

DVD ДАРОМ!  НОВЫЙ FEDORA 25

LINUX FORMAT

Главное в мире Linux

Февраль 2017 № 2 (219/220)

ВАШ ИДЕАЛ LINUX-МАШИНЫ

Переходите на Linux, и ваш ПК снова взорлит!

- » Время покидать Windows
- » Берите свободное ПО
- » Открывайте лучшее


ЕСТЬ
**ЭЛЕКТРОННАЯ
ВЕРСИЯ**
ДЛЯ ВАШЕГО
ПЛАНШЕТА!



Mozilla-фест

« Молодежь — в самом сердце технологии, это лидеры будущего, его хранители и устроители »

Дорин Файлз о празднике Open Source с. 36

Быстрейший Linux-ПК!

Сочетайте Linux и SSD для супер-скорости



Графика

Переход на Wayland

» Нырнем в графику нового поколения — прощай, X11



Сравнение

Быстрые и легкие

» Дистрибутивы-легковесы: ОЗУ меньше, скорость больше



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

Linux center
www.linuxcenter.ru



iTeleRadio

ИНТЕРАКТИВНОЕ РАДИО&ТЕЛЕВИДЕНИЕ



Что мы делаем

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

Кто мы

На сей раз мы спросили наших экспертов: Как затасчить в Linux новичка?



Джонни Бидвелл

Как только они завдыхают, что очередное обновление Windows снова сочло подходящим моментом встрять со злополучной серией «ожидайте-перезагружайте», надо как бы невзначай упомянуть, что не все ОС такие. И показать простейшие операции в терминале, чтобы они могли притворяться крутым хакером из *Mr. Robot*.



Нейл Ботвик

Посоветовать им забыть, как всё делается в Windows. Главная причина проблем новичков в Linux — то, что они пытаются применять методы Windows. Linux — это не «то же самое, что Windows, только лучше и бесплатно»; Linux просто лучше (и свободный). Если это не поможет, прокрадитесь к ним в дом, когда они спят, и поставьте на всех их машинах Linux.



Нейт Дрейк

Сперва я говорю людям, у которых смартфон на Android, что Linux у них уже есть, хотя они об этом и не подозревают! Потом иду на YouTube и показываю им особо яркое видео на тему Мощества Linux с огненным курсором, вроде <https://youtu.be/rw82sepw50E>, чтобы вышибить их из состояния самодовольства.



Ник Пирс

Большинство моих знакомых новичков в Linux хотели бы чередовать работу с Windows, так что я им советую параллельно с Windows установить Linux Mint, с двойной загрузкой. Рабочий стол Cinnamon в Linux Mint больше всех напоминает Windows, и это облегчает переход.



Лес 'Pi' Паундер

Не хотелось бы звучать, как заезженная пластинка, но идеальное введение в Linux — это Raspberry Pi. Этот дешевый компьютер столько всего умеет! А под капотом мы видим отличную машину с Linux, способную стать web-сервером, сервером печати или файлов, или тонким клиентом для обширной сети.



Валентин Синицын

Самое правильное — подsunуть ему Linux еще до того, как он увидит Windows. Если этот момент упущен, ненавязчиво показывайте ему на WSL, чтобы он потихоньку привыкал, а затем вдруг наемкните, что ASP.NET уже давно открытое ПО, так что в Windows ему делать и нечего.



Мюнхенский сговор

» Весь XX век столица Баварии попадала в какие-то политические истории — от «пивного путча» и «мюнхенского сговора» до теракта на Олимпиаде. В веке нынешнем город прославился тем, что одним из первых в Европе перевел все свои службы на свободное ПО, для чего даже был создан дистрибутив под названием LiMux (Linux для München). В 2013 г. муниципалитет объявил об успешном завершении миграции.

Только вот неделю назад городской совет одобрил решение снова вернуться на ОС и офисный пакет от Microsoft. По сообщениям прессы, инициатива исходила от мэра и была поддержана депутатами от правящей партии. Цена вопроса — примерно сотня миллионов евро (из кармана налогоплательщиков). При этом совсем не очевидно, что городское ИТ от того значительно улучшится. Зато каждый достигает своей персональной цели: менеджеры по продажам из Microsoft могут доложить о «сделке века» и получить бонусы, а городские чиновники могут доказать начальству, что не зря просиживают кресла в своих кабинетах, и заработать благодарность в приказе. Что-то сделали... только непонятно, зачем. Но как красиво доложили! Бюрократическое рвение в очередной раз победило здравый смысл. К сожалению, такое случается всё чаще, и не только в Баварии. Тенденция, однако...

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Сайт: www.linuxformat.ru

Группа «ВКонтакте»: vk.com/linuxform

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

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

Знаете всё о Linux?

ДА

НЕТ

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

Зарегистрируйтесь
на сайте
shop.linuxformat.ru



СОМНЕНИЯ

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



Версия для iPad
и iPhone доступна
в App Store

Выберите вид подписки

PDF-версия
на 6 месяцев

990 ₺

PDF-версия
на 12 месяцев

1800 ₺

Печатная версия
на 6 месяцев

1890 ₺

Печатная версия
на 12 месяцев

3480 ₺

Выберите вид доставки

Оплатите

Читайте Linux Format!

Станьте Linux-гуру

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

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

Печатная версия Linux Format понравится читателям, которые:

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

Способы доставки

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

Способы оплаты

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



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

Содержание

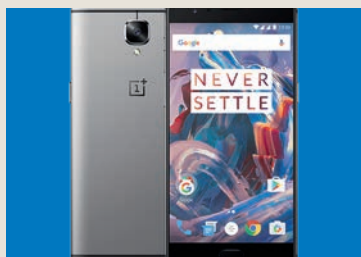
«Все хотят изменить мир, но никто не хочет измениться сам.» Лев Толстой

Обзоры

Bodhi Linux 4.0 12
Ubuntu LTS и рабочий стол Elementary впервые вместе! А может, и нет, но новый релиз дистрибутива, который чуть было не умер, на сей раз превзошел сам себя.

Zorin OS 12 Core 13
Не назван ли этот дистрибутив в честь вымышленного архизлодея из фильма *Вид на убийство*? Во всяком случае, он красив, как тот киноактер.

Один Плюс Три 14
Nexus умер, Pixel переоценили — сумеет ли OnePlus3 стать телефоном для хакеров свободного кода или это так, красивая штука?



➤ **Могучий убийца флагманов, чисто случайно более открытый, чем многие.**

Jolla C Sailfish 15
Рассматриваем новый открытый смартфон Jolla C и копаемся в Sailfish OS, чтобы понять, способен ли он потягаться с большими пацанами.

GamePad Digital 16
Эмулятор мини-геймпада совершенно в стиле Nintendo DS [Ред.: — Это, часом, не оскорбление?], но при всей гибкости открытого исходного кода Android.

Mad Max 17
Вашему компьютеру хватит мощности, чтобы поиграть в *Mad Max*? Вряд ли, но мы уже стараемся угнездить в Linux самую свежую AAA-игру.



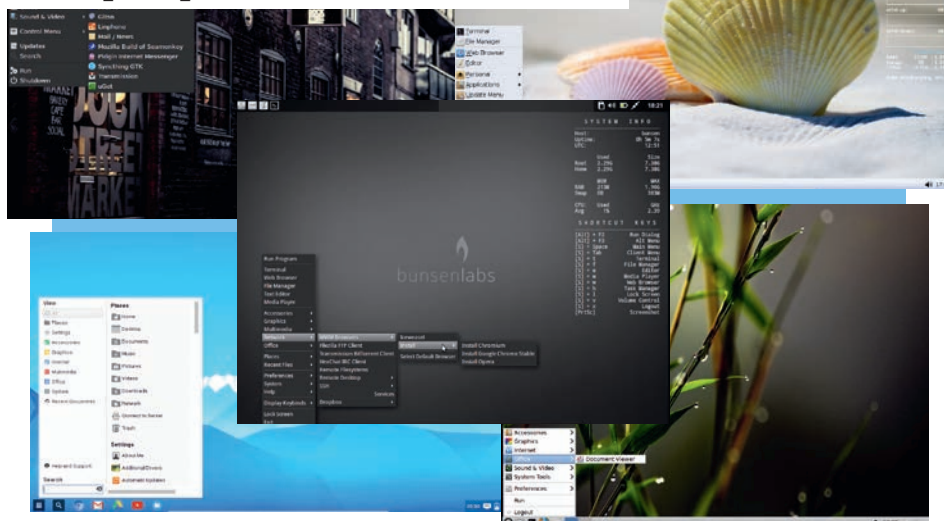
➤ **Макс безумен? Да не настолько, чтобы купить электромобиль.**

Ваша идеальная машина с Linux

Под нашим чутким руководством переходите на Linux — это лучшая в мире ОС! А руководство см. на с. 28



Сравнение: Дистрибутивы-легковесы с. 22



Интервью



« Не важно, пять лет участникам или восемьдесят — важно, что у них в голове »

Наш репортаж с MozFest-2016 с. 36

На вашем бесплатном DVD

LIVE-ДИСК: ЗАГРУЗИ > ОПРОБУЙ > УСТАНОВИ!

FEDORA 25
Обновится до самой свежей и популярной ОС Linux

openSUSE 42.2
Настоящая система класса предприятия

Bodhi 4.0
Красивая оболочка на базе Ubuntu

antiX 16
Быстрый, легковесный и идеальный для старых ПК

LINUX LIVE-ДИСК: ГОТОВ К РАБОТЕ
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

Fedora 25 Workstation 64-битный
Bodhi Linux 4.0 64-битный
antiX 16 32-битный, **Manjaro 16** 32-битный
openSUSE Leap 42.2 64-битный

» Каждый месяц — только лучшие дистрибутивы
ПЛЮС: HotPicks, Сравнение, и не только! **с. 106**

Побалуйте себя и любимых подпиской на LXF!

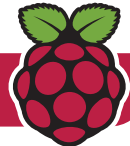


Доступно в AppStore!



www.linuxformat.ru/subscribe

Пользователям Raspberry Pi



- Новости Pi** 88
Pi догоняет Commodore x86 и становится корпоративным со SLES, а также превращается в точки доступа в Африке.
- Хаб Ethernet и USB** 89
Сообщество потрудились над созданием платы для Pi Zero. Жалеть на нее денег или нет?
- Whiptail** 90
«Хлыстохвост» Whiptail — система меню, способная украсить ваши приключения в Raspberry Pi.
- Соберем TalkiPi** 92
Отличный проект: смастерим свою собственную рацию «воки-токи» в ретро-стиле на нескольких Pi и сервере Mumble.



Ищите в номере

- Архитектура Tizen** 20
Чтобы грамотно программировать под любую ОС, следует вникнуть в ее строение. Чем и займемся.
- Самый быстрый ПК** 40
Весь чайный бюджет LXF на 2017 г. ушел на взятки коллегам из Maximim PC, которых мы попросили собрать максимально быстрый компьютер.
- Эти упрямые финны** 44
Jolla идет своим путем, не пасуя перед трудностями...

Академия кодига

- Python SQLite 3** 80
Нырнем с утеса в бесконечную пучину управления базы данных с невероятно полезным Python SQLite 3 — правда-правда полезным!
- Обработка текста в Python** 84
«Слова, слова, слова!» — заметил философический принц Датский. Разберемся, как в Python обрабатывать текст и всячески им манипулировать.

```

msos@msos:~$ python
Python 2.7.9 (default, Mar 1 2015, 12:57:24)
[[GCC 4.9.2] on linux2
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>> line = "12/13/1968 01/02/2000"
>>> print re.sub(r'(\d{2}/\d{2}/\d{4})', r'12-13-11', line)
12-13-1968 01/02/2000
File "stdin", line 1, in <module>
NameError: name 're' is not defined
>>> import re
File "stdin", line 1, in <module>
NameError: no module named re
>>> print re.sub(r'(\d{2}/\d{2}/\d{4})', r'12-13-11', line)
12-13-1968-12-02-2000-01
>>> import re
File "stdin", line 1, in <module>
NameError: no module named re
>>> bool(re.search('12[345]', '123'))
True
>>> bool(re.search('12[345]', '123'))
True
  
```

Постоянные рубрики

- Новости** 6
Asus выпустила ПК-одноплатник, выходило МФУ «Катюша», Raspberry Pi продвигает математику, в Питере создана ОС реального времени, выпущены ноутбук с KDE и высокопроизводительный сервер на Ubuntu, HP отзывает опасные батареи, а Olimex предлагает собирать ноутбуки самим.
- Вести мобильных ОС** 18
Xiaomi осиротела, Acer и Asus борются за студентов, Intel братается с Google, а Lenovo строит трансформер.
- Сравнение** 22
Мы только что собрали самый быстрый компьютер, а значит, надо немедленно протестировать самые легкие дистрибутивы! antiX, Bunsen Labs, Cub Linux, LXLE, WattOS.
- Интервью LXF** 36
Праздник на улице Mozilla привлек участников всех возрастов, но упор сделал на молодое поколение: ведь это им жить в будущем, которое готовится сейчас.
- Рубрика сисадмина** 46
М-р Джилион Браун пытается уместить в мозг факт вступления Microsoft в Linux Foundation и находит утешение в уютном интерьере распределенной GlusterFS.
- Ответы** 96
ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик — про переход с Mac OS и Windows, отказ одного диска монтироваться, медленную систему, перехват текста с консоли, Windows и Mint вместе, создание отдельного раздела /home.
- HotPicks** 100
Отдавайте горяченького! Лучшие в мире новинки свободного ПО: DevilsPie2, Digikam, Featherpad, Freeciv, Homebank, Krusader, Kup, Mouse Boat, Neural Enhance, Spek, Tuxguitar.
- Диск Linux Format** 106
Содержимое двустороннего DVD этого месяца.
- Пропустили номер?** ... 108
Пока это не превратилось в проблему, поспешите заказать. У вас остались кое-какие шансы.
- Через месяц** 112
Мы всегда предупреждаем, чтобы вы ни в коем случае не отдавали свои драгоценные данные в чужие руки. А теперь на деле покажем, как об этом позаботиться.

Учебники

- Основы терминала**
Экосистема переменных 52
Выучимся импортировать, экспортировать и задавать переменные окружения.
- Основы терминала**
Альтернативные оболочки ... 54
Ubuntu Terminal на свете отнюдь не один...
- Безопасность**
Сканирование с Nmap 56
Как прокрасться внутрь беззащитной ОС.
- Файловые системы**
Snapper 58
Систему моментальных снимков openSUSE можно заставить работать для всех.
- Графика**
Внутри Wayland 60
Добудем Wayland и внедрим его в любой дистрибутив: немножко компиляции и ведра чая.
- Серверы**
Польза от Cockpit 64
Cockpit задуман для управления серверами — системных администраторов это обрадует.
- Контроль версий**
Освоим Git 68
Исследуем одну из популярнейших систем контроля версий, чтобы употребить ее в быту.
- Интернет вещей**
Сообщение, дубль два 70
Для надежности добавим к IoT еще один канал связи; мобильные операторы нам в помощь.
- Умные дома**
Домом правит Linux 74
Теперь это безопасно! Идем дальше и объединяем все свои умные устройства.
- CRUI**
Тестирование 78
«Проверки на дорогах» — в разработке без подсистемы тестирования никуда.

В ЭТОМ НОМЕРЕ: ПК на плате » Катюша-многостаночница » Счетная Малинка » Наша ОС » KDE на ноутбуке » Сервер с Ubuntu » На всякий пожарный » Ноутбук своими руками

ОДНОПЛАТНЫЕ КОМПЬЮТЕРЫ

И ты, Asus?!

Tinker Board стал новым конкурентом Raspberry Pi.

В сегменте одноплатных компьютеров «размером с кредитную карту» свой вариант представила компания Asus. Спецификации новинки, Tinker Board, впечатляют: она способна обрабатывать видео 4K (включая декодирование H.265, благодаря наличию GPU Mali-T764 в составе ее 4-ядерного SoC Rockchip RK3288 1,8 ГГц) и аудио 192 кГц/24 бит, т.е. может стать основой для создания НTPС; оснащена 2 ГБ ОЗУ, Bluetooth 4.0, четырьмя портами USB 2.0, выходом HDMI, слотом microSD, Gigabit Ethernet, Wi-Fi 802.11b/g/n, и комплектуется 40-контактным внутренним разъемом с 28-ю контактами GPIO, портом CSI для камеры и портом DSI. Для сравнения, компонентами Raspberry Pi 3 являются процессор

ARM Cortex-A53 1,2 ГГц, 1 ГБ ОЗУ и HDMI 1.3. Tinker Board работает под управлением новой Linux OS от Asus, но возможность установки Ubuntu, openSUSE и *Kodi*, как обещают создатели, «на подходе».

А недостатки? Даже при поддержке видео 4K, позволяющей смотреть любой скачанный/созданный контент этого формата, стриминг Netflix пока недоступен из-за отсутствия аппаратных и программных средств для DRM-декодирования потока 4K Netflix (для этого придется задействовать устройства типа Chromecast Ultra или Nvidia Shield). И цена: Tinker Board предлагается за £55 (\$68), а Pi 3 — за £33 (\$35). Плюсы Pi 3 — наличие сообщества и поддержка со стороны Raspberry Pi Foundation.



» Поддержка видео 4K и 24-битного аудио делает ASUS Tinker Board едва ли не лучшим вариантом для медиа-центра DIY.



» Рубрику готовил
**АНДРЕЙ
ГОНДАРЕНКОВ**

ИМПОРТОЗАМЕЩЕНИЕ

Выходила на берег... Катюша

Компания ИТ Системы планирует выпуск серии многофункциональных устройств.

В России создали первый отечественный принтер-сканер-копир-факс «Катюша» с уникальной технологией полностраничной высокоэнергетической капиллярной печати (ПВК). Эта новейшая платформа реализует принцип неподвижной матрицы из печатных головок, что сводит к минимуму шум при печати при сверхвысокой скорости. Российские химики разработали пигментные чернила на полимерной основе, которые мгновенно закрепляются в толще листа и сохраняют отпечаток навсегда. Лист в выходном лотке абсолютно сухой. Благодаря простоте и надежности этих инженерных решений, расходы на сервис стремятся к нулю.

Отечественное ПО гарантирует конфиденциальность рабочих процессов. Модульный принцип сборки позволяет конфигурировать принтер для нужд различных заказчиков. Применение в производстве отечественных технологий и материалов дает возможность отказаться от закупки импортной печатной техники и позволяет предприятиям формировать прогнозируемые ИТ-бюджеты в российских рублях.

Основные параметры МФУ «Катюша»: черно-белый/полноцвет; ВК-печать со скоростью до 75 страниц/мин., обеспечивающая качество до 2400×2400 dpi; высокоемкие картриджи с ресурсом до 50 тыс. страниц; сенсорный дисплей управления.

» МФУ «Катюша» — пример успешной реализации научно-технического и производственного потенциала России в области электроники и информационных технологий.



Источник: www.computerworld.com; katyusha-print.ru

МИНИ-КОМПЬЮТЕРЫ

Новый формат «Малинки»

В продажу поступила плата Raspberry Pi Compute Module 3.

Новый одноплатный компьютер Raspberry Pi стал меньше и дешевле предшественника, но его создатели не ждут такого наплыва покупателей, как у предыдущих моделей. По мнению основателя Raspberry Pi Foundation Эбена Аптона [Eben Upton], производители, использующие в своих продуктах Compute Module, перейдут на Compute Module 3, продажа которого в первый год планируется на уровне нескольких сотен тысяч (далеко от миллионов Raspberry Pi 3). Compute Module 3 оснащен тем же 4-ядерным 64-битным процессором Broadcom BCM2837 и 1 ГБ ОЗУ, что и Raspberry Pi 3, но вдвое меньше размером (67,6×30 мм) и не имеет разъемов для Ethernet, USB, SD Card и дисплея; нет и поддержки Wi-Fi. Вместо отсутствующих разъемов применен краевой коннектор, соответствующий сокету SODIMM, используемый в ноутбуках для установки дополнительной памяти. Через слот SODIMM Compute Module 3 также получает питание от основной платы, в которой установлен.

По мнению Аптона, интересным проектом являются широкоформатные дисплеи NEC с возможностью подключения Compute Module, что добавляет «интеллекта» пассивному дисплею. Системы Smart signage для демонстрации рекламы компания NEC представила в октябре 2016 г., задолго до того, как Compute Module 3 поступил в продажу (16 января 2017 г.); но поскольку новый модуль pin-совместим с оригиналом Compute Module, выпущенным в 2014 г., затруднений процесс модернизации дисплеев не вызовет.

Compute Module 3 в 10–12 раз мощнее предшественника, но отличается повышенным энергопотреблением (до 4 Вт) при пиковом быстродействии. Даже когда мультимедиа-ядро отключено, а процессор ARM находится в ждущем режиме, плата все равно потребляет около 100 мВт: Compute Module 3 явно не предназначен для устройств, работающих от батарей.

Глобальный менеджер по управлению продуктами компании RS Components



От варианта Compute Module 2014 г. новинка отличается удвоенным ОЗУ и увеличенной в 10 раз вычислительной мощностью.

(дистрибьютора Raspberry Pi) Роберт Мэйкрофт считает, что Compute Module 3 положит начало новой тенденции в мире промышленных компьютеров: число клиентов, способных изготовить сложную конструкцию с нуля, сокращается, поэтому модульность становится нормой.

В США цена модуля \$30; модификация Lite без встроенной флеш-памяти предлагается за \$25.

Чтобы упростить клиентам процесс проектирования, Raspberry Pi Foundation по открытой лицензии предлагает собственную конструкцию платы — Compute Module IO Board.

ЗНАЙ НАШИХ

Для IT-оборудования и IoT

Компания из Санкт-Петербурга создала собственную ОС реального времени.

Основанная в 1991 г. и более 25 лет занимающаяся разработкой и поставкой ПО компания АстроСофт объявила о создании новой отечественной операционной системы реального времени — ОСРВ МАКС (Операционная Система Реального Времени для МультиАгентных Когерентных Систем).

ОСРВ МАКС — универсальная платформа, которая позволяет значительно ускорить разработку встраиваемого ПО при создании новых устройств на основе микроконтроллеров, в том числе устройств Интернета вещей, и встроенных систем. Новая ОСРВ не только реализует всю классическую функциональность ОС данного типа, но и обладает уникальными возможностями по организации взаимодействия множества устройств, позволяющих заметно упростить создание традиционно непростых, но необходимых во встраиваемых системах механизмов

(резервирование, горячая замена оборудования и др.). Данный продукт — целиком российская разработка, без сторонних решений и компонентов. В данный момент идут работы по сертификации ОСРВ в Федеральной службе по техническому и экспортному контролю (ФСТЭК). Параллельно ведутся исследования в области Mesh, IoT, которые привнесут в систему дополнительные возможности, особенно востребованные распределенными системами; среди них — отказоустойчивое сетевое взаимодействие, самоорганизующаяся структура сети, автономность устройств (низкое энергопотребление, беспроводная связь).

ОСРВ МАКС официально поддерживает оборудование, производимое АО «ПКК Миландр» (это один из ведущих российских разработчиков интегральных микросхем различного функционального назначения — микропроцессоров, микроконтроллеров, схем памяти и пр.), а также



ОСРВ МАКС воплощает классическую функциональность зарубежных ОС данного типа и обладает рядом преимуществ, ускоряющих разработку встраиваемого ПО при создании новых устройств на основе микроконтроллеров.

наиболее распространенные зарубежные платформы (целевые — ARM Cortex M0, M3/M4 (+FPU); планируемые — ARM Cortex M0+/M1, Analog Devices TigerSHARC, Xilinx MicroBlaze и др.).

АстроСофт предлагает свой продукт для использования в качестве готового решения в основе ПО для встраиваемых устройств, к которым применяются требования реального времени.

НОУТБУКИ

Нашего полку прибыло!

Сообщество KDE и ритейлер Slimbook выпустили ноутбук для фанатов KDE.

Сообщество разработчиков KDE совместно с компанией Slimbook (Испания) представили ноутбук KDE Slimbook с предустановленной ОС на основе сборки KDE neon. Начальная цена модели с процессором Intel i5 2,3 ГГц — \$779, модели с процессором Intel i7 2,5 ГГц — \$907. И хотя спецификации Slimbook'a не самые выдающиеся, ключевым аргументом в его пользу является предварительное тестирование разработчиками KDE предлагаемого для этого ноутбука ПО, что гарантирует высокую стабильность и совместимость с оборудованием. Ноутбук поставляется в серебристом алюминиевом корпусе, оснащается клавиатурой с подсветкой, 4 ГБ ОЗУ, SSD-накопителем 120 ГБ, двумя портами USB 3.0, портом mini HDMI, кард-ридером SD и MMC, аккумулятором 6800 мА·ч, дисплеем full HD 13,3-дюйма с разрешением 1920×1080 пикс. Комплектации обеих моделей предусматривают возможность

увеличения ОЗУ до 8 или 16 ГБ, а SSD — до 250 или 500 ГБ. Раскладка клавиатуры — английская, испанская, немецкая или французская. На сайте проекта (kde.slimbook.es) принимаются предзаказы, которые будут выполняться после 15 марта.

Slimbook под брендом KDE являет кульминацию усилий сообщества по решению проблем аппаратного уровня, над которым «мы имели нулевой контроль», — отметил один из разработчиков, Томас Пфайфер [Thomas Pfeiffer]. KDE neon — это не дистрибутив KDE как таковой, а проект KDE, который содержит последнее ПО KDE на основе Ubuntu 16.04 LTS. Проект KDE neon присоединился к сообществу KDE в прошлом году, а в конце того же года испанский ритейлер Slimbook выступил с предложением о создании фирменных ноутбуков KDE с предустановленными рабочим столом Plasma и приложениями KDE Applications. В настоящее время Slimbook доступен только с KDE



► KDE Slimbook: внешность MacBook Air, современная аппаратная начинка и непривычная для многих операционная система.

neon, но разработчики «открыты для предложений от любого другого дистрибутива, который поставляется с предварительно установленной Plasma».

ПРЕДСТАВЛЯЕТ SYSTEM76

Дорого и... производительно!

Ibex Pro объединяет мощь CPU Intel Xeon и GPU Nvidia Tesla.

Компания System76 представила Ibex Pro — высокопроизводительный сервер под управлением Ubuntu Server 16.10, с новейшими 22-ядерными процессорами Intel Xeon E5 v4 и GPU Nvidia Tesla. Начальная комплектация Ibex Pro предлагается за \$9575 и включает 6-ядерный процессор Intel Xeon E5-2603v4, 16 ГБ ОЗУ, GPU Tesla K40 и хранилище 250 ГБ. Система в полной комплектации предлагается за \$114725 и поставляется с двумя 22-ядерными процессорами Xeon E5-2699v4, 1,5 ТБ DDR4 DRAM, семью GPU Tesla P100, хранилищем NVMe SSD на 2 ТБ и дополнительным хранилищем SSD на 32 ТБ.

В максимальной комплектации у новинки то же количество и тот же тип GPU, что и у применяемого для глубинного обучения суперкомпьютера Nvidia DGX-1 (для этого рынка System76 позиционирует и свой rack-сервер Ibex Pro). По сравнению с DGX-1, количество CPU у сервера меньше, но они более нового типа.

Ibex Pro в своих GPU Tesla P100 не задействует возможности высокопроизводительной шины NVLink; эта функциональность пока зарезервирована только для DGX-1 и серверов Power8 (технология Nvidia NVLink обеспечивает значительно более высокоскоростное соединение, чем стандарт PCI-Express 3.0, используемый Ibex Pro). В любом случае, GPU Nvidia Tesla делают Ibex Pro значительно мощнее, чем системы с двухпроцессорными конфигурациями, но они же и влияют на его высокую цену.

Для работы с GPU и компонентами Ibex Pro в System76 протестировали Ubuntu, но возможна поставка сервера с другими дистрибутивами Linux (в частности, Red Hat) — в этом случае клиенту придется взаимодействовать с компанией, чтобы настроить все компоненты ОС для работы с аппаратными средствами. System76 известна как производитель ПК с Ubuntu, среди которых Ibex Pro — одна из немногих моделей серверов.



► Оснащенный графическими процессорами Nvidia Tesla «стоечный» сервер Ibex Pro будет отличным выбором при работе с мультимедиа или приложениями компьютерного зрения.

Новинка ориентирована на те компании, которые реализуют машинное обучение для таких задач, как распознавание речи и изображений. Например, Google и Facebook задействуют высококачественные графические процессоры для анализа и классификации изображений с последующей идентификацией объектов на снимке.

ПОЖАРОБЕЗОПАСНОСТЬ

Пироманам — нет!

HP отзывает еще 100 тыс. склонных к самовоспламенению аккумуляторов для ноутбуков.

История с самовозгорающимися аккумуляторами смартфонов Samsung Galaxy Note 7 заставила производителей уделить больше внимания безопасности средств электропитания своих устройств. Так, компания HP расширила программу по отзыву аккумуляторов, имеющих проблемы с перегревом, что может привести к повреждению устройства и даже пожару: дополнительно отзывается 101 тыс. батарей, реализованных в некоторых моделях ноутбуков между мартом 2013 г. и октябрем 2016 г. Напомним, в июне 2016 г. HP уже отозвала 41 тыс. аккумуляторов. Речь идет о батареях, используемых в моделях HP, Compaq, ProBook, Envy, Compaq Presario и Pavilion, а также поставленных отдельно. Реализация устройств велась через крупные розничные сети

и онлайн, бóльшая их часть продана в США, около 3 тыс. в Канаде и 4 тыс. в Мексике.

Под действие программы отзыва попадают батареи черного цвета, имеющие коды 6BZLU, 6CGFK, 6CGFQ, 6CZMB, 6DEMA, 6DEMH, 6DGAL или 6EBVA, напечатанные на обратной стороне батареи. Проверить, попадает ли батарея в число отзываемых, можно с помощью утилиты, предлагаемой на сайте HP (<https://h30686.www3.hp.com/>).

Пока в HP поступило одно сообщение о воспламенении ноутбука в Канаде и восемь сообщений из США о перегреве, возгорании или плавлении батарей. В одном случае, по данным Комиссии по безопасности потребительских товаров США, «перегрев, плавление и обугливание батареи оказались причиной материального ущерба в \$1 тыс.».

ДЛЯ УМЕЛЫХ РУК

Сделай себе... ноутбук!

Olimex предлагает полностью открытый ноутбук-конструктор за \$240.

Болгарская компания — производитель открытого оборудования Olimex выпустила DIY-комплект, из конструкций и компонентов которого пользователь сможет собрать собственный модульный ноутбук с Linux. При этом можно либо скачать с GitHub файлы CAD от Olimex со схемотехникой компонентов, таких как печатные платы (PCB), клавиатура и т.д., а также все программные компоненты Linux, требуемые для управления аппаратными средствами, и самостоятельно сделать всё «с нуля», либо купить за €224 (\$240) полуфабрикат ноутбука Teres 1, который поставляется в виде комплекта для сборки. Такой комплект содержит клавиатуру, тачпад, камеру, динамики, батарею, антенны, кабели и компоненты дисплея, а также винты, необходимые для крепления печатных плат, разъем для наушников, слот microSD и порт USB, кнопку включения питания, материнскую плату и плату управления клавиатурой. В итоге у пользователя должен получиться ноутбук в корпусе черного или белого цвета весом 980 г,



➤ Плюсами Teres 1, как и любого модульного оборудования, является гибкость в выборе спецификаций и возможность замены, по мере необходимости, используемых компонентов.

с четырехъядерным 64-битным процессором Allwinner A64 Cortex-A53, 11,6-дюймовым LCD дисплеем, флэш-накопителем 4 Гб, с Wi-Fi и Bluetooth, камерой и аккумулятором на 7000 мА·ч.

По словам представителя Olimex, проект Teres 1 основан на предназначенном для разработки электрических схем и печатных плат открытом программном комплексе KiCad. KiCad поддерживается CERN KiCad initiative, за которой, в свою очередь, стоят Raspberry Pi и Arduino. **LXF**

Новости короткой строкой

➤ Минкомсвязи РФ перевело сегмент госсистемы миграционного и регистрационного учета «Мир» на отечественную аппаратную платформу (серверы с процессорами «Эльбрус») и СПО. Источник: minsvyaz.ru

➤ В новой версии защищенного корпоративного мессенджера «Сибрус» общение по видеосвязи и в формате видеоконференций на мобильных и настольных устройствах — HD-качества. Источник: www.cybrus.ru

➤ «Россотрудничество» закупило лицензии на отечественную Alfa OS (на базе свободных компонентов) для автоматизированных рабочих мест. Источник: www.alfa.vision

➤ Грег Кроа-Хартман [Greg Kroah-Hartman] официально сообщил о придании ядру Linux версии 4.9 статуса LTS, с обновлениями до января 2019 г. Источник: lkml.org

➤ Владельцы хромбуков от 2017 г. и позже смогут использовать Play Store Google и приложения для Android. Источник: sites.google.com

➤ К 2018 г. компании Skillaz и Vision Labs представят систему компьютерного распознавания, оценивающую на собеседовании поведение соискателя и выбирающую подходящего. Источник: www.cnews.ru

➤ РФ заняла в международном рейтинге «Индекс готовности к Сетевому обществу» 2-е место по сотовой связи, 10-е — по широкополосному подключению, и 41-е — в общем зачете. Источник: www.cnews.ru

➤ Российский производитель IRBIS объявил о старте продаж бюджетного 5-дюймового смартфона IRBIS SP21. Источник: treolan.ru

➤ Стабильный релиз Wine 2.0 запускает «из коробки» 4450 приложений Windows; еще 3828 потребуют настройки и внешних DLL. Источник: www.winehq.org

➤ Рабочие места врачей поликлиник Москвы в течение 2017 г. снабдят компьютерами Acer с российским Linux. Источник: www.mskagency.ru

➤ Специалисты компании «Доктор Веб» обнаружили Windows-троянца, заражающего Linux-устройства. Источник: news.drweb.ru

СКАЖИ УЧЕБЕ

ДА!



СЕРВЕРНЫЕ РЕШЕНИЯ
LINUX
SOLARIS
ХРАНЕНИЕ ДАННЫХ
JAVA
ANDROID
БЕЗОПАСНОСТЬ
ЗАЩИТА ДАННЫХ

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

UnixEducationCenter



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Про Neon: зачем он?

Назначение Neon'a — не только демонстрация новейших достижений строителей KDE: он может служить основой систем для практического применения в разных сферах. Так, упомянутый в прошлой колонке (см. LXF218) дистрибутив Maui 2 олицетворяет собой единение программ, основанных на библиотеках Qt/KDE, с одной стороны, и GTK — с другой, в лице лучших их представителей, разумеется.

К сборке таких производных систем можно подойти и с другой стороны. Например, по тематическому принципу: невзирая, так сказать, на «морды» (т. е. библиотеки), комплектовать их софтом определенного назначения. Например, под лозунгом «Всё для сочинителя» собрать дистрибутив, оснащенный всеми инструментами работы с текстами.

А под годными к случаю девизами укомплектовать систему всевозможными приложениями для работы с графикой. Или, страшно сказать, собирать системы для научных работников — геологов, антропологов etc. Чем по мере сил и занимается автор этих строк. Руководствуясь, прежде всего, востребованностью такого рода штуквин у широких масс своих родных, друзей и близких.

Правда, открою две страшные тайны: одну военную, а вторую — государственную. Суть первой: да, такие системы можно собирать на базе Neon'a. Или положить в основу практически любой другой дистрибутив общего назначения. Вторая же — в том, что об этом еще на заре компьютеризации говорили большевики, меньшевики и научные работники...
alv@posix.ru

Сегодня мы рассматриваем:

Bodhi Linux 4.0 12

Обычно легковесность дистрибутива создается за счет снижения удобства, обеднения функциональности или пренебрежения внешними эффектами. Но это не про Bodhi.

Zorin OS 12 Core 13

Сходство с Windows — для ОС, конечно, фактор скорее компрометирующий, но ведь надо как-то смягчить новичкам трудности переходного периода... и потом, она красива.

OnePlus3 14

Лозунг изготовителя — «переводные смартфоны за полцены». И правда: камерой почти невозможно испортить кадр, корпус сде-

лан из цельного алюминия, ОЗУ — 6 ГБ, процессор — Snapdragon 820, SIM-карт — две... Берем!

Jolla C 15

Маленькая, но гордая пти... финская компания упрямо идет своим путем, разрабатывая собственную ОС и внедряя ее на смартфоны. Что же у нее получилось?

GamePad Digital XD 16

Надпись «Сделано в Китае» не должна отвращать вас от этого добротного устройства, подражания знаменитому Nintendo. У него есть все шансы стать классикой, но пока что требуется доработка.

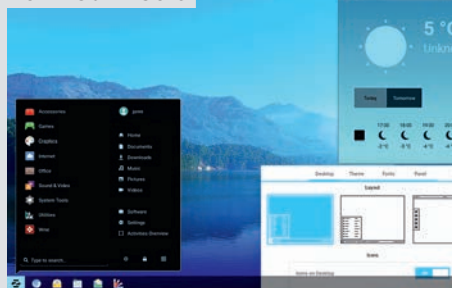


➤ Этот корпус не хрустнет в руках.

Mad Max 17

Переменяя кулачные бои с ристалищами на автомобилях, наш герой режется на открытой пустоши. К чему он стремится? Цель для него, похоже, ничто, а движение — всё.

Zorin OS 12 Core



➤ Кажется, нам случалось подобное где-то видеть. Не припомните, где?

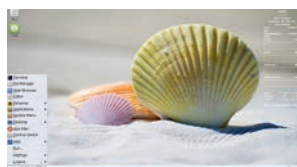
Mad Max



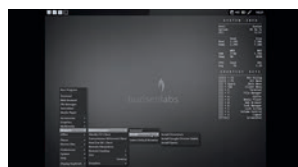
➤ Ощетинившись колючими, боевая машина Безумного Макса мчится на врагов...

