 5.



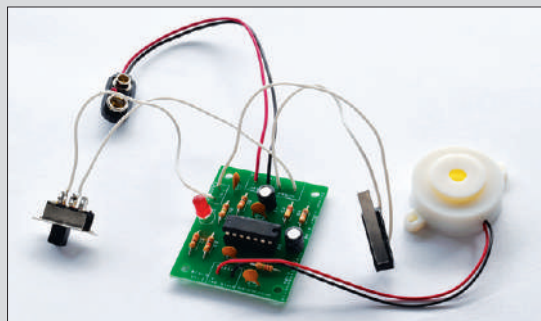
4 Припаяйте детали

Переверните печатную плату и положите ее на плоскую поверхность выводами деталей вверх. Припаяйте каждый вывод деталей к своим контактным площадкам, следуя инструкциям по пайке в нашей статье. Детали менее подвержены тепловому повреждению, чем было раньше, но поскольку дорожки на плате могут передавать тепло, рекомендуется сначала припаять резисторы и конденсаторы, оставив более чувствительные детали напоследок.



5 Обрежьте выводы деталей

Выводы деталей почти всегда длиннее, чем необходимо, поэтому последняя задача – обрезать лишнюю длину. А именно, вы должны обрезать все выводы чуть выше паянного соединения. Это делается небольшими кусачками. Имейте в виду, что излишки выводов при отрезании иногда отлетают на некоторое расстояние, и к тому же они острые. Поэтому во время выполнения этой работы рекомендуется использовать защитные очки и позаботиться о подходящей обстановке, вдали от маленьких детей и домашних животных.



6 Проверьте свою работу

Прежде чем испытать схему, проверьте свою работу визуально (зоркие глаза могут, например, заметить, что изначально неправильно подсоединены переключатели). Визуально контролируйте, что вся пайка выполнена качественно и нет случайных соединений двух соседних выводов. Устраните все замеченные неисправности. Затем испытайте, и если это не работает, немедленно отключите питание и вернитесь к визуальной проверке. В этом случае также убедитесь, что все детали находятся в должном месте и расположены правильно.

Ответы

Вас жгут вопросы по открытому коду или ядру? Каким бы ни был ваш уровень, присылайте их на lxformat@futurenet.com



Джонни Бидвелл

Пытается уладить ваши разборки с Туксом.

В Не покрутить

Многие из нас годами росли в стандартной пользовательской парадигме: полосы прокрутки (scroll bars) со связанными с ними стрелками вверх/вниз. Нажимайте на стрелку вверх/вниз, чтобы переместиться вверх/вниз на одну строку. Нажмите на желоб над или под ползунком, чтобы переместиться на высоту страницы/экрана. Это применялось в большинстве операционных систем, включая Windows и множество систем Linux, которые я использовал: Mandrake, SUSE, Fedora, Ubuntu. Пока, в один роковой день Ubuntu не внедрил Unity! В этот момент многие из нас соскочили с лодки ради удобства хорошо знакомого и любимого интерфейса — самым знакомым был Linux Mint с рабочим столом Mate, предлагающим внешний вид и способ работы, как в Gnome 2.

Представьте же мое удивление, когда я обновил ноутбук жены до Linux Mint 18 с Mate

и обнаружил, что стрелки прокрутки исчезли. Я, конечно, знаю, что можно достичь тех же результатов, используя стрелки вверх/вниз на клавиатуре, однако использовать мышь в ровной и управляемой манере теперь практически невозможно! Исследовав Интернет, я обнаружил ряд знаков, которые надо ввести в файлы настройки gtk, чтобы вернуть стрелки назад.

Теперь я обновил свою установку Mint до версии 19, и обнаружил, что отладка, работавшая в версии 18, больше не работает. Почему нечто, отлично работавшее, вдруг оказывается сломанным — и, похоже, намеренно? Я полагаю, что эта минималистская идея выросла из GTK3, но, как я понимаю, предполагалось, что Mate возьмет всё лучшее от Gnome 2 и добавит лучшее от Gnome 3, а не будет выбрасывать лучшие части Gnome 2!

Это меня одного раздражает данное изменение? Общая ли это проблема всех рабочих

столов, или, может быть, только всех рабочих столов на GTK (включая Mate)? Может, мне лучше перейти на другой рабочий стол или менеджер окон? Реально ли поставить KDE поверх Linux Mint с Mate и ожидать, что он будет работать?

*Дерек Барлоу (Derek Barlow),
по электронной почте*

О Я тоже большой фанат Mate. Единственное, что в нем чуть-чуть огорчает, это немалая нагрузка на ресурсы, так что на действительно старом оборудовании это не самый лучший выбор. Но что касается сохранения традиционной метафоры рабочего стола (и напоминания мне о счастливых днях Gnome 2), он лучший.

Во многом проблема усечения полос прокрутки на самом деле не в Mate; как Вы и подозреваете, это связано с GTK. Новые узкие панели, которые появляются, когда Вы наводите на них мышь, называются оверлейными полосами прокрутки; они и по внешнему виду, и по поведению отличаются от своих предшественников. Есть переменная среды для возврата к старому дизайну, но она признается не всеми приложениями, и можно полагать, что чем дальше, тем меньше она будет приниматься. Вероятно, в Mint 19 она менее эффективна, чем в его предшественнике.

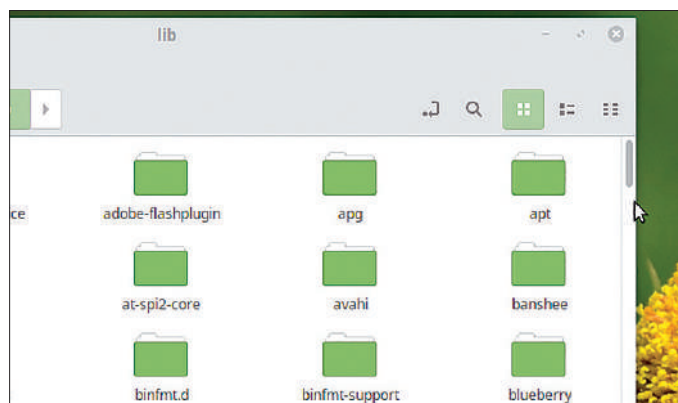
Можете поэкспериментировать, добавив строку `export GTK_OVERLAY_SCROLLING=0` в свой файл `.xinitrc`. Или запуская графические приложения из терминала малоизвестной командой `GTK_OVERLAY_SCROLLING=0 caja`. Эта переменная не исправляет поведение новой полосы прокрутки, а именно, не прокручивает до положения, где было щелкнуто, но, вместо этого, прокручивает страницу вверх или вниз, как в былые дни. Чтобы это исправить, Вы можете создать или изменить `~/.config/gtk-3.0/settings.ini` таким разделом:

```
[Settings]
gtk-primary-button-warps-slider = false
```

Возможно, Вы уже открыли для себя хитрость включения маленьких стрелочек, известных также как степперы, вручную, изменяя тему CSS. Если нет, попробуйте изменить/создать файл `~/.config/gtk-3.0/gtk.css` со следующим блоком:

```
.scrollbar {
-GtkScrollbar-has-forward-stepper: true;
-GtkScrollbar-has-backward-stepper: true;
}
```

Минимизация полосы прокрутки, о которой Вы говорите, является частью большой эволюции



А ты еще прокручиваешь, брат? Вот так дети сегодня осуществляют навигацию по окнам.

» ГОРЕСТИ САЙТА

В Что вообще происходит с вашим сайтом?
Анонимный, но озабоченный пользователь

О Хороший вопрос. После долгих лет пренебрежения наш бедняга-сервер рухнул, и мы собираемся всё у себя перенести. Печальный день, особенно с учетом его невероятной (десятилетней) службы, но нам больше не имеет смысла полагаться на щедрость UKFast по обеспечению ресурсов,

на поддержку которых у команды *Linux Format*, находящейся в тяжелом финансовом положении, нет средств. Сейчас старый сервер условно вернули к жизни, но не удивляйтесь, если он опять рухнет, ну или превратится в тыкву. Вдобавок кто-то [Ред.: — Джонни?] должен перекодировать наш архив подписчиков, что отнюдь не тривиально. Поэтому когда новый сайт запустится, на нем, вероятно, не будет портала доступа к историческим PDF.

«современного» рабочего стола. Баг на https://bugzilla.gnome.org/show_bug.cgi?id=790677 рисует мрачную картину будущего классических полос прокрутки в GTK4, но, возможно, они возродятся на уровне приложений, если разработчики осознают их нужность, или если люди просто достаточно активно будут теребить разработчиков.

Добавить еще один рабочий стол к Вашему дистрибутиву — совершенно нормально. Они не будут конфликтовать друг с другом, и Вы сможете выбрать требуемый на экране приглашения. Единственная проблема (которая куда мягче, чем раньше) — стиль программ Qt в Gnome и программ GTK3 в KDE Plasma иногда выглядит странно. Заменить менеджер окон также возможно (например, некоторые вполне нормально используют в LXQt вместо Openbox — Kwin), но это никак не повлияет на полосы прокрутки. Браузеры сейчас имеют тенденцию создавать собственные полосы прокрутки и проявлять уважение к старому стилю, поэтому не всё так плохо.

В Могучие Ge-force

Мой удобный и расположенный рядом компьютерный магазин собрал мне i7 с очень

красивым корпусом Fractal Design R6. Он очень тихо работает, и я без проблем настроил Kubuntu и Gentoo (даже незаметно, что он запустился, настолько он тихий). Кулер Noctua D15 позволяет ему работать всего при пяти-шести градусах выше температуры окружающей среды. В нем нет видеокарты, и пока мне приходится обходиться встроенной графикой Coffee Lake для всех моих радостей FlightGear.

Меня интересуют новые карты Nvidia. Я вообще не притрагиваюсь к играм FPS, потому что меня укачивает при движении. Поэтому мой вопрос заключается в следующем: нужна ли мне 2080 GTX (не-Ti) или я могу подождать 2070 и использовать ее несколько лет? Некоторые карты 2080 имеют издания 'Turbo' или 'OC' — не разрушат ли они мою тихую сборку? Меня привлекает 1070, но опять же, дотянет ли она до 2022 г.? Я начал со Slackware 7.1 в другой стране, и купил набор в коробке Red Hat 5.2, чтобы начать использовать FlightGear. Приведет ли драйвер Nvidia серии 400 к отличиям в работе моей сборки Gentoo? Gentoo Portage наложил маску на драйверы сверх 340.107. И опять же, я не смотрю на FPS или сравнительные тесты, но меня интересует

хорошая тихая система, тогда как мой кошелек просит пощады.

Рoger Андреассен [Roger Andreassen], по электронной почте

О Я бы сказал — побалуйте себя и не думайте о новых картах Nvidia. На самом деле они не намного мощнее серии 10x0, если исключить их возможности трассировки лучей. Если играть лишь изредка, я бы сказал, даже 1070 будет чересчур, 1060 прекрасно подойдет, а Ваш кошелек скажет Вам за это «спасибо». Однако Вы лишитесь своего тихого запуска: когда карта включается, вентиляторы на несколько секунд запускаются на полную мощность, пока не включится управление питанием. Что касается срока службы, я до сих пор пользуюсь GTX680 из 2012 г., и она прекрасно работает для моих обычных игр. Она даже отлично справилась с Nier:Automata на Steam Play. Возможно, стоит задуматься о более старых картах, однако под нагрузкой они больше греются и работают громче.

Прошло немало времени с тех пор, как я сражался с Gentoo, но картам серии 1000 нужны как минимум драйверы 396, а 2000 потребуют новой серии 410. В теории всё это должно быть просто: есть eclass, который должен определить, с какой версии драйвера снять маску. Проприетарный драйвер означает, что Вам придется компилировать еще одну вещь при каждом обновлении ядра, но Ваш компьютер достаточно мощный, чтобы Вам стоило об этом переживать. Всё же мы предпочитаем открытый подход, практикуемый AMDGPU, так что Вам, возможно, стоит обратить внимание на их предложения. LXF

» УБЕЙ МЕНЯ СРАЗУ!

В Такое ощущение, что я что-то пропустил. Мне частенько приходится использовать команды kill и killall для остановки плохо работающих команд. Чаше всего причина в том, что игры в Steam Play выдают сбой и просто сгорают. Метод, который я практикую сейчас — выяснить по top или ps aux, какие процессы еще не вычищены, затем применить к ним killall или kill (если это только один процесс). Если это не работает, тогда я применяю массовое поражение, добавляя к вышеупомянутым командам -9.

Не могу отрицать, что при этом испытываю некое удовлетворение, однако бывает, что требуется несколько итераций. Иногда приходится убивать один процесс (часто это wineserver.exe), и тогда все остальные тихо

исчезают, однако куда чаще я вижу, что нужно избавиться от целой группы программ.

Серийный убийца (программ)

О У программы wineserver имеется удобная опция -k, которая должна убирать все экземпляры себя, и вы можете добавить к ней 9, если она не слушается. Однако часто при сбое WINE приходится убирать куда больше, прежде чем она снова нормально запустится (особенно из Steam Play). Однако программа top обладает всеми нужными возможностями, пока она работает, нажмите k — и вы сможете убить программы по PID. Вам даже подскажут, каким сигналом их можно убить. Команда killall остается лучшим выбором для эвтаназии целой группы процессов, но с одинаковыми именами.

```
top - 14:26:21 up 3 min, 1 user, load average: 0.15, 0.29, 0.
Tasks: 221 total, 3 running, 218 sleeping, 0 stopped, 0 z
%Cpu(s): 1.9 us, 3.6 sy, 0.0 ni, 94.4 id, 0.0 wa, 0.2 hi,
MiB Mem : 16822.4 total, 14541.2 free, 933.4 used, 547.
MiB Swap: 0.0 total, 0.0 free, 0.0 used, 14816.
PID to signal/kill [default pid = 497]
  PID USER      PPID  CPU   MEM   VSZ   RSS   TTY    TIME  COMMAND
  497 root        20    0 R   15.2  0.3  0:15.83 Xorg
  989 jonni      20    0 S   15.2  1.4  0:18.14 cool-retro-term
  721 jonni      20    0 R   8.8   0.9  0:14.88 kwin_x11
  499 root       -51    0 S   2.5   0.0  0:05.32 irq/30-nvidia
 1814 jonni      20    0 R   2.5   0.0  0:00.98 top
  681 jonni      20    0 S   1.2   0.4  0:00.42 kded5
    1 root        20    0 S   0.0   0.1  0:00.00 systemd
    2 root        20    0 S   0.0   0.0  0:00.00 kthreadd
    3 root         0 -20 I  0.0   0.0  0:00.00 rcu_gp
    4 root         0 -20 I  0.0   0.0  0:00.00 rcu_par_gp
    5 root        20    0 I  0.0   0.0  0:00.04 kworker/0:0-ev+
```

Теперь можно убить (или просто TERM) своих демонов за несколько нажатий из программы Top.

ПОЛУЧИТЕ ПОМОЩЬ!

Мы рады ответить на все вопросы по Linux, которые вы присылаете на lxf.answers@futurenet.com, независимо от их уровня.

У нас у всех когда-то были проблемы, так что не стесняйтесь. Однако мы всего лишь люди (хотя многие подозревают, что Джонни — робот), и очень важно, чтобы вы указывали максимум информации. Если что-то работает на одном дистрибутиве, а на другом — нет, сообщите нам. Если вы получили сообщение об ошибке, пожалуйста, сообщите нам его точный текст и из-за чего оно появилось.

Если у вас есть, или вам кажется, что есть проблема с оборудованием, расскажите нам о нем. Подумайте об установке Hardinfo или lshw. Эти программы показывают оборудование на вашем компьютере. Если вы не хотите или не можете установить эти программы, запустите следующие команды в терминале root и пришлите нам также файл system.txt.

```
uname -a > system.txt
lsipc >> system.txt
lsipc -vv >> system.txt
```


HotPicks

Haiku » Eolie » Heimer » Acid Cam » Audacity » Treeline
» GrafX2 » EtherApe » Gaupol » Zabuyaki » uRacer



Александр Толстой
переходит на микроядро
и отказывается от Linux
в пользу Haiku — по крайней
мере, для серии HotPicks
этого месяца.

ОПЕРАЦИОННАЯ СИСТЕМА

Haiku

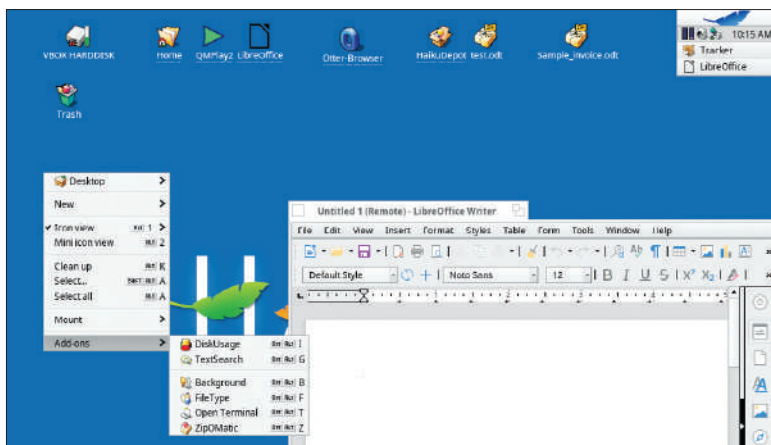
Версия: R1 Сайт: www.haiku-os.org

Наiku участвовала в *Сравнении* альтернативных ОС с открытым исходным кодом в **LXF200** и заняла второе место. Хотя эта ОС, навеянная BeOS, с момента ее создания в начале 2000-х гг. вызывала лишь скромный интерес сообщества разработчиков свободного ПО, сейчас настало время перемен.

В 2018 г. проект Haiku ускорил свое развитие и предъявил несколько замечательных новых функций. В Haiku был портирован целый ряд программ с открытым исходным кодом; все они доступны через чудесное хранилище Haiku Depot — нововведенный менеджер пакетов и магазин программного обеспечения. Признанным узким местом раньше было отсутствие офисных приложений, но теперь в Haiku могут работать *LibreOffice*, *Krita*, *Blender*, *Qmply2* и десятки других известных программ OSS.

Новый выпуск R1 был долгожданным, ведь предыдущая официальная версия — R1 Alpha — вышла еще в 2012 г. Она была склонна к сбоям, имела ограниченную аппаратную поддержку и не поставлялась с Haiku Depot, не оставляя пользователям иного выбора, кроме как вручную прочесывать Интернет в поиске пакетов *.hpkg или охотиться за древними программами BeOS. Одним из обходов этой проблемы было использование Haiku Nightly, что, хотя и давало свежий код, могло быть даже менее стабильным, чем R1 Alpha. Теперь, когда Haiku R1 находится в свободном доступе, для новичков это будет достаточно веской причиной ее попробовать.

Haiku не основана на UNIX, но имеет слой совместимости с POSIX и содержит множество базовых утилит GNU, предлагаемых из коробки. Эта микроядерная ОС, несколько похожая на macOS, уже выглядит как готовый продукт. При установке операционной системы выберите образ 'anyboot' и запишите его на USB-брелок или CD. Установочный образ также предоставляет систему live, которая даст прекрасную возможность увидеть, как Haiku работает на вашем оборудовании, до ее фактической установки. Мы прекрасно провели время, тестируя Haiku R1, и пришли к выводу, что Haiku Depot работает хорошо. Он сочетает в себе преимущества менеджера пакетов (вроде *Synaptic*) с магазином, где представлен ассортимент лучших предложений программного обеспечения (во многом как в App Center в elementary OS).