Сравнение: Дистрибутивы-легковесы с. 22

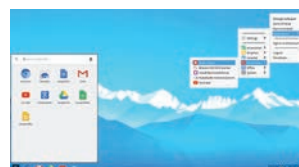
antiX



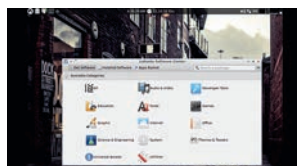
Bunsen Labs



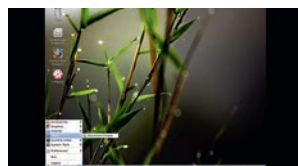
Cub Linux



LXLE



WattOS



Не все дистрибутивы настолько требовательны до ресурсов, что высасывают до капли все соки из процессора! Есть и такие, с которыми заплещут даже престарелые 32-битные системы — так почему бы не продлить ветеранам жизнь?

Bodhi Linux 4.0

Говорят, Будда достиг просветления (bodhi), сидя под фиговым деревом. Джонни Бидвелл надеется, что Bodhi Linux тоже просветит его жизнь.

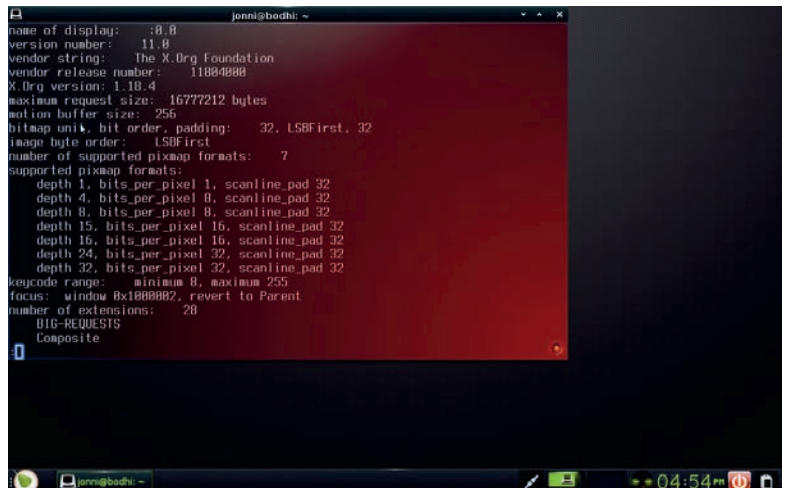
Вкратце

» Дистрибутив-легковес на базе Ubuntu, с собственным рабочим столом *Moksha*. Очень легкий в плане предлагаемых приложений, но идеальный для тех, кому нравится создавать рабочий стол с нуля. Также доступен на хромбуках с SeaBIOS. См. также: LXLE, Lubuntu, BunsenLabs.

Вodhi Linux — легковесный дистрибутив на базе Ubuntu, вышедший на сцену в 2011 г. Его системные требования — из наинизших среди рабочих столов Linux. Он может работать даже на процессоре не-PAE с 128 МБ ОЗУ и 300 МГц. У нас не было такого под рукой, но компьютер Джонни для обзоров достаточно старый и пыльный. Мы позаимствовали его и обнаружили простой процесс установки *Ubiquity* (такой же, как на Ubuntu), который обеспечил нам вполне пригодный к работе и вовсе не малосимпатичный с виду дистрибутив.

Bodhi основан на Ubuntu 16.04 LTS, так что мы получаем ядро 4.4 и всю стабильность этой системы. Но, в отличие от Ubuntu, новая установка занимает всего 2,5 ГБ. Экономия места достигается тем, что вместо рабочего стола Unity взят *Moksha* (что означает «освобождение») и задействуются только основные функции: терминал, текстовый редактор, менеджер файлов, программа просмотра изображений и веб-браузер. Также доступен более крупный ISO AppPack, включающий *LibreOffice*, *VLC* и *Pinta*, и т. д. *Moksha* использует ответвление *Enlightenment E17* (текущая версия — *E21*), имеющего блестящую репутацию среди хакеров рабочего стола. Все включенные приложения используют Enlightenment Foundation Libraries (EFL) вместо *GTK* или *Qt* и т. д., поэтому у них уникальный, но при этом гармоничный внешний вид.

Обычно легковесность означала жертвование удобством, функциональностью или привлекательностью рабочего стола.



» Bodhi Linux 4.0 выглядит стильно и современно, и все ваши друзья обзавидуются на ваш визуальный эффект звонка терминала.

Но не для Bodhi. Здесь не надо зубрить обязательные горячие клавиши, USB-диски не требуют ручного монтирования, и многие скажут, что *Moksha* выглядит куда лучше, чем Unity. Браузер *Midori* (известный благодаря Raspbian) может понравиться не всем, но это один из лучших браузеров-легковесов. Точно так же *PCManFM* — взятый из *LXDE* — может быть простоват в управлении файлами для запросов отдельных личностей. Репозитории Ubuntu, конечно, полны объемистых альтернатив, но они будут слишком тяжелы для старого оборудования.

Легкий и приятный

Нам особенно понравилась *Terminology*. По умолчанию задан шрифт *Terminus*, и пульсирующий курсор выглядит вполне современно: к тому же имеется прекрасный визуальный эффект колокольчика терминала, показанный на рис. вверху. Но постойте, это еще не всё: терминалы со вкладками — последний писк моды, но много ли вы видели терминалов, которые делятся по горизонтали и по вертикали? [Ред.: — Да, и он называется *Terminator*, см. *HotPicks*, стр. 102 **LXF204**.] Есть также и эффекты обратной связи при нажатии на кнопки мыши плюс при перемещении или изменении размеров окон. Bodhi сохраняет некоторые привлекательные функции из предыдущего релиза: вежливое напоминание о существовании Режимы Презентации [Presentation Mode] при быстром включении скринсейвера и поддержку URI

arturi:// (так что эти пакеты можно установить из браузера).

Рабочий стол *Moksha* также можно настроить индивидуально, добавляя Модули [Modules]. Это небольшие гаджеты, обеспечивающие программы запуска приложений, управление громкостью, задней подсветкой и системный лоток. Модули живут в контейнерах под названием *Shelves*, и настройка по умолчанию объединяет один shelf внизу с меню, пейджером рабочего стола, списком приложений и прочим содержимым, которому полагается там быть. Дополнительные shelves с дополнительными виджетами можно добавить по краям экрана. Это делает возможной исключительную персонализацию, но временами способно выглядеть громоздко. **LXF**

Свойства новскидку



Bodhibuilder
Он автоматизирует процесс создания индивидуальных ISO Bodhi. Можно сохранить пользовательские данные в удобной портативной резервной ОС.



EepDater
Здесь нет предупреждений на рабочем столе при появлении новых пакетов, но этот инструмент облегчает ручное обновление системы.

LINUX FORMAT Вердикт

Bodhi 4.0.0

Разработчик: Джефф Хугланд (Jeff Hoogland)
Сайт: www.bodhilinux.com
Лицензия: Разные

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

» Возможно, не совсем для новичков, но один из лучших подарков для вашего старого оборудования.

Рейтинг 8/10

Zorin OS 12 Core

Джонни Бидвелл с подозрением относится ко всему, что упрощает сложные вещи. Например, к Zorin OS, которая делает рабочий стол Linux простым и красивым.

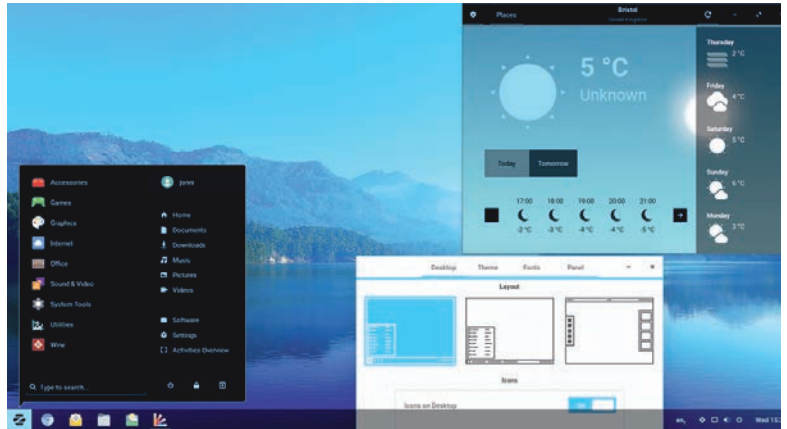
Вкратце

» Дистрибутив на базе Ubuntu, прямоиком нацеленный на пользователей, которым удобно в Windows или MacOS. Включение Wine и Play On Linux делает его соблазнительной опцией для тех, кто не удовлетворен текущим положением дел с играми в Linux; однако не ждите, что он сможет запускать GTA V по умолчанию. См. также: Linux Mint, KaOS, elementary OS.

Финальный выход Роджера Мура [Roger Moore] в роли Джеймса Бонда (*Вид на убийство*) продемонстрировал нам, как он побеждает злодея-блондина Зорина, со всеми своими характерными приподниманиями бровей, скверными каламбурами и находящимся на грани сексизмом, столь любимым зрителями 1980-х. Всё это никак не касается рассматриваемой нами Zorin OS, но те, кто не помнит историю, обречены на ее повторение, а с нас уже хватит бровей Роджера.

В отличие от агента 007, блистательная история Zorin OS простирается всего лишь до 2009 г., однако этого вполне достаточно для создания преданного сообщества поклонников (он занимает 8-е место в *DistroWatch*) и доведения до совершенства весьма впечатляющей среды рабочего стола. Вид по умолчанию напоминает тему Windows 10, разве что несколько более стильную: плоскую, внятную и минимальную, с несколькими значками рядом с меню.

У тех, кому нужен другой вид, есть две альтернативы: более стандартный вид Gnome 3 (только верхняя панель, а все остальное доступно из Activities Overview) и более классический вид Windows с заголовками окон на панели задач. Ее внешний вид будет и далее индивидуально настраиваться (отдельные нетипичные личности способны даже переместить кнопки окон влево), и можно настроить основную цветовую схему. Те, кто жаждут выложить €19 за издание Премиум (см. *Свойства навскидку*), могут дополнительно выбрать стиль Ubuntu, Gnome 2 или Mac OS X.



» Zorin OS симпатичный, интуитивный и имеет логотип, похожий на дружинника с мечом. Что тут может не понравиться? Ну, разве что модель freemium...

Установка через очищенную программу установки *Ubiquity* была, как мы и ожидали, очень легкой, и стандартная установка заняла менее 5,5 ГБ. Она основана на Ubuntu 16.04, а рабочий стол — на Gnome 3.18, и включает отличную подборку основных приложений Gnome.

Особого упоминания достойно включение *Déjà Dup* для резервного копирования и *Seahorse* для управления ключами и паролями. Zorin также демонстрирует современные инструменты коммуникации, такие как клиент электронной почты *Geary* (поставляемый также с elementary OS) и программа обмена мгновенными сообщениями *Empathy*. Еще есть приложение *Activity Journal*, которое умеет подключаться к календарям Google, Microsoft или ownCloud.

по крайней мере, не больше, чем базовый Gnome 3. Однако он определенно не для старого оборудования, *Chromium* и *LibreOffice* здесь точно так же прожорливы в плане ресурсов, как и везде.

Как сообщается в блоге Zorin, это «самый крупный релиз», и он определенно впечатляет. В отличие от предыдущих редакций, теперь у Zorin будут постоянные точечные обновления, предположительно в соответствии с оборудованием Ubuntu и обновлениями безопасности. Мы не считаем, что дистрибутив обязан быть похож на Windows или Mac OS X, поскольку тут есть риск скопировать недостатки этих систем. Но всё, что облегчает переход на настольный Linux, нас радует. И если, подобно Zorin OS, при этом можно еще и выглядеть красиво и не прятать основных инструментов настройки и Terminal, тем лучше. **LXF**

Четкий дизайн

Zorin когда-то предлагал множество эффектов *Compiz*, а именно пульсирующие окна и рабочий стол в форме куба. К счастью, вкусы изменились, и все эти прелести остались в прошлом. Оказалось, что для симпатичного рабочего стола нам совсем не нужно, чтобы окна исчезали в облаке губительного для драйверов дыма. Здесь есть несколько приятных эффектов теней и прозрачность панели задач, и это не ущемляет удобство в использовании в угоду внешней привлекательности.

Чистая загрузка на нашей скромной машине для обзоров (второе поколение Intel Core i5 с 4 ГБ ОЗУ) заняла около 20 секунд, и Zorin занял менее 600 МБ ОЗУ. Так что стили рабочего стола, похоже, не слишком нагружают системные ресурсы, или,

Свойства навскидку



Максимальное издание
За €15 вы получаете разные схемы рабочего стола, игры и поддержку. Увы, наш скудный бюджет не позволил нам всего этого протестировать.



Играйте в Linux
Играйте в игры Windows легко, или играйте в игры Windows с отличающимся опытом, по крайней мере. Теперь доступен ряд игр GOG.com.

LINUX FORMAT Вердикт

Zorin OS 12

Разработчик: Команда Zorin
Сайт: <http://zorinos.com>
Лицензия: Разные (€19 ред. Premium)

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

» Сначала мы были настроены скептически, но этот дистрибутив — усадка для глаз и абсолютное удовольствие в использовании.

Рейтинг 9/10

OnePlus3

У OnePlus наконец-то вышел по-настоящему убойный смартфон, считает **Джон Мак-Канн**.

Спецификация

- » **ОС** Android 6.0.1
- » **Процессор** Snapdragon 820 4-ядерный 2×2,15 ГГц Kryo, 2×1,6 ГГц Kryo
- » **ОЗУ** 6 ГБ
- » **Диск SSD** 64 ГБ
- » **Экран** 5,5" 1080×1920 AMOLED
- » **Емкость батареи** 3000 мА·ч
- » **Камера** 16 Мп основная, 8 Мп тыльная
- » **Связь** Wi-Fi 802.11 ac, bluetooth 4.2 LE, NFC, USB-C
- » **Датчики** Отпечатков пальцев, акселерометр, гироскоп, близости, компас
- » **Габариты** 152,7×74,7×7,4 мм
- » **Вес** 158 г

Три ведь магическое число, так? Видимо, на это и надеялись в OnePlus, выпуская третье поколение своего смартфона, OnePlus3. Верные своему девизу «Никогда не останавливайся», OnePlus идут вперед, в соответствии с планом «передовые смартфоны за полцены», и с новым OnePlus 3 это, кажется, удалось.

Здесь нет разрешения QHD, как и слота MicroSD, и для внутреннего хранения информации пользователям выдается только 64 ГБ. Но если, невзирая на эти мелкие недостатки, вы сможете оценить мощь, прекрасный дизайн и производительность OnePlus 3, то поймете, что это лучшее устройство данной фирмы.

С процессором Snapdragon 820, 6 ГБ оперативной памяти, задней камерой на 16 Мп, и передней на 8 Мп, сканером отпечатков пальцев и батареей с функцией быстрой зарядки на 3000 мА·ч, в OnePlus 3 есть все инструменты, чтобы — по крайней мере на бумаге — называться выдающимся.

Чего стоит уже один дизайн — OnePlus отреагировали на запросы пользователей его улучшить, создав свой первый смартфон OnePlus 3 в монолитном металлическом корпусе из алюминия.

Сканер отпечатков теперь распознает ввод для разблокировки гораздо быстрее, чем в OnePlus 2, всего за каких-то 0,3 секунды. OnePlus заявляют, что он быстрее, чем Touch ID от Apple, и мы склонны с этим согласиться.

В нем по-прежнему есть поддержка двух SIM-карт, но нет слота microSD; его функцию, в стиле Huawei P9, выполняет второй слот SIM. В OnePlus объясняют это решение дизайном.

Дисплея QuadHD — которым хвастаются многие «передовые» смартфоны — здесь нет. OnePlus 3 остался верен 5,5-дюймовому HD-экрану, но при этом всё же перешел на панель AMOLED, давая более яркие цвета, с двойной поляризацией, для просмотра при солнечном свете.

На полной мощности

В OnePlus утверждают, что с их 16-то мегапиксельной камерой трудно сделать плохое фото, учитывая автоматический HDR, автофокусировку и динамическое удаление шума. Вы также можете захватывать фотографии в формате RAW, а ручной режим дает больший контроль над параметрами, такими как ISO, баланс белого и выдержка.



» Верные своему девизу не останавливаться на достигнутом, OnePlus предлагают новый, более мощный и стильный телефон.

Мощности в OnePlus 3 хоть отбавляй. Первоклассный процессор Snapdragon 820 от компании Qualcomm дополнен огромным объемом ОЗУ: 6 ГБ! Производительность очень хорошая, с плавной навигацией и быстрой загрузкой приложений — но конкурентов превосходит не сильно. В тесте Geekbench 3 общий результат составил 5425, помещая трубку в середину списка флагманов 2016 г. Кроме того, он легко превзошел показатели OnePlus 2 (4795) и iPhone 6S (4417) 2015 г.

Даже при умеренном использовании я обнаружил, что переключаться в режим экономии аккумулятора мне приходилось почти ежедневно, поскольку к началу вечера заряд падал до 15%. После нашего теста с просмотром 90-минутного HD видео он составил 23%, и это является средним показателем. Здорово, что есть возможность быстрой зарядки, и наличие технологии Dash Charge позволяет наполовину зарядить аккумулятор за полчаса. Для работы телефону требуется находящийся в комплекте зарядник и толстый кабель USB типа C.

NFC также приносит пользу, давая возможность осуществлять бесконтактные платежи через Android Pay. Android 6 сопровождается его собственным интерфейсом

Oxygen, и OnePlus содержит инструменты для быстрого перехода к root и добавления альтернативных загрузчиков; радует и активная разработка CyanogenMod OS. К моменту написания статьи также был выпущен OnePlus 3T с улучшенным аккумулятором, процессором и различными вариантами хранения — правда, на £70 дороже, чем OnePlus 3. **LXF**

LINUX FORMAT Вердикт

OnePlus 3

Разработчик: OnePlus
Сайт: <https://oneplus.net/uk/3>
Цена: £ 329 64 Гб

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

» Став лучше по ряду показателей и при этом сумев удержать низкую цену, составит серьезную конкуренцию именитым образцам.

Рейтинг 9/10

Jolla C

Компания-разработчик ОС Sailfish выпустила второе поколение своего смартфона, и **Александр Толстому** он пришелся по душе.

Спецификация

- » **ОС** Sailfish OS 2.x
- » **Экран** 720×1280 IPS LCD
- » **Процессор** Snapdragon 212, 4-ядерный 1,3 ГГц Cortex-A7
- » **Графика** Adreno 304
- » **ОЗУ** 2 ГБ
- » **Внутренняя память** 64 ГБ
- » **Камера** 8 Мп тыльная, 2 Мп фронтальная
- » **Коммуникации** Wi-Fi 802.11 b/g/n, Bluetooth v4.0, A2DP, EDR, A-GPS, ГЛОНАСС
- » **Связь** Dual SIM GSM, HSPA, LTE
- » **Порты** micro USB
- » **Аккумулятор** Li-Ion 2500 мА·ч
- » **Габариты** 142×72×9,6 мм
- » **Вес** 150 г

Jolla C — второй, долгожданный смартфон от финской компании Jolla, выпущенный в мае 2016 г. на базе ОС Sailfish. Первый смартфон Jolla вышел в 2013 г., имел слабоватые аппаратные характеристики и выглядел как продолжение инновационного Nokia N9 из 2011 г., в основе которого была ОС MeeGo, производная от Linux. Тогда у Jolla были большие амбиции, например, сменные крышки «второй половины», превращаемые в клавиатуру QWERTY, или большой аккумулятор. Из-за финансовых проблем некоторые запланированные нововведения Jolla вынуждены были отменить или отложить, но сумели произвести новенький Jolla C в сотрудничестве с Intex.

На самом деле, Jolla C — это полный аналог Intex Aqua Fish, телефона среднего уровня, созданного для индийского рынка; в нем яркий 5-дюймовый дисплей, шустрый четырехъядерный процессор и поддержка практически всех современных стандартов связи. Обе модели Intex доступны в ограниченных количествах, в основном в Индии, а на ярко-красном корпусе устройства марки Jolla есть крошечная метка 'Limited Edition'. Телефон Jolla имеет мощный сменный аккумулятор, слоты для двух SIM-карт, слот MicroSD и практичный дизайн корпуса. Корпус пока что доступен только в одном цвете, который дополняется красной темой в настройках по умолчанию в ОС Sailfish. Выбирая Jolla C, вы немного переплачиваете за возможность доступа к редкой операционной системе на базе Linux, которой нет больше практически нигде. В операционной системе используются *Systemd*, *PulseAudio*, Linux ядро (версии 3.4) и другие программы, встречающиеся в обычном настольном Linux. Интерфейс создан с использованием библиотек *QML* и *Qt5* — подобных тем, что имеются в Plasma 5 (но в варианте Jolla более стабильных).

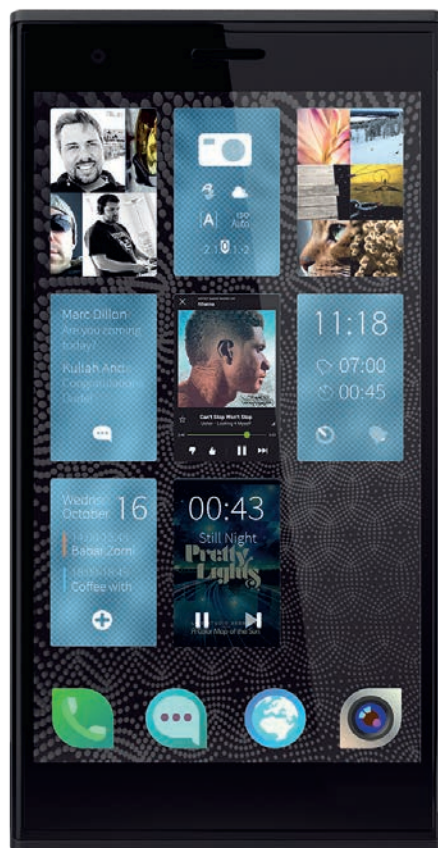
Sailfish OS не требуются никакие аппаратные кнопки: все действия выполняются жестами. Понадобится пара дней, чтобы к этому привыкнуть, но через некоторое время всё уже кажется легко, и уже трудно вернуться к привычному телефону на Android.

Больше плюсов

Благодаря четырехъядерному Cortex A7 и 2 ГБ ОЗУ, Jolla C кажется очень быстрой и отзывчивой. ОС, со своей стороны, помогает продлить срок службы батареи: Jolla C имеет аккумулятор на 2500 мА·ч, который при умеренном использовании протянет практически три дня. Конечно, ваш пробег может отличаться, но возможность играть, просматривать Интернет на скорости 4G и несколько часов изучать меню, не ставя телефон на зарядку каждую ночь — это уже хорошо. Стандартные предустановленные приложения включают обычный набор аксессуаров и очень приличный web-браузер, на базе движка Gecko от Mozilla. Новые приложения добываются из трех источников: официального магазина Jolla C, через клиент Warehouse для <https://openrepos.net> и некоторые магазины Android, такие как Yandex Store, Aptoidе Appstore и Anzhi Market.

Sailfish OS включает среду времени выполнения Alien Dalvik 4.4.4, которая эмулирует окружение Android и позволяет счастливым обладателям Jolla C наслаждаться огромным количеством приложений и игр Android. Среду времени выполнения Aliendalvik в Sailfish OS можно вручную выключить в настройках операционной системы, но даже когда она тихо работает в фоновом режиме, это не влияет на производительность или срок службы батареи. Приложениями можно, кроме того, управлять из командной строки. Помимо стандартной утилиты *pkgcon*, вы можете установить *zypper*, чтобы проделывать всё то же, что и в openSUSE либо SLE.

Говоря о недостатках, мы не можем закончить этот обзор, не упомянув о разочаровании от камеры Jolla C, шума и размытости в которой даже больше, чем в камере старого телефона производства Jolla. Комплект поставки Jolla C тоже очень скромный: зарядного устройства и наушников нет,



» Многозадачность в Jolla C поистине на уровне настольного Linux. Вообще ничего не тормозит!

предлагаются только крышка, аккумулятор, телефон и смехотворно короткий кабель микро-USB. Но это не портит картину вокруг великолепной Sailfish OS и ее легко доступного терминала root. **LXF**

LINUX FORMAT Вердикт

Jolla C

Разработчик: Jolla Oy
Сайт: <http://jolla.com>
Цена: € 190

Функциональность	8/10
Производительность	9/10
Удобство в работе	9/10
Документация	6/10

» Jolla C — бойкое сочетание технологичный Linux и современного смартфона с удобным интерфейсом.

Рейтинг 9/10



GamePad Digital XD

Джон Найт тестирует игровой эмулятор, встроенный в геймпад, который стал в Сети предметом культа, но стоит ли игра свеч?

Спецификация

- » **Экран** Тачскрин 5" 1280×720, IPS
- » **Процессор** Rockchip RK3288 4-ядерный (ARM Cortex-A17)
- » **Графика** Mali-T764 600 МГц
- » **ОЗУ** 2 ГБ
- » **Диск SSD** 16/32/64 ГБ
- » **Связь** 802.11b/g/n/ac 2,4 ГГц
- » **Порты** 3,5-мм аудио, HDMI, microUSB, microSD
- » **ОС** Android 4.4.4 (KitKat)
- » **Габариты** 155×89×24 мм
- » **Вес** 300 г

Вы, наверное, и слыхом не слышали о GamePad Digital (GPD) XD, но вокруг этого устройства в Интернете сложился уже самый настоящий культ. Оно встроено в оболочку, в целом напоминающую Nintendo DS, но работает на Android с программным пакетом, оптимизированным для игр (в частности, эмуляции). Несмотря на то, что изготовлено оно в Китае, качество сборки действительно приличное: выглядит прочным, нет ощущения, что оно вот-вот сломается, и аналоговый джойстик весьма неплох.

Цены на новое устройство на eBuy варьируются от £160–190, в зависимости от того, какой слот microSD вы выбираете: на 16, 32 или 64 ГБ. Низкая цена обусловлена отсутствием Bluetooth и камеры. Android старый, версии Kit-Kat 4.4.4.

Зато не поспешили на гораздо более важные вещи: четырехъядерный процессор, графический процессор, HDMI-выход, и целых 2 ГБ ОЗУ. Эта комбинация многократно увеличивает производительность —

XD способен эмулировать удивительно современное оборудование, и современные игры Android на нем идут превосходно.

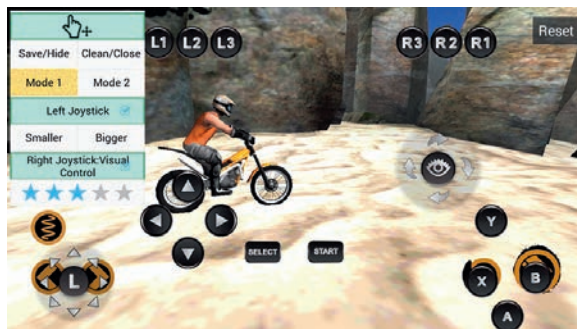
Срок службы аккумулятора приличный. Можете твердо рассчитывать на 8 часов полноценной игры с приличной нагрузкой на процессор и на 10+ часов при менее интенсивном использовании. За этой штукой действительно можно провести целый день.

Недостатки? Ну, как уже упоминалось выше, это разве что использование Android 4.4.4 — прекрасный повод для оптимизации; но только время покажет, появятся ли необходимые обновления ОС. Помимо этого, есть еще бредовый интерфейс в стиле Windows 8, который порадует разве что полных неопитов, но любому человеку с опытом тут же захочется переключиться на стандартный интерфейс Android. Вдобавок почти нет документации: официальный сайт уморителен (на ломаном английском), и чтобы получить хоть какую-то информацию, надо рыться на YouTube, в блогах и на досках объявлений.



дешевле и более мощные, но GPD XD является одной из тех редких вещей, когда целое куда лучше, чем сумма его составляющих. Людям оно нужно, и его можно полюбить. Зайдя в компьютерный магазин, мы увидели, что все сотрудники столпились вокруг него, желая попробовать — и многие говорили, что купят его себе.

По совести говоря, пока это твердая семерка, поскольку многовато недочетов: XD дорог (совершенно новый Nintendo Switch стоит £199), нуждается в качественной документации, написанной носителем английского языка; нужны более внятные эмуляторы с пользовательскими настройками; и существуют серьезные проблемы с «железом», в плане подзарядки и прошивок. Если в GPD это исправят, мы добавим еще балл-другой. Так что, если хотите, в перспективе это восемь! **LXF**



» Есть кнопка, позволяющая в любой момент назначить физические элементы управления, вместо неудобных экранных, используемых в играх для планшетов Android. Только представьте ужасы *Quake 3* с этими дурацкими кнопками на экране.

Эмулятор радости

Также XD опирается на громоздкие монолитные программы, вроде *Happy Chick*, для почти мгновенного запуска игр и эмуляции. Для чего-либо 8- или 16-битного это прекрасно, но со сколько-нибудь более современным справляется плохо, и кроме того, представляется весьма небезопасным: вещи, которые не должны быть доступны, таковыми являлись. Мы бы рекомендовали забыть о *Happy Chick* — и установить нормальные знакомые эмуляторы.

Как он поладит с новыми консолями? Возможно, на вашей любимой машине вам будет играть прекрасно, но, вероятно, придется настроить параметры. С 3DO все отлично. С DS — тоже. PS1 работает великолепно, а вот PS2 — с переменным успехом. PSP попробовать стоит: одни игры шли прилично, другие — ужасно (*Gran Turismo*, например, не повезло). *N64* в целом тянет, если по уму выбрать эмулятор и настройки. *Dreamcast* способен глючить в плане графики, но в остальном работает красиво.

Мы рискуем заявить, что в будущем это устройство станет классикой, за которой будут охотиться коллекционеры. Есть карманные компьютеры меньше, быстрее,

LINUX FORMAT Вердикт

GamePad Digital — GPD XD

Разработчик: GamePad Digital
Сайт: www.gpdxd.com
Цена: £160–190 (16–64 ГБ)

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

» Хотя он и близок к культовому, как клон похожий на Nintendo DS, но требует доработки.

Рейтинг **7/10**

Свойства навскидку



Playstation XD

При наличии выхода HDMI и внешнего контроллера, многие используют XD в качестве компактных игровых консолей и для Netflix.



Windows (h)8

По умолчанию в XD используется интерфейс в стиле Windows 8; советуем его поменять, чтобы сохранить рассудок.

Mad Max

Разбивая машины, освистывая бойцов и кулачные бои не на жизнь, а на смерть, **Кристофер Ливингстон** пролагает свой обычный маршрут к центру Бата.

Спецификация

Минимальные:

» **ОС** Ubuntu 16.04, Steam OS 2.0 (только 64-битная)

» **Процессор** Intel Core i5 3,4 ГГц, AMD FX 8350

» **ОЗУ** 8 ГБ

» **Графика** Nvidia GeForce GTX 660Ti с 2 ГБ (драйвер: 367.35)

» **Места на диске** 35 ГБ

ВНИМАНИЕ: Графика AMD и Intel не поддерживается

Макс обнаруживает себя брошенным в сердце пустыни — хуже того, его легендарный Interceptor украден — и после схватки с неуклюжим воителем объединяет усилия с услужливым механиком по имени Чамбакет [Chumbucket], согласившимся помочь Макс в постройке нового автомобиля, чтобы Макс смог отомстить и вернуть свою родную машину.

Добро пожаловать на открытую пустошь! Вы будете постоянно наталкиваться на одиночные машины или многомашинные воинские подразделения, оцетинившиеся копьями, загруженные взрывчаткой или перевозящие команды свистящих и улюлюкающих Бойцов [War Boys]. К счастью, у вас есть то, чего нет у них: мощный гарпун, установленный на машине. Прицельтесь в двери машины, и вы можете оторвать их. Прицельтесь в задний бампер, и вы можете прицепиться к нему и использовать эту машину в качестве тарана. Оторвите броню по кусочку, чтобы обнаружить топливный резервуар, который вы сможете потом взорвать выстрелом из дробовика.

Улучшив свой гарпун, вы приобретаете возможность отрывать колеса машины, полностью нейтрализуя ее. Самое лучшее — то, что вы можете выдрать водителя прямо с его сиденья, через небронированное лобовое стекло или через оторванную вами дверь, а потом катапультировать его в воздух. Потрясающе.

Гарпун полезен не только против других машин. На пустоши вражеские группировки создали устрашающие металлические башни-«пугала», которые можно загарпунить и снести. На этой местности также полно снайперских точек, и вы можете снести и их, или зацепить снайпера и вытащить его прямо из укрытия.



» Игра демонстрирует длинные, хаотичные и жестокие сцены погони из фильмов.



Кульминацией боев на машинах является встреча с топливным конвоем, большой ведущей машиной со множеством машин эскорта. Именно здесь игра демонстрирует длинные, хаотичные и жестокие сцены погони из фильмов. Когда вы гонитесь за конвоем и начинаете устранять защитные машины, вас со всех сторон будут таранить бойцы, жаждущие прыгнуть в вашу машину и пытающиеся достать вас через ваши окна, пока вы их не стряхнете.

Пешая битва

Конечно, вам придется выходить из машины; тут-то и начинается интересная часть *Mad Max*. Захват вражеских лагерей всегда происходит на пешем ходу; рукопашная работает так же, как мы видели в играх Arkham, хотя не такая гладкая и быстрая. Во время боя камера способна работать ужасно, иногда полностью заслоняя Макса и его врагов. У нас есть множество забавных историй о битвах на машинах, но ни одной — о кулачном бое.

Да и около десятка сражений с боссами тоже не выдающиеся. Лагерные боссы однообразны: неуклюжие здоровяки с огромными молотками, которых надо быстро атаковать, чтобы укоротить их линии здоровья, а затем увернуться от их атаки. Пара других боссов чуть более быстры и подвижны, однако стратегия остается той же.

Помимо машин и рукопашной, на пустоши есть чем заняться. Мы играли 45 часов, и карта всё еще была полна значками, хотя, честно говоря, эту деятельность вряд ли назовешь разнообразной.

По крайней мере, сама по себе пустошь весьма привлекательна. Вы найдете здесь

выбеленный солнцем песок, желтые подвижные дюны, темные горы и утесы, обширные маслянистые болота, а также возвышенности и валуны цвета ржавчины. Далее там есть Газоград [Gas Town] с облаками дыма и горами мусора, создающими впечатление мира, похожего на индустриальный Мордор. Жаль, что сама история не соответствует визуальным эффектам; большинство персонажей, включая и самого Макса, довольно неинтересны.

Недостатков в *Mad Max* имеется множество, но выдающиеся и потрясающие битвы на машинах солидно их компенсируют. Как только нам надоедали бесконечные кулачные бои, мы быстро прыгивали обратно в машину и начинали нарезать круги в поисках многообещающих облаков пыли, индикаторов перемещающегося конвоя. И тогда мы готовили гарпун, впрыскивали нитро и разогнались. **LXF**

LINUX FORMAT Вердикт

Mad Max

Разработчик: Feral Interactive
Сайт: <https://store.feralinteractive.com>
Цена: £ 25

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

» Увлекательные сражения на машинах и красивая пустошь в общем компенсируют однообразные ближние бои.

Рейтинг 8/10

ПЕРСОНАЛИИ

Уго Барра покинул Xiaomi

Потому что всегда мы должны возвращаться...

В сентябре 2013 г. Уго Барра [Hugo Barra] перешел из Google, где он занимал пост вице-президента по управлению продуктами Android, в малоизвестную тогда китайскую компанию Xiaomi. За время пребывания Барры в Пекине стратегия продаж стильных телефонов на базе Android с малой прибылью позволила Xiaomi выйти далеко за пределы внутреннего рынка. На CES 2017 в Лас-Вегасе презентация Xiaomi вызвала немалый интерес, а продукты компании экспортируются в более чем 20 стран, включая Индию, Индонезию, Сингапур, Малайзию, Россию, Мексику и Польшу. Но после этого Уго объявил о планах возвратиться в Силиконовую долину «по личным причинам» — глобальный бизнес компании «уже

не просто домашний стартап», и ему пора покинуть Xiaomi: «Несколько лет в такой своеобразной среде привели к серьезным последствиям в моей жизни и начали влиять на здоровье». Он также добавил, что хочет быть ближе к друзьям, семье и месту, которое считает своим домом (хотя родился не в Калифорнии, а в бразильском Белу-Оризонти, а в США переехал в 1996 г. и поступил в МТИ). Об этом Барра объявил в Twitter и Facebook, тогда как об уходе из Google сообщал в Google Plus (где его профиль не обновлялся уже 18 месяцев).

CEO Xiaomi Лей Цзюнь [Lei Jun] попросил Барру остаться советником компании. Вслед за тем Марк Цукерберг сообщил, что Барра занял пост руководителя VR-подразделения Facebook, где будет отвечать



► На CES 2017 вице-президент Xiaomi Уго Барра представляет смартфон Mi Mix в белом корпусе.

в т.ч. за разработку шлема виртуальной реальности Oculus. По мнению Цукерберга, Барра разделяет его уверенность, что виртуальная и дополненная реальность в будущем станут одной из крупнейших отраслей.

ХРОМБУКИ

Двое из ларца

Acer и Asus выпустили новые модели для образовательных учреждений.

В ходе Bett 2017 (международной конференции и выставки учебных, образовательных, информационных и коммуникационных технологий), 24 января Google представила два новых хромбука — естественно, ориентированных на сферу образования. Обе новинки, Acer Chromebook Spin 11 и Asus Chromebook C213, являются конвертируемыми, т.е. могут складываться в планшетоподобное устройство; при их разработке учли обратную связь от учителей и преподавателей, и в сотрудничестве с производителями реализовали поступившие предложения.