Интерфейс выглядит слегка архаичным из-за его наследия BeOS, но зато производителен.

Прогуляемся по Haiku OS



1 Рабочий стол

На рабочем столе можно размещать файлы, папки и ярлыки приложений [Ред.: – Возмутительно!], как и в любой классической рабочей среде.

2 Табличные границы окна

Это дизайнерское решение для BeOS и Haiku, которое экономит место на экране.

3 Познакомьтесь с панелью

Скоро вы привыкнете к этой компактной панели, которая

сочетает в себе пускатель, панель задач и системный лоток.

4 Настройте систему

Haiku глубоко настраивается и позволяет изменять различные настройки оборудования.

5 Контекстные меню

Tracker, файловый менеджер Haiku, позволяет управлять файлами и задачами через меню, вызываемое правой кнопкой мыши.

WEB-БРАУЗЕР

Eolie

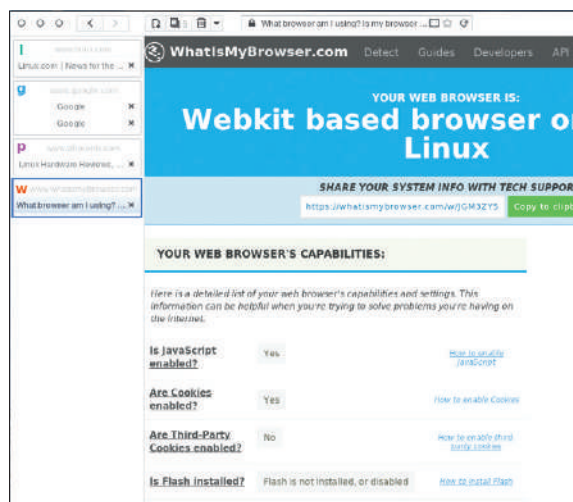
Версия: 0.9.37

Сайт: <https://gitlab.gnome.org/world/eolie>

Вкладки поддерживаются всеми современными браузерами, и если такие вкладки не представлены особым новым способом, эту функцию прославлять не приходится. Шаг вперед, *Eolie* — с оригинальным способом держать несколько web-страниц открытыми в одном окне; мы считаем, что этот подход к дизайну понравится многим людям. Вместо горизонтального отображения вкладок *Eolie* рисует вертикальную черту, где каждый сайт представлен в виде небольшого прямоугольника. Это похоже на вертикальную панель задач, которую предпочитают многие пользователи Linux.

Eolie основан на Webkit2, GTK3 и нескольких модулях Python3. Он разработан с целью идеально вписаться в стандартный рабочий стол Gnome и, возможно, заменить *Gnome Web* (урожденный *Eiphanu*). Мы попытались найти различия между *Eolie* и *Eiphanu*, поскольку оба используют WebkitGTK и имеют много общих функций, например, поддержку Sync, как в *Firefox*. Однако *Eolie* может похвастаться более активным развитием; и он явно предназначен для людей, которые недовольны интерфейсом *Eiphanu*.

Eolie любит, чтобы всё было просто. Он предлагает универсальную адресную строку со встроенной историей просмотра,



отображаемой в строке заголовка окна, полноэкранный режим и возможность отключить его заметную боковую панель.

Тем не менее, в этом браузере доступно множество функций. У *Eolie* есть полноценный блокировщик рекламы, режим «чтения», режим приватного просмотра и даже инструмент для загрузки видео. Последний работает прекрасно: щелкните правой кнопкой мыши web-страницу и выберите Save videos [Сохранить видео]. Затем откиньтесь на спинку кресла и смотрите, как ваше любимое видео с YouTube загружается в формате MP4. *Eolie* также умеет сохранять изображения со страницы или экспортировать саму страницу как изображение.

Eolie можно попробовать во многих дистрибутивах Linux, а также получить его с Flathub.

Eolie экономит много места на экране и под капотом скрывает продвинутые функции.

ИНСТРУМЕНТ КАРТ ПАМЯТИ

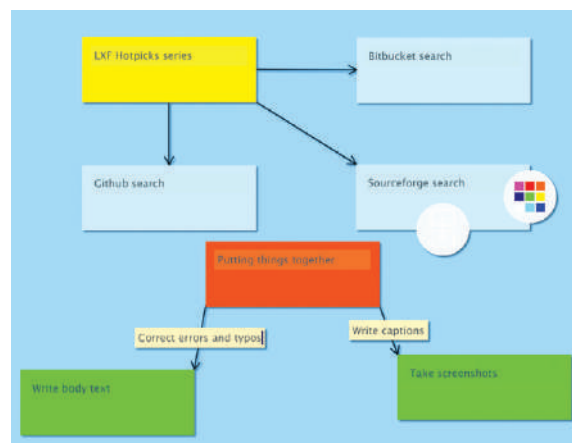
Heimer

Версия: 1.2.0 Сайт: <https://github.com/juzzlin/heimer>

Надоело вечно всё забывать? Нам тоже, поэтому мы накопили пару инструментов для создания в Linux карт памяти (они же — интеллект-карты): *Minder* (<https://github.com/phase1geo/Minder>) и *Heimer*. Когда мы попробовали их сравнить, на холсте *Minder* понадобилось слишком долго выяснять, как расположить дочерние узлы. *Heimer* выполнил данную работу быстро, по каковой причине и очутился среди HotPicks этого месяца.

Обычно от инструмента карт памяти требуется простота использования, и именно здесь *Heimer* блистает. Это инструмент с очень низким барьером входа, т.е. любой пользователь Linux может с помощью *Heimer* нарисовать красивую карту памяти за считанные минуты. Функций предлагается не так уж много: можно добавить сколько угодно узлов, надписать их названия, применить цвет и снабдить текст соединительными стрелками. Вот и всё; но вы не ощутите, что программе недостает функциональности.

Каждый узел может быть родительским для любого количества дочерних узлов, но в узел может входить только одна стрелка. Если вам надо нарисовать карту, где много узлов указывают на другой, примените такой обходной маневр: просто



создайте несколько узлов назначения и сложите их стопкой, чтобы они выглядели как один узел.

Heimer обладает достаточной мощностью, чтобы рисовать классические карты, где один объект происходит от другого и генерирует свои собственные дочерние объекты. Стрелки могут иметь подписи, помогающие добавить карте логики. Если вам не нужно соединять узлы, просто щелкните правой кнопкой мыши на холсте и выберите Create floating node [Создать плавающий узел], чтобы создать независимый узел без стрелок. Закончив, вы можете экспортировать свою карту в виде изображения PNG или сохранить ее в собственном внутреннем формате *Heimer*, который не без ехидства назван *.alz*.

Если это раздражило ваш аппетит, можете попробовать *Heimer*, взяв AppImage со страницы проекта, сделав его исполняемым и запустив его.

Создание красочной карты памяти в *Heimer* — пара пустяков!

ПРИЛОЖЕНИЕ ДЛЯ ОБРАБОТКИ ВИДЕО

Acid Cam

Версия: 1.19.1

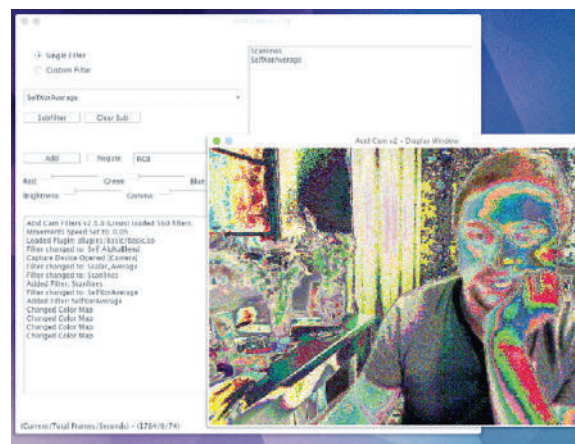
Сайт: <https://github.com/lostjared/acid.cam.v2.qt>

Дистрибутивы Linux поставляются с *Cheese*. Это приложение Gnome по умолчанию для web-камеры, и оно имеет множество видеоэффектов. Играть с различными видами искажений и инверсии цветов очень забавно, но на сей раз мы получили нечто получше.

Acid Cam — передовое приложение для web-камеры, и основное внимание в нем уделяется видеоэффектам, самый яркий из которых — «кислотная [acid]» раскраска. Это версия проекта второго поколения. Номер версии 1.19.1 — внутренний для порта *Acid Cam* на Qt для Linux и Windows (для версии macOS имеется отдельный проект).

Запустив *Acid Cam*, вы увидите, что здесь присутствует целый ряд эффектов и фильтров. Полный список содержит около 80 вариантов инструментов, позволяющих исказить цвета, смешивать, зеркально отображать пиксели и накладывать причудливые узоры. Эффекты превосходят то, чего можно достичь в *Cheese* или *Kamoso*.

Версия второго поколения этой программы позволяет выбирать источник видео. Можно использовать web-камеру с поддержкой V4L, а также выбрать видеофайл и приказать *Acid Cam* применить видеофильтр в режиме реального времени. Всё,



что вы видите на экране, обрабатывается на лету, и программа может записать результаты в локальный файл. Главное окно *Acid Cam* позволяет выбрать несколько фильтров и применить их одновременно, хотя это может швырнуть ваш графический процессор на колени. Меню Options [Параметры] позволяет управлять динамикой вывода, например регулировать скорость и поведение фильтра.

Мы отлично провели время с *Acid Cam*, и хотя он не всегда превращает обычный видеофайл в произведение искусства, его эффекты просто потрясающие. Независимо от того, что вы скормили *Acid Cam* — живое изображение с web-камеры или локальный медиа-файл — обработанный результат стоит того, чтобы его посмотреть. И при каждом нажатии кнопки Start [Пуск] сессия всегда сохраняется на вашем жестком диске!

Войдите в мир волшебных видеоэффектов и создайте уникальный опыт с *Acid Cam*.

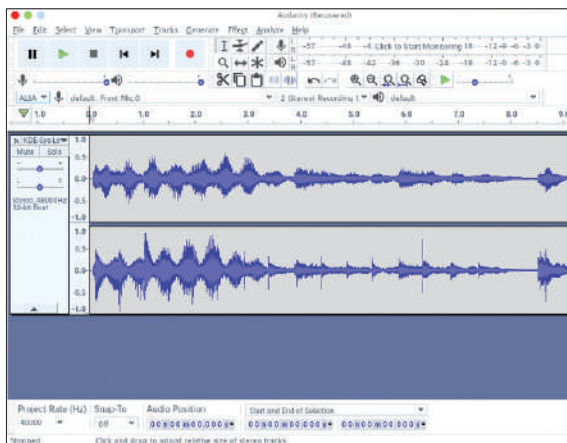
РЕДАКТОР ЗВУКА

Audacity

Версия: 2.3 Сайт: www.audacityteam.org

Отличная и мощная программа для записи подкастов, с открытым исходным кодом, способная помочь вам освоить многодорожечный аудиопроjekt, и, конечно, один из лучших примеров высококачественного FOSS — это, вкратце, *Audacity*. Мы писали о версии 2.2 год назад, в **LXF232**, но теперь, в версии 2.3, появились еще более восхитительные новые функции.

Мы взяли последний выпуск *Audacity* в основном из-за его рекламируемой функции Punch and Roll [вставь и продолжи]. Она позволяет вам, наконец, забыть про копирование и вставку фрагментов аудио в случае, если вам нужно срastить [splice] аудио из нарезки (например, при создании мастер-записи из нескольких удачных дублей). Функция Punch and Roll — простое решение для исправления ошибок. Как только вы почувствуете, что вам лучше бы переделать запись, остановите ее и с помощью линейки выберите положение точки слияния. Затем выберите Transport > Recording > Punch and Roll Record [Подача > Запись > Запись Punch and Roll], чтобы возобновить запись. Эта функция не вставляет фрагмент в дорожку; вместо этого будет перезаписываться всё, что следует за точкой слияния, пока вы специально не остановите его.



Audacity 2.3 улучшает звукозапись, являющуюся одной из самых популярных функций.

Функция Punch and Roll — гигантский скачок вперед для всех, кто использует *Audacity* для записи подкастов или музыки, поскольку экономит много времени на попытках создать идеальную звуковую дорожку. Но в *Audacity 3.0* есть и другие новые функции. Например, теперь можно менять размеры панелей управления воспроизведением и громкостью; растягивание этих панелей помогает добиться более точной настройки звука. С учетом других мелких улучшений интерфейса пользователя, *Audacity* остается, пожалуй, лучшим звуковым редактором для Linux — мощным, но простым в использовании, интуитивно понятным и гостеприимным. Нормализация звука, удаление фонового шума, упорядочение пары треков с эффектами затухания и экспорт проекта в OGG или MP3 не отнимут у вас много времени.

Audacity широко доступен практически во всех дистрибутивах Linux, а также в магазинах Flathub и Snap.

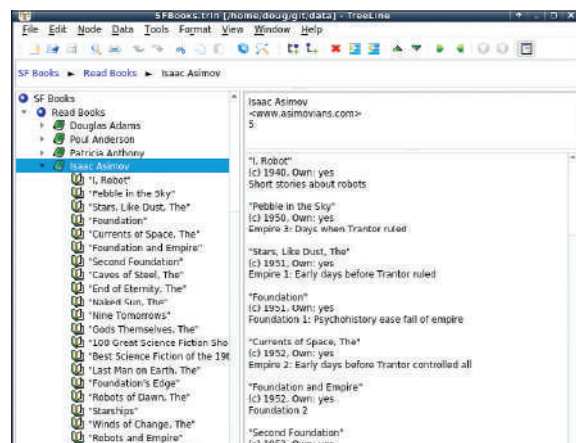
МЕНЕДЖЕР ПЕРСОНАЛЬНОЙ ИНФОРМАЦИИ

TreeLine

Версия: 3.0.1 Сайт: <http://treeline.bellz.org>

Нам всем надо следить за множеством логинов, паролей, контактов и всяких других учетных данных. Менеджеры персональной информации (PIM) помогают организовать подобные данные, но большинство их ориентированы на конкретные типы информации: электронные письма, контакты и т. д. В отличие от них, *TreeLine* охватывает практически всё: это приложение PIM, позволяющее хранить информацию любого вида.

TreeLine — приложение на *PyQt*, которое существует уже много лет, хотя при виде последней версии вам это и в голову не придет. Всё отшлифовано и ускорено. Интерфейс четкий и интуитивно понятный, с панелью дерева слева и панелью контента рядом с ней. Можно создавать сложные ветви деревьев со многими уровнями подстановки, и доступ к ним упрощается дорожками [breadcrumbs]. Каждая запись может хранить не просто текст, а много полей информации. Поддерживаются форматирование текста и разметка HTML. Каждый узел верхнего уровня в *TreeLine* называется файлом и технически является файлом JSON. Файлы могут содержать дочерние узлы с разными наборами полей, поэтому *TreeLine* не ограничивается хранением библиотеки или музыкальных альбомов.



Интерфейс программы позволяет настраивать узлы и содержащиеся в них поля (через диалоговое окно Data Types [Типы данных]), благодаря чему приложение можно точно настроить под любое применение. Оно работает как нереляционная мини-база данных. Независимо от содержимого, которое вы хотите сохранить в *TreeLine*, нижняя часть основной области содержимого предлагает три вкладки: Data output [Вывод данных], Data Edit [Редактирование данных] и Title list [Список заголовков]. Это напоминает HTML-редакторы WYSIWYG, где можно мгновенно увидеть, как изменения в коде или настройках отражаются в окончательном отображении страницы.

Таким образом, хотя в *TreeLine* нет особенно хитрых функций, это надежная и очень стабильная программа, и она отлично справляется со своей задачей. На нее определенно стоит поглядеть поближе.

Четкая и логичная программа для поддержки администратора.

РЕДАКТОР ИЗОБРАЖЕНИЙ

GrafX2

Версия: 2.5 Сайт: <http://grafx2.tk>

Это один из самых необычных редакторов изображений, которые мы когда-либо рассматривали. *GrafX2* выглядит ретро, как *LibreSprite* (LXF242), и далек от обычных редакторов изображений вроде *AzPainter* (LXF223). Все эти творческие программы имеют одну общую черту: ради максимальной отдачи от каждой программы пользователь должен потратить немало времени и усилий. Мы засучили рукава и нырнули прямо в *GrafX2* — и обнаружили, что он стоит трудов.

GrafX2 — это редактор для рисования пиксельных иллюстраций, и как только вы начнете что-то делать на холсте, вы заметите, насколько мощные его инструменты. Несмотря на намеренно блоковидный интерфейс, *GrafX2* более продвинуто, чем *Paint*-подобные программы прошлых лет. Он имеет настраиваемую историю отмен [Undo], различные кисти, меню градиента, слои, инструменты выделения и даже ряд великолепных эффектов в диалоге режима рисования [Drawing].

Даже когда вы ознакомитесь с тем, как работают инструменты программы, вам потребуется больше времени для выполнения сложных задач, таких как рисование узоров и плиток, или выбора части изображения и использования его в качестве художественной кисти. Размер холста выглядит фиксированным, но на самом



Режимы рисования позволяют создавать более сложные пиксельные картинки.

деле определяется размером окна *GrafX2*. Однако существуют слои с различными свойствами, включая прозрачность, а также страницы. По умолчанию *GrafX2* предлагает сохранять рисунки в формате GIF, где страницы превращаются в кадры, но предусмотрен широкий выбор других знакомых форматов, включая PNG и BMP.

В самой нижней части окна *GrafX2* показаны советы и описания для инструментов выше. Во многих случаях это реальная помощь, потому что кнопки инструментов часто не слишком проясняют выполняемую ими функцию.

GrafX2 — тот тип приложения, который мотивирует и вдохновляет даже неподготовленных пользователей создать нечто творческое. Настраиваемый набор мощных инструментов рисования в сочетании с разумно разработанным графическим интерфейсом, где доступны все функции — вот ключ к успеху!

ИГРА-ВОЙНУШКА

Zabuyaki

Версия: Git Сайт: www.zabuyaki.com

Луа — многообещающий скриптовый язык, идеальный для создания небольших игр, главным образом благодаря LOVE, игровому движку на Lua. Наша текущая находка — популярная игра на Lua под названием *Zabuyaki*, что в переводе на русский означает «уличные драчуны». Это веселая игра, полная энергии, иронии и великолепных работ в стиле ретро.

Zabuyaki сочетает в себе элементы аркадной прокрутки и классической видеоигры с драками. Доступно четыре персонажа: Rick [Рик] (дюжий блондин), Kisa [Киса] (женщина), Chai [Чай] (азиатский мастер кунг-фу) и Yag [Яр] (медведь в красных кроссовках). Однако, поскольку игра находится в стадии разработки, пока в нее могут играть только Рик и Чай. Оба имеют свои уникальные и отличительные черты, удары и атаки, и все они выглядят очень плавными и естественными.

Действие *Zabuyaki* происходит в центре Санкт-Петербурга. Оконные решетки на первом этаже, реклама фаст-фуда и обветшавшие водосточные трубы выглядят одновременно забавно и мило, создавая неповторимую атмосферу утрированного образа вымышленной зоны «сюда не ходи». В любом случае, не стойте и не смотрите слишком долго, потому что местная банда негодяев



Перенесите свои боевые навыки из *Double Dragon* на улицы современного Петербурга.

готова измолотить вас в кашу. Их контролирует продвинутый ИИ, но и Рик, и Чай тоже калачи тертые.