Обе модели имеют дисплеи диагональю 11,6 дюйма с разрешением 1366×768 пикс. Известно, что в Acer используется матрица IPS, а экран Asus закрыт стеклом Gorilla Glass 4. Комплектация Acer Spin 11 включает CPU Celeron N3450 или Celeron N3350, 4 или 8 ГБ ОЗУ, 32 или 64 ГБ флэш-памяти; емкости аккумулятора хватает на 10 часов

работы (у Asus — 11 часов). У обеих моделей влагозащищенные клавиатуры и порт USB-C, который также служит для зарядки.

Чуть ранее в январе Samsung представила первый хромбук с поддержкой стилуса, теперь этому примеру последовали Acer и Asus. Обе модели оснащены двумя фронтальными камерами: одна традиционно расположена над дисплеем, а вторая — над клавиатурой; такое решение дает возможность пользоваться камерой, когда устройство находится в режиме планшета. Точная цена и дата выпуска не названы; Google обещает, что оба хромбука выйдут «в конце весны».

Новые «хромбуки для образования» поддерживают приложения для Android; Google предоставила администраторам возможность создавать перечень разрешенных для использования в учебном заведении приложений. А компания Adobe объявила о предстоящих обновлениях



► Конвертируемый Acer Chromebook Spin 11 с IPS-матрицей продолжает успешную линейку «школьных» хромбуков Google.

некоторых своих креативных приложений для устройств на Android, которые теперь будут оптимизированы и для хромбуков (в списке приложений Adobe, которые получают совместимость с хромбуками — Photoshop Mix, Lightroom Mobile, Illustrator Draw, Photoshop Sketch, Adobe Comp CC, Creative Cloud Mobile).

ИНТЕРНЕТ ВЕЩЕЙ

Меч и орало: Союз Intel и Google

Плата Joule 570x будет работать под управлением ОС Android Things.

Плата Intel Joule 570x сможет работать с Android Things, специальной версией Android для «умных» устройств. Такое программно-аппаратное сочетание позволит создавать гаджеты на Android или «умные» устройства для использования в домашних условиях, в розничной торговле или на производстве. Решения на основе платы Intel получат значительный прирост вычислительной и графической мощности: поддержка графических возможностей 4K делает 570x незаменимой для экранов или устройств с компьютерным зрением, таких как роботы и дроны. Intel уже продемонстрировала робота-бармена с этой платой.

Замечательной особенностью Joule 570x является наличие камеры RealSense 3D, способной определять объекты и измерять расстояние. Плата оснащена процессором Atom T5700, 4 ГБ LPDDR4 ОЗУ, накопителем 16 ГБ и Wi-Fi 802.11ac.

Пока Android Things поддерживается только тремя платами — Raspberry Pi, Intel

Edison и NXP Pico i.MX6UL. «На бумаге» спецификации Joule 570x лучше, чем у Raspberry Pi 3, однако для Android Things они могут оказаться избыточными: например, для работы с сенсорными устройствами Android Things достаточно базовых возможностей таких процессоров, как Quark на Intel Edison.

Распространение Android Things в разнообразных устройствах поможет Google эффективно конкурировать с используемым в большом количестве гаджетов и бытовых приборов голосовым ассистентом Amazon Alexa. Ранее от Google поступила информация, что производители смогут создавать устройства, использующие технологии машинного обучения и распознавания речи компании, выполняемые в облаке. Позднее в этом году Google предоставит производителям доступ к TensorFlow API. Android Things — еще один аргумент в поединке Google с Microsoft Windows 10 IoT Core, Ubuntu Snappy Core и другими встроенными ОС на основе Linux. Для



► Плата Joule 570x с Android Things IoT OS на борту как свидетельство укрепления партнерства Intel и Google, направленного на расширение позиций Android в мире Интернета вещей.

IoT в ближайшие годы будут предложены миллиарды устройств, и битвы ОС в этой области обещают быть не менее жаркими, чем на рынке ПК и серверов. Пока же Android Things находится в стадии preview, и дата выхода стабильной версии этой ОС неизвестна.

ТРАНСФОРМЕРЫ

Атомная Йога

Lenovo предлагает бюджетный трансформер на Android.

При создании трансформера Yoga A12 2-в-1 компания Lenovo отказалась от традиционной жесткой клавиатуры в пользу сенсорной панели Halo. Продажи начались 7 февраля по цене от \$299,99; новинка оснащается универсальной сенсорной панелью с подсветкой, которая уже знакома пользователям по вышедшей в 2016 г. модели Yoga Book, а также 12,2-дюймовым сенсорным экраном с разрешением 1200×800 пикс. (однако, в отличие от Yoga Book, панель Yoga A12 невозможно использовать для создания графических объектов). Время автономной работы без подзарядки — до 13 часов. Имея накопитель 32 ГБ и 2 ГБ ОЗУ, функционально Yoga A12 подобен хромбуку — кроме окружения Android.

Комплектуется Yoga A12 морально устаревшим процессором Intel Atom X5 (кодовое наименование Cherry Trail). Intel заменяет процессоры Atom X5 более новыми чипами

Celeron и Pentium с кодовым именем Apollo Lake, которые в основном устанавливаются в планшеты и устройства 2-в-1, работающие под управлением Windows. Использование в Yoga A12 чипа Atom X5 Lenovo объясняет тем, что когда весной прошлого года Intel объявила о прекращении производства чипов Atom, разработка Yoga A12 была в самом разгаре. Вдобавок Intel приняла на себя обязательства сделать все необходимое для поддержки продуктов, использующих Atom X5. Lenovo также заявляет о наличии четкой процедуры обновления процессоров с Atom до Apollo Lake (возможно, в преемнике Yoga A12 будет задействован Apollo Lake).

В большинстве устройств на Android сейчас используются чипы ARM; лишь немногие планшеты с Android работают на Cherry Trail (в их числе Lenovo Yoga Book и Panasonic ToughPad FZ-A2). От производства планшетов на Android с чипами



► Yoga A12 размывает границы: что мы получаем за \$300 — ноутбук на Android или планшет?

Atom отказалась Dell, а Microsoft прекратила выпуск своего планшета Surface 3, тоже на чипе Cherry Trail.

На Yoga A12 предустановлена ОС Android 6.0.1, которую Lenovo планирует обновить до Android 7.0 «в ближайшие пару месяцев». При этом Intel, ответственная за разработку обновлений для Android, сократила развитие ОС для мобильных устройств, сосредоточив основное внимание на Windows и хромбуках. Вероятно, Lenovo придется взять на себя сложную задачу развития индивидуальной версии Android 7.0 для Yoga A12. Нынешние мобильные Android-устройства с чипами Atom в массе своей работают на Android 5 или 6, и речь об их обновлении до Android 7.0 не идет. **LXF**

Архитектура платформы

Не зная броду, не суйтесь в воду. **Лада Шерышова** описывает особенности ОС Tizen, чтобы вы смогли приняться за создание приложений.



Наш эксперт

Лада Шерышова долгие годы работала на коммерческие корпорации, создавая промышленные высоконадежные информационные системы. Но пришло время сбросить оковы и применить свои знания и опыт в работе со свободным ПО.

Как мы уже знаем, Tizen — это операционная система с открытым исходным кодом на базе Linux, которая позволяет создавать многофункциональные приложения для различных категорий устройств — смартфонов, планшетов, носимых устройств, «умной» техники и т.д. Перед тем, как начать разрабатывать свои собственные приложения под Tizen, кратко ознакомимся с архитектурой платформы.

Типы приложений в ОС Tizen

Платформа Tizen поддерживает 2 основных типа приложений — нативные приложения и Web-приложения.

Нативное приложение — это приложение, которое написано на «родном» (от англ. «native» — родной) для определенной платформы языке программирования и оптимизировано под конкретную ОС. Такие приложения загружаются через магазины приложений (например, App Store, Google Play и т.д.) и отличаются высокой скоростью работы и производительностью. Также они имеют доступ к аппаратной части устройства (камере, плееру, микрофону, акселерометру и т.д.) и экономно расходуют его ресурсы — аккумулятор и память. Кроме того, нативные приложения могут полностью или частично работать при отсутствующем Интернет-соединении. Однако разработка нативных приложений — достаточно трудоемкий процесс, т.к. требует от разработчика специальных знаний и умений для работы в конкретной среде разработки.

В Tizen нативные приложения разрабатываются на языке C++ и могут получать доступ к таким аппаратным возможностям устройств, как камера, GPS, акселерометр и т.д. Приложения используют Native API, который предоставляет различные интерфейсы ко многим аппаратным средствам, встречающимся в современных мобильных и носимых устройствах, а также обеспечивает управление памятью и производительностью. Native API включает десятки модулей, которые охватывают широкий спектр функциональности.

Web-приложение — это, по сути, мобильная версия сайта, хранящегося на устройстве и разработанного с использованием web-языков, таких как HTML5, CSS и JavaScript. Web-приложения иначе называют html5-приложениями. Пользовательский интерфейс в таких приложениях создается при помощи стандартных web-технологий. Их не нужно загружать из магазина приложений, т.к. они работают через браузер, но при этом требуют подключения к Интернету. Главной особенностью web-приложений является их кроссплатформенность — они могут работать на всех устройствах без дополнительной адаптации. Однако скорость работы приложения, как правило, ограничивается качеством Интернет-соединения. Кроме того, независимо от платформы, web-приложения не могут использовать ПО устройства, на котором работают.

В Tizen web-приложения используют Web API, который реализует стандартную структуру проекта web-приложения с базовыми элементами. То есть, приложение, написанное с использованием Web API, представляет собой стандартный сайт с файлом **index.html** и отдельными директориями для ресурсов — JavaScript, CSS, картинок и звуковых ресурсов. Такой подход делает разработку web-приложений под Tizen максимально интуитивной и позволяет без особого труда писать простые приложения с использованием высокоуровневых языков.

Платформа Tizen позволяет также разрабатывать и гибридные приложения — нативные приложения со встроенным HTML. Они сочетают в себе такие функции нативных и web-приложений, как кроссплатформенность и возможность использования ПО устройства. Гибридные приложения могут быть загружены через магазины приложений и при этом имеют возможность независимого обновления, поэтому требуют подключения к Интернету.

Архитектура Tizen

Кратко рассмотрим архитектуру Tizen. На рис. 1 представлена архитектура Tizen, поддерживающая 2 типа приложений.

В основе архитектуры Tizen лежит ядро Linux и драйверы устройств. Основная подсистема состоит из библиотек с открытым исходным кодом и набора API для обеспечения работы Web и нативного окружения.

Компоненты ядра представлены следующими подсистемами:

» **Application Framework** Реализует управление приложениями, в том числе запуск других приложений. Обеспечивает уведомление приложений об основных системных событиях, таких как нехватка памяти, низкий уровень заряда батареи, поворот экрана и push-уведомления.

» **Base** Содержит основные системные библиотеки Linux, реализующие такие ключевые функции, как поддержка баз данных, интернационализация и XML-парсинг, а также базовые функции, которые включают управление общими файлами и архивными ZIP-файлами, обработку событий и организацию данных.

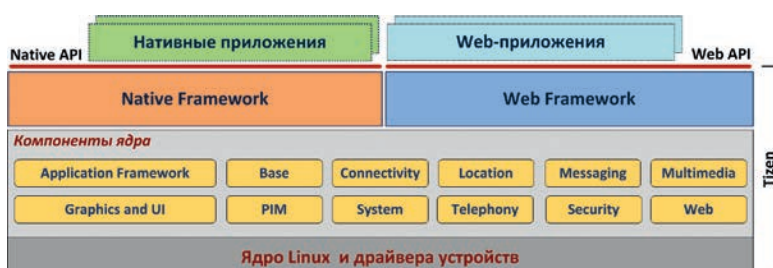
» **Messaging** Реализует функциональность, связанную с SMS, MMS и email: обмен сообщениями, включающие их создание, отправку и чтение, а также управление сообщениями в хранилище сообщений. Поддерживается только для мобильных приложений.

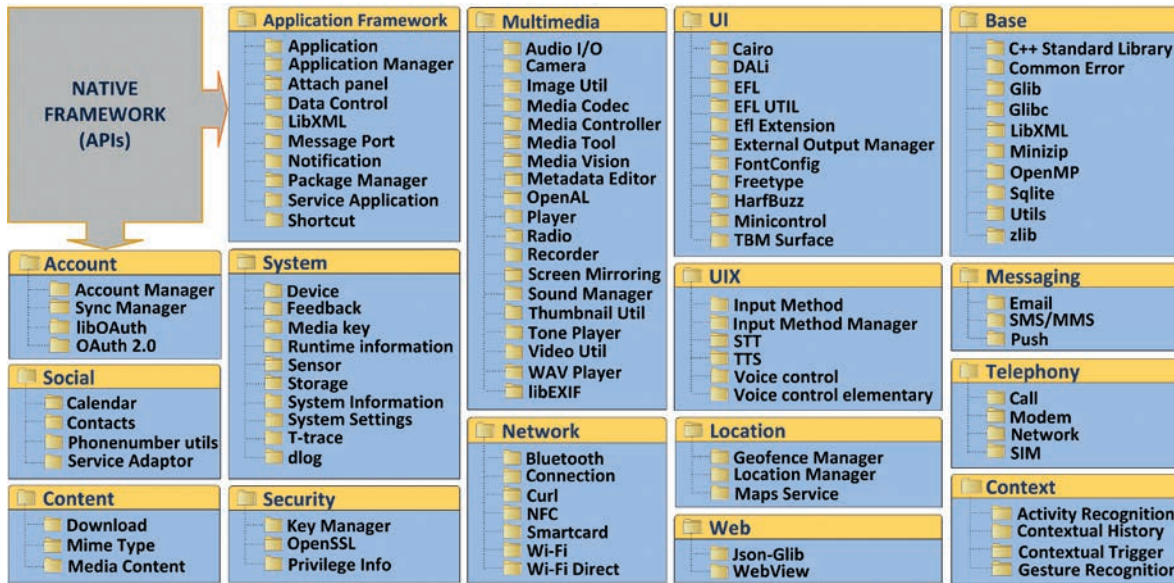
» **Connectivity** Предоставляет функциональность для управления беспроводными подключениями к другим устройствам и сетям с использованием 3G, Wi-Fi, HTTP, Bluetooth и NFC (Near Field Communication). Также обеспечивает возможность загрузки контента и обмена данными с приложением.

» **System** Обеспечивает управление различными устройствами и получение информации о системе. Реализует следующие функции:

- » предоставление интерфейсов для доступа к таким устройствам, как сенсоры и дисплей;
- » управление электропитанием;
- » обновление системы;
- » мониторинг подключаемых устройств (USB, гарнитура, зарядное устройство и т.д.) и обработка связанных с ними событий;
- » предоставление доступа и отслеживание информации об устройстве, такой как уровень заряда батареи, доступная память, номер версии, наименование модели и используемой сотовой сети.

» Рис. 1. Архитектура Tizen, поддерживающая 2 типа приложений.





► Рис. 2. Структура Tizen Native Framework.

- **Location** Предоставляет сервисы LBS — сервисы, основанные на местоположении (Location-based services). Включает получение информации о местоположении, геокодирование, спутниковую информацию и GPS. Подсистема основана на сервисе GeoClue, который передает информацию о местоположении от различных источников позиционирования, таких как GPS, WPS (Wi-Fi Positioning System), Cell ID и сенсоров.
- **Multimedia** Обеспечивает поддержку мультимедиа-контента (видео, аудио, изображений и VoIP), а кроме того, управление метаданными.
- **Security** Отвечает за безопасность в системе и поддерживает реализацию таких функций, как контроль доступа, управление сертификатами и т. д.
- **Graphics and UI** Функциональность, связанная с графикой и пользовательским интерфейсом, включая набор графических библиотек EFL (Enlightenment Foundation Libraries) и системы управления окнами на основе X11, а также OpenGL.
- **Telephony** Реализует функциональность, относящуюся к связи и работе модема (UMTS, CDMA, GSM).
- **Web** Реализует Tizen Web API и включает движок WebKit для разработки и поддержки web-приложений.
- **PIM (Personal Information Management)** Реализует управление такими компонентами, как календарь, контакты и задачи, а также отвечает за синхронизацию данных в устройстве.

Доступ к перечисленным выше подсистемам из приложений осуществляется посредством двух платформ: Web Framework и Native Framework.

Tizen Native Framework

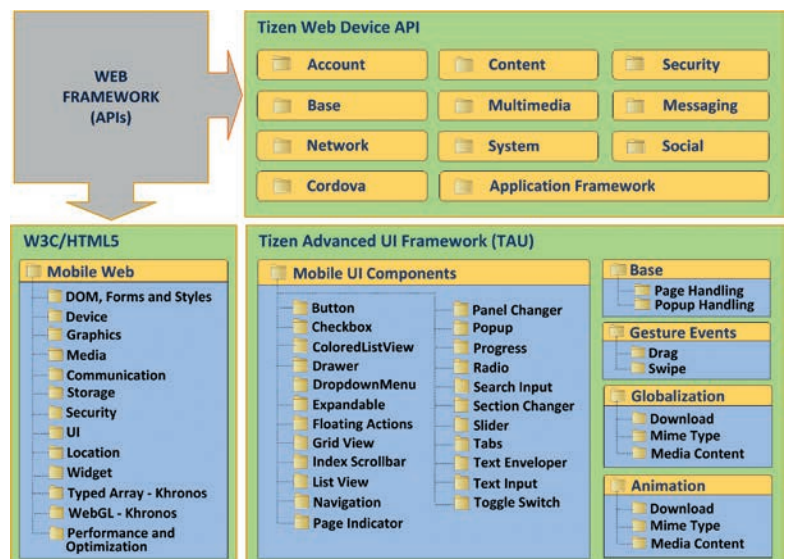
Tizen Native Framework обеспечивает доступ нативных приложений к функциональности описанных выше компонент ядра с помощью набора интерфейсов (Native APIs), реализующих: управление жизненным циклом приложения; работу с основными типами данных; управление контентом; работу с 2D- и 3D-графикой, текстом, изображениями, файлами, базами данных, аудио и видео, сообщениями различного типа; геопозиционирование; сетевое взаимодействие; криптографию; управление доступом и работу с сертификатами; управление внешним интерфейсом устройства, а также встроенными аппаратными возможностями; телефонию; управление графическим интерфейсом как на стандартном, так и на продвинутом уровнях; работу с сенсорами; управление учетными записями; работу с Web; и многое другое.

Полная функциональность Tizen Native Framework представлена на рис. 2.

Tizen Web Framework

Tizen Web Framework — платформа, предоставляющая набор интерфейсов (Web APIs) для разработки Web-приложений, включающий W3C (HTML5), Khronos WebGL и интерфейсы для доступа к функциональности устройства. Tizen Web Framework включает следующие компоненты (рис. 3):

- **Tizen Web Device API** Обеспечивает доступ к различным функциям устройства из Web-приложений, позволяя управлять жизненным циклом приложения, обмениваться данными, совершать платежи через NFC и т. д. Основан на JavaScript.
- **W3C/HTML5 API** Позволяет включать в приложение графику, мультимедиа, данные о местоположении, управлять соединениями и безопасностью, а также создавать пользовательский интерфейс приложения с использованием платформы Tizen Advanced UI Framework или HTML и CSS. Использует движок WebKit.
- **Tizen Advanced UI Framework (TAU)** Платформа, предназначенная для разработки пользовательского интерфейса приложения. Предоставляет такие инструментальные средства, как компоненты интерфейса (кнопки, флажки, переключатели, выпадающие меню, списки, текстовые поля и многие другие), события, различные эффекты и анимации, которые могут понадобиться при разработке Web-приложений. **LXF**



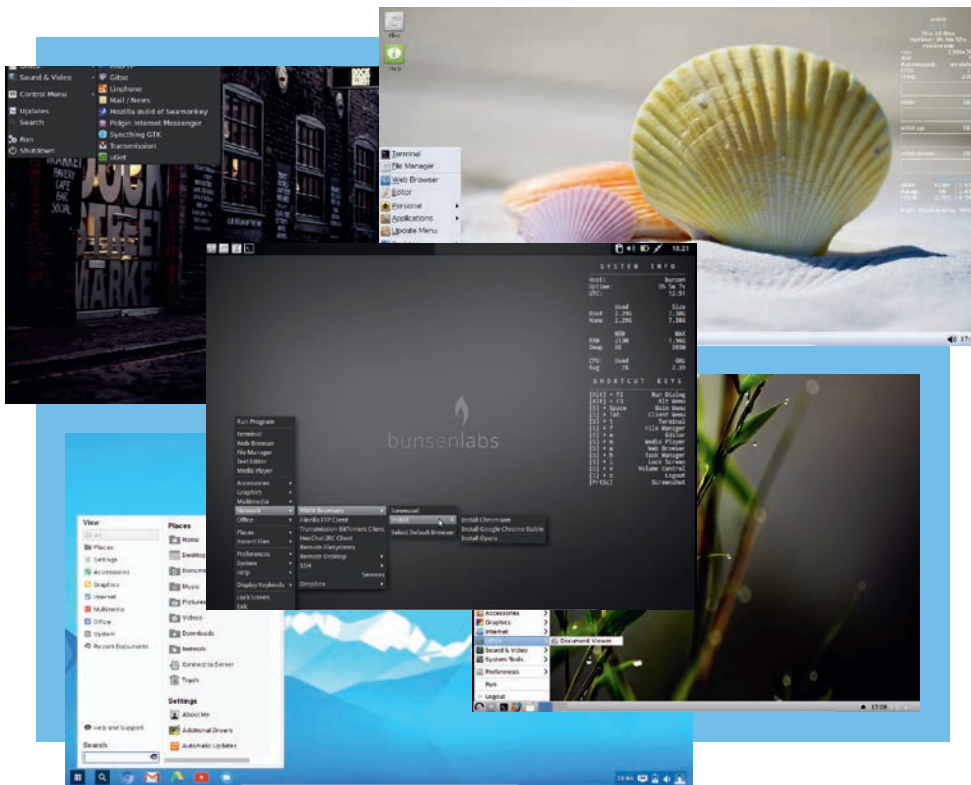
► Рис. 3. Структура Tizen Web Framework.

Сравнение

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

Дистрибутивы-легковесы

Отчаявшись заставить пахать свою рабочую клячонку, Маяк Шарма разыскивает дистрибутив, нетребовательный к ресурсам.



Про наш тест...

Каждый раз, когда заходит речь об операционных системах для старых компьютеров, неизбежно размывается грань между понятиями «минимальный» и «легковесный». Легковесность дистрибутива просто означает количество требуемых ему системных ресурсов. А минимальность относится к размеру файла ISO и означает, что в нем установлено очень мало приложений по умолчанию. Правда и то, что многие легковесные ОС часто минимальны. В нашем Сравнении мы ищем легковесные дистрибутивы, которые могут быть, а могут и не быть минимальными в плане своего выбора пакетов, но при этом весьма скромны в плане использования ресурсов компьютера. Чтобы найти самое подходящее оборудование для нашего Сравнения, мы покопались в темных недрах нашего гаража и нашли одноядерный настольный ПК 2,3 ГГц с 40-ГБ диском и 2 Гб ОЗУ. [Ред.: — Да это роскошь!]

Наша подборка

- » antiX
- » Bunsen Labs
- » Cub Linux
- » LXLE
- » WattOS

На легковесные альтернативы как отдельных приложений, так и целых дистрибутивов спрос существовал всегда. Но недавнее появление многофункциональных и алчных до ресурсов дистрибутивов подстегнули усилия по поиску возможностей применения старых и в ином случае непригодных рабочих лошадей. Даже современный рабочий стол Linux потребляет немало ресурсов, и у вашего оборудования может не хватить силенок для работы Unity или Gnome 3.

К пользователям, владеющим относительно недавно устаревшим оборудованием, присоединились те, кто по-прежнему

верен своим компьютерам прошлого десятилетия. Вряд ли справедливо заставлять их выбрасывать свои устаревшие машины, если им всего-то и надо, что снова по Интернету, редактировать текст да смотреть видео. Вбивать им в глотку самый современный многоядерник — это просто неправильно.

Многие разработчики открытого кода ищут возможность снова использовать

старое оборудование. Существуют легковесные приложения, потребляющие лишь малую толику запросов на ресурсы своих полнофункциональных родичей, и дистрибутивы, отлично работающие на компьютерах со скромными спецификациями. В нашем Сравнении мы рассмотрим некоторые из лучших дистрибутивов, с самого начала разработанных для работы со скудными ресурсами устаревших компьютеров.

Многие разработчики открытого кода ищут возможность снова использовать старое оборудование.

Программы по умолчанию

Избавят от общения с менеджером пакетов?

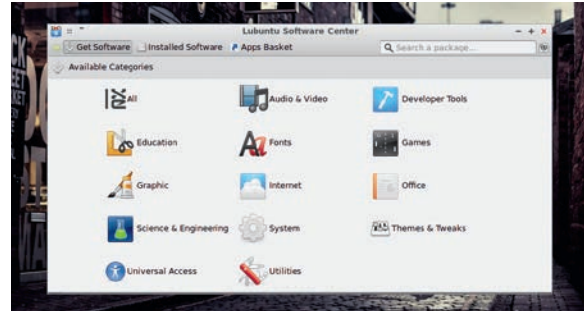
Как и популярные дистрибутивы, специализированные дистрибутивы могут предлагать ряд предустановленных программ. Однако, в отличие от обычных настольных дистрибутивов, просто предлагающих последнюю стабильную версию самых популярных приложений, дистрибутивы для оборудования постарше должны тщательно продумывать, какие приложения включать, и выбор программ обычно диктуется их целями.

Например, Cub Linux предлагает очень мало приложений, поскольку он создает среду наподобие хромбука. Дистрибутив включает браузер *Chromium* и пару оффлайн-приложений, плюс ряд основных приложений Google, таких, как Gmail, Docs, Drive и YouTube.

WattOS тоже содержит ровно столько приложений, чтобы стать пригодным, ведь цель дистрибутива — предоставить

базовую систему, далее настраиваемую согласно вашим нуждам. Bunsen Labs немного лучше, поскольку предлагает несколько приложений, обычно встречающихся в главных дистрибутивах. В дистрибутиве нет программы запуска приложений, но имеются такие полнофункциональные приложения, как *VLC* и *LibreOffice Writer*, и опционно можно установить и другие компоненты полноценного офисного комплекта, если облегченный *Gnumeric* или онлайнный Google Docs вам не подходят. Резкий контраст им, два других дистрибутива в нашем Сравнении применяют совершенно иной подход, наполняя свои закрома приложениями, чтобы по умолчанию обеспечить полноценную работу. И LXLE, и antiX не только включают тонны приложений, но и находят место для нескольких игр.

Список приложений LXLE включает несколько полнофункциональных, таких



» LXLE подходит к выбору приложений утилитарно, поэтому может не подойти для более старого оборудования.

как *LibreOffice*, *GIMP* и видеоредактор *OpenShot*, наряду с легковесными альтернативами, такими как музыкальный плеер *Guayadeque*, программа просмотра изображений *Mirage*, перекодировщик *Arista* и т.д. AntiX тоже включает популярные приложения вроде *Firefox* и *LibreOffice*, но дополняет их легковесными: *mplayer* из Gnome, клиент электронной почты *Claws* и web-браузер *Dillo*, и заходит еще на шаг дальше, включая редкие, но удобные приложения: например, сервер общего сетевого доступа к файлам *Droppy*. Хотя antiX, возможно, превосходит LXLE по числу приложений, подборка в обоих дистрибутивах удовлетворит потребности самых разных пользователей.

Вердикт

- antiX ★★★★★
- LXLE ★★★★★
- Bunsen Labs ★★★★★
- WattOS ★★★★★
- Cub Linux ★★★★★

» Дистрибутивы многое предусмотрели, исключение составили Cub Linux и WattOS.

Пригодность

Годятся ли они для старого оборудования?

AntiX и WattOS имеются в разных вариациях. WattOS производит издание *MicroWatt* для тех, кому нужен еще более легкий и минимальный дистрибутив, собираемый с нуля. AntiX предлагает в своих разновидностях ряд легковесных рабочих столов и менеджеров окон, а в его основном издании достаточно

приложений, чтобы помочь вам собрать свой дистрибутив с нуля. Полное же издание использует *IceWM* вместе с менеджером файлов *Rox* — это один из самых легких дистрибутивов, дополненный специально отобранной коллекцией приложений.

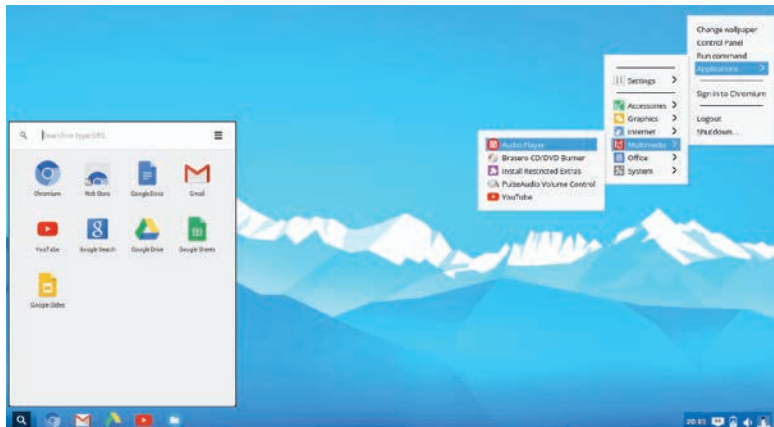
WattOS создан из Ubuntu 16.04.1 LTS посредством очень вдумчивого добавления

пакетов, потребляющих не слишком много ресурсов, но создающих вполне пригодный к использованию дистрибутив. Он использует *LXDE* и менеджер окон *i3*, экономии ради. WattOS тоже сосредоточен на малом расходе ресурсов и включает такие утилиты, как *PowerTOP*, для оптимизации энергопотребления.

Bunsen Labs, с другой стороны, использует менеджер окон *Openbox* с панелью *tint2* и системным монитором *Conky* на рабочем столе. Проект предлагает дополнительную 32-битную сборку, не требующую поддержки PAE в CPU, что делает ее дружелюбной к старым компьютерам.

Аскетичный Cub Linux, сосредоточенный в облаке, также использует *Openbox*, однако в индивидуально настроенной версии, и дополняет его инструментами из *LXDE*, Gnome и *Xfce* с рядом других легковесных утилит.

LXLE, предназначенный в первую очередь для устаревших компьютеров, основан на *Lubuntu*, и дополнительно оптимизирует *LXDE*. Он использует программу запуска приложений *Fehstart* вместе с легковесными, но полнофункциональными приложениями из других проектов, например, текстовым редактором *Pluma* из рабочего стола *Mate*. В нем также имеется *iCareSystem Core*, который автоматизирует основные процессы администрирования системы.



» Cub Linux изначально был известен как *Chromixium OS*, пока в 2015 г. Google не вынудил его изменить название.

Вердикт

- antiX ★★★★★
- Bunsen Labs ★★★★★
- Cub Linux ★★★★★
- LXLE ★★★★★
- WattOS ★★★★★

» Все дистрибутивы немало потрудились, чтобы стодиться для старого оборудования с малыми ресурсами.

Производительность и удобство

Хороши ли они при повседневной работе?

Вы найдете множество дистрибутивов с компонентами, которые не слишком напрягут ваше старое оборудование. Однако требуются некоторые усилия, чтобы сколотить эти компоненты вместе без зазоров и превратить старый ржавый хлам в настольный ПК, который вы сможете использовать для повседневной работы.

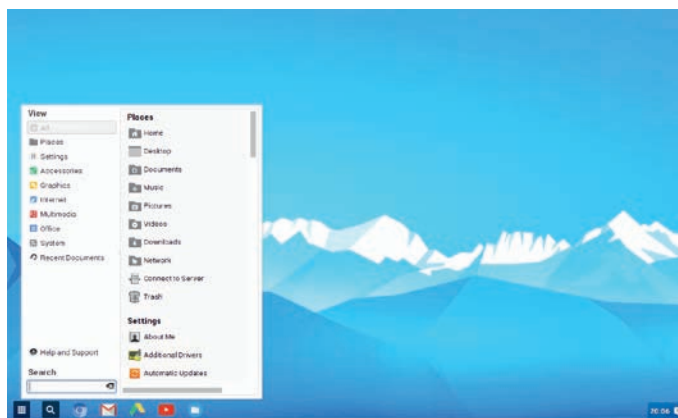
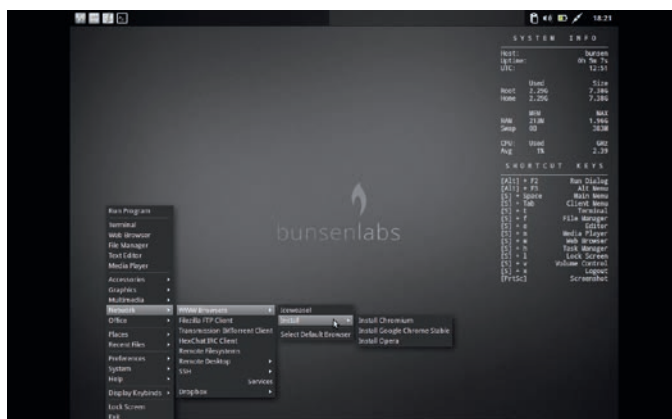
Для обычного дистрибутива количество приложений в нем напрямую сказывается на его удобстве в использовании. Однако дистрибутив для более старого оборудования — это нечто большее, чем просто сумма приложений. Вот почему мы ищем дистрибутив, который настроил бы все возможные аспекты базовой системы для обеспечения

максимума производительности, а это важный показатель при измерении пригодности дистрибутива для использования на старом оборудовании. Кроме того, будем реалистами: дистрибутивы, которые усложняют для пользователей возможность поиграть, транслировать поток в Интернет или получить доступ к содержимому диска, подходят не всем.

Bunsen Labs ★★★★★

Дистрибутив является продолжением проекта CrunchBang Linux, разрабатываемым сообществом. Bunsen Labs использует на рабочем столе серую тему с панелью того же цвета и виджетом статуса *Conky*, расположенным справа. Прелесть Bunsen в том, что вы можете управлять всем дистрибутивом с клавиатуры, с помощью списка часто используемых горячих клавиш, который также отображается на рабочем столе.

Здесь нет программы запуска приложений, и для их вызова вам придется щелкать правой кнопкой по рабочему столу. Удобству дистрибутива сильно способствуют разнообразные пункты меню, обеспечивающие установку в одно нажатие. Bunsen использует скрипт *welcome* для настройки разных аспектов новой установки. Скрипт очень продуманный и многословный, чтобы пользователи могли понять, что им предлагается, но работает пошагово, что отнимет у вас некоторое время, в зависимости от вашего выбора.



Cub Linux ★★★★★

Цель дистрибутива Cub Linux — воссоздать Chrome OS, но без предложений от Google и ряда ограничений, и он потрясающе хорошо справляется с этой задачей. Приятный рабочий стол минималистичен и включает строку состояния внизу, с программой запуска приложений *Chromium* вместе со ссылками на популярные web-приложения Google ярлык для уместно названного менеджера файлов *File*. Web-приложения работают отлично, и некоторые, например, Google Docs, позволяют работать также и оффлайн. Можно щелкнуть правой кнопкой по рабочему столу, чтобы вывести меню приложений, и добавить приложение в строку состояния. Хотя дистрибутив предназначен для использования онлайн-приложений, в нем есть и обычные настольные приложения для выполнения необходимых задач и администрирования. Чтобы извлечь максимум пользы из дистрибутива, вы можете использовать *Cub Software Center*, с целью оснастить его добавочными настольными приложениями.

Связь с истоками

Как они используют наследие предшественников?

Все дистрибутивы в этом Сравнении основаны на одной из ветвей проектов Ubuntu или Debian. Для начала, *antiX* использует по умолчанию репозитории Debian Stable, но вы можете настроить его на Debian Testing и Unstable. Дистрибутив использует большие репозитории программ Debian, но, в отличие от других производных дистрибутивов в этом Сравнении, *antiX* использует не программу установки Debian, а свою собственную.

Bunsen Labs тоже основан на Debian Stable, но, в отличие от *antiX*, взял программу

установки Debian. Bunsen предусматривает устанавливаемый носитель live для 32- и 64-битных систем, но в его репозиториях отыщутся и пакеты для архитектуры Armhf (ARM).

Три других дистрибутива все основаны на Ubuntu и используют его программу установки. *WattOS* включает среду ядра DKMS и поддержку унаследованных устройств Broadcom для более широкой поддержки оборудования, и использует *Gufw* для настройки брандмауэра. *Cub Linux* использует легковесные компоненты

из таких официальных ответвлений Ubuntu, как *Lubuntu*, *Xubuntu* и *Gnome Ubuntu*, и предоставляет пользователям возможность скачать дополнительные кодеки одним щелчком.

LXLE основан на минимальной основе *Lubuntu 16.04.1* и использует механизм обновления безопасности Ubuntu. Он также использует более широкую экосистему Ubuntu и включает приложения из других разновидностей, например, *bodhibuilder* из *Bodhi Linux* для резервного копирования установки в дистрибутив live.

Вердикт

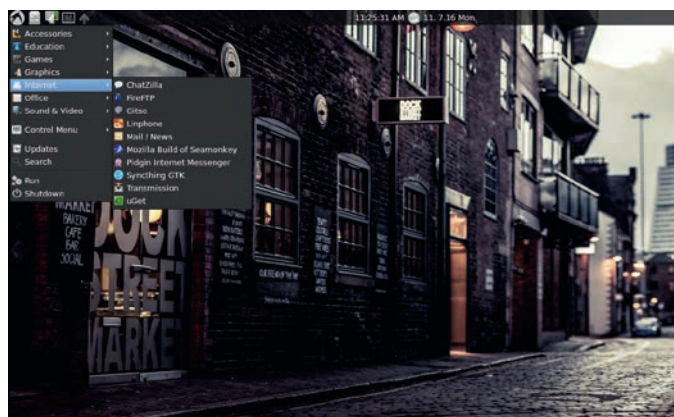
- Bunsen Labs ★★★★★
- Cub Linux ★★★★★
- LXLE ★★★★★
- antiX ★★★★★
- WattOS ★★★★★

» Все дистрибутивы отлично используют выбранный ими вышележащий дистрибутив, и все делают это по-своему.