Много заботы и внимания уделено деталям хулиганов, уличных объектов и игровых персонажей. Ваш герой управляется клавишами со стрелками плюс X для атаки и C для прыжка, но существует множество комбинаций этих клавиш, которые выполняют различные супер-удары и сложные атаки. Чтобы победить врагов, можно сломать дорожный знак и вспрыгнуть на мусорный бак, но учтите и их сильные стороны. Они могут задушить вас, напасть на вас с боксом или неожиданно пнуть. Попытка убежать от группы из них иногда приводит к тому, что ваши противники путаются в ногах и падают на тротуар.

Сложность средняя, и мы предсказываем, что *Zabuyaki* будут скорее игрой-забавой, чем жесткой задачей на выживание. Но всё равно поиграть стоит!

ГОНКИ

uRacer: The King Of The Drift

Версия: 0.9.4 Сайт: <https://github.com/manuelbua/uracer-kotd>

Давайте устроим автослалом, а не борьбу. Эта привлекательная игра в виде сверху является хорошим кандидатом для тех, кто любит гонки и красивую графику в играх с открытым исходным кодом для Linux. *uRacer: The King Of the Drift* предлагает слалом как основной элемент игровой механики.

На первый взгляд, у *uRacer* много общего с *Dust Racing 2D (LXF200)*, которая тоже проходит в виде сверху; но затем игра вступает в свои права. Она отличается очень впечатляющей и качественной графикой. *uRacer* зависит от производительности вашего видеочипа и на оборудовании низкого качества способна тормозить, но при приличной отдельной видеокарте — блистательна. Автомобиль, трасса и пейзаж выглядят очень гладко, и игра просматривается через старый эффект ЭЛТ-фильтра с линиями развертки в сочетании с затемнением окружающей среды.

Правда, элементы управления довольно необычные. Помимо клавиш со стрелками, автомобиль управляется мышью. Положение курсора задает направление, а щелчок левой кнопкой ускоряет автомобиль. Итак, вам надо нажать и удерживать левую кнопку



мышью, а затем постараться не пропустить поворот. Машина — зверь в управлении, и у вас очень мало времени, чтобы реагировать на повороты. Очевидно, что неопытный игрок в *uRacer* имеет мало шансов удержать машину на трассе, не врезаясь в дорожное ограждение, но это отнюдь не проблема: автомобиль невосприимчив к повреждениям, и у вас есть неограниченная свобода для безопасной гонки по трассе. Каждый удар по барьеру сопровождается интенсивным эффектом а-ля «солнечный ветер», который стоит увидеть сам по себе.

Мы прекрасно провели время, исследуя каждый из шести треков *uRacer*, где можно соревноваться с «призраком» вашего лучшего результата. Это-таки потребовало времени, но в конце концов мы развили необходимые навыки, сравнимые с полученными после некоторого времени игры в *Neverball*. Прохождение разворотов с использованием ваших навыков автослалома кажется очень полезным, даже если смотреть сверху.

В игре потрясающие визуальные эффекты, которые выходят на первый план, когда вы сталкиваетесь с дорожным барьером.

ДИАГНОСТИКА СЕТИ

EtherApe

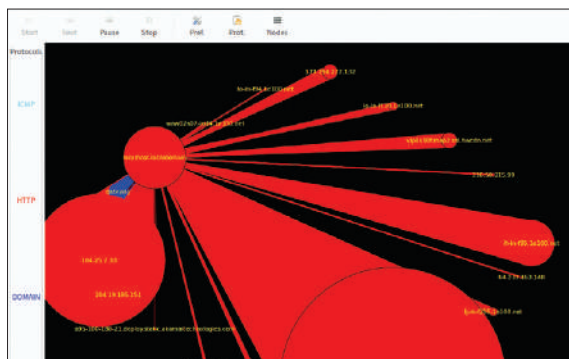
Версия: 0.9.18

Сайт: <https://etherape.sourceforge.io>

ЕtherApe — инструмент мониторинга и анализа, который покажет вам, что происходит за кулисами вашей локальной сети. Не будучи полноценным инструментом криминальной цифровой экспертизы, *EtherApe* способен фильтровать все сетевые протоколы и предоставлять снимок того, какие биты проходят по сети, где и с какой скоростью.

Настраивать *EtherApe* для получения понятных результатов нет необходимости. С момента первого запуска программа начинает прослушивать активный сетевой интерфейс, рисуя подключенные узлы в режиме реального времени. Размер кружка указывает объем трафика на этом узле, а цвет указывает тип трафика (HTTP, HTTPS, FTP и т.д.).

График постоянно обновляется, поэтому для просмотра истории сети в более неторопливом темпе лучше всего нажать кнопку Nodes [Узлы] на панели инструментов *EtherApe*, чтобы просмотреть список адресов. Каждая строка показывает объем накопленного трафика для соответствующего адреса, и легко увидеть основные потоки. Меню Capture [Захват] позволяет выбрать режим прослушивания (Link Layer, IP или TCP), а также определенный сетевой интерфейс.



Пока вы беззаботно фланируете по Сети, явно происходит много необъяснимого...

Опытные линуксоиды знают, как открыть инструменты разработчика в *Firefox* или *Chromium* (нажмите F12), чтобы увидеть сетевые подключения, но *EtherApe* предоставляет вам доступ ко всей вашей системе, а не только к тому, что творится в браузере. Различные CDN [сети доставки данных], рекламные платформы, телеметрические сервисы и социальные сети — всё это отлично видно. Кстати, сейчас хороший момент занести в черный список `/etc/hosts` хотя бы часть из них.

EtherApe 0.9.18 примечателен благодаря своему долгожданному переходу на *GTK3*, но стоит отметить, что при запуске без прав root (например, когда вы просто щелкаете по панели запуска *EtherApe*) возможности приложения ограничены. Запуск от имени root приложений с графическим интерфейсом считается ныне предосудительным, и вот приемчик, который можно употребить:

```
$ xhost + local: && sudo etherape
```

Не бойтесь, этот обходной путь не перекроет всевидящее божество *EtherApe*.

РЕДАКТОР СУБТИТРОВ

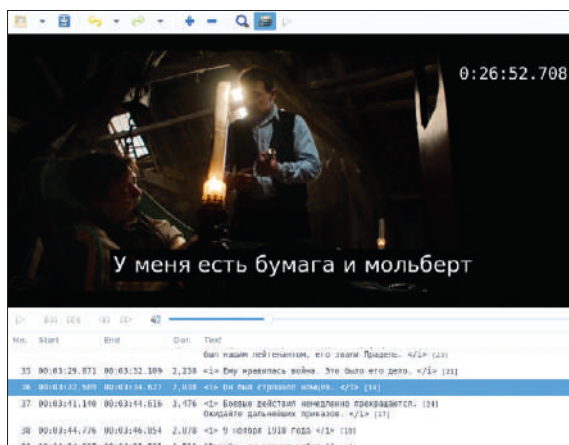
Gaupol

Версия: 1.4.1

Сайт: <https://otsaloma.io/gaupol>

Мы часто пишем о программном обеспечении, связанном с воспроизведением или преобразованием медиа-файлов, но существует много других программ, попадающих в категорию мультимедиа. Инструмент *Gaupol* разработан в помощь пользователям Linux при создании и редактировании субтитров для видеофайлов. Хотя цель этого приложения, очевидно, нишевая, она следует давней философии UNIX, которая подразумевает, что приложение должно выполнять одну задачу, но делать это хорошо.

Gaupol — компактное приложение на *GTK3*, сочетающее функции текстового редактора и видеоплеера. Стандартные функции Load/Save [загрузки/сохранения] относятся не к видео-файлам, а к субтитрам. *Gaupol* поддерживает широкий спектр популярных форматов субтитров, включая SUB, SRT, LRC, SSA, VTT и другие — нам не удалось найти файл, который *Gaupol* не смог обработать. Загруженные субтитры выглядят как таблица со столбцами для нумерации, времени начала/окончания, продолжительности и текстовых записей. Выберите Video > Load Video [Видео > Загрузить видео] или нажмите Ctrl+L, чтобы выбрать видеофайл, соответствующий вашим субтитрам. Вообще-то *Gaupol* отлично работает еще и как видеоплеер, хотя и лишенный полноэкранного режима.



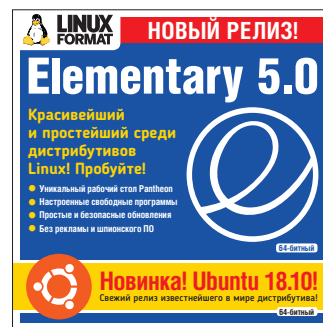
Наслаждайтесь фильмами на иностранных языках и исправляйте субтитры в *Gaupol*!

Программа способна корректировать субтитры, если они не соответствуют сцене. С помощью меню Tools [Инструменты] в *Gaupol* можно запускать проверку орфографии для столбца Text [Текст], сдвигать субтитры или менять их продолжительность. В программе даже можно автоматизировать рутинные задачи. Например, функция Correct Texts [Исправить тексты] (F8) умеет объединять/разбивать слова, разбивать строки, менять регистр шрифтов и даже исправлять опечатки без ручного вмешательства. Этот метод редко работает идеально, но всё равно экономит вам массу времени. Есть даже возможность изменить частоту кадров файла фильма и функция Preview [Предпросмотр], которая запускает плеер *MPV* (убедитесь, что он установлен).

В настоящее время *Gaupol* является наиболее мощным инструментом для освоения перевода иностранного фильма или исправления распространенных ошибок в готовых субтитрах. Он обязательно нужен всем любителям кино! **LXF**

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...