LXLE ★★★★★

LXLE расшифровывается как *Lubuntu Extra Life Extension*; это расширение дистрибутива *Lubuntu* с помощью ряда дополнительных настольных приложений. И хотя LXLE довольно-таки шустрый по сравнению с обычными настольными дистрибутивами, на компьютерах с 1 ГБ ОЗУ он способен периодически впадать в легкую летаргию.

Рабочий стол дистрибутива размещает панель сверху экрана, где находится программа запуска приложений, система уведомлений, системные значки и ярлыки для некоторых приложений. Есть также скрытая панель внизу, где размещается переключатель рабочих столов и где отслеживаются открытые окна и отображаются в стиле *Expose*. Чтобы увеличить удобство рабочего стола в использовании, вы можете добавить часто используемые приложения на нижнюю панель, что особенно радует с учетом перегруженного списка приложений LXLE.

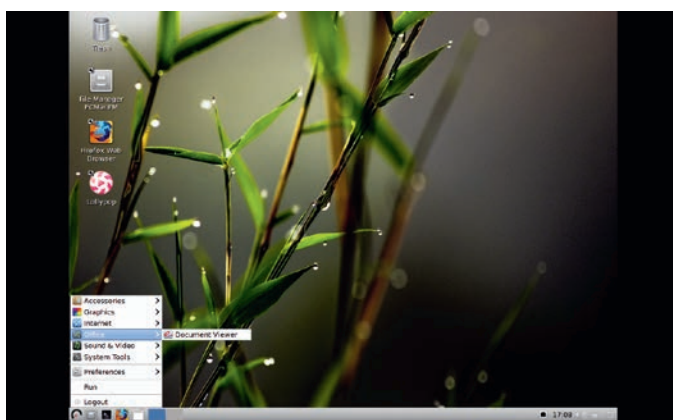


antiX ★★★★★

Основанный на *Debian* дистрибутив совершает чудо, загружаясь в симпатичный менеджер окон *IceWM* со значками на рабочем столе. Внизу экрана расположены меню приложений, переключатель задач и системный лоток. Можно также щелкнуть правой кнопкой по рабочему столу, чтобы вывести меню приложений. Дистрибутив использует индивидуальную программу установки, достаточно многословную, чтобы ее мог использовать неопытный пользователь, несмотря на то, что для разбиения диска на разделы она использует *GParted*. Правда, двойную загрузку на старой машине сделают немногие, и с программой установки очень приятно иметь дело, если вы переустанавливаете *antiX* на весь диск. Единственная оплошность, способная стать проблемой для новичков — отсутствие подсказок в меню *Applications*. Это особенно важно, поскольку *antiX* использует немало нестандартных (но полезных) приложений, которые не похожи на популярные.

WattOS ★★★★★

Дистрибутив загружает симпатичный рабочий стол со значками для некоторых приложений. Это стандартная раскладка для *LXDE* с панелью внизу, где размещается меню приложений. В дистрибутиве нет какой-то явной индивидуальной настройки, которая внешне отличала бы его от обычного дистрибутива на базе *LXDE*. Навигация по меню не оставляет затруднений, поскольку в первичной установке приложений почти нет. Однако вы не можете разместить приложения на панели, а это несколько неудобно. Основная проблема с использованием — отсутствие дружелюбной к пользователю программы установки приложений для настройки дистрибутива. *Synaptic* — прекрасное приложение, но принесет мало пользы в руках неопытного новичка. Дистрибутив использует модифицированную программу установки *Ubuntu*, которая новичкам понравится. После установки *WattOS* работает, как заявлено, и его время загрузки и выключения намного меньше, чем у обычного настольного дистрибутива.



В поисках поддержки

У вас проблемы? И куда обратиться за помощью?

Некоторые дистрибутивы активно делятся информацией и помогают своим пользователям решать проблемы. Например, *WattOS* размещает в своем блоге подробности о релизе, и по конкретным релизам можно найти поддержку на форумах, которые весьма активны, и пообщаться с разработчиками, используя форму помощи на сайте.

Документация отсутствует и у *Cub Linux*, но у него весьма активные форумы с удобным пересечением тем форума, включая раздел распределения помощи и еще один

для компоновки документации. Есть раздел на сайте LXLE со ссылками на внешние видео и обзоры. Раздел поддержки указывает на форумы, wiki, списки рассылки и каналы IRC дистрибутива LXLE и рабочего стола LXDE. Форумы LXLE являются лучшим источником документации, где помимо способов решения проблем есть подсказки, хитрости и руководства.

И *Bunsen*, и *antiX* включают указатели на документацию внутри самого дистрибутива. В *Bunsen* есть подменю ссылок, которое укажет пользователям на форумы

поддержки, *Debian Handbook*, *wiki Arch Linux* и другие источники вспомоществования в трудную минуту.

Весь сайт *antiX* являет собой *wiki*, которая подробно объясняет все изменения в релизах. Есть также множество статей *HowTo* для ознакомления пользователей с основными компонентами и ссылки на видео, созданные сообществом и командой и объясняющие разные функции. Пользователи могут задать вопросы на форумах, где есть темы для вопросов новых пользователей.

Вердикт

- antiX ★★★★★
- Bunsen Labs ★★★★★
- Cub Linux ★★★★★
- LXLE ★★★★★
- WattOS ★★★★★

» У всех проектов есть ресурсы поддержки, в помощь решению проблемных ситуаций.

Управление пакетами

Надо ли вам расширять дистрибутив?

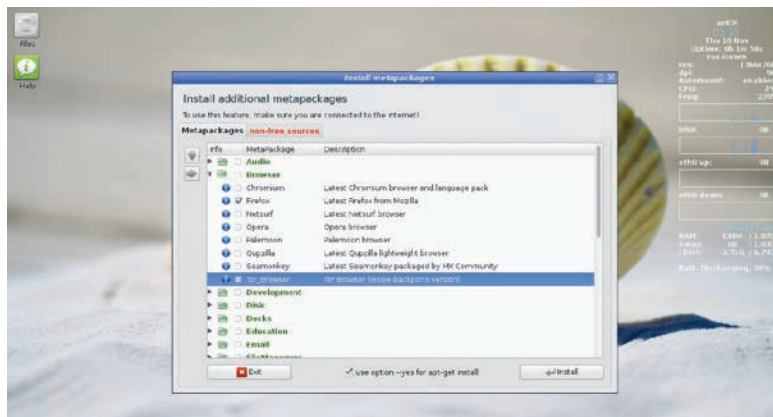
Поскольку все эти дистрибутивы опираются на две самых популярных основы, вы можете расширить их самыми разными приложениями. Увы, некоторые из них будут работать прекрасно, а некоторые — нет. Например, WattOS включает только менеджер пакетов *Synaptic*, что странно для дистрибутива, предназначенного для расширения

пользователем. Основанные на Ubuntu Cub Linux и LXLE включают центры приложений попроще, а также *Synaptic*. Кроме как в репозиториях Ubuntu, дистрибутивы находят программы в своих собственных репозиториях.

Bunsen также содержит собственные репозитории и облегчает установку популярных приложений через добавление

пунктов для поиска и установки их в само меню приложений. Кроме того, дистрибутив не забывает сообщать пользователям, что они не должны устанавливать программы извне официальных источников, и, в частности, не рекомендует добавлять Ubuntu PPA.

Как и другие, antiX использует собственные репозитории вместе с репозиториями Debian. Он тоже использует менеджер пакетов *Synaptic* и не предлагает привлекательного центра приложений, но всё же получает более высокий рейтинг по сравнению с другими благодаря своей программе установки метапакетов. Она поможет установить популярные приложения в более чем двух десятках категорий. Благодаря этому доморощенному инструменту можно установить отдельные приложения, такие как плеер Spotify и клиент Steam, а также полный web-сервер Apache вместе с PHP и MySQL, одним нажатием. Инструмент настолько широк, что вам просто незачем обращаться к менеджеру пакетов, если только вы не решите установить какую-то определенную библиотеку.



Чудесные метапакеты делают antiX подходящим для многих аппаратных конфигураций.

Вердикт

antiX ★★★★★
 Cub Linux ★★★★★
 Bunsen Labs ★★★★★
 LXLE ★★★★★
 WattOS ★★★★★

» Программа установки метапакетов antiX делает его доступным для новых пользователей.

Индивидуальная настройка

Что делает их уникальными?

Все дистрибутивы в нашем Сравнении прошли некий путь, чтобы добиться наилучшего соответствия ПК с ограниченными аппаратными ресурсами. Например, WattOS тщательно потрудился над тем, чтобы убрать всё лишнее и предложить минимальное число приложений, сохраняя работоспособность дистрибутива. Он использует модифицированный рабочий стол LXDE с менеджером окон i3 и предлагает утилиту PowerTOP, которая помогает замерить, объяснить и минимизировать энергопотребление компьютера.

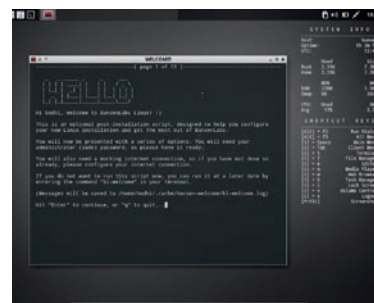
Cub Linux предлагает свой индивидуальный рабочий стол с компонентами из разных рабочих столов и менеджеров окон, а также позволяет добавлять традиционные меню из проекта Cardapio и включает индивидуальную панель управления, дающую доступ к некоторым утилитам для расширенных задач администрирования.

Дистрибутиву LXLE также сопутствует отлично поддающийся индивидуальной настройке рабочий стол LXDE, который использует индивидуальные скрипты для

функций типа Expose и Quick launch apps [Быстрый запуск приложений]. Для пушера удобства менеджер файлов PCManFM в дистрибутиве также оснащен несколькими удобными расширениями, например, способностью рассчитывать MD5sum файла и помещать изображение в качестве обоев.

Bunsen Labs идет с менеджером окон OpenBox и позволяет изменять привязки клавиш для управления им. Все аспекты дистрибутива настраиваются в меню Preferences. Bunsen содержит целый ассортимент приятных тем GTK 2/3, обоев и настроек Conky, и отлично поддается индивидуальной подстройке к вашим рабочим потребностям. В темах форума проекта даже есть раздел, где пользователи делятся своими рабочими столами и настройками Conky. Более того, имеются индивидуальные ссылки для установки дополнительных программ в разных категориях в меню приложений Bunsen, что очень удобно и сразу выделяет дистрибутив среди его собратьев.

И, наконец, antiX, который тоже использует солидную индивидуальную настройку,



Bunsen включает очень полезный (но довольно сложный) пост-установочный скрипт приветствия, который поможет индивидуально настроить важные аспекты установки.

от предложения нескольких рабочих столов и менеджеров окон до отдельных элементов в его собственной панели управления. Фактически, одно из самых главных преимуществ дистрибутива — доморощенная панель управления antiX, используя которую, вы можете модифицировать буквально все аспекты своего компьютера и операционной системы. Например, можно модифицировать внешний вид вашей установки, в том числе темы, меню, обои, а также настроить блокировщик рекламы antiX, создать образ раздела и настроить automount с помощью индивидуальных модулей, предусмотренных на контрольной панели.

Вердикт

antiX ★★★★★
 Bunsen Labs ★★★★★
 LXLE ★★★★★
 Cub Linux ★★★★★
 WattOS ★★★★★

» Все дистрибутивы по-своему допускают индивидуальную настройку.

Легковесные дистрибутивы

Вердикт

Все дистрибутивы в нашем Сравнении можно использовать, чтобы вернуть неиспользуемый компьютер к активной жизни. Выделяет их то, как они реализуют функции на ограниченном оборудовании.

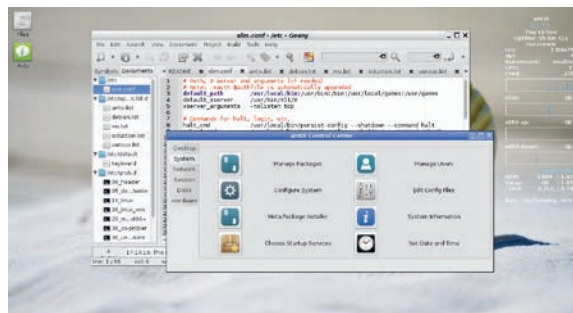
Например, Cub Linux лучше всего работает для тех, кому достаточно облачных приложениях. Дистрибутив расширяется без особого труда, но его разрабатывали без учета излишнего багажа. То же самое касается и WattOS. Дистрибутив создает функциональную систему, но в первую очередь благодаря мудрому выбору приложений. Он быстро набирает вес и впадает в летаргию, как только вы начинаете нагружать в него приложения.

Отчасти это верно и для LXLE. Дистрибутив расширяет удобство в использовании Lubuntu, отнимая при этом меньше ресурсов, чем, скажем, Xubuntu. Но, отлично работая на недавно устаревших компьютерах, на действительно старых он не слишком хорош.

Вы не столкнетесь с этой проблемой в Bunsen Labs или antiX. Оба дистрибутива располагают достойным числом приложений и при этом умудряются оставаться довольно умеренными в потреблении ресурсов. Оба дистрибутива также отлично поддаются индивидуальной настройке, и к их скромным рабочим столам надо привыкнуть, однако это не самая высокая цена за то, чтобы вернуть вашу старую рабочую лошадку к активной жизни.

В Bunsen можно без особых проблем установить любые полезные приложения благодаря его разумным пунктам меню. Во внешнем виде дистрибутива также присутствует логичность, которая позволяет ему выглядеть более изящно. У проекта сильное наследие и очень преданное сообщество пользователей и разработчиков.

Однако Bunsen уступает antiX, потому что antiX умудряется соблюсти верный баланс между легковесностью и дружелюбием к пользователю. Он прост в использовании, но при этом не слишком



» antiX легко использовать, подстраивать и индивидуально настраивать, независимо от вашего опыта работы онлайн.

нагружает оборудование. AntiX работает в рамках весьма жестких ограничений, но предлагает достаточно гибкости, чтобы адаптировать установку к требованиям пользователя. Выбор пакетов в установке по умолчанию достаточно широк, чтобы позаботиться о потребностях большого числа пользователей. Для всего остального есть очень удобная программа установки метапакетов, которая на ближайшее обозримое будущее избавит вас от необходимости обращаться к менеджеру пакетов.

AntiX: баланс между легковесностью и дружелюбием к пользователю.

I antiX ★★★★★
 Версия: 16.0 Сайт: www.antix.mepis.org Лицензия: GPL v2
 » Идеальный дистрибутив типа «установите и забудьте» для всех видов старого оборудования.

IV WattOS ★★★☆☆
 Версия: R10 Сайт: www.planetwatt.com Лицензия: Разные
 » Хорошая опция для любителей всё делать самим, желающих заняться своей установкой.

II Bunsen Labs ★★★★★☆
 Версия: Hydrogen Сайт: www.bunsenlabs.org Лицензия: Разные
 » Может быть легко расширен добавочными популярными приложениями.

V Cub Linux ★★★☆☆
 Версия: 1.0RC Сайт: www.cublinux.com Лицензия: Разные
 » Если вас не смущают онлайн-приложения Google, то этот дистрибутив для вас.

III LXLE ★★★★★☆
 Версия: 16.04.1 Сайт: www.lxle.net Лицензия: Разные
 » Утилитарный дистрибутив, отлично смотрится на недавно устаревших компьютерах.

Обратная связь
 Вы пытались воскресить старый ПК с помощью легковесного дистрибутива? Расскажите нам о своем опыте на lx.f.letters@futurenet.com.

Рассмотрите также...

Есть множество проектов, работающих над возвратом к жизни старых компьютеров. Если вы готовы выйти из своей зоны комфорта, то есть ряд нестандартных опций, например, SliTaz, Tiny Core Linux и Puppy Linux.

Имеется еще VectorLinux, доступный в нескольких вариантах, и среди них — Light Edition с IceWM.

Не стоит пропускать также Lubuntu и Xubuntu, версии проекта Ubuntu.

Существует и Emmabuntus, созданный для использования на восстановленных компьютерах, которые собирают благотворительные организации из пожертвованных запчастей. В числе альтернатив облачному Cub Linux — Peppermint OS, где намного

больше онлайн-приложений, чем в Cub. Помимо обычных Google apps, разработано несколько таких онлайн-инструментов, как Editor от pixlr.com и ряд игр. Кроме того, прилагается замороженное приложение под названием Ice — это простая утилита, способная превращать сайты в web-приложения и включать их в меню. **LXF**

Идем в Linux

Джонни Бидвелл поставил Linux на всех своих системах и впадает в лирику, объясняя, почему вам тоже надо это сделать и как это делается. Поставьте-ка чайник.



Пользователем Linux может стать любой. По сути, любой счастливый обладатель какой-либо потребительской технологии или посетитель Интернет,

вероятнее всего, уже пользователь Linux. Linux находится в сердцевине телефонов Android, умных телевизоров, устройств IoT и большинства web-серверов мира. Linux также работает на 99% крупнейших в мире суперкомпьютеров. Но вы можете использовать Linux и в качестве настольной операционной системы — это гораздо проще, чем было раньше.

Можно без всякого риска попробовать Linux прямо с нашего диска или на виртуальной машине. Есть рабочие столы, способные удовлетворить потребности любителей Windows и MacOS, и есть ра-

Можно без всякого риска попробовать Linux прямо с нашего диска или на виртуальной машине.

бочие столы для людей со вкусом [Ред.: — Сразу за пустой треп]. Свобода ПО никогда еще не была так важна, поскольку корпорации пытаются запереть нас в своих закрытых экосистемах, ущемить

нашу конфиденциальность и ограничить наши эксперименты со своим оборудованием. Linux не заставит вас скачать и установить целую новую ОС, за которой вы не гнались, и не прекратит под-

держки вашего более старого оборудования. При установке приложений в Linux вам не всучат нежелательных довесков, и не придется в отчаянии носиться по Сети в поисках официальной ссылки

на скачивание, поскольку управление программами делается куда более связно. Тысячи превосходных свободных программ можно скачивать прямо с рабочего стола. Давайте же исследуем этот мир...

Что такое Linux?

Перейти на Linux — это, конечно, хорошо, но будет разумнее, если сначала мы в точности объясним, во что же мы предлагаем вам встрять.

В 1991 г. молодой талантливый финн по имени Линус Торвальдс обратился к новостной рассылке, чтобы сообщить о своем проекте, любительской операционной системе, вскоре получившей название Linux. Проект привлек интерес во всем мире, и (возможно, из-за того, что это совпало по времени с судебным процессом SCO-BSD) начиная с середины и конца 1990-х его начали всерьез воспринимать в деловых и академических кругах и считать его угрозой для Microsoft (поищите «Хэллоуинские документы [Halloween documents]»).

Было даже несколько безумцев, которые использовали его в качестве настольной ОС. Строго говоря, сам по себе 'Linux' — не вполне операционная система, а скорее ее центральная часть, которая является мостом между аппаратной и программной частями. Она известна как ядро Linux, и состоит из аппаратных драйверов, сетевых протоколов, файловых систем — и низкоуровневых штук, о которых вам лучше даже не думать. Чтобы оно хоть что-то могло сделать, нужна программа, которая говорит на языке системных вызовов ядра, и в самом начале это были инструменты GNU. Проект GNU также предназначен для создания свободной операционной системы, но ему так и не удалось создать рабочее ядро. Зато ему удавалось создавать свободные версии ряда инструментов Unix (включая редактор, компилятор и оболочку), что совершенно необходимо для любой Unix-образной ОС. Линус портировал их для работы со своим встроенным ядром, и полученное в результате слияние часто именуют GNU/Linux. В основном именно эта комбинация (или некоторые ее расширения) и имеется в виду, когда сегодня люди используют слово "Linux".

Однако сегодня люди ожидают от ОС чуть больше, чем возможность редактировать и компилировать исходный код. На самом деле, у них имеются совершенно безумные и дикие ожидания: подавай им надежную систему для установки программ, симпатичный GUI, приложения для просмотра фильмов и фотографий и воспроизведения музыки, графические драйверы с 3D-ускорением. И главный лозунг масс — чтобы всё это просто работало. Ради удовлетворения такого множества запросов и потребностей в жизнь воплотилось несколько дистрибутивов Linux. Каждый дистрибутив создается и поддерживается так, чтобы все его части отлично сработались вместе, и у каждого — собственное видение того, чем должна быть ОС Linux. У каждого дистрибутива своя доля фанатичных приверженцев, которые ни перед чем не остановятся, пытаясь обратить вас в свою веру.

Чем Linux не является

Некоторые утверждают, что однажды Linux свергнет некую известную ОС из Редмонда, которая на данный момент (по данным Netmarketshare) установлена на 90% подключаемых к сети ПК. В 2003 г. Линус сказал: «Вообще-то я не собираюсь разрушать Microsoft. Это будет совершенно непреднамеренным побочным эффектом». Вскоре после этого появилась ОС под названием Ubuntu, которая активно продвигалась как «Linux для людей». В то время (и это может подтвердить любой, уже бывший тогда в игре) это казалось неким противоречием в терминологии, однако при финансовой поддержке дот-комовского миллионера Ubuntu приняла весьма широко и в очень сжатые сроки. Launchpad, программа отслеживания ошибок дистрибутива, получила сообщение об ошибке № 1 в августе 2004 г. Оно гласило: «Microsoft занимает большую долю на рынке новых ПК. Это ошибка, которую должны исправить



» К счастью, Linux так уже не выглядит (если только вы сами этого не захотите), и знание командной строки не является обязательным.

Ubuntu и другие проекты». Да, на тот момент Microsoft представлялся таким Голиафом, которого должен был повергнуть маленький Давид из своей пращи (работающей в командной строке). Это было время юношеских надежд и веры в то, что лучшие дни впереди. Каких бы взглядов вы ни придерживались, сразить Microsoft Linux не удалось — и вряд ли удастся (хотя в апреле прошлого года Торвальдс заявил, что он «всё еще над этим работает» и надеется «взять их измором»). Рынок ПК переживает спад, для своих прогулок в Интернете люди все больше используют мобильные устройства, а если говорить о плохих парнях, то сейчас есть кан-

Каждый дистрибутив создается и поддерживается так, чтобы его части сработались.

дидатуры и похуже, чем Microsoft. У новых злодеев совсем иная стратегия: ухватить наши любимые прелести открытого кода и создать поверх них проприетарные сервисы. Поэтому им нужен FOSS, им незачем вовлекать его в патентные битвы, и им на самом деле наплевать, какая операционная система работает у людей на десктопах и ноутбуках, поскольку это всего лишь «немые» терминалы: сейчас всё уходит в облака.

Это не к тому, что вам не надо работать в Linux на своем настольном ПК: вам как раз надо это делать. Почему? Ну, во-первых, это бесплатно. Да, на приобретенных в магазине ПК и Mac'ax стоит ОС от Microsoft или Apple, но она прикована именно к этой машине. Если вы соберете собственную из всяких обломков и старых компонентов — ну или из компонентов с полки для запчастей — вам придется платить. Кроме того, поддаться соблазну этих закрытых экосистем — дело нездоровое, это приводит вас к определенному складу ума: вам придется принять программы и все навязанные вам условия. Не каждый хочет быть разработчиком, но использование ПО с открытым кодом позволяет всем узнать, как рождаются приложения (намекнем: там есть гигантский двоичный аист и огромное поле кремневой капусты) и как работает операционная система, и получить свободу делать что-то самостоятельно.



О дивная новая ОС

Делать первые шаги в любой ОС страшно, но вооружившись знаниями об основных концепциях, вы почувствуете себя увереннее.

Многие откладывают знакомство с Linux из опасения повредить уже установленную операционную систему. Теоретически установка вместе с существующей ОС должна пройти нормально, однако в Сети полно историй о том, что всё получилось иначе. Обычно это происходит из-за ошибки пользователя, или из-за того, что программа загрузки Windows начинает вести себя, как капризный ребенок, которого заставляют поделиться конфеткой. Однако всё это не очень актуально, поскольку есть возможность познакомиться с Linux, не устанавливая его: либо используя среду live со средствами установки, либо установив его на виртуальную машину (см. ниже).

Оба этих способа не обеспечат столь же ровной и гладкой работы, как настоящая установка, однако оба абсолютно безопасны. У обоих есть преимущества и недостатки — использование носителя live не предоставляет постоянного хранения данных, а установка на *VirtualBox* может ограничить производительность графики.

Дистрибутивов Linux сотни. Но не тратьте время на то, чтобы охнуться в них с головой, если вы ищете, с чего начать. У вас будет

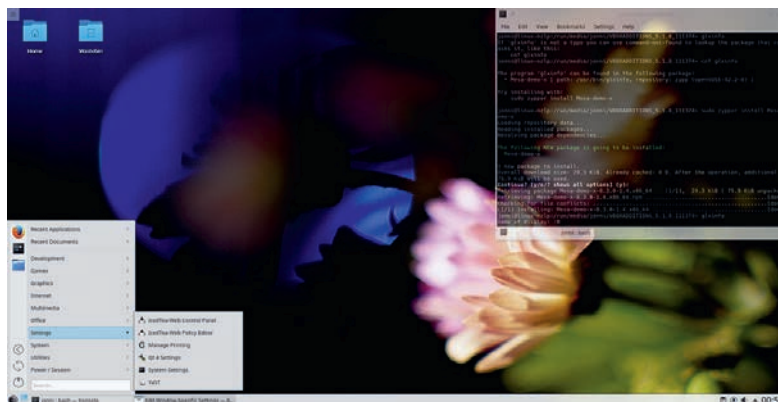
достаточно времени для туманных хипстерских дистрибутивов потом, а в самом начале важнее всего поддержка. Несмотря на оптимизм данной статьи, рано или поздно вы столкнетесь с проблемами. Если дистрибутив популярный, то велики шансы, что подобная проблема у кого-то уже была, и всё, что вам нужно для получения решения — задать правильный поиск в Сети. Есть немало претендентов на роль первого дистрибутива — когда мы упоминаем Ubuntu, нас каждый раз ругают, однако над ним работает множество людей, обеспечивая широкую совместимость, и как ни смотри, ни один дистрибутив не сделал больше для удобства пользователя.

Быть популярным неплохо

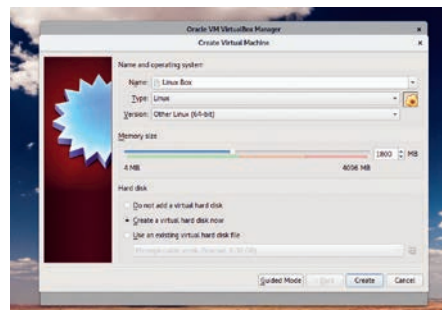
Есть также Linux Mint, который с изнанки оказывается Ubuntu, но с более традиционной средой рабочего стола. Если у вас достаточно храбрости и вы считаете себя технически подкованным, вы, возможно, решите выбрать Fedora. Fedora предпочитает более новые версии пакетов и старается внедрить новые функции ядра быстрее, чем остальные. Еще один вариант, который часто остается незамеченным — openSUSE Leap. Этот дистрибутив с логотипом-хамелеоном практикует новаторский подход к сохранению более старых, более стабильных версий ядра и низкоуровневых библиотек, одновременно предоставляя новейшие приложения. Он весит аж 4,7 Гб, но зато сразу после установки обеспечит вас всем необходимым пользователю настольного ПК.

Все эти дистрибутивы должны отлично работать на оборудовании, произведенном за последние семь лет, но если ваше оборудование старше или вы хотите дать своей запыленной видекарте перевести дух, мы советуем вам Bodhi Linux. Bodhi — еще одно производное Ubuntu, где используется прекрасный легковесный рабочий стол под названием *Moksha*. Не считайте себя обязанными следовать нашим рекомендациям, есть множество других вариантов — разнообразные вариации на основе Ubuntu хороши, потому что вы получаете стабильность базовой системы совершенно

➤ openSUSE Leap — дистрибутив со стильным рабочим столом KDE Plasma 5.



Устанавливаем Linux на VirtualBox



1 Установим VirtualBox

VirtualBox — бесплатная программа (и с открытым кодом). Версии для Windows и Mac вы можете скачать с официального сайта www.virtualbox.org. Используйте соответствующую ссылку на скачивание, скачайте и запустите бинарник. *VirtualBox* установится стандартным образом, подобно любому другому приложению.

2 Выберем свой дистрибутив

Дистрибутивы поставляются в виде файлов ISO (образы устанавливаемых дисков), которые в виртуальной машине ведут себя, как настоящие диски, так что вам понадобится один из них. ISO-файлы могут быть большими, так что, возможно, вы решите использовать один из файлов с **LXFVDVD** и сначала скопировать его на жесткий диск.

3 Настроим новую VM

Откройте *VirtualBox*, выберите New, дайте VM [Виртуальной машине] имя и настройте ее тип на 'Linux'. Если ваш дистрибутив находится в окне Version, выделите его, в ином случае выберите Other Linux. Вы должны выделить своей VM как можно больше памяти, но вполне хватит около 30% общего количества ОЗУ. Нажмите Create [Создать].

бесплатно, и по большей части инструкции/отладки Ubuntu в равной мере применяются и к его производным. Те, кто не верит, что Linux может быть красив, пусть взглянут на elementary OS, а тем, кто хочет получить традиционный легковесный рабочий стол, советуем Ubuntu Mate.

Установив Linux на своем компьютере, вы, возможно, обнаружите знакомые элементы: кнопку меню, способную запускать программы; панель задач, которая отслеживает, какие программы открыты; регулировку громкости; ну и простой способ выключить компьютер. Но там могут оказаться и элементы непривычного вида, особенно в дистрибутивах с Gnome, например, в Fedora, где используется новая и минималистская парадигма рабочего стола. Интерфейс Ubuntu Unity не настолько сбивает с толку, хотя тоже подойдет не каждому. И в Unity, и в Gnome имеется строка состояния вверху, с иконками управления вверху справа. Это практически всё, что видно в Gnome. Unity идет дальше, обеспечивая боковую панель слева с доступом к популярным программам и к Dash — инструменту Ubuntu для поиска программ, музыки, документов; считайте ее эволюцией меню Старт. Доступ к Dash также можно получить через кнопку Windows. В Gnome есть похожая функция, доступная через ту же кнопку или по движению курсора мыши вверх влево. Другие рабочие столы (например, Plasma 5, Cinnamon, Mate, Moksha) более традиционные, но Linux — не Windows, и не стоит рассчитывать, что всё будет там же, где оно было в Windows.

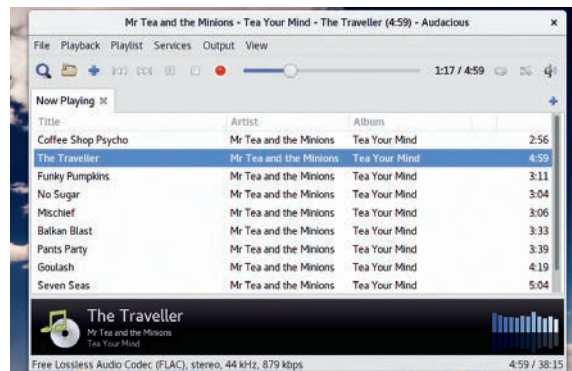
Модульные радости Linux

Одно из самых замечательных свойств Linux — его возможности индивидуальной настройки: поскольку определенный дистрибутив имеет рабочий стол по умолчанию, легко установить другой, совершенно преобразив внешность и работу своей системы. Вы можете выбрать требуемый рабочий стол на экране приглашения, и все они будут неплохо ладить друг с другом. Итак, мы видим, что рабочий стол — всего лишь надстройка основного дистрибутива. Большая часть Linux модульная: части можно удалять, обновлять, смешивать, и это очень увлекательно. Например, все рекомендуемые нами дистрибутивы поставляются с медиа-плеером, и при наличии должных вспомогательных пакетов (т.е. если вы пометите окно, где говорится об установке сторонних кодеков в дистрибутивах на базе Ubuntu) он сможет воспроизвести любой аудио- (или видео-) файл, который вы ему укажете. В ряде дистрибутивов предусмотрено достойное приложение для воспроизведения музыки,

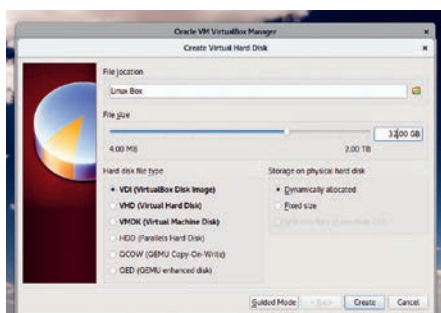
способное работать с медиа и с оформлением обложки — например, *Rhythmbox*; но есть люди, предпочитающие более традиционные приложения типа *Winamp*, поэтому давайте установим замечательный плейер *Audacious*.

Audacious — приложение с открытым кодом, и его можно установить в любом дистрибутиве Linux. Откройте программное приложение (в большинстве дистрибутивов оно называется *Software*, и если открыть меню и ввести его первые буквы, оно определит местонахождение. В openSUSE оно называется *YaST*). Используйте функцию поиска, чтобы найти *Audacious*, и следуйте подсказкам, чтобы его установить. Вам придется ввести пароль администратора, поскольку установка пакетов — это операция, требующая особых привилегий. После этого будет вызван менеджер пакетов и установлен *Audacious* (и его зависимости). Вы должны обнаружить значок для него в меню системы, как раньше.

При наличии правильного рабочего стола и некоторого везения можно будет получить систему, которая сама о себе позаботится. Периодически будет возникать напоминание об установке обновлений, но это можно отложить, и обычно для этого не требуется перезагрузка. Это хорошая новость, если вам нужно сделать настройку для не очень технически подкованных пользователей или если вам самим не хватает навыков. Если вы установили один из рекомендованных нами дистрибутивов, вы найдете там также знакомые программы: в большинстве дистрибутивов установлен *Firefox*, *LibreOffice Writer* выглядит и работает во многом так же, как *Microsoft Word* в начале нулевых, да и менеджер файлов всегда будет у вас под рукой.

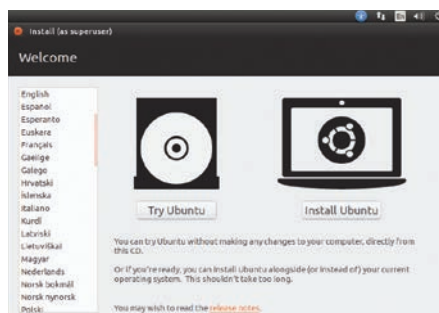


» *Audacious* — традиционный музыкальный плеер; он может выглядеть именно так или как *Winamp*.



4 Создадим виртуальный диск

Типичная установка требует около 10 ГБ, но лучше будет расщедриться до 30 ГБ. Имя и размещение виртуального диска будет указано вверху. Оставьте отмеченными опции VDI и Dynamically allocated [Динамически отведенный], если у вас нет причин поступить иначе, и снова нажмите Create. Итак, ваша виртуальная машина (VM) родилась!



5 Загрузим VM

Нам нужно установить наш виртуальный DVD. Выберите новую VM и перейдите в Settings > Storage. Выберите значок CD из дерева Controller. Нажмите на выпадающее меню CD, выберите Choose Virtual Optical Disc [Выбрать Виртуальный Оптический Диск] и укажите местоположение нужного вам Linux-ISO. Нажмите OK, затем запустите VM.



6 Установим, и... наслаждайтесь Linux!

Некоторые дистрибутивы сразу предлагают опцию установки, но некоторые требуют сначала загрузить рабочий стол. Оказавшись на нем, выберите install и выберите Erase Entire Disk [Стереть весь диск]. По завершении выключите VM, извлеките ISO, увеличьте видеопамять и включите 3D.

Покидая Редмонд

Linux всё делает не так, как другие ОС; мы расскажем об основных отличиях и объясним, почему он делает лучше.

Полазив по Linux некоторое время, легко понять, что он несколько отличается от macOS и сильно отличается от Windows. Все три ОС (и все остальные) должны делать одно и то же: обеспечить пользователю взаимодействие с оборудованием так, чтобы можно было получить некий значимый или полезный результат. Но все они делают это по-разному. В некотором смысле Windows становится больше похожа на Linux — начиная с Windows 7, появляется нормальное разделение привилегий между смиренными пользователями и всемогущими администраторами, плюс в Windows 10 имеется собственный менеджер пакетов и настоящая подсистема Linux.

Одно из самых глубоких отличий лежит в установке программ. Вполне возможно найти в Сети программы Linux — как в виде исходного кода, так и в виде бинарников. Их можно скомпилировать и установить, и они даже могут работать, но делать этого не стоит, пока не получат большее распространение не зависящие от дистрибутива схемы работы с пакетами (такие, как Flatpak или Ubuntu Snaps). Все дистрибутивы Linux разные — они используют разные схемы файловых систем, разные ядра и разные версии библиотек. Чтобы всё это сработало, у дистрибутивов имеются целые армии пользователей, которые берут приложения и аккуратно упаковывают их в пакеты, чтобы их можно было установить одним нажатием кнопки (или несколькими нажатиями на клавиши в командной

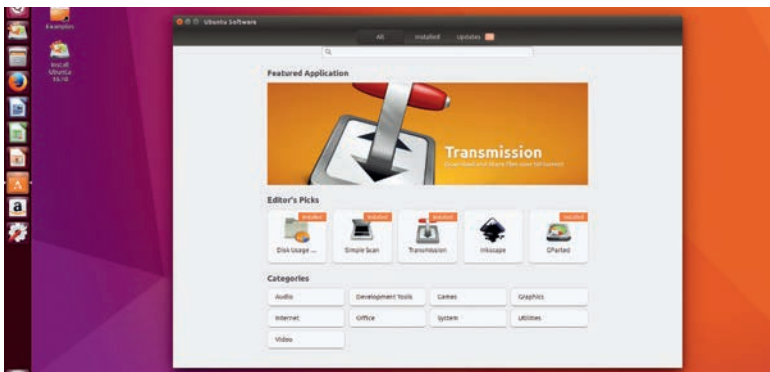
строке). Это и правда значительно облегчает жизнь, поскольку позволяет регулярно обновлять программы или дочиста удалять их. Разные дистрибутивы используют разные менеджеры пакетов (Debian и его производные используют *APT*, Fedora использует *DNF*, а openSUSE использует *zypper*) и форматы пакетов (*.deb* и *.rpm*), но все они служат одной цели.