» НАЧНИТЕ ОТСЮДА

РАБОТА С ДИСКОМ

Первое применение Linux может озадачить. Linux отличен от того, что у вас, видимо, было раньше, особенно если вы привыкли к MS Windows или MacOS от Apple. Обычно у наших DVD прямой запуск, т.е. при первом включении ваш ПК (или Mac) должен «загружаться» с DVD, и еще до загрузки Windows или MacOS прямо с DVD запустится Linux. Этот трюк известен как Live DVD и позволяет попробовать разные версии Linux, ничего не устанавливая или изменяя на своем ПК. Просто извлеките DVD, перезагрузитесь, и ПК станет таким, каким был до того. Многие системы, обнаружив DVD, загрузятся с него, но многие — нет. Ниже приведен стандартный процесс включения загрузки с DVD на разных настольных ПК и ноутбуках.

Альтернатива — найти на DVD файл ISO, записать его на отдельный USB-накопитель и попытаться запустить его. Вам в помощь — *Etcher* с <https://etcher.io>, доступный для Windows, MacOS и Linux. Удачи!

ЗАГРУЗКА ДИСКА

Если ваш ПК при включении с DVD в дисковом не загружается автоматически, м.б. он предлагает ранее «загрузочное меню» по нажатию клавиши при включении: F9 (HP), F12 (Dell, Lenovo), F8 (Amibios) или F11 (Award BIOS). Или, чтобы начать работу с оптического привода, настройте порядок загрузки через BIOS/UEFI. В BIOS/UEFI вы попадете, нажав при включении питания клавишу — обычно Del, но иногда F1 или F2. Для ряда новых ПК с UEFI нужен доступ сперва из Windows: удерживая Shift, выберите параметр *Restart* [Перезапуск]. Если проблемы с DVD остались, см. www.linuxformat.com/dvdsupport.

У вас Mac? Для загрузки с DVD, включаясь, удержите клавишу C.

КРАСИВЫЙ ДИСТРИБУТИВ ДЛЯ НОВИЧКОВ

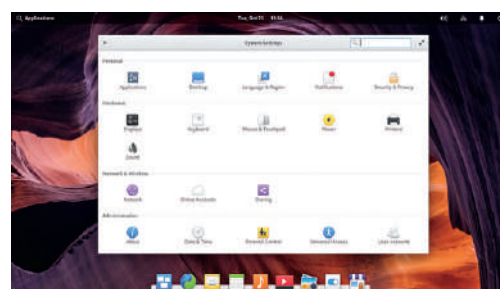
Elementary 5.0 Juno

64-БИТНЫЙ

Команда Elementary OS разработала один из самых лучших дистрибутивов, когда-либо созданных для Linux. В своем последнем выпуске они это закрепили и утверждают, что, перейдя с v0.4 на v5.0, Elementary OS из незавершенного продукта превратилась в готовую к повседневному применению.

Elementary OS основана на Ubuntu (узнайте больше о нашей свежей версии Ubuntu на стр. 107), но выделяется из толпы, предоставляя более изысканный, простой в работе и более гармоничный рабочий стол. Ключевой ее частью является индивидуальный рабочий стол под названием Pantheon, который разрабатывался годами с использованием новейших технологий открытого исходного кода.

Elementary 5.0 включает множество интересных улучшений. Night Light [ночной свет] вводит контроль цветовой температуры дисплея вручную или на основе местного времени заката/восхода. Плиточная раскладка окон автоматически привязывает их к областям переменного размера для облегчения рабочего процесса. Picture-in-picture использует трюк Android и web-воспроизведения видео: оно привязано к углу, не мешая вам параллельно выполнять другие задачи. Системные панели умеют определять яркость фоновых обоев и настраиваться так, чтобы их было легче увидеть. И более 1000 иконок были изменены с целью сделать их более согласованными.



Внешность значков Elementary была стандартизирована ради более опрятного и гармоничного вида.

Немало и других поправок. Не можете вспомнить ту самую комбинацию клавиш? Нажмите Super Key (некоторые называют ее клавишей Windows), чтобы получить список наиболее распространенных сочетаний клавиш. Встроен улучшенный инструмент экранных снимков [screengrabber], и предлагается поддержка эмодзи в классическом наборе Noto Color Emoji. Данный выпуск Elementary OS также проявил заботу о мире слуха, перенастроив звуки уведомлений.

Областью, над улучшением которой Elementary OS долго трудилась, является ее заказываемый выбор приложений. Они устанавливаются либо по умолчанию, либо через AppCenter за плату по вашему усмотрению, что

позволяет вам поддерживать разработку Elementary OS, одновременно улучшая свои вычислительные возможности. Вот какие программы доступны по умолчанию: *Code* (специализированная среда IDE), *Files* (файловый менеджер), *Music* (для управления вашими мелодиями) и *Photos*, позволяющий настраивать библиотеку фотографий.



Заполните свой дистрибутив по-своему, с помощью AppCenter.

» ВНИМАНИЕ! ЭТО ВАЖНО!

ПОРЧЕННЫЕ ДИСКИ: В маловероятном случае какого-либо дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Ubuntu 18.10

64-БИТНЫЙ

U buntu — одна из популярнейших версий Linux в мире. Классически, Ubuntu любит выпускать два основных релиза ежегодно, а Ubuntu 18.10 является вторым в 2018 г. по 10-месячному графику. В отличие от выпусков с долгосрочной поддержкой, эти промежуточные выпуски получают обновления девять месяцев, поэтому они предназначены скорее для людей, желающих опробовать Ubuntu, чем для установки ОС для долгосрочного использования.

Новинкой для Ubuntu 18.10 является свежая версия рабочего стола Gnome 3.30. Предыдущий релиз, 18.04, перешел на рабочий стол Gnome 3.x. Более интересным является введение темы сообщества Yaru, которая установлена по умолчанию — освежающее изменение стандартного оранжевого интерфейса.

Ubuntu теперь имеет передовые алгоритмы сжатия, называемые LZ4 и zstd. Усовершенствованная система позволяет загружать и устанавливать Ubuntu быстрее этак на 10%. Компания, стоящая за Ubuntu, также продолжает развивать свою технологию доставки Snap-пакетов, это новый способ упаковки установочных файлов приложений. Улучшения в 18.10 помогают быстрее запускать Snaps и упрощают их установку. Отчасти

это связано с тем, что Snap applicationz интегрированы в Центр программного обеспечения.

Внутри этого релиза Ubuntu находится Kernel 4.18, последняя версия ядра, доступная на момент выпуска. Список изменений огромен; обновления включают поддержку графического процессора AMD Radeon RX Vega M и полную поддержку Raspberry Pi 3 B и 3 B+ с улучшенной поддержкой Thunderbolt и USB Type-C. Если вы любитель игр, теперь поддерживается Steam Controller, вместе с Open GL 4.4 и обновлением Mesa 18.2. Да, и если вам также интересно, что это за безумное фоновое изображение — оно вытекает из кодового названия данного релиза: Cosmic Cuttlefish [Космическая Каракатица]. Не спрашивайте...



» НОВИЧОК В LINUX? НАЧНИТЕ ОТСЮДА!

Мы ответим на ваши вопросы, установите DVD и откройте [index.html](#).

- Что такое Linux? Как его установить?
- Есть ли эквивалент MS Office?
- Для чего нужна командная строка?

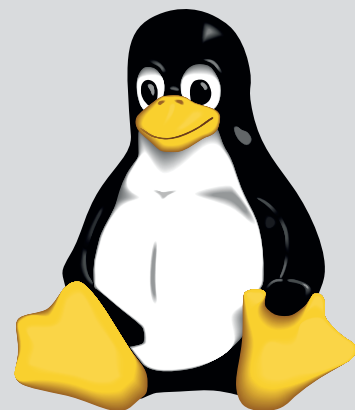
**СКАЧАЙТЕ
СВОИ ОБРАЗЫ
DVD!**

Найдите код и образы DVD
на [www.linuxformat.com/
archives](http://www.linuxformat.com/archives)

» И ЕЩЕ!

КНИЖНАЯ ПОЛКА

- **Расширенное руководство по скриптам Bash** Улучшите написание скриптов.
- **Руководство по Bash для начинающих** Осваивайте скриптописание в Bash.
- **Скрипты Bourne Shell** Начиная 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 сисадминов.
- **Нырните в Python** Всё, что вам надо знать.
- **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.
- **Справочник по GNU Emacs** Шестьсот страниц важной информации!
- **Создание программ с открытым кодом** Всё, что вам надо знать.
- **Программирование с основ** Сделайте первые шаги.



ПРОШЛЫЕ НОМЕРА » ПРОПУСТИЛИ?

LXF243

Ноябрь 2018

- **Прощай, Windows** Есть операционка дружелюбнее
- **Голливуд ваш** Ищем лучший видеоредактор
- **Децентрализуемся** Как не попасться в соцсеть
- **Android в Linux** Рабочая настройка — легко



LXFDVD

Austrumi, Endless OS, Hyperbola, Mint, Peppermint, Slax, 15 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_243/
PDF-версия: shop.linuxformat.ru/elxf_243/

LXF242

Октябрь 2018

- **Хит-парад дистрибутивов** Ищем лучший
- **Вечно юные** Само-обновляемые дистрибутивы
- **Клеточные автоматы** Стихийное поведение
- **Живи... играй!** Консоль на Raspberry Pi



LXFDVD

Antix, Bodhi, deepin, KaOS, Manjaro, Mint, Puppy, Solus, 17 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_242/
PDF-версия: shop.linuxformat.ru/elxf_242/

LXF241

Сентябрь 2018

- **Врага надо знать** Переодеваемся хакерами
- **Музыке — не гнить** Оживим вашу фонотеку
- **Бот ученый** Пусть анализирует журналы web-сервера
- **Глобализация Raspberry Pi** на Linux, Windows и macOS



LXFDVD

BunsenLabs, Feren OS 2018, GhostBSD, Kali Linux, MX Linux, 17 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_241/
PDF-версия: shop.linuxformat.ru/elxf_241/

LXF239/240

Август 2018

- **Прощайте, облака** Запустим свои
- **Резервные копии** Чтобы с вашими данными не случилось страшное
- **Плюнет, поцелует?** Машинный анализ эмоций
- **Аудио по всему дому** Дешево и сердито



LXFDVD

Devuan, Linux Mint с Cinnamon и Mate, openSUSE Leap, Q4OS, Sparky Linux LXQt, Trisquel Mini, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_239-240/
PDF-версия: shop.linuxformat.ru/elxf_239-240/

LXF238

Июль 2018

- **Сделаем сами** Рабочий стол мечты
- **Редактируем изображения** Как создать фотоскейпер
- **Децентрализуем Интернет** IPFS — межпланетная файловая система
- **Гадаем на ромашках** Raspberry Pi различает цветы!



LXFDVD

Fedora, NixOS, Pinguy, Salix, Ubuntu, Void OS, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_238/
PDF-версия: shop.linuxformat.ru/elxf_238/

LXF237

Июнь 2018

- **Бионический Бобер** Ubuntu долгосрочный
- **CMS** Кому доверить управление контентом
- **Коты, кубиты и кванты** Передний край технологий
- **Запускаем мультисайт** Удобная иерархия



LXFDVD

Linux Lite, Pinguy, Rosa, Sculpt, Slax, Ubuntu, Zorin, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_237/
PDF-версия: shop.linuxformat.ru/elxf_237/

Закажите на сайте www.linuxformat.ru

Архивные версии журнала вы найдете на www.linuxformat.ru или www.linuxcenter.ru

Телефоны: (812) 309-0686 – Санкт-Петербург, (499) 283-8606 – Москва

Linux Format ВКонтакте: vk.com/linuxform

Содержание

Страница 0

ДИСТРИБУТИВЫ

Elementary 5.0 (64-битный)
Настольный дистрибутив на базе Ubuntu, сделавший своим приоритетом простоту. Предлагает множество готовых пользовательских приложений, включая Photos, Music, Videos, Calendar, Terminal, Files и т.д., состав которых можно пополнить из репозитория Ubuntu и Debian.

Ubuntu 18.10

(64-битный)
Полноценная ОС Linux на базе Debian, для настольных ПК, свободно доступная с поддержкой сообщества и профессионалов. Сообщество Ubuntu построено на идеалах, закрепленных Манифестом Ubuntu: ПО должно быть доступно, программные инструменты должны предоставляться людям на их родном языке и невзирая на ограничения их физических возможностей, и пользователи должны иметь право настраивать и изменять свое ПО любым способом на свое усмотрение.

УЧЕБНИКИ

HTTPS
WebStorm

ГЛАВНОЕ

Grid
Metasploit

ДОКУМЕНТАЦИЯ: 17 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
Bits and bytes Открытый исходник — это не только про ПО
Bourne Shell Scripting Начальное руководство по программированию на Bash
The Cathedral and the Bazaar Классический текст Эрика Реймонда [Eric S. Raymond] «Сбор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Dive Into Python Учебник по программированию на Python
Intro to Linux Начальное руководство по Linux
Linux Dictionary Словарь Linux, объясняющий специфическую терминологию

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. И 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, упомянут 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 всё же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Новинка! Ubuntu 18.10!

Свежий релиз известнейшего в мире дистрибутива!

64-битный



Elementary 5.0

Красивейший и простейший среди дистрибутивов Linux! Пробуйте!

- Уникальный рабочий стол Pantheon
- Настраиваемые свободные программы
- Простые и безопасные обновления
- Без рекламы и шпионского ПО

64-битный

Содержание

Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Край-Хартманом [Greg Kroah-Hartman]
Producing OSS Как выпускается СПО
Programming from the Ground Up Программирование с самых основ
Python 3.7.0 Официальная документация
System Administrators Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU
Ultimate Linux Newbie Guide Руководство по Linux для новичков, но не только
Understanding Open Source And Free Software Licensing Как лицензируется открытое и свободное ПО

Сторона 2

ДИСТРИБУТИВЫ

Fedora 29 (64-битный)
 Дистрибутив Linux, разрабатываемый сообществом при поддержке проекта Fedora. Принадлежит Red Hat и содержит открытое ПО, распространяемое под свободной лицензией. Fedora имеет репутацию дистрибутива с упором на инновации, интеграцию новых технологий на ранней стадии и тесное сотрудничество с другими сообществами Linux.
Fedora 29 Netinstall (32-битный)
 Предлагает меню, где можно сначала выбрать базовую среду, а затем добавить ей функции. Базовая среда может быть конкретным рабочим столом, минимальной установкой или web-сервером. К этому вы можете добавить инструменты для разработки, игры и пр., всего за несколько щелчков создав индивидуальную установку, где загружены только необходимые пакеты.

Пожалуйста, перед использованием данного диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109.

КОММЕНТАРИЙ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru
ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru



Manjaro 18.0 KDE (64-битный)
 Дружелюбный к пользователю дистрибутив Linux на базе независимо разработанной самообновляемой ОС Arch предоставляет все преимущества Arch и уделяет особое внимание удобству и доступности. Подходит и новичкам, и опытным пользователям.

УЧЕБНИКИ

Sudoku

Все дистрибутивы представлены ISO-образами, который можно записать на одноплатный носитель, и загрузить в live-режиме прямо с LXFDVD. У всех присутствует возможность установки на жесткий диск.

Настоящий диск тщательно протестирован и превратился на всех стадиях производства, однако, как и в случае с любым другим ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу загрузки с нашего единственного сайта.
 Тираж изданий ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИДТР ВАО N 77-03.

» СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке — например, 0,3,0. Теперь вы можете с легкостью записать образ на диск: *cdrecord dev=0,3,0 -v /путь к образу/image.iso*

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле */etc/default/cdrecord*. Добавьте по одной строке для каждого устройства записи (скорее всего, в вашей системе присутствует только одно такое устройство): *Plextor=0,3,0 12 16M*

Первое слово в этой строке — метка; затем после адреса SCSI-устройства вы должны указать скорость и объем буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Всё будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Для записи ISO-образа вам осталось набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку *Burn* и *ISO 9660 Image* в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на *Combust*! Пока ваш образ прожигается на диск, можете выпить чашечку кофе.

ДРУГАЯ ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто перенесут 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 сайт разработчика дистрибутива.

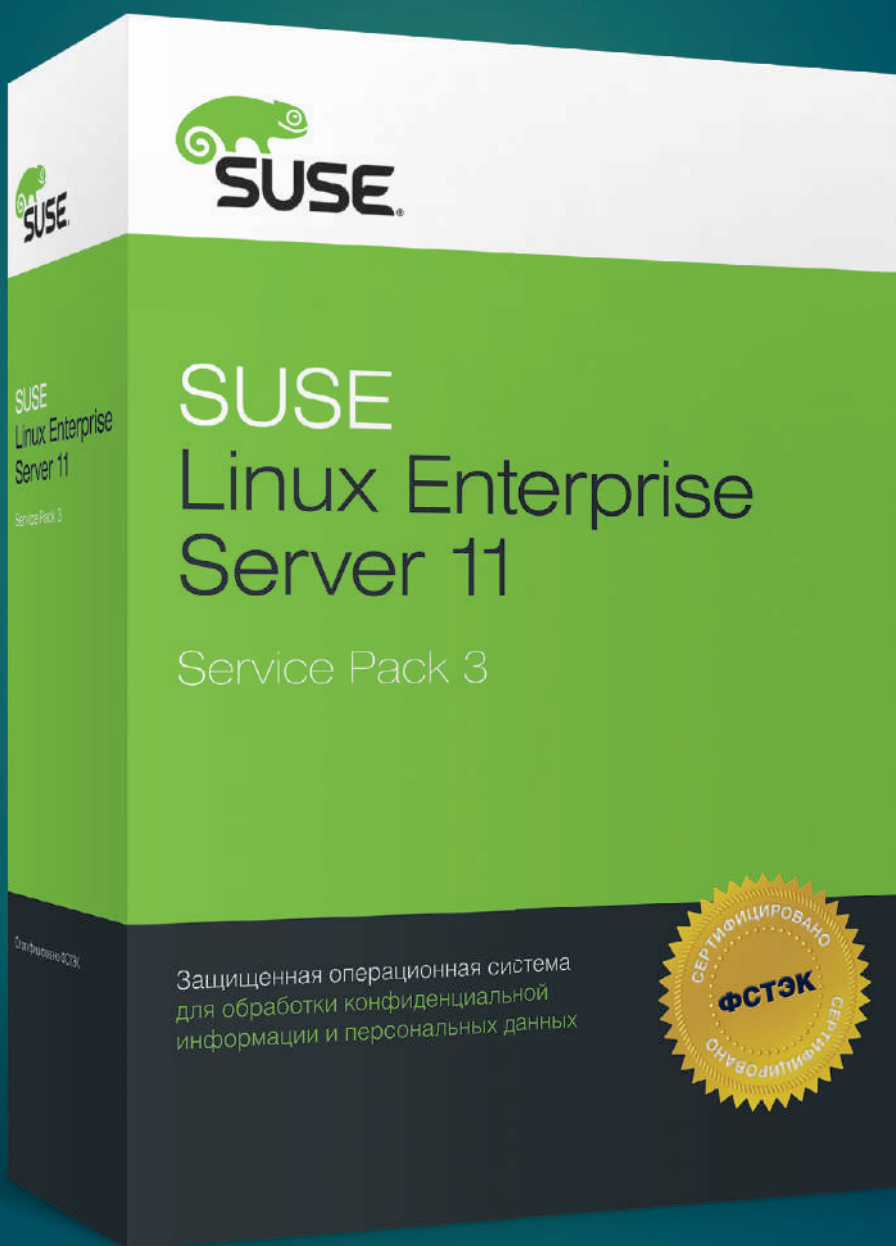




КОМПАНИЯ «ГНУ/ЛИНУКСЦЕНТР» — ОФИЦИАЛЬНЫЙ
ПОСТАВЩИК СЕРТИФИЦИРОВАННОГО ФСТЭК
ДИСТРИБУТИВА SUSE

SUSE Linux Enterprise Server 11

SUSE Linux Enterprise Server 11 Service Pack 3 — высоконадежная, масштабируемая и безопасная серверная операционная система для предоставления IT-услуг корпоративного уровня в физической и виртуальной инфраструктурах. Операционная система создана для работы в смешанных окружениях и предлагает лучшую в своем классе производительность.