Помимо официально поддерживаемых репозиториях пакетов, разработчики могут создать свои собственные, чтобы можно было установить неофициальные или более новые программы. Пакеты Ubuntu можно находить через Personal Package Archives (PPA), а RPM для Fedora — через сторонние репозитории. Например, если вы хотите установить несвободное ПО на Fedora, понадобится добавить репозиторий RPM Fusion. Но тут следует предупредить, что подобные программы не подвергались столь же тщательному тестированию, как программы в официальных репозиториях, так что примите это во внимание.

Детальное управление

Тем, кто уже потянулся к браузерам, чтобы скачать новые улучшенные драйверы, советуем попридержаться коней. Да, некоторые производители предлагают на своих сайтах драйверы для Linux, но использовать их — не лучшая идея. И особенно это касается драйверов для принтеров. В отличие от Windows, где охота на драйверы является основной частью ритуала установки нового оборудования, ядро Linux уже включает невероятное количество оборудования. Если это оборудование распознается, то загружается соответствующий модуль, и всё в полном порядке. Это, конечно, теория, но есть несколько примечательных исключений — некоторые Wi-Fi-адаптеры требуют проприетарных прошивок, игровые мыши требуют руководства по раскладке кнопок, и далее мы отвели целую страницу на графические драйверы.

Linux предлагает куда более тонкое управление, чем Windows: от сервера отображения, *X.org* или *Wayland*, который отправляет пиксели на монитор через драйвер отображения, до менеджера окон, отвечающего за рамки вокруг приложений, наборов инструментов (например, *GTK* и *Qt*) для рисования меню и виджетов и тематических движков для скинов. Похожее на всё это подсистемы существуют в Windows и macOS, но в большинстве своем у пользователей нет к ним доступа.



➤ Старый Ubuntu Software Centre сменился на Ubuntu Software. Здесь можно найти все сокровища, и без лишних панелей инструментов.

Путеводитель по жаргону

Нового пользователя Linux может сильно напугать непонятный жаргон, с которым придется столкнуться. Терминов, номенклатуры и трехбуквенных сокращений (TLA — three-letter acronyms) здесь просто как грязи. Чтобы это вас не огорчало, вот вам наш путеводитель по терминологии:

➤ **Boot loader** Небольшая программа, загружаемая сразу по окончании инициализации системы посредством BIOS или UEFI. Предлагает меню, откуда можно загружать несколько ОС. Ее основная задача — довести систему до состояния загрузки ядра. Самая распространенная такой загрузчик именуется *Grub* (GRand Unified Bootloader).

➤ **Shell [Оболочка]** Приложение, позволяющее набрать команду и запустить ее в системе. Рабочие столы и серверы Linux в качестве оболочки по умолчанию в основном используют *Bash*.

➤ **Terminal [Терминал]** Строго говоря, это устройство с клавиатурой и экраном, но часто используется в значении графической программы, которая дает доступ к оболочке (по сути, это виртуальный терминал).

➤ **Systemd** После загрузки ядра Linux первой обычно загружается именно эта программа. Она управляет сервисами, логинами, объемом хранилища и многим другим.

➤ **Filesystem [Файловая система]** Способ организации данных на разделе или логическом разделе. Самая распространенная файловая система для рабочих столов Linux — *ext4*, но сейчас все чаще используются более новые файловые системы, типа *Btrfs* и *ZFS*. Большинство дистрибутивов способны читать файловую систему Windows NTFS, а используя программу *ntfs-3g*, можно также и записывать ее.

➤ **RTFM [Read the f... friendly manual]** Этот совет вы, вероятнее всего, получите, если разместите вопрос, не удосужившись сначала прочитать руководство пользователя.

Жизнь в командной строке

Иногда наводить мышь и щелкать — то же, что общаться через пылкую жестикуляцию. Уж лучше тщательно выбранные и ясно высказанные слова.

Некотрые читатели сохранили воспоминания о работе в MS-DOS, и они, вероятно, не самые приятные. Другие читатели могут прийти в замешательстве — зачем, в нашу эпоху блестящей графики и изобилия ОЗУ, кому-то приспичит управлять своим компьютером, вводя непонятный текст. А оказывается, зная нужные заклинания, из командной строки можно сделать практически все. Итак, давайте пройдем курс освоения терминала.

Вы запросто отыщите приложение терминала в меню рабочего стола (или панели Dash или Activities). В дистрибутивах на базе Ubuntu и Gnome оно называется Terminal, а KDE Plasma называет его Konsole. Если его запустить, откроется страшное окно со подсказкой вроде `user@host:~$`, где `user` и `host` — имена, выбранные для пользователя и компьютера при установке. Одна из наиболее распространенных команд — `ls`, она отображает содержимое текущей директории. Введите эти две магические буквы, затем нажмите Enter. Terminal открывается в нашей директории `home`, которую мы можем сокращенно назвать с помощью символа `~`, поэтому возвращенный `ls` результат должен согласовываться с тем, что мы видим при открытии менеджера файлов. Можно передать в `ls` дополнительные аргументы: например, `ls /` покажет нам директорию `root`, и во врезке внизу имеются детали того, как она организована. Помимо других путей к директориям, мы можем также передать опции, влияющие на выводимый `ls` результат. Например, попробовав `ls -l`, вы получите полную информацию, включая размеры файлов, разрешения и информацию о владельце. А `ls -lh` выдаст удобные для чтения размеры файлов (килобайты и мегабайты вместо байтов).

Магия терминала

Навигацию по файловой системе `root` можно осуществлять командой `cd` (change directory). Например, мы можем перейти в директорию `root` с помощью `cd /`. Если напечатать здесь `ls`, оно покажет использование тех же директорий, которые мы видели, когда задавали команду `ls /` выше, но поскольку сейчас мы уже находимся «внутри» директории `/`, нам уже не нужен дополнительный слэш. Как уже упоминалось ранее, `~` — ярлык для нашей директории `home`, поэтому `cd ~` вернет нас туда. Но даже просто сама по себе `cd` тоже может это сделать. Мы можем просмотреть ранее выполненные команды с помощью стрелок вверх и вниз, что

```

jonn@acerofspades ~1$ ls /bin/ip
ip          ipx         iptables
ip6tables  ip6tables-compat
iptables-compat-restore
iptables-compat-save
iptables-restore
iptables-save
ipcmk      iptables-xm1
ipcrm      iptc
ipcs
jonn@acerofspades ~1$ ls /
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot etc  lib  lost+found  opt  root  sbin  sys  usr
jonn@acerofspades ~1$
  
```

► Используйте *Guake*, чтобы оживить свой Terminal. Добавьте яркое выпадающее меню, полупрозрачную консоль и затейливый фон.

избавит нас от необходимости повторять уже запущенную последовательность команд. Вместо того, чтобы вводить длинные имена файлов (и делать печальные ошибки), мы можем использовать автозаполнение с помощью Tab. Например, из директории `home` введите `cd D` (в Linux важен регистр) и затем дважды нажмите на Tab. Выведется все возможные директории, начинающиеся на D (Desktop, Documents, Downloads). Нажмите на клавишу `e`, чтобы команда выглядела, как `cd Desktop`, и снова нажмите на Tab. Команда будет автоматически дополнена до `cd Desktop/`, поскольку теперь это — единственная опция.

В Ubuntu мы можем установить программы по команде `apt-get`:

```
$ sudo apt-get install audacious
```

Команда `sudo` (super user do) используется для того, чтобы повысить наши привилегии до пользователя-администратора (`root`). Некоторые дистрибутивы предпочитают отдельную учетную запись `root` с собственным паролем, однако использование `sudo` при необходимости является чуть более безопасным способом работы. Fedora и openSUSE используют другие менеджеры пакетов, но идея та же; просто замените `apt-get` на `dnf` или `zypper` соответственно. Пакеты можно удалить, подставив вместо `install` опцию `remove`. Имена пакетов нередко не зависят от дистрибутива, и если в каком-нибудь учебнике мы даем инструкции по установке только для версии с `apt-get`, они должны легко переводиться и на другие менеджеры пакетов.



Всё — файл

Это странная концепция, но Linux действительно всё рассматривает как файл. Устройства хранения, например, жесткие диски, можно найти в `/dev/sda`, `/dev/sdb` и т. д., и на их разделы ссылаются как на `sda1`, `sda2`. Информацию о разделах можно вывести командой `lsblk`. Техническую информацию о процессоре системы можно получить с помощью `cat /proc/cpuinfo`, или `cat /proc/meminfo` для статистики памяти. Раздел, где установлен Linux, монтируется как директория `root /`. Это довольно близко к диску C: в Windows, но здесь есть ключевые различия. Во-первых, другие разделы монтируются как

директории внутри иерархии `/`: например, жесткий диск может монтироваться в `/mnt/backup`, а автоматизированная флэшка USB — на `/run/media/user/thumb`. Более того, имеются псевдо-файловые системы, например, `/dev`, `/proc` и `/sys`, которые не хранятся ни на одном жестком диске. Они обеспечивают доступ к большому количеству информации по оборудованию, системным ресурсам и работе процессов. Некоторые из них могут записываться, что позволяет контролировать параметры ядра. Команда `ls /` отображает содержимое директории `root`. Директории верхнего уровня именуются

в позабытых традициях Unix. Файлы загрузчика хранятся в `/boot`, файлы настройки — в `/etc`, программы устанавливаются в `/usr`, лог-файлы и данные рабочей среды проживают в `/var`. У каждого пользователя есть своя поддиректория в `/home`, где он является полноправным хозяином. Другие файлы и директории имеют свои разрешения, чтобы не позволить пользователю вмешиваться. Некоторые предпочитают, чтобы `/home` был отдельным разделом, возможно, на отдельном устройстве. Всё это можно настроить прямо из программы установки, но лучше оставить на потом.

Графические драйверы

Карта Nvidia серии Pascal? Неутолимая жажда FPS? Читайте дальше...

Часто проклинаемым пугалом является невероятная сложность и запутанность видеодрайверов в Linux. А точнее, подготовка и ритуалы по установке проприетарных драйверов от AMD и Nvidia, которые по идеологическим причинам (как в случае с Debian или Fedora) могут не распространяться по обычным каналам обновления дистрибутива.

Многие дистрибутивы предлагают для них пакеты, но поскольку им требуется расширенное и активное тестирование перед включением, они часто опаздывают по сравнению с релизами производителей GPU. Невероятно удачный проект — драйвер с открытым кодом Nouveau для видеокарты Nvidia, да и поддержка FOSS для карт AMD через драйверы Radeon (более старые карты) и AMDGPU (более новые) очень выиграла от содействия AMD. Intel тоже предлагает драйвер с открытым кодом для своих интегрированных GPU. Итак, если не брать в расчет какое-то особенное или доисторическое оборудование, в большинстве установок Linux имеется достойная встроенная поддержка графики — колоссальная разница по сравнению с временами, когда приходилось редактировать `xorg.conf` и молиться, чтобы монитор не задымился.

Как бы то ни было, если вы хотите серьезно играть, вам, вероятно, стоит обдумать установку соответствующего проприетарного

драйвера. При ментальности Windows, это вполне нормально и стандартно — вы найдете драйверы для Linux (и FreeBSD) на сайтах производителей, вместе с инструкциями по скачиванию большого бинарника, запуску от имени root и перезагрузке. Такова теория, но порой всё идет наперекос, и наш отчаявшийся пользователь безнадежно пялится в черный экран. Как мы уже обсудили ранее, разрешать чему-либо, кроме вашего менеджера пакетов, устанавливать программы — не лучшая идея, и это особенно верно для критически важных компонентов системы. Так делали многие — и даже остались живыми свидетелями этих сказок, но мы не советуем идти их путем. Видеодрайверы загружаются как модули ядра, и поскольку ядра в разных экосистемах отличаются, следует скомпилировать модификатор, который соединит видеодрайвер с ядром. И даже если всё работает после первичной перезагрузки, могут быть скрытые проблемы, которые тихо ждут обновления ядра.

Обновить драйвер безопаснее

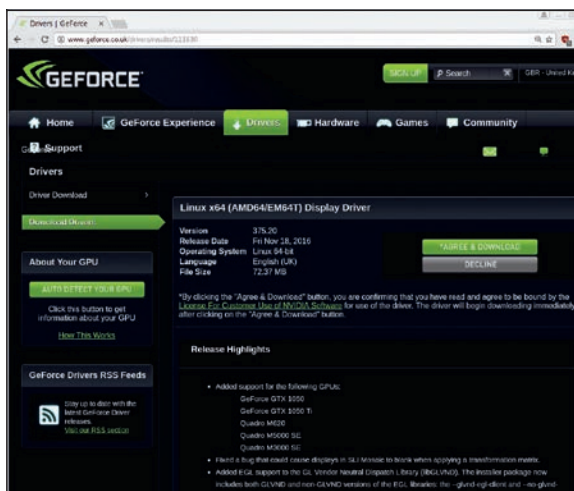
Итак, дабы предотвратить эту драму, мы рассмотрим более безопасные опции, которые питают достаточное уважение к менеджеру пакетов. На момент написания, используйте Ubuntu 16.04 или 16.10 с видеокартой Nvidia производства 2010 (серия GeForce 400) или позднее, легко установить долгоиграющий драйвер типа 367 от Nvidia. Всё, что для этого потребуется — несколько щелчков в утилите *Software and Updates*. Однако лучшая производительность достигается при использовании более позднего типа 370. В Ubuntu есть простой способ сделать это благодаря добрым душам, поддерживающим PPA `graphics-drivers`. Вы можете добавить его и обновить список пакетов с помощью:

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

После этого новый драйвер должен стать виден во вкладке *Additional Hardware* утилиты *Software and Updates*. Помните, что это пока экспериментальный PPA, и не удивляйтесь, если что-то пойдет не совсем по плану. Вы можете вернуться к старому драйверу и удалить более новый из системы с помощью

```
ppa-purge ppa:graphics-drivers/ppa.
```

Самый новый графический стек с открытым кодом (Mesa 13) можно отыскать в PPA `oibaf` (`ppa:oibaf/graphics-drivers`) в Ubuntu, и вновь вышедший Fedora 25 скоро его получит в качестве обычного обновления. Так что экспериментируйте также с драйверами FOSS, какой бы ни была ваша карта.



➤ Свежайшие проприетарные драйверы для Linux доступны на сайте Nvidia, но не советуем вам устанавливать их этим способом, если только вы не на все сто уверены в том, что вы делаете.

Ситуация с AMD

Несмотря на то, что AMD передает спецификации оборудования и выделяет время разработчикам для Linux, текущая ситуация немного странная в зависимости от вашего оборудования. Их более старый проприетарный драйвер, `fglrx`, ликвидируется и больше не поддерживается сервером *X.org*, поэтому пользователям более старых карт ATI приходится использовать драйвер с открытым кодом `radeon`.

Более современные чипсеты ('Volcanic Islands' и новее) поддерживаются новенькой средой

AMDGPU (вообще-то эта поддержка еще более давняя, но надо подправлять ядро), где есть обычный модуль ядра с открытым кодом и опционный проприетарный пользовательский драйвер, заманчиво именуемый AMDGPU-Pro. Это должно способствовать появлению более прочного драйвера, так как наличие исходника модуля включает куда более простую интеграцию с каждым дистрибутивом.

Однако AMDGPU-Pro всё еще на стадии бета, и поддерживает только Ubuntu и RHEL. Самые

отважные могут установить его с сайта AMD — <http://bit.ly/AMDRadeonGPUProLinuxDriver>, но мы бы рекомендовали оставаться с пользовательским Mesa с открытым кодом. В свое время AMDGPU-Pro станет доступнее, и поддержка AMDGPU карт GCN1.0 будет делом обычным. Но более новые карты отлично работают с драйверами FOSS, а большую производительность из них можно выжать с помощью `oibaf` PPA, упомянутого в основной статье (*выше*).

Обрывающая связи

Переход на Linux — дело непростое, а некоторые вещи вообще способны вызывать недоумение.

Многих отталкивают от Linux крупные софтверные компании, которые не выпускают Linux-версий своих популярных коммерческих продуктов. Самые известные среди них — *Microsoft Office*, пакет *Creative Cloud* от Adobe и многие игры AAA (хотя тут ситуация меняется). При этом многим они и не нужны: *LibreOffice* справляется с большей частью файлов *Office*, а *GIMP*, *Inkscape*, *darktable* и *Krita* вполне могут конкурировать с *Photoshop*, *Lightroom* и *Illustrator*.

Сервисы облачного хранения, такие как *Dropbox*, могут интегрироваться в рабочий стол Linux, но мы могли бы пойти еще дальше и настроить собственный экземпляр *Nextcloud*. Если вам действительно нужно общаться со своими коллегами, имеется клиент *Slack*, а *Microsoft* даже трудится над новым клиентом *Skype* для Linux. Старый тем временем тоже более или менее фурычит. Большинство дистрибутивов предлагают браузер *Chromium* с действительно открытым кодом — тот же *Google Chrome*, но без неприятных моментов. В некоторых дистрибутивах и он, и *Firefox* теперь включают модуль дешифрации контента *Widevine*, чтобы воспроизводить контент с *Netflix* и других сайтов потокового вещания. Однако в разных дистрибутивах поддержка разная, и если это не работает, то простейшим решением будет установить *Chrome* из RPM- (*Fedora*, *openSUSE*) или Deb-пакетов, доступных на <https://www.google.com/chrome/browser/desktop>.

Ситуация с играми тоже улучшается: игры можно скачивать со встроенного клиента *Steam for Linux* или DRM-free с *GOG.com*. Популярные игровые движки теперь поддерживают Linux, так что вы теперь можете исследовать целый сундук с сокровищами инди-игр. Кроме того, благодаря работе по портированию *Aspyr Media* и *Feral Interactive* многие большие игровые наименования приходят в Linux, но иногда на пару месяцев, а то и лет позже. В основном эти порты не отличаются такой же производительностью, как в *Windows*, поскольку используют надстройку для перевода вызовов *DirectX* в *OpenGL*. Однако по мере внедрения *Vulkan*, нового кросс-платформенного графического API, эта ситуация, скорее всего, улучшится. *Steam* можно найти в приложении *Ubuntu Software* или в *Build Service* от *openSUSE*, на <http://software.opensuse.org/package/steam>. Пользователи *Fedora* найдут его в *RPM Fusion*. Клиент *Steam* работает не так, как другие пакеты. Пакет просто содержит программу установки, которая скачивает клиент и помещает



► В Linux немало чудесных игр. На очередной распродаже в *Steam* или на *GOG.com* пингвинистам придется наступать на горло своей песне, чтобы не вылететь в трубу...

его в вашу директорию *home*. При его запуске *Steam* обновляется, что не соответствует обычной практике управления пакетами.

Двойная загрузка

Мы знаем, что некоторых читателей разочарует отсутствие в нашей статье темы двойной загрузки. Мы рассказывали о ней [Учебники, стр. 70 LXF217], и в целом этот процесс надежен. Но данная статья предназначена для начинающих, и как таковая, должна рассказывать только о том, что не несет риска. Могут возникнуть проблемы при изменении размера разделов *Mac* и *Windows* или из-за их загрузки. Мы не говорим, что двойную загрузку не стоит настраивать и что этому присущи некие врожденные проблемы (кроме того, что половина ваших ОС — не Linux); надо только сделать перед этим соответствующие резервные копии. Для старых ПК с классическим BIOS лучше установить *Windows* до Linux, поскольку *Windows* любит съедать *Grub*. Расскажите нам о своих достижениях, только, пожалуйста, не обращайтесь за техподдержкой. LXF



Помощь

Если только вы не бог, то рано или поздно Linux начнет вести себя не как положено или не так, как вы от него ждете. Первое, что надо сделать — поискать решение в Сети; при этом могут оказаться полезными сообщения об ошибках и номер модели оборудования. Это может быть неприятной ситуацией; и, к сожалению, дезинформации тоже хватает.

В общем, не лучшая идея — слепо копировать и вставлять случайный код, найденный на форумах, если вы не знаете, что он делает. Вы очень быстро обнаружите, что в Интернете немало устаревшей или просто неверной информации. Потратите

некоторое время на изучение проблемы (или предложенного решения) — весьма неплохо вообще отойти от экрана и потратить на обдумывание ночь. И если вы ни к чему не пришли, тогда можно просить о помощи.

В Linux нет единого центра оказания помощи. Linux бесплатен, поэтому нет линии техподдержки. Производители оборудования вам вряд ли помогут, и довольно подло беспокоить продавцов (например, спрашивать, совместимо ли устройство с Linux) всеми этими странными разговорами о Linux. Помните, большинство о Linux даже не слышали. Однако помощь, безусловно, найдется. Сообщество

Linux по большей части очень дружелюбное. У всех дистрибутивов есть собственные форумы; есть и форумы сообществ, такие как *Phoronix* или *LXF Forums* (www.linuxformat.com/forums). На форумах есть свои правила, нарушать которые можно только на свой страх и риск.

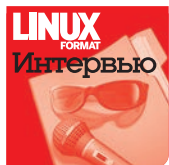
В частности, разместив часто задаваемый вопрос, вы можете нарваться на резкий отказ. Столь же мало помогают общие описания. Куда полезнее предоставить лог-файлы и сообщения об ошибках. Читайте инструкции, читайте FAQ, будьте вежливы, и люди вам помогут (так же, как и в жизни).



Фестиваль Mozilla 2016

Лес Паундер поехал в Лондон на MozFest, и в тени Арена 02 обнаружил уникальное здание, где куется будущее Открытого Интернета.

ДОРИН ФАЙЛЗ



Linux Format: Пожалуйста, расскажите нашим читателям немного о себе и о своем участии в MozFest.

Дорин Файлз [Dorine Files]: Я — мама, работаю в отделе кадров в компании, занимающейся кибербезопасностью. А здесь я «главный по площадке», продюсер Youth Zone [Молодежной зоны]. Как правило, на MozFest у нас бывает от семи до девяти продюсеров. Каждый отвечает за докладчиков и за общую концепцию своей площадки в контексте всего мероприятия. На нас основная часть планирования логистики, составления расписания и идейного оформления той или иной площадки; последнее связано с неким сюжетом или предметом, которым мы хотим поделиться с нашими единомышленниками.

LXF: А кто эти единомышленники?

ДФ: Все, кто приехал на фестиваль, будь то участники — те, кто ведет мастер-классы — или те, кто платит за их посещение. И не важно, пять им лет или семьдесят, важно только, что у них в голове.

LXF: Вы упомянули, что за каждой площадкой стоит своя идея; а какова цель Youth Zone?

ДФ: Наша идея — это привлечение молодежи, а цель — взрастить таланты для следующего поколения, которому жить в том мире, который создается сейчас. Они имеют полное право участвовать в диалоге о том, каким будет мир, где им придется жить. Необходимо иметь определенный уровень зрелости, чтобы понимать, как и почему надо что-то менять на благо общества, и дети в Youth Zone показали, что они его достигли.

LXF: Похоже, что площадка Youth Zone на фестивале MozFest имеет ключевое значение. Как давно она существует?

ДФ: В 2016-м это уже третья итерация. Мы начали в 2014 г., тогда это была площадка, которую взрослые организовывали для детей, в чем лично мне виделась проблема, поскольку все остальные



площадки MozFest проводятся взрослыми для взрослых. Мне же хотелось, чтобы YouthZone стала местом, созданным для детей совместно детьми и взрослыми. Наша цель в том, чтобы среди организаторов MozFest было 10% детей, и в этом году мы ее почти наполовину достигли.

LXF: Какое место отводится молодежи в рамках столь масштабного события, как MozFest?

ДФ: Youth Zone занимает один из девяти этажей MozFest, с каждым годом наше мероприятие рас-

LXF: Такое впечатление, что нынешнее поколение детей умеет гораздо больше, чем предыдущее — у них гораздо больше навыков в более раннем возрасте?

ДФ: У детей есть доступ к безграничному знанию через Интернет, в том числе и в области программирования. Нам необходимо вооружить их пониманием способов создания безопасного кода, а также тех приемов, которые используют хакеры.

Например, мне известен случай, когда ребенка исключили из школы за то, что он взломал школь-

О МОЛОДЕЖИ НА MOZFEST

Молодежь — в самом сердце технологии, это лидеры будущего, его хранители и устроители.



тет, оккупируя всё больше пространства, которое нам любезно предоставляет колледж Рэйвенсборн [Ravensbourne College]. Молодежь — в самом сердце технологий, это лидеры будущего, его хранители и устроители. Но дети очень категоричны и скорее прислушаются ко мнению сверстников, чем взрослых. Поэтому меня интересует, как сделать так, чтобы и взрослый тоже стал «единомышленником». За рамками MozFest ребенка определяют как ребенка в силу возраста и недостатка жизненного опыта. Но этот ребенок может быть одаренным математиком/программистом. И нам следует признать их стремления, дать им увидеть в лице взрослых своих союзников, которые помогут им создать и понять то, за чем ожидается будущее.

ную сеть. Родители даже не представляли масштаб взломов. А ребенок потом взломал еще компанию по заказу пиццы онлайн и отправил всем своим друзьям по пицце. Тогда пришлось вмешаться полицейским, которые, помимо прочего, посоветовали родителям избавиться от компьютеров. Если бы эта школа сотрудничала с сообществом программистов/компьютерных мастеров, они бы поняли, что этому ребенку требуется особая помощь в развитии его уникальных навыков и что в сообществе есть люди, способные ее оказать. Но, к сожалению, похоже, что здесь мы потеряли талантливую хакера, который мог бы когда-нибудь стать ценным кадром для компаний, занимающихся информационной безопасностью. »

СЕРИС ЛОК



LXF: Привет, Серис, сегодня ты одна из докладчиков на MozFest; можешь рассказать нашим читателям немного о себе?

Серис Лок [Cerys Lock]:

С февраля 2016 г. я провожу Stafford Raspberry Jam. На MozFest я тоже помогаю с организацией, а заодно веду свой мастер-класс.

LXF: И как тебе в роли докладчика?

СЛ: Youth Zone — чудесное место. Очень много выставочных проектов, в том числе практических мастер-классов, где можно опробовать новые идеи и навыки. Я посетила два: о том, как использовать GitHub, и как взломать Wi-Fi с помощью Kali Linux. Мы также разместили всех юных участников одном месте, чтобы они могли совместно работать над новыми проектами и улучшать существующие идеи.

LXF: А зачем ты приехала на MozFest?



СЛ: Меня пригласили выступить Дорин Файлз (см. стр. 37) и Эндрю Малхолланд [Andrew Mulholland], они руководят площадками. Мой мастер-класс посвящается плате Sense HAT, той самой, что использовалась в проекте Astro Pi [Ред.: — Pi в космосе!]. Я буду показывать, как с ее помощью создается

мира. Женщин среди них существенный процент, можно сказать, выше, чем на других конференциях. Способствует ли такое разнообразие состава расширению сообщества в целом?

СЛ: Здесь очень пестрая публика, посетители говорят на нескольких языках, приносят свое новое

О РАЗНООБРАЗИИ MOZFEST

Нам надо, чтобы этого становилось побольше, и показывало: разнообразие — главный залог будущего.

простая анимация. Это несложный, но интересный способ научить пользоваться данной платой, позволяющий любому оценить ее возможности.

LXF: Как ты сама, будучи юным участником Youth Zone, чувствуешь себя на MozFest, и могут ли дети повлиять на будущее Интернета и открытого ПО?

СЛ: На MozFest я себя ощущаю чужою странно. Я в первый раз вдали от дома, но я в окружении чудесных людей, в безопасной среде, все разделяют страсть к созданию нового. У детей, безусловно, есть право голоса и своя роль в будущем Интернета и Open Source. Нам надо прививать им интерес к коду. Например, благодаря недавнему проекту BBC micro:bit удалось добиться того, что больше учеников 7-го класса [Ред.: — В Британии это первый класс средней школы] заинтересовались компьютерами. И больше девочек захотели кодировать.

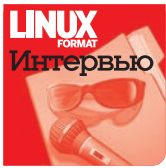
LXF: Фестиваль Mozilla — широкое мероприятие, в этом году мы видим здесь участников со всего

видение. Я вижу, что посетители MozFest более открыты к новым идеям и к узнаванию нового; и нам надо, чтобы этого становилось побольше, и показывало: разнообразие — главный залог будущего.

LXF: В старшей школе ты будешь изучать программирование в рамках обязательной школьной программы. Но как ты думаешь, надо ли детям учиться писать код, даже если они никогда не будут работать в этой сфере?

СЛ: Информатика учит детей думать. Понимать ту логику и те процессы, которые мы воспринимаем как данность. Учит делить более крупные проблемы на более мелкие составляющие, чтобы их стало проще понять, а это, по-моему, полезно уметь всем и в любом предмете. Выбирая себе профессию на будущее, я наверняка захочу работать в сфере технологий, но прямо сейчас еще не знаю, в каком именно направлении. Возможно, работа моей мечты еще не изобретена, и если это так, то мне самой и прокладывать этот путь.

ФИЛИП КОЛЛИГАН



LXF: На это у нас ушел год, но все-таки, наконец, мы вас нашли, Филип; расскажите, пожалуйста, немного о себе. **Филип Коллиган [Philip Colligan]:** Здравствуйте, я — исполнительный директор Raspberry Pi Foundation.

LXF: Зачем вы приехали на MozFest?

ФК: Совместить приятное с полезным. Совместно сказать, я только второй раз на MozFest, но мне тут очень нравится. Я давний поклонник Mozilla, и не только их технологий, хотя они потрясающие. На мой взгляд, сила Mozilla в том, что это именно движение, к которому может присоединиться каждый и в основе которого лежит идея Открытого Интернета — она мне очень дорога, и я страстно ее отстаиваю. Радует то, что Raspberry Pi тоже является частью этого движения, и конечно же, у нас есть свое сообщество, которое по большей части пересекается с сообществом Mozilla. И здорово видеть на MozFest столько проектов Raspberry Pi. Я бы приехал сюда, даже не имея отношения к Фонду, но я счастлив быть здесь.

LXF: За эти выходные на MozFest представили множество проектов, и очевидно, что многие выполнены под влиянием Raspberry Pi — при создании устройства такого не предполагалось.

ФК: Я, конечно, не стоял у истоков Raspberry Pi, но изначально амбиции у проекта были довольно скромные. Идея состояла в том, чтобы попытаться увлечь несколько тысяч детей изучением информатики, а получилось так, что мы дали толчок к развитию более широкого движения, суть которого заключается в демократизации доступа к технологиям. В том, как научиться делать что-то с компьютерами, а не просто использовать устройства для потребления контента. Это очень важно, и просто поразительно, какое разнообразие творческих и оригинальных идей повлекло за собой создание платформы Raspberry Pi. Семилетние дети MozFest организуют собственные мастер-классы для детей — это невероятное достижение. И они не только учатся информатике, но и приобретают целый ряд жизненных навыков, таких как навык решения проблем, ведения мастер-классов, умение думать о других. Это счастье — быть к такому приобщенным.

LXF: Это те самые «социальные навыки», которые так нужны работодателям?

ФК: Да, такие навыки работодателям требуются, и по их словам, официальная система образования неправильно готовит детей к будущему. Но это больше касается воспитания предпринимателей и навыка решения проблем: в век, когда все определяют цифровые технологии, молодые люди должны многое уметь.



LXF: Демократизация доступа к технологии означает, что пользоваться компьютером должны все или почти все.

ФК: Интересно это слышать от вас. Если бы завтра мы вдруг исчезли — чему, разумеется, не бы-

настоящий настольный Linux, целиком состоящий из открытых программ, и если через 20 лет нас спростят «Что вы считаете своим самым большим достижением?» — возможно, как раз то, что кабы не мы, многие и не узнали бы про Open Source.

О РАСТУЩИХ АМБИЦИЯХ

Raspberry Pi — это настоящий настольный Linux, целиком состоящий из открытых программ.

вать! — но если вдруг... я думаю, что есть вещи, которые произошли благодаря Фонду и которые останутся навсегда. Во-первых, изменилось представление, что за мощный компьютер нужно много платить. Цена в \$35 устоялась, и теперь перед производителями стоит задача снижения стоимости технологий. Во-вторых, мы рассказали миллионам людей об Open Source. Raspberry Pi — это

LXF: Чтобы у нового поколения был более широкий выбор открытых инструментов?

ФК: И это, и понимание того, как устроен мир. Конечно, для платных программ тоже останется место, но огромная популярность открытого ПО, такого как Mozilla и Linux, сегодня имеет немалый вес в мире, и важно донести до детей, что такие общественные начинания надо понимать и уметь в них участвовать. И на мероприятиях, подобных MozFest, собираются люди, стремящиеся сделать Интернет лучше; именно эта идея привела к созданию Raspberry Pi.

LXF: Своих целей Raspberry Pi достиг во многом благодаря сообществу, и оно является лишь одним из многих других, также разделяющих цели Mozilla.

ФК: Представлена куча диаграмм Венна, по которым можно определить, сколько сообществ съехало сюда на MozFest. Но главное, есть то, что объединяет всех, откуда бы они ни были, и это любовь к Открытому Интернету. **LXF**



НЕТ БЫСТРЕЕ LINUX-ПК

Зак Стори собирает быстрый системный блок Linux, совместив последнюю архитектуру Intel Skylake с лучшим вариантом SSD, который можно купить.



Компоненты

- » Корпус Bitfenix Nova ATX midi £ 30
- » Мobo Asus B150M-A/M.2 mATX £ 70
- » CPU Intel Core i5-6500 £ 200
- » ОЗУ Team Elite Plus 2×4 ГБ £ 42
- » PSU Kolink KL-500W £ 33
- » SSD Samsung PM961 128 ГБ M.2 PCIe SSD OEM £ 59
- » HDD WD Blue 2 TB 5400 RPM £ 62
- » Итого £ 496

Как мило наблюдать загрузку системы почти сразу после тычка в кнопку питания! Прошли те времена, когда вы успевали испить кофею, пока с жужжанием ворочаются эти диски и оживает дисплей, лениво перетаскивая вас в GUI, который вам дом родной.

И, конечно, эту роскошь невероятной скорости берегут для тех, кто готов угрохать на новую систему £1000+? На наше счастье, это уже не так. Благодаря прорывам в технологии за последние шесть лет и неослабевающим усилиям Intel по постоянному взбодрению своих чипсетов с каждым поколением, мы видим, как постепенно появляются на рынке высокоскоростные компоненты по цене, приемлемой для среднестатистического обывателя.

Жесткие диски сменились на SSD, предлагающие загрузку за время до 10 секунд и невероятную скорость передачи файлов. Однако за прошедшие три года мы увидели взлет и падение шины SATA III. Это был стандарт, существование которого

ого ожидалось вплоть до 2020 г., но уже полностью исчерпавший себя, и только стойкие HDD по-прежнему используют эти возможности соединения. К счастью для нас, в постоянном поиске увеличения производительности и скорости нашлись обходные пути. Применение для передачи данных

между жестким диском и процессором шины PCI, как оказалось, имеет почти безграничный потенциал, когда речь идет о скорости передачи данных — по крайней мере, на данный момент. И по мере достижения зрелости M.2 и U.2 PCI SSD, то же произошло с их скоростью чтения и записи, вместе с неуклонным снижением стоимости их производства. В итоге мы получили возможность собрать систему за чуть больше чем £400.

Вот по такой системе мы и намерены провести вас сегодня. Система включает четырехъядерный процессор Intel Core i5-6500, один из последних чипсетов с молниеносным и безумно энергоэффективным стандартом памяти DDR4, и любой SSD с PCIe M.2. В нашем случае мы выбрали один из дисков Samsung OEM PM961 M.2, на 128 ГБ.

Мы видим высоко- скоростные компоненты по приемлемой цене.

Выбор правильного оборудования

Выбор оборудования для новой системы — довольно жуткая перспектива для любого из нас, особенно если вы временно оказались не в теме: ведь прогресс в мире оборудования не прекращается ни на секунду. Платформы, процессоры, сокет, стандарты подключения изменяются так часто, что за ними не поспеть.

Неплохо начать с оценки своего бюджета. Определите, сколько вы готовы потратить на приобретение новой системы, загляните на www.pcpartpicker.com, измените язык на симпатичный вам [русского, увы, там пока нет, — прим. ред.], нажмите View Your System Build и принимайтесь собирать то, что вам требуется. Систему всегда следует собирать вокруг нескольких основных компонентов. В нашем случае мы знали, что требуется один из самых последних четырехъядерных процессоров Intel и какой-нибудь M.2 SSD. Прелесть PC Part Picker в том, что при выборе определенного продукта для вас заодно выделяются совместимые с ним компоненты из других списков. Укажите процессор, подходящий к сокету 1151, и сайт позаботится, чтобы вы видели только материнские платы, совместимые с этим чипом.

Product	Availability	Quantity	Unit price	Total value
Intel Core (i5-6500 3.20GHz (Skylake) Socket LGA1151 Processor - Retail Stock Code CP-586-IN	Out of stock	1	£199.99*	£199.99*
Asus B150M-AM2 Intel B150 (Socket 1151) DDR4 Micro ATX Motherboard Stock Code MB-88P-AS	In stock	1	£69.95*	£69.95*
Team Group Elite 8GB (2x4GB) DDR4 PC4-19200C16 2400MHz Dual Channel Kit - Black (TPK4D48GM2400HC16DC0)	In stock	1	£49.99*	£49.99*

➤ Естественно, поищите надежного продавца. У Overclockers UK фантастическая служба работы с потребителями, и без них нам было бы не написать эту статью.

Другой хитроумный трюк — заглянуть на www.overclockers.co.uk, это один из крупнейших специализированных розничных магазинов оборудования ПК в Великобритании. Мы поняли, как удобно

найти физическое представление того, что вы собираетесь купить. Вдобавок все ваши заказы находятся в одном месте, что облегчает обслуживание — а это явный выигрыш.

Начальная подготовка и установка БП

Итак, вы готовы создать свою новую скоростную систему. Первым делом подготовьте себе рабочее место. Вам, разумеется, понадобятся инструменты. Мы предлагаем взять набор насадок для отвертки, соединительные кабели и ножницы. Затем позаботьтесь, чтобы не наводить при сборке системы статических зарядов: не надевайте шерстяных носков и свитеров. Теперь распакуйте свой корпус, откройте боковую панель и достаньте оттуда картонную коробку с болтами и соединительными проводами. После этого надо установить блок питания с вентилятором — направленным вверх, если вы хотите отвести нагретый воздух от шасси, или вниз, если поток воздуха — не главная проблема. Закрепите блок питания четырьмя болтами. Затем уложите провода в шасси.



➤ Направление вентилятора на вашем блоке питания ему никак не повредит, но если вы ставите системный блок на пол, не забудьте про толщину ковра.

К игре готовы

Четыре ядра в Intel Core i5-6500 обеспечивают достаточную мощность для современных рабочих задач, но игроманам стоит раскошелиться на новую видеокарту. Коротко говоря, сейчас покупать новую видеокарту самое время. AMD и Nvidia переходят от 28-нм узла обработки к 14- и 16-нм соответственно, и мощность в графических лошадиных силах увеличивается экспоненциально; AMD RX 480 и Nvidia GeForce GTX 1060 феноменально оправдывают свою цену [Ред.: — Лично мне

по карману только GTX 1050]. В наших тестах GTX 1060 3 Гб оказалась самой эффективной по соотношению цены и производительности картой, которую сейчас можно приобрести. Если вы собираетесь играть при разрешении 1080p, то это две лучшие карты в данном отношении, которые можно купить сегодня. Однако если вы хотите использовать свой ПК для домашнего кинотеатра, Core i5-6500 будет более чем достаточно, чтобы осуществить ваши 4К-мечты.



➤ GTX 1060 обеспечивает отличную производительность при цене менее £180.

Установка материнской платы

Далее по списку идет материнская плата. Вытащите Asus B150M-A из антистатической упаковки и поместите на свое рабочее место. Потом найдите заднюю плату ввода-вывода внутри коробки от материнской платы. Надо правильно ее расположить и установить в заднюю часть шасси. Это защитит вашу систему от излишней пыли.

Небольшой совет: дешевые платы ввода-вывода довольно затруднительны в установке. Рекомендуем согнуть или срезать металлические скобы, закрывающие каждый из тыльных портов ввода-вывода на плате. Затем, когда вы ее установите, разместите шасси вдоль нее и поместите материнскую плату в корпус. Потом надо прикрепить материнскую плату к стойкам, уже установленным на шасси. Достаньте болты из коробки, которая была внутри корпуса Nova, и аккуратно прикрутите плату на место. Чтобы расположить ее ровно, необходимо применить некое осторожное усилие, но как только два болта окажутся на месте, всё пойдет хорошо.



► Очень удобна магнитная отвертка, если она у вас есть. Болтик, уроненный в шасси, гарантирует вам приступ злости, а то и ободранные костяшки пальцев.

Установка и охлаждение процессора

Следом идет процессор. Поднимите удерживающую лапку гнезда в середине материнской платы. Оставьте на месте пластиковую крышечку и не трогайте выводы. Возьмите CPU и найдите золотой треугольник в верхнем углу процессора. Совместите его с треугольником на крепежной скобе материнской платы и осторожно поместите процессор в гнездо. Затем верните крепежную скобу на место, вниз под болт, и снова замкните удерживающую лапку. Выскочит пластиковая заслонка.

Далее надо добавить достаточное охлаждение. Для его установки просто снимите покрытие с основания радиатора CPU (он находится в коробке от вашего CPU). Не повредите термопасту, убедитесь, что четыре запирающих штифта на внешней стороне вентилятора открыты. Затем выровняйте вентилятор относительно процессора и осторожно поместите монтажные штифты

вентилятора в отверстия на материнской плате. Надавите на каждый угол, пока не услышите щелчок, затем закройте все четыре фиксирующих штифта.

Останется только проложить провод до головки вентилятора CPU, расположенной на верхней части материнской платы, и всё готово.



► Величие Intel Core i5-6500. Совместите золотой треугольник с треугольником на скобе и на защитном покрытии, и шансов на ошибку не останется.

Будущее — Zen?

Старинная мантра гласит: если нужен дешевый и бодрый процессор, выбирайте AMD. Если нужен мощный процессор, выбирайте Intel. По крайней мере, именно так всё было лет семь-восемь назад. Увы, правила игры изменились, и стареющей архитектуре AMD уже не угнаться за такой фабрикой мощности, какой является Intel.

Skylake лидирует по удельной производительности (на ватт), и AMD просто

за ним не успевает. С его устаревшей архитектурой и чипсетами, которые сами по себе не обеспечивают ни одной из современных функций, AMD срочно нуждается в обновлении.

Здесь-то и выходит на сцену Zen. Новая линия процессоров, которая должна появиться в начале 2017 г., будет подходить гнезду AM4+, сведя все предыдущие выборы материнских плат к одной

плате. Кроме того, AMD перейдет от 32-нм производственного процесса к 14-нм, идентичному Skylake. Пока что показатели производительности выглядят весьма привлекательно, так что высокотехнологичные чипы AMD легко смогут составить конкуренцию Intel. Это должно снизить цены на процессоры Intel и оживить рынок, и это означает большую мощь по более приемлемым ценам.



► Сможет ли AMD наконец-то обогнать Intel благодаря новому чипсету, новому гнезду и усиленной мощности?

Установка жестких дисков

Займемся хранилищем. Во-первых, вам нужен SSD Samsung PM961 на 128 Гб. Поскольку это диск OEM, он будет просто завернут в упаковочный полиэтилен. Стоит отметить, что это довольно маленькое устройство. Найдите слот M.2 на материнской плате (в нашем случае — прямо под процессором) и открутите верхний болт на самой дальней стороне от слота M.2. Он крошечный и серебристый. Далее поместите диск M.2 в порт, совместив коннекторы с пазом на диске. Диск должен оказаться под очень странным углом, но это нормально. Осторожно нажмите на диск и закрутите болт, который мы только что вывинтили, на свое место, зафиксировав диск M.2.

Затем перейдем к 2-Тб жесткому диску. Просто установите его в отсек под 5,25-дюймовой нишей, чтобы коннекторы были обращены к блоку питания, и закрепите двумя болтами с каждой стороны. Теперь присоедините один из кабелей SATA, идущих с материнской платой, к L-образному коннектору и подключите его к верхнему соответствующему по форме L-образному SATA порту на плате.



› Жесткий диск размещается в самом низу, чтобы, если позднее мы решим поставить вентиляторы, этот беспокойный 3,5-дюймовик их не заблокировал.

Память и кабели

Следующий важный компонент — ОЗУ. Для его установки откройте защелки на слотах памяти, расположенных справа от процессора. Надо разместить каждую планку памяти в соответствии с пазами на плате и установить каждую планку на место, подталкивая под оба конца, пока не услышите заветный щелчок. Теперь кабели. Надо подключить самый большой 24-контактный кабель ATX к материнской плате. Установите 8-контактный блок питания CPU поверх платы (контакты похожи на применяемые в платах расширения и GPU, но у каждого своя форма, чтобы вы не поместили блок куда не надо). После этого подключите питание SATA к жесткому диску. Найдите то, что похоже на чуть удлиненный L-образный кабель SATA, и установите его рядом с кабелем

SATA, установленным ранее. И, наконец, передняя панель ввода-вывода. Найдите кабели, убранные в 5,25-дюймовую нишу, и вытащите их из корпуса. Вам нужны кабели с маленькими контактами квадратного сечения, с пометками 'Power', 'Reset', 'LED +', 'LED -' и 'HDD LED'. Откройте руководство к материнской плате и найдите раздел про передние коннекторы ввода-вывода. Подключите кабели к соответствующим контактам на материнской плате, подключите гнездо USB 3.0 на передней панели USB 3.0 к коннектору на плате, как и передний вывод аудио, и у вас получится отличная система. LXF

› Проложите кабели как можно аккуратнее: если они не будут перепутаны, впоследствии это избавит вас от лишнего раздражения.



Будущее хранения данных

SSD впервые появились более шести лет назад, в 2010 г. Они были недолговечными, не слишком быстрыми и до смешного дорогими. С тех пор мы наблюдали, как увеличивается их емкость, падает цена, а скорость и стабильность возрастают, и сейчас можно купить 120-Гб SSD, способный на последовательное чтение и запись при скорости до 550 МБ/с, менее чем за £ 50. Однако вскоре после того, как возможности этой шины были исчерпаны, заступилась M.2.

У преемника mSATA, M.2, идея была проста: полностью обойти SATA, передавая важные данные

через шину PCIe. Таким образом вы можете обойти предел скорости 565 МБ/с и получить куда более быстрые диски в куда меньшем форм-факторе. Сначала эти диски PCIe шли как большие объемистые платы расширения, с радиаторами и чипами памяти. Они по-прежнему использовали шину PCIe, но им не удавалось подключиться к слоту M.2.

И так было, пока Samsung не создал свою первую серию дисков PM951 и SM951; тут мы и увидели сходную производительность при намного меньшем форм-факторе. Сегодня рынок перегрет множеством SSD с PCIe, размером меньше ладони

и с потрясающей скоростью. Самые популярные SSD сейчас 950 и 960 Pro и Evo, со скоростью чтения и записи до 3,5 Гб/с и емкостью 2 Тб.



› Благодаря агрессивной ценовой политике вы можете найти диски SSD с M.2 PCIe куда дешевле, чем традиционные 2,5-дюймовые SSD.



ПО ВОЛНАМ МЕЧТЫ JOLLA

Пролагайте путь по беспокойным водам на титанической Sailfish OS от Jolla с помощью Шона Кэмерона.

Святой Грааль для мобильных производителей — это безупречная со всех сторон бизнес-модель. В теории, она легко достижима: создайте свой собственный магазин приложений и операционную систему, установите их на ряд удобных первоклассных устройств и радуйтесь доходам.

На самом деле реализовать такое, разумеется, исключительно сложно. В начале эры смартфонов это был приз, на который теоретически претендовала любая компания; с тех пор многие потерпели поражение, некоторые преуспели, а некоторые вообще с трудом выжили, и Jolla — что переводится с ее родного финского как «шлюпка» — относится к последней группе.

Основанная в период спада, последовавшего за мучительной кончиной подразделения сервисов и устройств компании Nokia, Jolla цеплялась за жизнь с поразительным упорством. Другие, более крупные и богатые фирмы канули в Лету, но шлюпка Jolla, судя по всему, оказалась совершенно непотопляемой.

Шел 2013 год, когда Jolla впервые появилась на свет, подобрав последние тлеющие угольки MeeGo, последней попытки Nokia создать встроенную операционную систему, и превратив их в новую платформу под названием Sailfish OS.

Sailfish OS

Сами Пиенимакки [Sami Pienimäki], соучредитель финского стартапа, относится к MeeGo по-прежнему нежно, как и многие другие люди в Jolla, включая ряд бывших инженеров, работавших над незадачливой платформой Nokia.

Sailfish основана на внутреннем коде MeeGo с новым интерфейсом пользователя, разработанным под активное употребление жестов. Jolla гордится созданным в Sailfish OS, но тем не менее возникает упорное ощущение, что всё присутствующее там по-прежнему не доделано.

С самого начала Jolla отличали астрономические амбиции, несмотря на штат менее 200 сотрудников. Целью компании всегда было задревание пробелов, имеющихся в Android (таких как недостаток функций безопасности), плюс предложение расширенного взаимодействия с пользователем: «Наш главный конкурент — Android. Конечно, другие основные ОС тоже присутствуют в этом списке», сказал Пиенимакки в беседе с *TechRadar*, перед тем, как перейти к уникальному коммерческому аргументу ОС Sailfish.

«[Это] действительно независимая мобильная ОС, предлагающая миру реальный выбор на плоском/однородном рынке мобильных ОС».

➤ Sailfish OS от Jolla выросла из проекта Nokia MeeGo.



За недолгую жизнь Jolla кое-где создается ощущение, что компании не хватает стратегии. Цель существовала всегда, хотя по большей части выступала в роли мечты; однако дорога к этой мечте была полна препятствий и ошибок.

Возьмем изначальную мечту: используя начинку MeeGo, создать новую операционную систему с ультрасовременным вводом посредством жестов, красивой эстетикой и кучей новых идей. У ОС Sailfish был прочный фундамент, но ее масштабирование и превращение солидного старта в жизнеспособный и прибыльный продукт оказалось вне компетенции стартапа.

Первым официальным устройством с ОС Sailfish стал Jolla Phone, вышедший в 2013 г. Массивный и узнаваемый, он отличался неким неувловимым своеобразием, которого были лишены многие тогдашние устройства. Несмотря на некоторый успех у себя на родине, телефон не закрепился в других местах — и на этом проблемы Jolla только начались.

Столкнувшись с низкими продажами и таянием финансов, в Jolla переосмыслили всю свою бизнес-модель. Йоллафону позволили жить и дальше, но уже как вещи коллекционной. Тем временем внимание переместилось с продажи оборудования на лицензирование ПО, переход с весьма ощутимыми последствиями. Jolla вознамерилась завоевать мир. Открылись офисы в Гонконге, и появились партнерства в Африке за Сахарой, Индии, Южной Америке и в других местах.

Компания выискивала мелких производителей, предлагавших устройства, потенциально способные использовать ПО от Jolla по-новому и прославить обе стороны, как объясняет Пиенимакки: «Идеальными партнерами Jolla в основном являются пионеры в своей области; те, кто хочет выйти за привычные рамки и предложить нечто экстраординарное...».

Однако время шло, а эти устройства так и не материализовались, и цифровые мечты Jolla развеялись как дым. И только в 2016 г. появилось первое устройство не от Jolla с установленной ОС Sailfish: Intex Aqua Fish.

Но как же, несмотря на все свои неудачи, этот крошечный игрок на задворках северного полушария — и на задворках мобильного рынка — умудрился выжить?

Конечно, ностальгия и сентименты сыграли свою роль, хотя говорить подобное значит третировать тысячи тех, кто решил следовать за Jolla. Что ими двигало? Ответ прост: надежда. Надежда продала тысячи планшетов Jolla, и надежда заставила энтузиастов от венчурного капитала предоставлять ей средства в тощие годы. Надежда не остается без последствий, и у Jolla по-прежнему имеется огромное сообщество.

Оно состоит из групп энтузиастов открытого кода и бывших сотрудников Nokia, а также тех разработчиков-любителей, которые стремятся к чему-то отличающемуся от привычных норм. Одна группа дошла даже до того, что создала напечатанную на 3D-принтере накладку на клавиатуру для смартфона фирмы.

Понятно, что Jolla решила сосредоточиться на этом сообществе, спасительном фарватере в бурном море. Каждая новая



› Jolla C — ограниченное издание для разработчиков и сообщества.

сборка операционной системы подвергается совместному тестированию, и Jolla охотно сотрудничает со всеми, кто принимает в нем участие: «Наше сообщество разработчиков и фанатов всегда было в центре нашего рабочего процесса, — сказал нам Пиенимакки. — Пару месяцев назад мы запустили специальную программу Sailfish Community Device, включающую последний смартфон Jolla, 'Jolla C — Limited Community Edition' и прочую деятельность, например, мероприятия сообщества и сессии поддержки разработчиков».

Jolla C

Последнее устройство, выпущенное фирмой (см. Обзоры, стр. 15) — Jolla C, и оно необычно по целому ряду причин. Во-первых, оно не разработано и не выпущено Jolla, как это было с Jolla Phone, а является ребрендингом Intex Aqua Fish. Во-вторых, оно продается как модель только для разработчиков, производимая небольшими партиями по несколько сотен в каждой и продаваемая на флэш-распродаже. Весь имеющийся запас может быть полностью распродан, что создает ряд преимуществ, не последнее из которых — отсутствие необходимости вести официальную базу поддержки потребителей телефона, поскольку он предназначен только для использования в качестве устройства поддержки для разработчиков Sailfish.

Хотя настроение в Jolla, похоже, боевое, разделить их энтузиазм достаточно сложно. Многие из первых учредителей (включая лицо компании, Марка Диллона [Marc Dillon]) ушли вместе с десятками работников, уволенных после сокращения отдела оборудования. Бизнес-партнерства оказались неплодотворными, и финансирование иссякает. В декабре 2015 г. Jolla застолбила в фонде спасения \$12 миллионов, чтобы остаться на плаву, но не снят вопрос: что дальше? Между молотом iOS и наковальней Android, какие практические опции можно изыскать для своего расширения?

По этой части Пиенимакки явно предпочитает не раскрывать своих карт. «Сейчас у нас есть несколько клиентов-лицензиатов, с которыми мы работаем», говорит он. Одним из этих клиентов стало правительство России, которое использует SailfishOS для снижения зависимости от иностранных ОС. В таком случае может оказаться, что с этой бесстрашной маленькой шляпкой еще не покончено. LXF

› Читайте наш обзор про Jolla C на стр. 15.



Проблемы с планшетами

Анонсированный в конце 2014 года как «первый в мире планшет совместного финансирования», Jolla Tablet выглядел вполне достойно. Он обещал яркий 7-дюймовый экран 2048×1536, четырехъядерный 64-битный процессор Intel и много чего еще — и людей это привлекло. Планшет снискал огромный успех при сборе средств, собрав невероятную сумму в \$2571262 — 479% от изначальной цели. Понятное дело, что в Jolla были вне себя от радости.

Однако за этим последовала литания невыполненным обещаниям. Начальные задержки привели к месяцам ожидания с сообщениями о спорах с производителем, помимо прочих проблем. В конечном итоге отгрузили только несколько сотен планшетов, а отнюдь не ожидаемые тысячи.

Jolla признала свое поражение, и была запущена крупномасштабная программа рефинансирования — и официально эта ситуация рассматривалась

как ценный урок для компании, по крайней мере, по словам Пиенимакки: «Проект Jolla Tablet был для нас отличным уроком, и кроме того, это отличное устройство для Sailfish OS, — говорит он. — У небольшого стартапа типа Jolla обычно уходит несколько лет на поиск правильной стратегии и цели, и проект Jolla Tablet оказался ступенькой на пути к более четкой лицензионной стратегии бизнеса».



По советам м-ра Брауна

Джوليон Браун

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

Эзотерическое системное администрирование из таинственных закоулков серверной.

Это я-то параноик?!

Интересно, открыл ли кто-нибудь из причастных пару бутылочек шампанского Voërl & Kroff Brut (каких-то £1957 за бутылёк) по случаю принятия в ноябре Акта о полномочиях следствия? В Open Rights Group (www.openrightsgroup.org) назвали этот закон, предыдущие версии которого не могли пройти через парламент (отсюда и повод выпить шампанского), «одним из самых суровых законов по слежке за гражданами, принятых в демократических странах».

Я уверен, что изначальные намерения по принятию этого закона были самыми благими. Однако с учетом разоблачений Сноудена и недавнего судебного решения, вынесенного следственным трибуналом, о том, что службы безопасности несколько лет незаконно собирали массу данных об общении, меня можно простить за дурные предчувствия. «Но какое отношение это имеет к нам, сисадминам и инженерам?» — слышу я ваш вопрос. Так вот, самое прямое. Видите ли, сисадмины были многолетней целью для сотрудников спецслужб (наберите в поиске “I hunt sysadmins” для файлов Сноудена). Но принятый в ноябре закон имеет ряд пунктов, способных обеспокоить почти всех, кто выполняет задачи системного администрирования повседневно. Например, если вы или ваш работодатель вдруг получите «Уведомление об изменении технических характеристик», то с вас могут потребовать, например, отключить шифрование для службы, с которой вы работаете, чтобы госбезопасность получила доступ к данным или трафику, которые иначе не были им доступны (рамки в законе очень широки). Хуже того, вам грозят тюрьма или штраф всего лишь за извещение о том, что подобное «уведомление» имело место. Пожалуйста, сделайте одолжение, ознакомьтесь с этим законом и узнайте, как он влияет на вас, если в ближайшем будущем не хотите попасть в прескверную ситуацию.
jolyon.brown@gmail.com



Microsoft вошла в Linux Foundation

Платиновое членство для гиганта из Редмонда, вывод LUKS из строя нажатием одной кнопки и доминирование Linux на главных 500 суперкомпьютерах (снова).

Гигант из Редмонда продолжает двигаться по пути принятия Linux — да так, как всего несколько лет назад нельзя было себе представить! Microsoft объявила, что присоединяется к Linux Foundation как участник платинового уровня (максимального из доступных). Это может быть отражением того факта, что почти на каждой третьей виртуальной машине в облаке Azure (предоставляемом Microsoft) запущен Linux. Это большой шаг вперед по сравнению с временами, когда Стив Балмер [Steve Ballmer], бывший директор Microsoft, высмеивал Linux, называя его «раковой опухолью».

Первая публичная предварительная версия Microsoft SQL Server для Linux появилась в ноябре, и небольшие исследования показывают, что в ней используется ядро Windows, запущенное в «песочнице». Так Microsoft выполняет свое обещание сделать свою ведущую базу данных доступной тем

клиентам, которые пользуются только Linux. Однако корпорация, как сообщается, получила урок по безопасности Linux, когда программисту Яну Даффи [Jan Duffy] удалось обнаружить в используемой инфраструктуре RPM уязвимость, которая теоретически открывала root-доступ к каждой системе Linux, в которой эта инфраструктура использовалась для обновления и применения патчей. Microsoft исправила ошибку и заплатила премию за ее обнаружение.

Сам Linux тоже не избежал этой участи — не так давно в реализации спецификации Linux Unified Key Setup (LUKS) была обнаружена довольно нелепая уязвимость. Чтобы она проявилась, нужно нажать и удерживать клавишу Enter в системной консоли 70 секунд. LUKS используется в большинстве дистрибутивов для шифрования пользовательских данных. Однако при нажатии и удерживании клавиши в течение долгого времени злоумышленник неожиданно попадает в оболочку root initramfs. А в этой оболочке можно воспользоваться другими потенциальными уязвимостями. Также предполагается, что уязвимость способна затрагивать облачные системы, которые обычно предоставляют удаленный доступ с помощью консоли.

И — на закуску: Linux снова возглавил рейтинг операционных систем на 500 главных суперкомпьютерах, он используется на 498 из них. Лишь пара стойких упрямяцев на основе AIX не позволила Linux оккупировать весь список.



» В 2001 г. я не знал, что покажется более невероятным — что Дональд Трамп станет президентом США или что Microsoft будет спонсировать Linux.

Red Hat GlusterFS

Из старых серверов, пылившихся у него в гараже, Джолион создает распределенную ФС, которой по плечу серьезнейшие рабочие нагрузки.

Помню, много лет назад один из коллег прислал мне вывод команды, содержащий сведения о новой файловой системе. Не какой-то произвольной файловой системе, а о первой когда-либо созданной файловой системе, преодолевшей магический барьер в 1 ТБ. Мало того (жаль, что вас при этом не было), эта файловая система зеркалировалась между дата-центрами с помощью проприетарного ПО, которое мы тогда использовали, через выделенную линию обмена данными, которая могла обрабатывать огромный (по тем временам) объем данных, и SAN, с помощью которого наши до неприличия дорогие серверы подключались к столь же дорогим дисковым массивам. Счастливые времена, в самом деле. По крайней мере, для тех, кто продавал «железо». Сегодня всё иначе, и стоимость «железа» снизилась во много раз (если оно вообще кому-то нужно). К тому же у нас в распоряжении любимая операционная система и немыслимый выбор открытого ПО для создания таких систем. Но кое-что осталось прежним. Эти ужасные пользователи всё еще создают данные в форме файлов, думают, что смогут делать это вечно и пространство никогда не закончится, а заодно ожидают, что данные останутся на месте, когда они захотят обратиться к ним в следующий раз. Они также хотят, чтобы всё это было дешевле, чем раньше, если нетрудно. Спасибо!

Я не жду, что ныне моя статья сумеет восхитить кого-то так же, как меня в те дни восхитил этот великолепный монстр размером 1 ТБ. Но расскажу о файловой системе под названием GlusterFS, которой я пользовался довольно долго и восхищался в течение этого периода. Если вам нужно предоставить большой объем хранилища большому количеству конечных точек и пользователей, думаю, на нее определенно стоит взглянуть.

Прежде всего следует сказать, что эта файловая система не такая уж и новая. GlusterFS в различных версиях существует уже не первый год. Не знаю, как вы, а я это учитываю, когда дело касается хранения данных (особенно моих): версия 0.1a-pre — спасибо,

не надо. Первую версию файловой системы Gluster Inc разработала еще в 2007 г., а в 2011-м всю компанию и ее разработки поглотила Red Hat, заплатив за это чуть больше \$100 миллионов. Продукт Red Hat Gluster Storage можно приобрести с контрактом на поддержку (тем, кому она нужна). Но, конечно же, GlusterFS остается полностью открытой и доступной для всех желающих ей воспользоваться.

Основы

GlusterFS позиционируется (www.gluster.org) как «масштабируемая сетевая файловая система». Ее можно собрать на обычном доступном «железе». Она распределенная и масштабируется в диапазоне до нескольких петабайт (для счастливых владельцев таких объемов хранилища). Она поддерживает несколько тысяч клиентов, которые могут обращаться к ней различными способами: например, с помощью NFS, SMB/CIFS или небольшого специализированного драйвера для самой GlusterFS. Отработка отказа и восстановление могут выполняться автоматически. Если один из узлов на какое-то время станет недоступным, опасаться практически нечего. Эта файловая система работает наряду с известными файловыми системами Unix, такими как ext4, xfs (или даже zfs). У этой файловой системы нет «центральных» серверов, о которых следует беспокоиться. Читатели моей рубрики наверняка знакомы с такими. Это маленькие прямоугольнички, которые расположены сбоку от основного слайда в презентациях поставщиков и от которых зависит вся система. Эти серверы всегда оказываются автономными и не имеют готового решения для высокой доступности, которое «обязательно появится в следующей версии», как с лучезарной улыбкой говорит вам специалист по продажам (если вам кажется, что в моих словах звучит горечь, то так оно и есть).

Но я отвлекся. С GlusterFS все эти тревоги вам не угрожают. Прежде чем знакомиться с демо-версией, важно понимать, что

»

Gluster process	TCP Port	RDMA Port	Online	Pid
Brick gluster1:/export/brick1/gv0	49152	0	Y	10331
Brick gluster2:/export/brick1/gv0	49152	0	Y	4217
Brick gluster3:/export/brick1/gv0	49152	0	Y	3450
NFS Server on localhost	N/A	N/A	N	N/A
Self-heal Daemon on localhost	N/A	N/A	Y	10357
NFS Server on gluster2	N/A	N/A	N	N/A
Self-heal Daemon on gluster2	N/A	N/A	Y	4239
NFS Server on gluster3	N/A	N/A	N	N/A
Self-heal Daemon on gluster3	N/A	N/A	Y	3477

» Представьте себе, сколько в те дни могла стоить масштабируемая сетевая файловая система с такими возможностями, как у Gluster...

GlusterFS CLI 101

Как и все хорошие продукты, Gluster сильно завязан на командную строку (во всяком случае, открытая версия — я не проверял версию продукта с поддержкой Red Hat). Существует несколько основных команд, которые стоит знать с самого начала, особенно если что-то пойдет не так (это возможно, невзирая на отказоустойчивость Gluster). Команда `$ gluster volume info` делает именно то, что от нее ожидается: выводит информацию о состоянии томов в доверенном пуле. Чтобы убедиться, что узлы «видят» друг друга, скомандуйте `$ gluster peer`

`status`. С помощью команд `$ gluster volume profile <volname> start|info|stop` удобно получить метрики производительности, особенно если есть подозрения, что один из узлов каким-либо образом расходится с остальными.

Другая по-настоящему полезная команда — `$ gluster volume heal <volname> info`, она выводит некоторую информацию о текущем состоянии содержимого тома. Если файл надо «лечить», вы увидите это в списке. Чтобы «вылечить» файл, чаще всего достаточно просто прочесть файл

на клиентском компьютере Gluster. При этом запускается процесс «самолечения» (`glfsheal`), который в 95% случаев работает в фоновом режиме. Однако лечение можно запустить и вручную — командой `$ gluster volume heal <volname>`. Если всё закончится плохо, файл окажется в состоянии `'split brain` [раздвоение личности]! Исчерпывающее руководство по решению этой проблемы (для которого часто достаточно выбрать файл большего размера при сравнении различий в файлах) см. на официальном сайте <http://gluster.readthedocs.io>.

в Gluster хорошо и что плохо. Gluster прекрасно работает с неструктурированными данными. У вас есть несколько резервных копий, несколько электронных таблиц и огромный архив фотографий с котятками? Тогда GlusterFS — ваш друг. Подходит ли она для структурированных данных? Уже не так хорошо. Не стоит запускать на ней свою базу данных. Это во всех отношениях плохая идея. На наиболее фундаментальном уровне, GlusterFS — просто набор обычных файловых систем (в терминологии Gluster — «кирпичиков»), которые предоставляются для общего доступа или экспортируются из нескольких систем (узлов) и объединяются в один «том» Gluster; таким образом, операции чтения и записи данных становятся общими для этих систем. Конечно, это далеко не всё, и существует несколько вариантов организации таких систем, но базовый принцип именно таков.

В компаниях вроде Facebook (где Gluster применяется довольно широко — есть несколько интересных презентаций об используемой ими конфигурации) узел может состоять из 100 ТБ хранилища, распределенного по нескольким десяткам жестких дисков. Локально они могут быть организованы с помощью аппаратного или программного RAID. На каждом из «кирпичиков» будет использоваться xfs (или другая предпочтительная файловая система Linux). При обращении с этими узлами следует выполнять все обычные действия. Их надо подготавливать и обслуживать. Диски будут выходить из строя, и их надо будет заменять. Необходимо применять исправления. С таким масштабом «доверенный пул» узлов Gluster в каком-нибудь дата-центре может занимать несколько стоек, и тома могут состоять из нескольких очень больших «кирпичиков». У меня здесь цели гораздо менее амбициозные.

➤ На сайте масса документов, охватывающих все аспекты использования Gluster.



Возможно, вы уже заметили, что 'Gluster' и 'GlusterFS' часто заменяют друг друга при разговоре об этой программе (это заметно даже на сайте проекта). В командной строке используется `gluster`.

Мой пример работает на трех локальных вычислительных машинах (Ubuntu 16.04). У каждой из них есть небольшой дополнительный диск, который я употребляю для своего тома Gluster. Я настроил несколько записей в `/etc/hosts` вместо DNS. Также можно было использовать физические устройства или облачные провайдеры. Хосты имеют имена `gluster1`, `gluster2` и `gluster3`.

Первый шаг — установка самой GlusterFS. В Ubuntu надо установить пакет `glusterfs-server` (выполнив все следующие команды на всех трех узлах).

```
$ sudo apt-get install glusterfs-server
```

Тут пора установить файловую систему и смонтировать ее. Мой запасной диск — `/dev/sdb`. В рабочих средах чаще всего используется XFS, но у нас это просто пример, поэтому я воспользуюсь `ext4`.

```
$ sudo mkfs.ext4 /dev/sdb
```

```
$ sudo mkdir -p /export/brick1
```

Затем добавлю следующую строку в конец файла `/etc/fstab file`:

```
/dev/sdb /export/brick1 ext4 errors=remount-ro 0 1
```

Теперь сделаю свой новый «кирпичик» доступным на каждом сервере командой `$ sudo mount -a command`.

Попробуем выполнить несколько команд Gluster. С первого сервера я смог настроить доверенный пул с помощью простой команды `peer probe`:

```
$ sudo gluster peer probe gluster2
```

```
$ sudo gluster peer probe gluster3
```

Получив сообщение об успешном результате (`peer probe: success`), я могу выполнить команду `sudo gluster peer status` на каждом из других узлов, у каждого из которых есть два соседних сервера, идентификатор `uuid` и состояние `Peer in Cluster (connected)`. Пока всё хорошо.

Реплицированный том

Теперь я хочу создать на каждом сервере новый каталог `/export/brick1/gv0`. Сделав это, можно будет создать и запустить том Gluster с первого узла:

```
$ sudo mkdir -p /export/brick1/gv0 # do this on each node
```

```
$ sudo gluster volume create gv0 replica 3 gluster1:/export/brick1/gv0 gluster2:/export/brick1/gv0 gluster3:/export/brick1/gv0
```

```
$ sudo gluster volume start gv0
```

```
$ sudo gluster volume status
```

Команда `$ df -h` на любом из узлов ничего не выведет. Том может сообщать, что он запущен и работает, но как к нему обратиться? Пока я просто воспользуюсь одним из серверов (`gluster3`) в качестве клиента:

```
$ sudo mount -t glusterfs gluster1:/gv0 /mnt
```

Теперь на всех трех узлах во всей своей красе видно `/gv0`. Но работает ли он?

Типы томов

Наряду с реплицированным томом, который я создал в основной статье, GlusterFS предлагает множество других вариантов для учета различных обстоятельств (обо всех я рассказать не смогу). Большинство из них представляют собой компромисс между стоимостью, избыточностью/масштабируемостью и производительностью. Самый простой тип тома — «распределенный», при котором файлы распределяются по «кирпичикам» в поле, но не предоставляется никакой избыточности. В томах с чередованием данные перемежаются

по «кирпичикам» тома, что бывает удобно для сценариев с высокой степенью параллелизма, вовлекающих большие файлы. Также существуют сочетания базовых типов: «распределенные с чередованием», «распределенный с репликацией» (и, само собой, «распределенный с чередованием и репликацией»).

Обычно «распределенный» означает, что важнее всего масштабирование, а «репликация» должна обеспечить высокую доступность. Тома с чередованием обычно нацелены на хорошую

производительность. Лично мне не попадались «распределенные» тома, но они, очевидно, основаны на «кодах стирания», форме защиты данных, при которой файлы разбиваются на фрагменты. Они хранятся в каждом «кирпичике» таким образом, что для восстановления исходного файла требуется только их часть. При разработке сервиса с GlusterFS в качестве компонента важно понимать, как будут использоваться данные и как к ним будут обращаться, чтобы выбрать наилучший вариант из возможных.



redhat Technologies Services & support Success stories About Red Hat

STORAGE

Red Hat Gluster Storage

OPEN, SOFTWARE-DEFINED STORAGE

Scale-out NAS storage for modern enterprise workloads

Hardware-based storage segregates information, making it hard to find, access, and manage. Adding capacity to your traditional system often disrupts access to data. And if your hardware fails, it can bring your business to a standstill.

Red Hat® Gluster Storage is different. It's open, software-defined file storage that scales out as much as you need. You can easily and securely manage large, unstructured, and semi-structured data at a fraction of the cost of traditional, monolithic storage. And only Red Hat lets you deploy the same storage on premise; in private, public, or hybrid clouds; and in Linux® containers.

TRY BUY DOWNLOAD

Test-drive Red Hat Gluster Storage. Amazon Web Services makes it easy—and free.

TRY IT NOW

› Для желающих иметь службу поддержки, куда можно обратиться при возникновении проблемы, доступна версия с коммерческой поддержкой от Red Hat.

```
$ sudo touch /mnt/test # from the same machine as the mount command was run
```

```
$ sudo ls /export/brick1/gv0 # from one of the other nodes
```

Эти команды покажут, что мой тестовый файл реплицировался в соответствующие «кирпичики» программой GlusterFS — потому что при создании тома я выбрал параметр 'replica' (доступны и другие варианты). Данная базовая схема прекрасно подходит для простых ситуаций.

Самые внимательные из вас, возможно, уже заметили потенциальный недостаток моей великолепной схемы: я напрямую смонтировал GlusterFS из своего узла gluster1. Если узел выйдет из строя, не окажется ли мой злосчастный клиент у разбитого корыта? Проверить это довольно легко — достаточно выключить gluster1. После этого gluster3 покажет, что файловая система GlusterFS все еще смонтирована в /mnt и я могу просмотреть тестовый файл командой ls. Это вызвано тем, что клиент знает о других узлах в доверенном пуле и просто пользуется ими.

После этого можно поработать с gluster1 в автономном режиме, чтобы проверить репликацию данных на узлы при их возврате в рабочий режим. Создание другого файла на gluster3 командой `$ touch /mnt/test2` работает, как можно было ожидать. При включении gluster1 он довольно быстро переходит в состояние «узел в кластере [peer in cluster]», и отсутствующий файл моментально появляется в соответствующем «кирпичике» без вмешательства с моей стороны.

Различные клиенты и преобразователи

Используемый здесь клиент GlusterFS представляет собой файловую систему пространства пользователя. Очевидно, это решение было принято разработчиками GlusterFS изначально, так как добавление модулей в ядро считалось очень долгим и сложным. Клиент использует модуль ядра FUSE (файловая система в пространстве пользователя). По моему опыту, всё работает хорошо. Но некоторые из ваших клиентов не смогут воспользоваться этой системой (не везде работает Linux для новичков, верите вы или нет). Это добавляет несколько иные трудности, которые заключаются в том, что отработки отказа не получится, если пользователь не включит

какое-либо средство балансировки нагрузки. Мне очень повезло, и я с этим не столкнулся. Варианты могут включать широко известные механизмы VIP (virtual IP), например, периодической активации [keep-alive]. Однако я видел в Сети обсуждение использования базы данных Cluster Trivial Database (CTDB), возникшей из проекта Samba. С ее помощью можно сделать так, чтобы клиенты CIFS и NFS переключались на альтернативные серверы GlusterFS, если используемый ими сервер по какой-либо причине перешел в автономный режим.