Сертифицировано ФСТЭК

Сертифицированная ФСТЭК операционная система SUSE Linux Enterprise Server 11 Service Pack 3 является программным средством защиты от несанкционированного доступа к информации, не содержащей сведения, составляющие государственную тайну, и соответствует требованиям руководящих документов «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации» (Гостехкомиссия России, 1992) — по 5 классу защищенности и «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недеklarированных возможностей» (Гостехкомиссия России, 1999) — по 4 уровню контроля.

Операционная система разработана в соответствии с ТУ 501110-007-82487552-14.

Москва
+7 (499)

283-86-06

Санкт-Петербург
+7 (812)

309-06-86

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





Спасибо, что были с нами!

Уважаемые читатели *Linux Format*!

Прекратить выпуск печатного издания журнала с 13-летней историей в России — это сложное решение, которое далось мне очень трудно. Основная причина такого решения — значительное снижение спроса на печатные журналы как со стороны читателей, так и со стороны рекламодателей, а также бурный рост распространения пиратских копий электронной версии.

Мы осознали, что российское Linux-сообщество в основной массе не хочет и не будет платить за контент, а в *Linux Format* контент обходится дорого. Поэтому мы будем искать новые формы продвижения идей Свободного Программного и Аппаратного Обеспечения в Интернете под свободными лицензиями. Архивы *LinuxFormat* мы постараемся сохранить для сообщества — возможно, они кому-то пригодятся.

Благодарю всех, кто читал *LinuxFormat* и помогал в его продвижении!

Простите, что не справился с управлением этим проектом.

С уважением,
Павел Фролов,
издатель *LinuxFormat* в России

LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 1000 экз., распространение электронной версии 30 000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Галина Итальянская, Даниил Кривошеин, Светлана Кривошеина, Валерий Смирнов, Елена Толстякова, Ирина Шулакова

Редактор диска

Григорий Огилько

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 «Типографский комплекс „Девиз“» 195027, Санкт-Петербург, Якорная ул., 10, корп. 2, литер А, пом. 44

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор (Neil Mohr) neil.mohr@futurenet.com

Мистер Робот Джонни Бидвелл (Jonni Bidwell)

jonni.bidwell@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

(Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Клифф Хоуп (Cliff Hope)

clifford.hope@futurenet.com

Директор редакции Грэм Барлоу (Graham Barlow)

Старший художественный редактор Джо Гулливер (Joe Gulliver) (Jo Gulliver)

ПОДГОТОВКА МАТЕРИАЛОВ

Матс-Тэге Аксельссон (Mats Tage Axelsson), Майк Бедфорд (Mike Bedford), Джонни Бидвелл (Jonni Bidwell), Нейл Ботвик (Neil Bothwick), Фрейзер Браун (Fraser Brown), Нейт Дрейк (Nate Drake), Кент Ельчук (Kent Elchuk), Линдси Хандмер (Lindsay Handmer), Джон Найт (John Knight), Джереми Лэйд (Jeremy Laird), Джон Лэйн (John Lane), Аарон Питерс (Aaron Peters), Лес Паундер (Les Pounder), Маянк Шарма (Mayank Sharma), Шашанк Шарма (Shashank Sharma), Валентин Синицын (Valentine Sinityn), Крис Торнетт (Chris Thornett), Александр Толстой (Alexander Tolstoy), Михалис Цукалос (Mihalis Tsoukalos), Филиппа Уорр (Philippa Warr), Евгений Балдин, Андрей Гондаренков, Алексей Федорчук

Иллюстрации Шейн Коллиндж (Shane Collinge)

Иллюстрация на обложке 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 — Санкт-Петербург, +7 (499) 283-8606 — Москва

Электронная почта: info@linuxformat.ru

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания *Linux Format*, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция *Linux Format* получает исключительное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. *Linux Format* стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция *Linux Format* не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция *Linux Format* не несет ответственности за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса (Linus Torvalds). "GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт 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

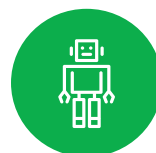
- Праздничные мастер-классы
- Городской лагерь на время каникул
- Курсы повышения квалификации **для педагогов**



Инновационная
программа обучения

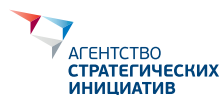


Собственная
методическая база



Российские технологии
и оборудование

ВООБРАЖАЙ! ПРОГРАММИРУЙ! СОЗДАВАЙ!



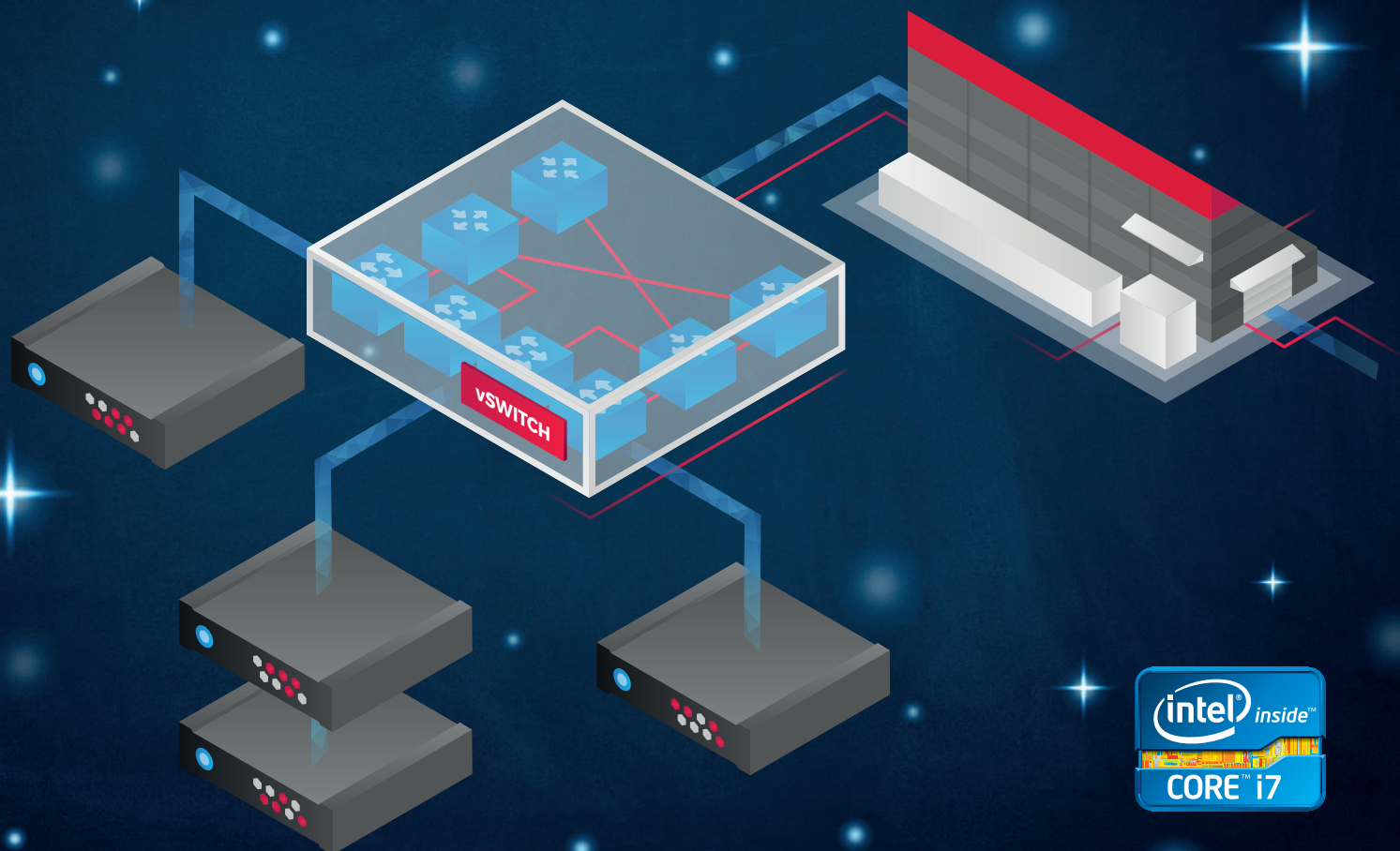
www.robboclub.ru

8 800 777 2985

vk.com/robboclub ● info@robboclub.ru

vSWITCH

ВИРТУАЛЬНЫЙ КОММУТАТОР
БЕЗ ДОПОЛНИТЕЛЬНОЙ ОПЛАТЫ



Выделенный сервер SX61

- ✓ Intel® Core™ i7-3770 Quad-Core
- ✓ 32 ГБ DDR3 RAM
- ✓ 4 x 6 TB SATA 3 Gb/s 7200 rpm
- ✓ 30 ТБ трафик*
- ✓ Без минимального контракта
- ✓ Установка 5400 рублей

5400 рублей в месяц

* Нет платы за превышение. При превышении 30 ТБ/месяц скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 80 руб. за каждый дополнительный ТБ.

vSwitch – Построй свою виртуальную сеть!

vSwitch использует технологию VLAN для объединения нескольких выделенных серверов находящихся в разных дата-центр парках Hetzner Online. Настройки vSwitch производятся через удобный веб-интерфейс панели Robot. Например, можно настроить частную сеть для безопасного обмена данными или заказать публичную подсеть и сделать систему отказоустойчивой.

ru.hetzner.com

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.