Увы, пространство в печатном журнале ограничено, но прежде чем закруглиться, стоит упомянуть преобразователи. Это элементы GlusterFS, которые выполняют различные задачи, такие как репликация, блокировка, проверка квот и т.д. Честно говоря, для простых схем они просто не нужны, но с (возможным) развитием системы стоит потратить некоторое время, чтобы понять, что делают такие преобразователи, как DHT (Distributed Hash Table — распределенная хэш-таблица) и AFR (Automatic File Replication — автоматическая репликация файлов). DHT занимается размещением файлов в подтомах, и это — настоящее сердце GlusterFS. Тогда как AFR поддерживает согласованность реплик и может обеспечить восстановление данных при сбоях (если одна копия данных осталась доступной).

Существует масса вещей, о которых я не успел рассказать — например, моментальные снимки, которые обладают довольно большим набором функций, или более экзотические применения Gluster, вроде SwiftOnFile (в качестве серверной части OpenStack Swift). Можно даже использовать ZFS для резервного хранилища файлов для Gluster, хотя в инструкциях есть не слишком вдохновляющее предупреждение 'Moniti estis [Готовьтесь к худшему]'.

Примерно в то же время, когда происходила история с файловой системой, о которой я рассказывал в начале, мой работодатель заплатил огромную сумму за кластеризованную версию проприетарной программы для создания монстра объемом 1 ТБ. Программа была ужасна! Она была медленной, сложной в настройке и часто «падала». Очень часто, придя в офис, мы видели: программа «рухнула» из-за того, что пользователи вошли в систему и создали для системы не очень и большую нагрузку. Эх, будь Gluster доступна мне тогда — возможно, сейчас у меня было бы меньше седых волос [Ред.: — Где там волосы?!] LXF

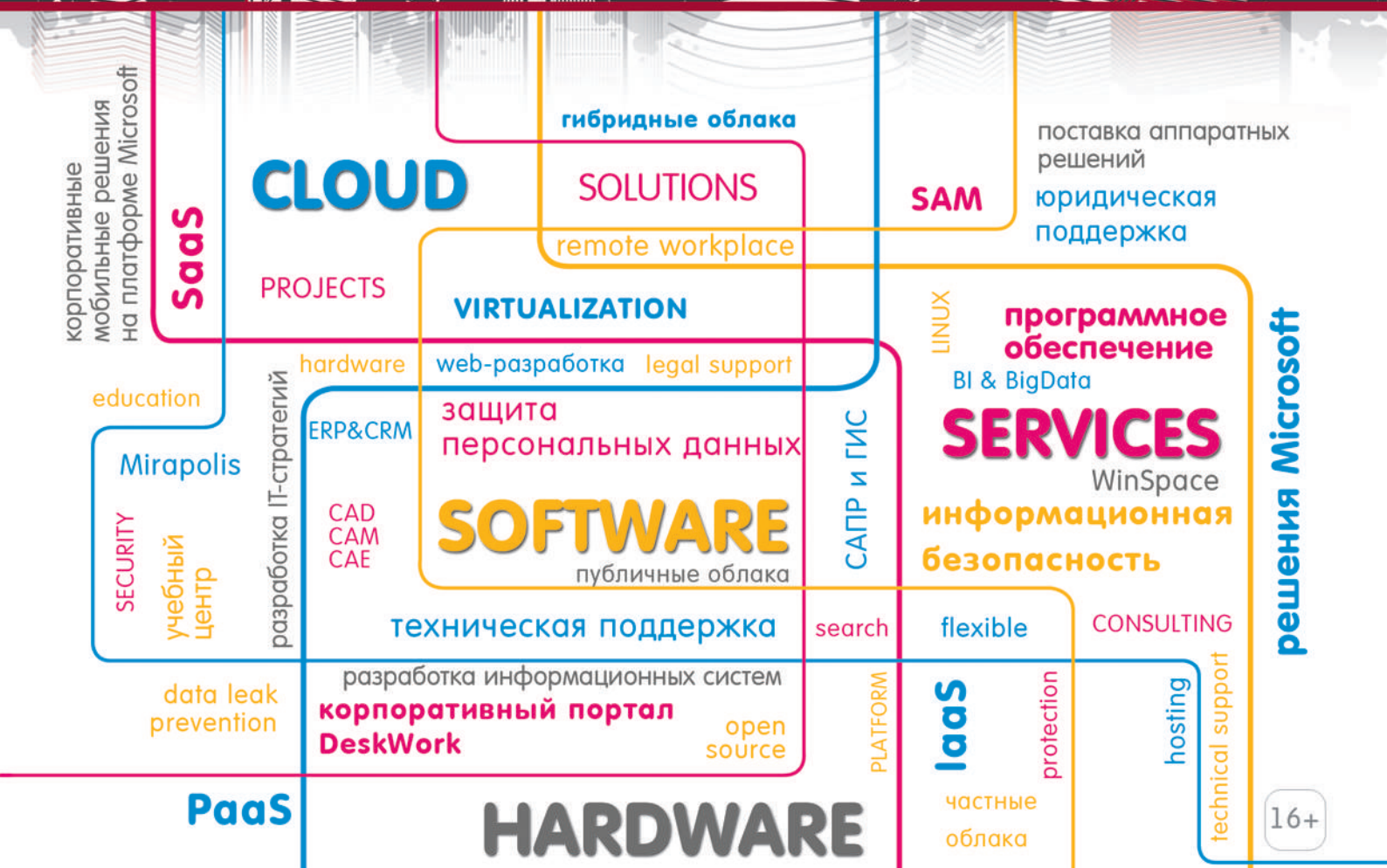
softline®

Cloud Software Hardware Services

20+

Years in IT

IT-архитектура вашего бизнеса



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

Свободные движки для инди

— Привет, чеволек!
The Ur-Quan Masters
("Hello, hunam", вариант перевода)

Мир меняется. Временами — пугающе, но приходится признавать, что «ётюберы» могут иметь аудиторию много большую, чем традиционные «профессиональные» СМИ; на чтение «снэпов» подростки тратят гораздо больше времени, чем на чтение книг; а игры соревнуются по кинематографичности с полноценным кино. Новые веяния не заменяют прежние средства передачи информации полностью, но весьма сурово дополняют прежнюю реальность, виртуальную и не очень.

Если глянуть в сторону инди-игроделов, то я уверен, что число шедевров, появляющихся в единицу времени, увеличивается, хотя в нагрузку идет «девятый вал» игровых проектов качеством «так себе» и «глаза бы не глядели». Но вкусы и планки у людей довольно разные, и в теории у каждого создателя найдутся свои поклонники.

Кан Гао своей миниатюрой *To The Moon* продемонстрировал, что «графоний — это далеко не все», а свободный проект tkxр сделал возможным лицезреть его произведение на свободных же платформах. Благодаря открытому проекту Adventure Game Studio нам доступен *Gemini Rue*, а питоновский движок Ren'Py дал путевку в мир повести *Analogue: A Hate Story*. Порог вхождения на «мировую арену» постоянно снижается, и часто это делается за счет развития свободного ПО. E.m.Baldin@inp.nsk.su



В этом месяце вы научитесь...

- | | |
|---|---|
| Работать с переменными 52 | Выбирать эмуляторы 54 |
| Чего только Ник Пирс с переменными ни выделывает! И создает, и экспортирует, и редактирует. Преображает, короче, среду окружения. | Никку Пирсу показалось мало стандартного Gnome Terminal, и он перебирает альтернативные варианты — среди них встречаются любопытные экземпляры. |
| Сканировать систему 56 | Снимать диски 58 |
| В заботе о безопасности, Таджидер Калси написал книгу спасительных рецептов. По его мнению, один скан в день уберет вас от злых хакеров. | Пострадавший от dd Шашанк Шарма на каждом шагу создает моментальные снимки своей системы, что и вам настоятельно рекомендует. |
| Отказываться от X 60 | Управлять серверами 64 |
| Узнаем от Матс-Таге Аксельссона , как перейти на <i>Wayland</i> и почему это следует сделать. Расставание с X-сервером неизбежно... | Шашанк Шарма жалеет измотанных сисадминов и намерен облегчить их тяжкий труд, предложив им инструмент web-администрирования. |
| Контролировать версии 68 | Повышать надежность 70 |
| <i>Git</i> — не только для разработчиков: с его помощью можно, например, вместе работать над документами. Для Шашанка Шармы это важно: он юрист. | Опасаясь сбоя у интернет-провайдера, Максим Черепанов добавляет к системе управления Послушным Домом сотовую связь. |
| Связываться с Pi 74 | Тестировать систему 78 |
| Кстати о связи: Ник Пирс обнаружил платформу с открытым кодом для надзора за подключенными устройствами. Интегрируемся с Интернетом вещей. | Павел Емельянов завершает серию статей о <i>CRIU</i> тщательной проверкой: действительно ли состояние процессов сохраняется и восстанавливается? |

АКАДЕМИЯ КОДИНГА



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

- | | |
|---|--|
| Python ушел на базу 80 | Python и тексты 84 |
| Михалис Цукалос деловито общается с базами данных на языке SQL, подключаясь к ним с помощью скриптов Python. | Если всё на свете — файл, то всё в файле — текст. Или нет? Так или иначе, Михалису Цукалосу захотелось автоматизировать обработку текста. |

Terminal: Переменные среды

Ник Пирс покажет, как управлять своим окружением, и откроет, как редактировать его с помощью переменных и зачем вам это надо.



Наш эксперт

Ник Пирс считает, что было бы здорово, если можно было бы назначать (и редактировать) переменные окружения в реальной жизни. БАЛАНС=1 000 000 и всё такое. Но как это сделать, он пока не придумал.

Переменные играют ключевую роль в повседневной компьютерной жизни, даже если вы об этом не подозреваете. Они могут быть индивидуальными, обеспечивающими быстрый доступ к информации или настройкам в отдельных скриптах или программах, или широкого применения, доступными в любом сеансе оболочки Terminal.

Выделяют два типа переменных, используемых в сеансах оболочки: переменные окружения (среды) и переменные оболочки. И в том, и в другом есть такие полезные функции, как языковая локаль (LANG), домашний каталог пользователя (HOME) и пути к командам (PATH). Наиболее очевидным их преимуществом является то, что они избавляют приложения от необходимости определять эти параметры индивидуально. А главное отличие между двумя типами в том, что переменные оболочки действуют только в ее пределах, а переменные окружения могут, кроме того, влиять на любые дочерние процессы, запускаемые из оболочки, например, программы.

В рамках нашего урока мы сосредоточимся только на переменных окружения, но во врезке вверху стр. 53 показано, как установить переменные оболочки, а после этого преобразовать их

в переменные окружения, или наоборот, по мере необходимости. Но не будем забегать вперед. Сначала посмотрим, как устроены переменные.

Анатомия переменной

Каждая переменная сопровождается значением и выражается в виде

```
VARIABLE=value
```

Имена переменных чувствительны к регистру: например, 'VARIABLE', 'Variable' и 'variable' — это три разные переменные. Однако, как правило, принято задавать все переменные заглавными буквами, при необходимости используя нижнее подчеркивание () для обозначения пробела. Значение может выражаться строкой или числом, например:

```
HOME=/home/username
```

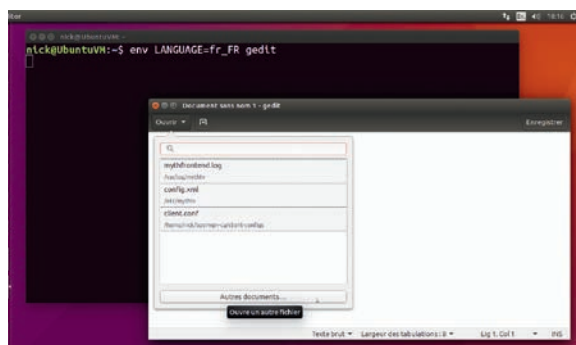
```
LANG=en_GB.UTF-8
```

```
HLVL=1
```

Некоторые переменные могут иметь несколько значений, которые выражаются в виде ПЕРЕМЕННАЯ=значение:значение:значение. В качестве примера можно привести переменную PATH, в значениях которой перечисляются все папки, содержащие команды. Увидеть полный перечень установленных на данный момент переменных окружения можно с помощью команды `$ printenv`.

Вы получите список всех переменных для текущего пользователя, а чтобы быстро просмотреть значение отдельной переменной, наберите `printenv VARIABLE` и нажмите Enter.

Можно применять переменные окружения к командам. Это позволяет быстро заменить многословные строки на более легко запоминающиеся переменные. Такая возможность есть и для переменных среды, и для переменных оболочки; всё, что вам надо сделать, это добавить префикс `$`, чтобы при выполнении команды вместо переменной подставлялось ее значение. Например, вы можете перейти откуда угодно в свой каталог `home` с помощью



» Вы можете назначать переменные окружения, действующие только для одного процесса или программы.

Часто используемые переменные окружения

Переменных окружения существует множество — к сожалению, в Ubuntu нет никакой документации, где они все были бы описаны, но полный перечень можно найти на <http://bit.ly/EnvironmentalVariables>. А пока рассмотрим наиболее используемые:

» **PWD** содержит путь к текущему рабочему каталогу и поэтому меняется, когда вы перемещаетесь по каталогам.

» **OLDPWD** содержит путь к предыдущему посещенному каталогу.

» **TERM** Тип используемого Terminal; адресована программам, эксплуатирующим специфические возможности Terminal.

» **COLUMNS/LINES** По умолчанию устанавливает число столбцов (80) и строк (25) для консоли. Попробуйте изменить их, если строки не переносятся.

» **PWXDG_CURRENT_DESKTOP** Используемое окружение рабочего стола.

» **USER/LOGNAME** В обеих этих переменных записывается имя текущего пользователя. В Ubuntu им

по умолчанию будет ваше имя пользователя. Это значение лучше не трогать.

» **TZ** Определяет часовой пояс. В системах больше не используется за ненадобностью, но можно ее установить, чтобы отменить отображение времени для конкретного сеанса оболочки. Для этого необходимо предварительно создать переменную TZ (например: `TZ=GMT+2`), а затем преобразовать ее в переменную окружения (выполнив команду `export TZ`).

Переменные оболочки

Если при создании новой переменной вы используете выражение присваивания, создается новая переменная оболочки. Этот тип переменной применяется только к текущей оболочке, и, в отличие от переменных окружения, не может влиять на поведение других приложений.

Тем не менее, можно преобразовать существующую переменную оболочки в переменную окружения с помощью команды `export`, которая позволит использовать ее, помимо оболочки, с программами и процессами: `$ export VARIABLE`. Кроме того, команда `export` поможет вам сэкономить время,

создав переменную оболочки и экспортировав ее за раз: `$ export VARIABLE=value`. Если позже вы захотите отменить этот шаг, не удаляя переменную навсегда, используйте флаг `-n`, чтобы снова ограничить ее применение оболочкой — например, `$ export -n VARIABLE`.

`$ cd $HOME` или вернуться к предыдущей директории с помощью `$ cd $OLDPWD`.

Можно изменить переменные окружения навсегда (следуйте указаниям во врезке *Часто используемые переменные окружения* внизу стр. 52), но пока мы займемся временными изменениями, влияющими только на текущий сеанс работы с оболочкой (и любые дочерние процессы, запускаемые из него). Можно убедиться, что изменения имеют преходящий характер, открыв вторую оболочку и воспользовавшись командой `printenv`, которая продемонстрирует, что оболочка сохраняет исходные значения всех переменных окружения.

Изменить значения переменной среды несложно — надо всего лишь переопределить их в командной строке, например, так: `$ VARIABLE=value`. Скажем, чтобы изменить язык любого приложения, которое вы запускаете из текущего сеанса оболочки, можно ввести `LANGUAGE=fr_FR` для изменения параметров языка и локализации на французский.

Помните, что изменение данной переменной окружения действует только для текущего сеанса. Если вы откроете другую оболочку, она будет использовать значения по умолчанию; при закрытии первой оболочки ваши изменения тоже будут потеряны. Вы можете пойти на шаг дальше, привязав пользовательские переменные окружения только к определенным командам. Для этого воспользуйтесь командой `env`, которая позволяет менять окружение программы, в которой вы планируете работать, не затрагивая другие процессы. Ее синтаксис выглядит следующим образом: `env VARIABLE=value имяпрограммы`. Вы можете добавлять аргументы после имени программы обычным способом, и, если захотите, применять несколько переменных окружения. Так, следующая команда запустит текстовый редактор *nano* на французском языке:

```
$ env LANGUAGE = fr_FR nano
```

Внесение постоянных изменений

Если вы хотите изменить параметры переменной окружения навсегда, необходимо отредактировать некоторые файлы настройки — либо внеся правки в существующие переменные, либо добавив новые, которые изменят все настройки, сделанные где-либо еще. Эти настройки хранятся в ряде конфигурационных файлов, разбросанных по вашей файловой системе. Некоторые такие файлы считываются при запуске, а затем применяются при входе в оболочку, но не в рабочий стол Ubuntu; другие считываются, когда вы запускаете оболочку с помощью Terminal. В обоих случаях используются два типа конфигурационных файлов: в файлах одного типа содержатся глобальные настройки, а в файлах другого типа — пользовательские.

При входе в систему прямо в оболочку это файл `/etc/profile` и один из `~/bash_profile`, `~/bash_login` или `~/profile` соответственно. Если — и скорее всего — для доступа к оболочке вы используете Terminal, то глобальный файл — это `/etc/bash.bashrc`, а пользовательские настройки хранятся в `~/bashrc`.

Давайте начнем с рассмотрения последнего: `$ nano ~/.bashrc`.

Вы увидите, что он похож на прочие файлы скриптов, где переменные окружения выражены как `'ПЕРЕМЕННАЯ=значение'`. Здесь вы найдете не все переменные, но если, например, добавить в конце строку `LANGUAGE=fr_FR`, все будущие сеансы Terminal будут на французском языке.

Сохраните файл и выйдите. Проверить это изменение можно, закрыв и снова открыв Terminal или введя следующую команду, которая перезагружает файл `.bashrc` без всяких перезапусков: `$ source ~/.bashrc`. Теперь при запуске любого сеанса Terminal он должен быть по умолчанию на французском языке, в подтверждение тому, что изменение постоянное. Чтобы вернуться к исходному языку, надо удалить эту строку, сохранить файл, а затем выйти из системы и войти обратно, чтобы снова применить исходные настройки языка.

Вы заметите, что некоторые переменные устанавливаются с помощью интерактивных циклов `'if else fi'`, позволяющих настроить переменные, изменяющиеся в зависимости от окружения. Четыре переменных окружения — `PAGER`, `EDITOR`, `VISUAL` и `BROWSER` — позволяют назначить предпочитаемые приложения для отображения длинного текста, редактирования текста в оболочке, редактирования текста в графическом интерфейсе и просмотра web-страниц соответственно. В следующем примере мы использовали цикл `'if else fi'`, чтобы сделать *Firefox* предпочтительным web-браузером, при наличии *X* или другой подобной графической среды; в противном случае будет использоваться текстовый браузер *Links* (если его нет, добудьте его, набрав `sudo apt-get install links`):

```
if [ -n "$DISPLAY" ]; then
export BROWSER=firefox
else
export BROWSER=links
fi
```

И, наконец, желая удалить переменную окружения, вы можете сделать это, просто установив пустое значение (`export VARIABLE=`). Это сработает во многих случаях, но не всех, так что лучше не рискуйте и сотрите всё полностью: `$ unset VARIABLE`. Опять же, это относится только к текущему сеансу; если вам надо удалить или отредактировать переменную окружения навсегда, вы должны отредактировать конфигурационный файл, где она находится. **LXF**

```
nick@ubuntuvm:~$ printenv
HOME=/home/nick
DESKTOP_SESSION_ID=this-is-deprecated
UPSTART_INSTANCE=
UPSTART_EVENTS=started starting
XDG_SESSION_DESKTOP=ubuntu
LOGNAME=nick
QT4_IM_MODULE=xim
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-31660hXB1k
LESSOPEN=| /usr/bin/lesspipe %s
INSTANCE=Unity
UPSTART_JOB=unity-settings-daemon
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
LESSCLOSE=/usr/bin/lesspipe %s %s
XAUTHORITY=/home/nick/.Xauthority
_/usr/bin/printenv
nick@ubuntuvm:~$ printenv HOME
/home/nick
nick@ubuntuvm:~$
```

► Увидеть переменные окружения в деле поможет команда `printenv`.

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

Terminal: Ваш НОВЫЙ ЭМУЛЯТОР

Gnome Terminal — лишь один из многих существующих эмуляторов.

Ник Пирс разбирается, есть ли между ними разница.



Наш эксперт

Ник Пирс занимается компьютерами уже более 30 лет, из них последние 10 балует с Linux.

Получая доступ к оболочке из рабочего стола Ubuntu, Unity, вы пользуетесь эмулятором терминала или «окном терминала». Попросту говоря, это эмуляция терминала командной строки в окне рабочего стола. Знать об этом полезно, но не то чтобы строго обязательно, поэтому стандартное окно терминала в Ubuntu привычно называется *Terminal*.

В нем полно полезных функций, и для многих это единственный известный им инструмент для управления Ubuntu из командной строки в Unity (а точнее, в любом рабочем столе на базе Gnome). Но это не единственный такой эмулятор, и мы подобрали пять вариантов, которые могут вам подойти.

Встроенные терминалы

Терминал по умолчанию в Ubuntu на самом деле называется *Gnome Terminal*. Не верите? Откройте окно терминала и выберите *Terminal > O* программе. У него есть собственный сайт (<https://wiki.gnome.org/Apps/Terminal>), и он, как мы уже убедились по всей этой серии, имеет богатую функциональность.

Он основан на эмуляторе терминала *XTerm*, который является стандартным эмулятором для *X Window System* и тоже присутствует в Ubuntu (наряду с *UXTerm*, также на базе *XTerm*, но с поддержкой Unicode). *XTerm* — более минималистичный вариант оболочки: здесь нет меню, и вся настройка выполняется непосредственно из командной строки.

В число характеристик, по которым *Gnome Terminal* превосходит *Xterm*, входит поддержка нескольких профилей во всех индивидуальных учетных записях пользователей, цветной текст, настраиваемый фон и ограниченная поддержка событий мыши (например, для вставки из буфера обмена щелкните правой кнопкой мыши).

Tilda

Tilda (<https://github.com/lanoox/tilda>) можно установить из *Terminal* (`sudo apt-get install tilda`), и его отличительная особенность — то, что он ведет себя как всплывающее окно, а не обычное окно

приложения. Чтобы переключить его внешний вид, используйте F1. (Примечание: если вы открываете несколько копий, то для второго нужно использовать F2, для третьего — F3, и так далее).

Еще одним преимуществом *Tilda* является настраиваемость. По умолчанию при запуске открывается экран настройки — или можно в любое время щелкнуть правой кнопкой в окне *Terminal* и выбрать Настройки — доступные опции будут разбиты на семь вкладок, в том числе для изменения внешнего вида, поведения *Tilda* (по умолчанию они всегда появляются вверху, но можно это изменить) и привязки клавиш. Кроме того, нам нравятся его всплывающее окно поиска (удерживайте Shift + Ctrl + F), которое находится под главным окном терминала, и использование *Monospace* в качестве шрифта по умолчанию.

Guake

Если вы фанат стрелялок, вы наверняка знаете всё о выпадающем терминале *Quake*, и *Guake* (<http://guake-project.org>) — это дань уважения к нему. Как и *Tilda*, его можно установить прямо из текущего терминала (`sudo apt-get install guake`).

В отличие от *Tilda*, вы можете запускать только одну копию *Guake* за один раз. Тем не менее, вы можете выбрать, на каком дисплее он будет открываться при вызове, с помощью клавиши F12. По умолчанию он будет следовать за мышью, но можно указать конкретный монитор, зайдя в Настройки > Общие и выбрав Появляться на дисплее.

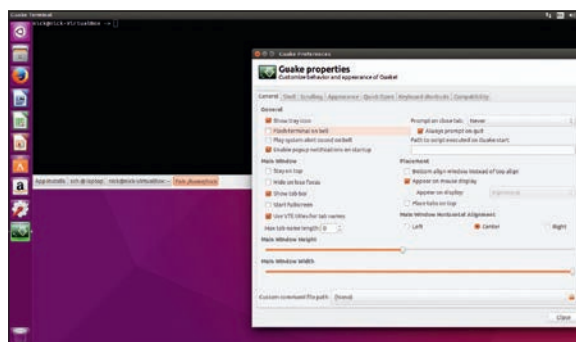
Guake поддерживает несколько вкладок, как и все хорошие эмуляторы (включая *Gnome Terminal*), но здесь есть приятный нюанс: вы можете настроить имена, чтобы было легче в них ориентироваться. Вкладки по умолчанию отображаются под главным окном терминала (переместить их наверх можно снова через Настройки > Общие).

Guake также позволяет выбрать, какую оболочку следует запустить — для этого зайдите в Настройки > Оболочка. Вы также можете открывать новые вкладки из каталога, заданного в данный момент. В числе прочих отличительных особенностей — 130 готовых цветовых схем, чтобы быстро выбрать оформление на свой вкус, а также возможность щелчком открывать файлы в текстовом редакторе по умолчанию.

Единственный минус в том, что *Guake* пока не поддерживает сохранение сеансов — существенный недостаток при таких возможностях настройки; но это можно обойти, задав скрипт, который будет по умолчанию выполняться при запуске и применять ваши настройки.

Terminator

Terminator (<https://gnometerminator.blogspot.co.uk>) — это производная от *Gnome Terminal* программа, призванная упростить жизнь тем, кому нужно управлять несколькими сеансами терминала.



» *Guake* — один из эмуляторов терминала выпадающего типа, а его параметры позволяют легко осуществить тонкую настройку.

Альтернатива Bash

На этом уроке мы сосредоточились на альтернативных эмуляторах терминала, и все они работают со стандартной оболочкой *Bash*, тем самым внутренним командным интерфейсом, которым вы пользуетесь за пределами рабочего стола (опять не верите? Наберите в терминале `echo $$SHELL`, и он укажет на `/bin/bash`).

Подобно тому, как в Ubuntu есть альтернативы эмулятору терминала по умолчанию, есть они и для оболочки *Bash*. Одна из наших любимых — это *fish*,

которая может сосуществовать с *Bash*, не заменяя его полностью. Устанавливается она следующим образом:

```
$ sudo apt-add-repository ppa:fish-shell/release-2
$ sudo apt-get update
$ sudo apt-get install fish
```

После установки запустите *fish* через эмулятор терминала, набрав `$ fish`.

Вы увидите, что командная подсказка изменилась на `->`; введите что-нибудь с целью убедиться,

что и подсветка синтаксиса, и автозаполнение работают. Не видите требуемую команду? Нажмите на правую стрелку, чтобы ее выделить, или нажмите [Tab] — тогда вы увидите перечень (историю) других команд, которые вы уже успели наводить до этого.

Посетите <https://fishshell.com>, там вы узнаете, как пользоваться *fish* — в том числе, как настроить *fish* из вашего браузера (подсказка: следует набрать в оболочке `fish_config`).

Во-первых, вы можете организовать ваши терминалы в виде сетки в пределах одного окна. Для этого щелкните правой кнопкой мыши в окне *Terminator* и выберите разделение по горизонтали или по вертикали. Затем вы можете продлевать то же самое для отдельных частей окна, пока не получите подобную сетке структуру. Размер частей тоже можно менять, нажав и перетаскивая их границы для создания более сложных структур; изменить расположение можно, нажав и перетаскивая заголовок.

Вы можете организовать несколько окон в группы, вводить один и тот же код в группы (выберите Broadcast group [Оповестить группу]) или все терминалы (Broadcast all [Оповестить все]) и намудрить еще больше, заведя несколько вкладок. Раскладку окон можно сохранять (правый щелчок в окне, и затем Настройки > Раскладка), и предусмотрена поддержка десятков настраиваемых сочетаний клавиш, профилей и даже плагинов, еще больше расширяющих функциональные возможности *Terminator*. Некоторые из них установлены изначально, например, меню команд пользователя; другие можно загрузить и добавить в *Terminator*, поместив их в папку `~/.config/terminator/plugins`. Одним из таких является плагин, осуществляющий поиск выделенного в терминале текста в Google — чтобы его получить, зайдите на <https://github.com/choffee/terminator-plugins>.

Хотя *Terminator* доступен из основных репозиториях Ubuntu, для получения последней версии выполните

```
$ sudo add-apt-repository ppa:gnome-terminator
$ sudo apt-get update
$ sudo apt-get install terminator
```

Учтите, что здесь, как и в *Terminal*, действует сочетание клавиш Ctrl+Alt+T.

Terminology

Этот терминал предназначен для тех, кому в жизни не хватает красок, и обещает быть одним из самых быстрых среди доступных эмуляторов в плане обработки ввода и вывода. Это потому, что данные в *Terminology* одновременно сжимаются и сохраняются в оперативной памяти, а не в файл. Это также наиболее приятный на вид эмулятор, и вы признаете это, как только щелкнете правой кнопкой мыши внутри окна терминала и увидите всплывающее меню.

Набор функций аналогичен *Terminator*, особенно по части разделяемых окон изменяемой величины (со вкладками внутри панелей для вящей путаницы). Легко также переименовывать отдельные окна (выберите окно, щелкните правой кнопкой мыши, выберите Задать заголовок). С точки зрения визуальной привлекательности *Terminology* элегантнее, чем *Terminator*, но не столь настраиваемый. Тем не менее, у него есть одна убойная функция: поддержка видео, изображений и миниатюр внутри терминала.



► Удобный интерфейс пользователя *Terminology* скрывает интересные функции, например, просмотр эскизов изображений прямо из командной строки.

Перейдите к папке с фотографиями, введите `tyls -m`, и вы поймете, что мы имеем в виду — щелкните на эскизе, и сможете просмотреть изображение или воспроизвести видео в окне *Enlightenment*. *Terminology* является частью оконного менеджера *Enlightenment*, а значит, перед его установкой вам надо добавить репозиторий *Enlightenment* (и придется загрузить аж целых 60 МБ).

```
$ sudo add-apt-repository ppa:enlightenment-git/ppa
$ sudo apt-get update
$ sudo apt-get install terminology
```

LXTerminal

У нашего последнего эмулятора терминала вся суть — в его минимализме. *LXTerminal* (<https://wiki.lxde.org/en/LXTerminal>) является частью легковесного рабочего стола *LXDE*; также этот эмулятор используется по умолчанию в Raspbian. Установите его через репозитории Ubuntu (`sudo apt-get install lxterminal`), и поразитесь, насколько он прост. Но в нем есть, по крайней мере, поддержка вкладок (которые можно именовать), и можно настроить прозрачность фона, если засилие черного вас слишком отвлекает.

Среди других эмуляторов терминала, снижавших популярность у пользователей, *Konsole* (в этом терминале по умолчанию в KDE можно создавать закладки каталогов), *Yakuake* (*Konsole* в стиле выпадающего *Guake*), *ROXTerm*, *xfce4-terminal* (облегченная альтернатива *Tilda*) и *PAC* (этот понравится в основном тем, кому приходится часто инициализировать удаленные соединения). Наберите в Google имя каждого из эмуляторов, чтобы разузнать о них побольше. **LXF**

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



Nmap: Сканы безопасности

Таджиндер Калси считает, что один скан в день спасет вас от хакеров; мы обрисуем, как обеспечить большую безопасность вашего Linux-ПК и сети.

Наш эксперт

Таджиндер Калси имеет более чем 9-летний опыт работы в области информационной безопасности, и в свое свободное время проводил семинары в более 125 колледжах Индии. В данный момент он работает независимым консультантом по информационной безопасности.

Безопасность — одна из самых больших проблем программного обеспечения, и одновременно самая скрытная. Мы могли бы месяцами обсуждать, почему это так, но давайте перейдем к практике. На этом уроке, почерпнутом из *Практической Поваренной Книги по Безопасности Linux* Таджиндера Калси [Tajinder Kalsi *Practical Linux Security Cookbook*], мы рассмотрим, как использовать *Nmap* для сканирования хостов и получить доступ администратора на уязвимой системе Linux.

Nmap — это действительно полезный инструмент для сканирования сетей. Он существует уже довольно давно, и, вероятно, стал одним из самых популярных инструментов для этой задачи. Если вы оцениваете уязвимость, вам определенно стоит задуматься об использовании *Nmap*. На данном уроке, взятом из *Практической Поваренной Книги по Безопасности Linux*, вы узнаете, как сканировать систему Linux, а потом обнаружить уязвимый сервис. После этого вы сможете получить доступ root к системе.

В большинстве версий Linux *Nmap* уже установлен. Первый шаг — это проверить, есть ли он у вас, с помощью команды

```
nmap -version
```

Если *Nmap* существует, вы должны увидеть вывод, подобный этому:

```
nmap -version
Nmap version 6.00 ( http://nmap.org )
```

Если *Nmap* не установлен, он должен быть доступен в репозиториях вашего дистрибутива, и его можно установить по `sudo apt-get install nmap` или скачать напрямую с <https://nmap.org/download.html>.

Неспокойные времена

Nmap проверяет сервисы, которые слушают посредством тестирования самых распространенных сетевых портов. Эта информация помогает администратору сети закрыть нежелательные или неиспользуемые сервисы. *Nmap* можно использовать для сканирования портов, и в нем есть мощные инструменты для изучения сети вокруг нас. В *Nmap* есть также несколько полезных функций скриптов, которые можно использовать для написания собственных скриптов и употреблять эти скрипты с *Nmap* для автоматизации и расширения его возможностей сканирования.

Сканирование хостов

```
$ nmap -vv -sP 103.46.192.2-100
Starting Nmap 6.00 ( http://nmap.org ) at 2015-07-09 10:14 IST
Installing Ping Scan at 21:24
Scanning 100 hosts (2 ports/host)
Completed Ping Scan at 21:24, 2.33s elapsed (100 total hosts)
Initiating Parallel DNS resolution of 100 hosts. at 21:24
Completed Parallel DNS resolution of 100 hosts. at 21:24, 4.28s elapsed
Nmap scan report for 103.46.192.2 (host down)
Nmap scan report for 103.46.192.3 (host down)
Nmap scan report for 103.46.192.4
Host is up (0.025s latency).
Nmap scan report for 103.46.192.7 (host down)
Nmap scan report for 103.46.192.11
Host is up (0.079s latency).
Nmap scan report for 103.46.192.19
Host is up (0.036s latency).
Nmap scan report for 103.46.192.20 (host down)
.....
Read data files from: /usr/bin/../share/nmap
Nmap done: 100 IP addresses (26 hosts up) scanned in 6.67 seconds
```

```
$ nmap -v -n -sP --max-rtt-timeout 500ms 103.46.192.2-100 -T4
Starting Nmap 6.00 ( http://nmap.org ) at 2015-07-09 21:34 IST
Initiating Ping Scan at 21:34
Scanning 100 hosts (2 ports/host)
Completed Ping Scan at 21:34, 1.97s elapsed (100 total hosts)
Nmap scan report for 103.46.192.2 (host down)
Nmap scan report for 103.46.192.2
Host is up (0.021s latency).
Nmap scan report for 103.46.192.2 (host down)
Nmap scan report for 103.46.192.3 (host down)
Nmap scan report for 103.46.192.4
Host is up (0.056s latency).
Nmap scan report for 103.46.192.5
Host is up (0.026s latency).
.....
Read data files from: /usr/bin/../share/nmap
Nmap done: 100 IP addresses (26 hosts up) scanned in 1.97 seconds
```

```
Host: 192.168.1.247 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.248 () Status: Up
Host: 192.168.1.248 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.249 () Status: Up
Host: 192.168.1.249 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.250 () Status: Up
Host: 192.168.1.250 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.251 () Status: Up
Host: 192.168.1.251 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.252 () Status: Up
Host: 192.168.1.252 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.253 () Status: Up
Host: 192.168.1.253 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.254 () Status: Up
Host: 192.168.1.254 () Ports: 21/filtered/tcp//ftp//
Host: 192.168.1.255 () Status: Up
Host: 192.168.1.255 () Ports: 21/filtered/tcp//ftp//
# Nmap done at Tue Nov 8 10:27:28 2016 -- 256 IP addresses
scanned in 26.16 seconds
```

1 Базовое сканирование

Самое распространенное использование *Nmap* — нахождение всех онлайн-хостов в заданном диапазоне IP-адресов. Используемая для этого команда по умолчанию требует времени на сканирование всей сети, которое зависит от количества хостов в сети.

Однако мы можем оптимизировать процесс, чтобы сканировать быстрее. Время на завершение сканирования здесь было 6,67 секунд при сканировании 100 хостов. Если надо просканировать целый диапазон IP-адресов для определенной сети, это займет намного больше времени.

```
$ nmap -vv -sP 103.46.192.2-100
```

2 Ускоренное сканирование

Теперь давайте попробуем ускорить процесс. Параметр `n` велит *Nmap* не выполнять DNS-разрешение IP-адресов, отсюда — увеличение скорости процесса. Параметр `T` задает *Nmap* скорость работы. Здесь `T1` — самая медленная, а `T5` — самая быстрая. Опция `--max-rtttimeout` указывает максимальное время ожидания ответа.

Получив приведенную ниже команду, *Nmap* просканировал весь диапазон IP-адресов за 1,97 секунд. Неплохо, правда?

```
$ nmap -v -n -sP --max-rtt-timeout 500ms 103.46.192.2-100 -T4
```

3 Открытые порты

Сканирование портов с помощью *Nmap* помогает обнаружить онлайн-сервисы — например, найти FTP-серверы. Для этого используйте команду

```
$ sudo nmap -sS -vv -n -Pn -p21 --max-rtt-timeout 500ms 192.168.1.1/24 -T4 -oG -l grep 'open'
```

Отобразится список всех IP-адресов с открытым портом 21. Не только FTP, но и другие сервисы могут быть обнаружены по соответствию номеру порта, на котором они работают. Например, *MySQL* работает на порте 3306. Теперь команда будет выглядеть так:

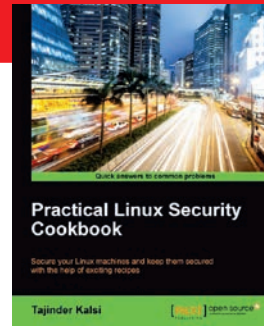
```
$ sudo nmap -sS -vv -n -Pn -p3306 --max-rtttimeout 500ms 192.168.1.1/24 -T4 -oG -l grep 'open'
```

Рецепты для админа

При растущей популярности Linux администраторы постоянно переходят на эту систему для создания сетей и серверов для самых разных задач. Однако это также означает, что Linux становится первостепенной целью для многих злоумышленников.

Практическая поваренная книга по безопасности Linux полна пошаговых рецептов, экранных снимков и описаний, чтобы расска-

зать вам обо всем необходимом для создания более безопасной системы Linux, начиная со вводной информации о разнообразных угрозках и пройдя далее по всему пути, вплоть до управления аутентификацией пользователя, как локально, так и удаленно, ослабления сетевых атак и изучения патчей уязвимости Bash и мониторинга системных журналов для повышения безопасности.



» Всё, что вам следует знать о безопасности ваших систем Linux.

Когда вы пытаетесь научиться сканировать и использовать компьютер с Linux, самая большая проблема заключается в том, чтобы выяснить, где именно вы можете его испытать. К счастью, команда Metasploit разработала полезную виртуальную машину под названием Metasploitable. Metasploitable была намеренно сделана уязвимой, с запуском многих незалатанных сервисов. И это отличная платформа для оттачивания ваших навыков тестирования на предмет несанкционированного доступа.

Вначале скачайте Metasploitable 2 (<https://sourceforge.net/projects/metasploitable/files/Metasploitable2>) с SourceForge. На нашем руководстве мы также будем использовать Backtrack 5 R2. Теперь обрисуете шаги по получению доступа root к уязвимой системе Linux.

Во-первых, откройте консоль Metasploit в системе Backtrack следующим образом: Main Menu > Backtrack > Exploitation Tools > Network Exploitation Tools > Metasploit Framework > Msfconsole. С помощью Nmap вам следует просканировать цель (в данном примере — 192.168.0.1),

```
$ nmap -sS -Pn -A 192.168.0.1
```

и вы должны увидеть нечто вроде этого:

```
hexec: nmap -sS -Pn -A 192.168.0.1
```

```
Starting Nmap 5.51SVN...
```

```
Nmap scan report for 192.168.0.1...
```

```
Not shown: 988 closed ports
```

```
PORT STATE SERVICE VERSION
```

```
21/tcp open  ftp ProFTPD 1.3.1
```

```
...
```

```
139/tcp open netbios-ssn Samba 3.x
```

```
445/tcp open netbios-ssn Samba 3.x
```

```
...
```

В предыдущей команде опция `-sS` позволяет выполнить скрытое сканирование, а опция `-A` пытается обнаружить информацию о версии операционной системы и сервиса. Также можно увидеть массу сервисов, использующих разные порты. Среди них — *Samba*, работающий на портах 139 и 445. Если вы не знаете, что такое *Samba* [Ред.: — Это бразильский танец родом из Африки, который наш арт-директор любит исполнять на корпоративе?], то это сервис, обеспечивающий SMB-файл и печать для систем Windows.

Знойная Samba

Определив местонахождение сервиса *Samba*, сфокусируемся на нем. Предыдущий вывод показал, что используется *Samba* версии 3.x. Теперь мы попытаемся получить более подробную информацию о сервисе. Для этого мы воспользуемся любым из вспомогательных модулей Metasploit, например, разделом сканера, и поищем протокол SMB.

Сканирование версии *Samba* позволит оценить подходящие векторы атаки. Мы видим, что в разделе сканера есть детектор версии

SMB. Теперь мы получим точную версию *Samba* с помощью программы-детектора. Поиск онлайн все уязвимости определенной версии *Samba*, мы найдем скрипт карты имени пользователя [map script username]. Теперь мы можем найти список эксплоитов, имеющихся в Metasploit, чтобы проверить, существует ли эксплоит для map script username, используя команду `search samba`:

```
Name Date Rank Description
```

```
...
```

```
exploit/linux/samba/nttrans 2003-04-07 average Samba 2.2.2 - 2.2.6 nttrans buffer overflow
```

```
exploit/linux/samba/usermap_script 2007-05-14 excellent Samba "username map script" Command execution
```

```
exploit/linux/samba/lsa_transnames heap 2007-05-14 average Samba lsa_io_trans_names Heap
```

```
Overflow
```

Все мы любим пятизвездочные эксплоиты, не так ли? Оценивая эксплоиты, как будто они участники программы X-Factor, а мы — Саймон Коуэлл [Simon Cowell] с до смешного высоко подтянутым брючным ремнем.

Мы обнаружили эксплоит для map script username, и у него превосходный рейтинг, а значит, можно его использовать. Используйте map script username, чтобы получить права root в оболочке системы. Выбрав эксплоит и определив цель, легко получить доступ root и управление.

Добудем доступ root с помощью предыдущего эксплоита. Выбрав эксплоит и настроив его на целевой IP-адрес (в данном случае — 192.168.0.1), мы выполним команду для запуска эксплоита. Это создаст и даст нам удаленную сессию на целевой системе и также откроет командную оболочку. Теперь введите команду `id` в удаленной оболочке. Вы получите результат

```
uid=0(root)gid=0(root)
```

Это подтверждает, что мы получили удаленный доступ root к целевой системе.

Порты прямо по борту

Вначале мы выполнили сканирование с *Nmap* на предмет поиска работающих сервисов и открытых портов, и обнаружили работающий сервис *Samba*. Затем мы попытались выяснить версию сервиса SMB. Сделав это, мы отыскали доступный для *Samba* эксплоит. С помощью эксплоита мы попытались атаковать целевую систему и получить доступ root к ней. Пакеты, отправляемые атакующим, обычно поддельные, что усложняет процесс расследования вторжений.

Будем надеяться, данные рецепты безопасности Linux были вам полезны. Для более подробной информации — и даже большего количества рецептов — убедитесь, что вы нашли полную Практическую поваренную книгу по безопасности Linux на www.packtpub.com. **123**

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



Snapper: ДИСКИ, ВАС СНИМАЮТ

Шашанк Шарма когда-то применил `dd` на разделе вместо флешки и с тех пор помешан на моментальных снимках.



Наш эксперт

Шашанк Шарма подвергнул свой характер испытанию с помощью `dd`, и в поисках новых приключений теперь проводит свои дни, слоняясь по залам судов Дели в качестве юриста.

Ничто не вызывает такого шока, боли и мучений, как поврежденная система, не дающая доступа к вашим драгоценным данным. Неважно, в чем причина — плохо настроенная видеокарта не позволяет запустить GUI или установка коварной программы, которая препятствует загрузке вашего дистрибутива. К счастью, во избежание подобного сценария дистрибутивы Linux предлагают ряд решений по резервному копированию данных и даже инструменты, разработанные для копирования дисков целиком. Познакомьтесь же со *Snapper*, позиционирующим себя как исключительный инструмент моментальных снимков системы для Linux, который использует мощь файловой системы Btrfs для создания моментальных снимков вашей системы.

Snapper — инструмент командной строки, позволяющий создавать моментальные снимки вашего компьютера перед внесением важных изменений. Например, вы можете создать моментальный снимок перед установкой нового ядра, заплатки или даже сервера LAMP. Если после установки что-то пойдет не так, то с помощью моментального снимка вы вернете свою систему к рабочему состоянию.

Разработанная одним из разработчиков SUSE, Арвином Шнеллем [Arvin Schnell], она, что неудивительно, отлично встает на установку openSUSE. Однако проект предлагает прекомпилированные бинарники для различных систем, таких как Debian, Ubuntu и т.д., и доступные в репозиториях многих других, например, Fedora и Arch. Но если вы никогда раньше не работали с Btrfs и не знакомы с ее уникальными функциями, лучше создать тестовую установку openSUSE, вместо того, чтобы пытаться управиться со *Snapper* на вашем дистрибутиве.

Другим преимуществом работы со *Snapper* в openSUSE является его интеграция с надежным инструментом *YaST*, позволяющим управлять моментальными снимками через графический интерфейс. В отличие от других утилит резервного копирования вы не можете просто решить, моментальные снимки каких файлов или директорий сделать с помощью *Snapper*. Помимо необходимости наличия системы Btrfs, еще одной странностью *Snapper* является то, что он работает только с подтомами, функцией файловой системы Btrfs.

Настройка Snapper

Для каждого подтома *Snapper* требует отдельного файла настройки. Вы можете запустить следующую команду для отображения списка доступных настроек:

```
sudo snapper list-configs
Config | Subvolume
-----+-----
root | /
```

```
home | /home
log | /var/log
```

Она показывает, что *Snapper* настроен для создания моментальных снимков перечисленных подтомов. Вы можете создать дальнейшие настройки с помощью команды `snapper -c configname create-config subvolume`. Так, если вы хотите создать настройку для директории `/usr/local`, запустите команду

```
sudo snapper -c usrlocal create-config /usr/local
```

Поскольку вы можете создавать настройки только для уже существующих подтомов, запустите следующую команду для отображения всех подтомов:

```
sudo btrfs subvolume list -pa /
root's password:
ID 257 gen 139 parent 5 top level 5 path <FS_TREE>/@
ID 258 gen 181 parent 257 top level 257 path <FS_TREE>/@/.snapshots
ID 259 gen 215 parent 258 top level 258 path <FS_TREE>/@/.snapshots/1/snapshot
ID 262 gen 215 parent 257 top level 257 path <FS_TREE>/@/home
ID 263 gen 171 parent 257 top level 257 path <FS_TREE>/@/opt
ID 265 gen 215 parent 257 top level 257 path <FS_TREE>/@/tmp
ID 266 gen 171 parent 257 top level 257 path <FS_TREE>/@/usr/local
ID 274 gen 215 parent 257 top level 257 path <FS_TREE>/@/var/log
ID 277 gen 215 parent 257 top level 257 path <FS_TREE>/@/var/tmp
```

Каждой конфигурации следует дать уникальное имя, чтобы вы могли отрегулировать некоторые настройки, такие как частота моментальных снимков и т.д., индивидуально для каждого. Например, снимки подтома `/` или даже `/home` не обязательно делать ежедневно. Вы можете получить доступ к этим настройкам, перейдя в директорию `/etc/snapper/configs`.

map-страница `snapper-configs` определяет различные параметры, включая временную шкалу, которую надо тщательно рассмотреть при отладке файлов настроек по умолчанию. Например, можно использовать файлы настройки для определения, сколько моментальных снимков в час/день/год вы собираетесь сохранить при запуске алгоритма автоматической очистки по заданию `cron`. Или же можно настроить *Snapper*, чтобы он удалял самые старые моментальные снимки, как только будет накоплено определенное их количество. Важно настроить тот или иной метод удаления старых моментальных снимков, иначе у вас скоро закончится место на диске.

Кстати о месте: обычные утилиты, такие как команда `df -h`, не особенно полезны при расчете потребления дискового пространства. Вы должны, следовательно, запустить следующую



Скорая помощь
Пользователи SUSE могут использовать отличный инструмент *YaST* для графического администрирования моментальных снимков с помощью *Snapper*, а для пользователей других дистрибутивов похожий графический интерфейс предлагает утилита *Snapper-GUI*, размещенная на GitHub (<http://bit.ly/2eaFdtK>).

Концепция подтомов

Каждый подтом — это часть файловой системы со своей собственной иерархией директорий, подражающей любой файловой системе Linux. Это означает, что, как и разделы, подтома монтируются, и они также похожи на директории в том, что каждый подтом может далее содержать файлы, директории и другие подтома. Вдобавок нет ограничения по количеству вложенных подтомов, которые можно создать.

Получить доступ к содержимому подтома можно таким же образом, как и к директории. Но вы можете, кроме того, смонтировать подтом в качестве

отдельной файловой системы, воспользовавшись командой `mount` и обозначив точку монтирования. В установке SUSE только что созданная файловая система `/` также является подтомом.

`Snapper` создает моментальные снимки только указанных подтомов, а не любой директории по вашему выбору. Нельзя, например, использовать `Snapper` для создания моментальных снимков вашей директории `/etc` на вашем текущем дистрибутиве. Вот почему важно отчетливо осознавать, что такое подтома и какие с ними допустимы операции. За более подробной информацией

по администрированию подтомов обращайтесь на <http://bit.ly/2fjvPUa>.

Если вы уже работаете с дистрибутивом Linux и хотите попробовать `Snapper`, мы предлагаем создать новый раздел с файловой системой `Btrfs`. Затем вы можете создать подтома внутри этого раздела и начать работу со `Snapper` и моментальными снимками. В `openSUSE` сама программа установки создает несколько основных подтомов, таких как `/var/log`, `/tmp`, `/usr/local` и т.д., оставляя на ваше усмотрение создание моментальных снимков этих важных директорий.

команду для определения использованного и доступного места на диске:

```
sudo btrfs filesystem show
Label: none uuid: db407f9b-c248-40cf-b14b-79d0fc8b2fe6
Total devices 1 FS bytes used 14.31GiB
devid 1 size 27.99GiB used 16.07GiB path /dev/sda2
btrfs-progs v4.1.2+20151002
```

Поснимаем моментально

Хотя установка `openSUSE` настроена создавать моментальные снимки / после каждой операции `YaST`, снимки можно создавать и вручную. Это следует делать до и после. Предварительный моментальный снимок создается перед тем, как вы приметесь за любую важную, но потенциально опасную задачу, такую как установка сервера LAMP или компиляция патча ядра. Пост-моментальный снимок делается по завершении этой задачи.

```
sudo snapper create --type pre --print-number --description
"Перед установкой LAMP"
root's password:
13
```

Данная команда выведет на экран цифру, которая станет номером вашего моментального снимка. Вы должны ссылаться на этот номер при создании пост-моментального снимка с помощью:

```
sudo snapper create --type post --pre-number X --description
"После установки LAMP"
```

Убедитесь, что вы заменили `X` в команде выше на номер предварительного моментального снимка. Теперь при запуске команды `sudo snapper list` вы заметите, что моментальный пост-снимок указывает ID соответствующего предварительного моментального снимка.

Для созданных вручную моментальных снимков следует указать алгоритм очистки — номер, временную шкалу и т.д., в противном случае они не будут автоматически удаляться. Для этого вместе с командой `snapper create` можно использовать опцию команды `--cleanup-algorithm <number|timeline|empty-pre-post>`:

```
sudo snapper create --type pre --print-number --description
"Перед установкой VirtualBox" --cleanup-algorithm Number
```

Теперь вы легко можете увидеть, какие изменения произошли в промежутке между двумя моментальными снимками. Запустите команду `sudo snapper status 13..14`, и этот инструмент выдаст длинный список всех файлов, которые за истекшее время были созданы, удалены или изменены. Напротив каждой строки в этом результате вы увидите значок '+', '-' или же букву 'c'. Это, соответственно, обозначает, что следующий файл был добавлен, удален или изменен. Для каждого символа с соответствующим 'c' перед ним вы можете даже запустить команду `diff`, чтобы просмотреть изменения этого файла: `sudo snapper diff 13..14 /etc/sysconfig/apache2`. При запуске данной команды без указания имени файла

`Snapper` создаст `diff` каждого изменения, сделанного между двумя моментальными снимками.

Когда в вашей операции что-то случилось неправильно и вы хотите отменить изменения, вы также можете сделать это с помощью команды `snapper undochange`. Например, желая восстановить файл `/etc/hosts` до его предыдущего состояния, вы легко это сделаете командой

```
sudo snapper -v undochange 7..8 /etc/hosts
```

Если запустить команду `undochange`, не указав имя файла, `Snapper` отменит все изменения и восстановит систему до состояния предыдущего моментального снимка.

Поскольку каждый моментальный снимок имеет свой уникальный номер ID, вы можете использовать `sudo snapper delete ID` для удаления конкретного моментального снимка.

`Snapper` — невероятно надежный инструмент командной строки, позволяющий создавать моментальные снимки, сравнивать их и даже отменять изменения, и всё это с помощью простых для понимания интуитивных команд. Простота в использовании `Snapper` делает его идеальным для людей, работающих с суперсовременными технологиями, и даже для тех, кто использует дистрибутивы с возобновляемыми релизами. **LXF**

```
linuxlala@linuxlala:~$ sudo snapper list -pa /
root's password:
ID 257 gen 139 parent 5 top level 5 path <FS_TREE>/#
ID 258 gen 274 parent 257 top level 257 path <FS_TREE>/#/snapshots
ID 259 gen 307 parent 258 top level 258 path <FS_TREE>/#/snapshots/1/snapshot
ID 260 gen 139 parent 257 top level 257 path <FS_TREE>/#/boot/grub2/1386-pc
ID 261 gen 139 parent 257 top level 257 path <FS_TREE>/#/boot/grub2/x86_64-efi
ID 262 gen 307 parent 257 top level 257 path <FS_TREE>/#/home
ID 263 gen 171 parent 257 top level 257 path <FS_TREE>/#/opt
ID 264 gen 139 parent 257 top level 257 path <FS_TREE>/#/srv
ID 265 gen 306 parent 257 top level 257 path <FS_TREE>/#/tmp
ID 266 gen 225 parent 257 top level 257 path <FS_TREE>/#/usr/local
ID 267 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/crash
ID 268 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/lib/libvirt/images
ID 269 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/lib/mailman
ID 270 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/lib/mariadb
ID 271 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/lib/mysql
ID 272 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/lib/named
ID 273 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/lib/pgsql
ID 274 gen 307 parent 257 top level 257 path <FS_TREE>/#/var/log
ID 275 gen 139 parent 257 top level 257 path <FS_TREE>/#/var/opt
ID 276 gen 306 parent 257 top level 257 path <FS_TREE>/#/var/spool
ID 277 gen 307 parent 257 top level 257 path <FS_TREE>/#/var/tmp
ID 278 gen 139 parent 258 top level 258 path <FS_TREE>/#/snapshots/2/snapshot
ID 285 gen 139 parent 258 top level 258 path <FS_TREE>/#/snapshots/5/snapshot
ID 286 gen 139 parent 258 top level 258 path <FS_TREE>/#/snapshots/6/snapshot
ID 287 gen 139 parent 258 top level 258 path <FS_TREE>/#/snapshots/7/snapshot
ID 288 gen 139 parent 258 top level 258 path <FS_TREE>/#/snapshots/8/snapshot
ID 292 gen 150 parent 258 top level 258 path <FS_TREE>/#/snapshots/9/snapshot
ID 293 gen 151 parent 258 top level 258 path <FS_TREE>/#/snapshots/10/snapshot
ID 294 gen 157 parent 258 top level 258 path <FS_TREE>/#/snapshots/11/snapshot
ID 295 gen 159 parent 258 top level 258 path <FS_TREE>/#/snapshots/12/snapshot
ID 296 gen 237 parent 262 top level 262 path <FS_TREE>/#/home/.snapshots
ID 297 gen 237 parent 274 top level 274 path <FS_TREE>/#/var/log/.snapshots
ID 299 gen 179 parent 296 top level 296 path <FS_TREE>/#/home/.snapshots/1/snapshot
ID 300 gen 180 parent 297 top level 297 path <FS_TREE>/#/var/log/.snapshots/1/snapshot
ID 301 gen 234 parent 296 top level 296 path <FS_TREE>/#/home/.snapshots/2/snapshot
ID 302 gen 235 parent 297 top level 297 path <FS_TREE>/#/var/log/.snapshots/2/snapshot
ID 303 gen 267 parent 258 top level 258 path <FS_TREE>/#/snapshots/13/snapshot
ID 304 gen 272 parent 258 top level 258 path <FS_TREE>/#/snapshots/14/snapshot
linuxlala@linuxlala:~$
```

► Подтома по умолчанию в `openSUSE` дают отличное представление о директориях, моментальные снимки которых вы, возможно, захотите сделать.

Wayland: Переходим с X

Вы уже наслушались о недостатках протокола X, а теперь **Матс-Таге Аксельссон** покажет вам, как и почему надо перейти на *Wayland*.



Наш эксперт

Матс-Таге Аксельссон проработал не один десяток лет, пытаясь заставить свои компьютеры работать на Linux. Его первой системой с Linux был ноутбук от IBM, еще в те времена, когда компания их выпускала.

Скорая помощь

Не подвергайте свою систему риску: *Wayland* можно и нужно настраивать от имени собственного пользователя. И у вас всегда должен быть *X11* на случай отката, поскольку *Wayland* все еще на стадии активной разработки. Установите его на всю систему только после того, как некоторое время с ним поработаете.

Запуская наши компьютеры, мы обычно оставляем в стороне сервер отображения и нацеливаемся на работу с GUI, настраивая внешний вид этого интерфейса и перетаскивая меню. Но с появлением других серверов отображения, типа *Wayland* и *Mir*, ребром встает вопрос: не пора ли отказаться от *X Windows System* (или коротко *X11*) и перейти на нечто более современное?

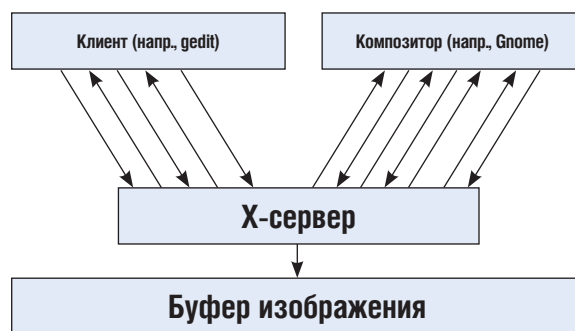
X11 появился аж в 1987 г. Этот протокол был реализован для множества систем — даже в Linux у нас есть несколько разных; но большинство использует версию *X.org*. В самом начале многие его функции предназначались для обеспечения прозрачности сети, но адаптация X к более быстрым видеокартам и 3D оказалась очень ограниченной. В конце концов, это только 2D-протокол. Любая поддержка 3D осуществляется через экзотические расширения. Ни у одного приложения, которые X рассматривает в качестве клиентов, нет доступа к графической памяти, отчего и возникают проблемы, как мы объясним позднее. Кроме того, есть некоторые ограничения, способные замедлить работу вашего компьютера или привести к возникновению потенциальных проблем с безопасностью.

Чтобы понять происходящее, сначала определим части системы X. Во-первых, клиент. Как уже упоминалось, его надо воспринимать как приложение. Затем — композитор, это обычно ваш менеджер окон; и X-сервер, чья задача — выводить графику на экран.

В X клиент подвергает рендерингу то, что выходит на экран, затем говорит X-серверу, что ему требуется для рисования чего-нибудь на экране (на рис. сверху справа). X-сервер выясняет у композитора, обычно это ваш менеджер окон, что надо сделать, если таковая потребность имеется. Менеджер окон решает, какие украшения и свойства необходимы и куда их разместить, и X-сервер копирует всю информацию из приложения в буфер изображения. Этот буфер используется графическими драйверами для представления конечного результата на экране.

На вид всё просто, однако проблема в том, что операция копирования отнимает много времени. Еще одна проблема — коммуникация с композитором обычно состоит из множества шагов. Например, при запуске *gedit*, текстового редактора, делают сотни запросов между *gedit*, клиентом, X-сервером и композитором. Это приводит к тому, что ваши приложения грузятся очень медленно, а также способны вызвать разрывы отображения на экране. Разрывы происходят, когда X-сервер пытается осуществить рендеринг, но не получает правильного сообщения-подтверждения. В результате некоторые части экрана остаются непрорисованными. Это исправляется без особого труда, но по большей части уже после того, как вы замечаете разрыв. Например, видео на экране может быть обкусано, и в результате испортится впечатление от всего просмотра.

В работе *X11* имеются также проблемы с безопасностью. Среди важных — та, что все клиенты могут слушать всё вводимое



► В нашем примере сотни запросов курсируют между *gedit*, клиентом, X-сервером и композитором, замедляя запуск.

с клавиатуры. Мы не собираемся ввергать вас в паранойю, но это может означать, что у вас есть клавиатурный шпион с окном нулевого размера, о котором вы даже не подозреваете.

Основное различие между *Wayland* и X в том, что в *Wayland* композитор и сервер — одно. Это экономит несколько мегабайт в рабочей памяти. Еще одно конструктивное решение в *Wayland* — предоставление клиентам доступа к графическому буферу. Они могут быть в совместной памяти Posix или в доступной памяти GPU. Благодаря такому решению весь рендеринг совершается клиентами и затем отправляется композитору; клиенты просто раздвигают указатель на буфер, используемый в данный момент времени. Фактически, клиенты могут использовать несколько буферов и опрашивать их в цикле. На этой стадии процесса композитор получает всё, что ему требуется — визуальный уровень, и отправляет его на экран. Поскольку данные уже доступны, никакого копирования не делается, и это повышает эффективность.

Украшательство окон тоже всегда было головной болью для X, но в *Wayland* рендеринг может осуществляться клиентом или сервером. Надо надеяться, это значит, что менеджеры окон позаботятся о рендеринге, не приковав нас ко всему лишь одному стилю для всех приложений. По части безопасности, *Wayland* требует от всех клиентов регистрировать, что они слушают, дабы все процессы отслеживались на случай поиска вредоносного процесса.

Установка Wayland

Прежде чем начать свое путешествие по *Wayland*, не забудьте проверить свое графическое оборудование. Лучше всего поддерживаются Intel i915, AMD и Nvidia. Для двух последних опций по-прежнему нужны драйверы с открытым кодом, поскольку проприетарные драйверы пока что не поддерживают *Wayland*. Есть также возможность компиляции для Raspberry Pi. Графическое оборудование — Broadcom VC4, и для него тоже требуются драйверы с открытым кодом.

Git: Постоянные обновления

Если вы хотите обновить свою локальную копию до самой последней версии, перейдите в директорию и «вытащите» разницу. *Git* работает с файлами и директориями в той директории, которую мы только что клонировали.

Помните, что все файлы, которые мы намерены добавить, должны быть добавлены в директорию, а затем в *git*.

В *git* есть масса функций, которые мы на нашем уроке не используем, но нам может понадобиться

использовать команду `pull` системы управления версиями:

```
$ cd wayland
$ git pull git://anongit.freedesktop.org/wayland/wayland
```

При запуске этой команды *git* начинает с проверки правильности дерева директории посредством проверки содержимого директории `.git`. Следующий шаг — определить, какие были сделаны изменения со времени последней синхронизации, и внести эти

изменения в вашу локальную копию. Фактически эта команда запускает две команды: `git fetch` и `git merge`. Эти две команды скачивают все изменения с сервера в вашу директорию и объединяют всё, что у вас есть, в одну ветвь.

При внесении кода надо быть осторожнее, чтобы не повредить своим собственным изменениям; наш случай проще, мы всего лишь следуем за разработкой. Данный процесс усложняется, когда вы вносите код, но это уже тема для другого учебника.

Вы увидите, что система для использования *Wayland* обычно поставляется с вашим дистрибутивом, хотя пока что не по умолчанию. Помните, что *Wayland* — это протокол, поэтому без композитора библиотеки *Wayland* бесполезны. Мы также будем использовать для начала базовую реализацию, *Weston*.

В Ubuntu 16.10 уже включен *Weston*. Пользователи Fedora будут рады узнать, что *Wayland* готов к тестированию в Fedora 24, а Fedora 25 намерен использовать его в качестве композитора по умолчанию с Gnome.

Любителям острых ощущений подойдет использование встроенного сервера отображения; исходный код для *Wayland* доступен на git://anongit.freedesktop.org/wayland.

Компиляция Wayland

Для самостоятельной компиляции следует обзавестись всеми библиотеками разработки для *Wayland*, и, вероятнее всего, понадобится *xwayland* для поддержки X поверх *Wayland*. При компиляции пакетов Debian вам понадобятся *gcc*, *cpp*, *binutils* и *libc-dev*, но не нужен *g++* для самого *Wayland*. Если всё это кажется вам слишком сложным, горю помогут удобные инструменты. В Ubuntu вы можете использовать в команде `apt` параметр `build-dep`:

```
$ sudo apt build-dep wayland Weston wayland-protocols libinput xserver-xorg
```

Она установит программы, необходимые для компиляции пакетов.

Поскольку мы компилируем из исходника, лучший способ не отстать от разработчика — получить код через *git*. Сначала проверьте наличие у себя *git* с помощью `$ apt install git`. При запуске *git* создает директорию в соответствии с тем, что вы скачали, поэтому в нашем случае мы создали директорию `install`:

```
$ mkdir install
$ cd install
$ git clone git://anongit.freedesktop.org/wayland/wayland
```

Созданная директория является клоном того, что имеется на сервере *git*, как понятно из подкоманды. В данном проекте мы будем повторять те же самые команды для этих пакетов:

```
git://anongit.freedesktop.org/wayland/wayland-protocols
git://anongit.freedesktop.org/wayland/libinput
git://anongit.freedesktop.org/xorg/lib/libXfont
git://anongit.freedesktop.org/xorg/xserver
git://anongit.freedesktop.org/wayland/Weston
```

В списке присутствуют два пакета для *X.org* — причина в том, что вам надо будет запускать и те приложения, которые пока что не поддерживают *Wayland*, с помощью *XWayland*.

Теперь всё готово для компиляции вашей версии *Wayland*. Первый шаг — настроить среду, чтобы у вас был отдельный исполняемый файл, который не запускается по умолчанию. Тогда вы

не перезапишете свою рабочую среду. Задайте переменную, содержащую путь ко всем нашим библиотекам:

```
$ export WLD=$home/install
```

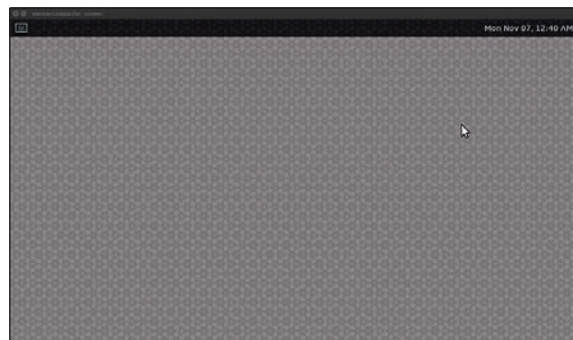
Мы здесь создали переменную `WLD`, которой и воспользуемся для настройки других переменных, управляющих процессом компиляции. Плюс указали путь к `install`, поскольку ранее скачали туда все программы. Теперь надо настроить другие переменные, которые направят наши инструменты компиляции в правильные директории.

```
$ export LD_LIBRARY_PATH=$WLD/lib
$ export PKG_CONFIG_PATH=$WLD/lib/pkgconfig:$WLD/share/pkgconfig/
$ export PATH=$WLD/bin:$PATH
$ export ACLOCAL_PATH=$WLD/share/aclocal
$ export ACLOCAL="aclocal -I $ACLOCAL_PATH"
$ mkdir -p $ACLOCAL_PATH
$ export MAKEFLAGS="j5"
```

Здесь требуется небольшое пояснение. `LD_LIBRARY_PATH` используется системой для поиска библиотек времени исполнения. Мы настраиваем ее, чтобы позаботиться о применении самых новых библиотек. `PKG_CONFIG_PATH` служит той же цели, но при поиске информации по библиотекам, установленным в системе. `PKG_CONFIG` будет использовать `autoconfig` при сборе метаданных для процесса компиляции. `ACLOCAL` поможет `autoconfig` в поиске имеющихся библиотек и пакетов, чтобы `PKG_CONFIG` мог собрать метаданные для этого процесса. В данном случае нам надо обеспечить, что мы делаем компиляцию для *Wayland*, а не *X.org*.

При компиляции мы запускаем приведенные ниже команды для каждой директории, созданной *git* ранее:

```
$ cd libinput
$ ./autogen.sh --prefix=$WLD
$ make check
$ make && make install
$ cd ..
```



Скорая помощь

О проекте *Wayland* имеется обширная информация на его официальной странице: <https://wayland.freedesktop.org>. *Wayland* также ведет список проектов, его поддерживающих, отмечает, насколько далеко продвинулись в плане поддержки крупные дистрибутивы, и приводит ссылки на инструментари.

»

» *Weston* начинается с простого, поскольку нет файла настройки. Украшение экрана с помощью `.config/weston.ini` является одной из целей нашего урока.

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

Последовательность будет точно такой же для всех пакетов, кроме *xserver*, где надо будет использовать параметры для отключения *X* и включения *Wayland*. Добавьте всё это в одну строку сразу после `./autogen.sh`:

```
$. /autogen.sh --prefix=$WLD --disable-docs --disable-devel-docs
--enable-xwayland --disable-xorg --disable-xvfb --disable-xnest
--disable-xquartz --disable-xwin
```

Это заставит *xserver* работать поверх *Wayland*, а не в качестве самостоятельной системы.

Вам нужно быть в директории пакета, который вы пытаетесь скомпилировать, когда вы запустите три другие команды, поэтому запустите `cd [path-name]`, чтобы добраться до пакета.

Следующая команда — `./autogen.sh --prefix=$WLD`. При запуске она ищет в вашей системе требуемые для компиляции библиотеки. Эта команда также пытается угадать определенные системные параметры, требуемые для компиляции. Префикс сообщит команде, где искать в первую очередь. Поскольку `LD_LIBRARY_PATH` и `PKG_CONFIG_PATH` тоже настроены в соответствии с префиксом, мы будем использовать всё, что есть в этой библиотеке. Префикс также покажет, где установлен ваш исполняемый файл, чтобы у вас был его оригинал для вашей системы на случай, если со вновь скомпилированной возникнут проблемы. По завершении этой команды вы увидите, как всё это сработало. У многих появится ошибка `No Package <имяпакета>`; если это произойдет, вам надо найти пакет и либо установить его на свою систему обычным образом, либо скомпилировать его в этой директории.

После успешного выполнения `autogen.sh` вы можете двигаться дальше и выполнить `make check`. Эта команда, как видно по ее названию, запускает самопроверку. И, наконец, запуск `make` скомпилирует код, создавая бинарники в текущей директории, а `make install` поместит бинарники в соответствующие директории. Здесь вступает в дело `PATH`, и поскольку путь использует наш префикс, бинарники в конечном итоге окажутся в нашей директории. Когда с бинарниками всё будет сделано, можете поменять префикс, переключиться на `root` и заново запустить `make install`, сделав бинарники системными. Однако старайтесь замечать и исправлять все ошибки. Общей оплошностью бывает компиляция в неверном порядке — и получение сообщений об ошибках, заключающихся в отсутствии пакета или старой версии. Если вы собираетесь употребить свою версию, указывайте путь туда, где вы ее компилировали. В нашем случае им был `$WLD/bin/weston`.

Теперь *Wayland* и *Weston* у нас готовы к работе. Чтобы запустить поставляемую версию, из *X*-терминала, скомандуйте `$ weston &/cend`. Если вы хотите запустить версию, которую только что сами скомпилировали, надо использовать путь к этому бинарнику: `$ HOME/bin/weston`.

➤ *Wayland* станет выглядеть лучше, если добавить *Firefox* в раздел `launcher` и установить симпатичный фон в разделе `shell`.



➤ **Hawaii** — дистрибутив, созданный для работы только с *Wayland* и динамически адаптирующийся к используемому устройству.

Когда *Weston* запустится впервые, он будет выглядеть довольно уныло, поскольку здесь нет файла `config`. *Weston* ищет этот файл в вашей директории `home` в директории `.config/`, а значит, надо создать `weston.ini`. Содержимое этого файла разбито на разделы с заголовками в квадратных скобках. Самые используемые разделы — ядро, клавиатура, вывод и программа запуска. Например, в минимальной основной раскладке он будет содержать только `modules=xwayland.so` и два других модуля: `cms-colorsd.so` для управления цветами и `screen-share.so`, применяемый для запуска второго экземпляра вашей сессии для удаленного доступа. Сначала добавим на рабочий стол фон:

```
[shell]
background-image=/home/matstage/Pictures/rod_stuga_GRIPEN.
jpg
background-type=scale
```

Желая добавить значок программы запуска в верхнюю панель экрана, мы обратимся к разделу программы запуска.

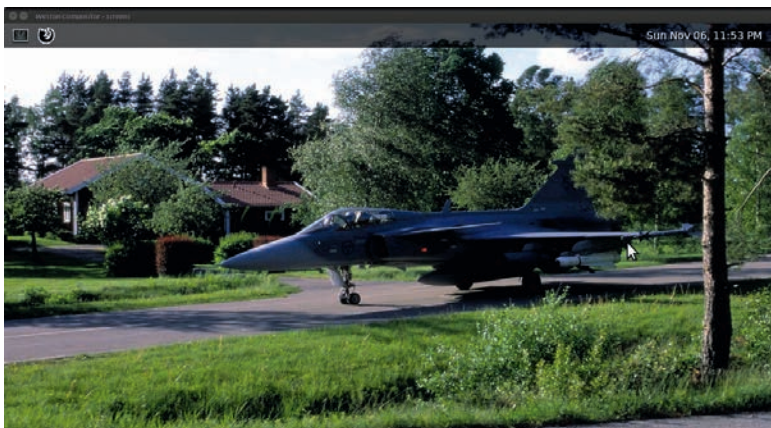
```
[launcher]
icon=/usr/share/icons/HighContrast/24x24/apps/firefox.png
path=/usr/bin/firefox
```

Мы использовали значок из темы `HighContrast`, которая была у нас на компьютере. Верхняя панель экрана по умолчанию имеет прозрачный серый цвет, что можно изменить в разделе оболочки с помощью `panel-color=0xff00ff00`. Цвет изменится на зеленый, это нас устраивает. Однако, заглянув в справочник по *Weston*, вы обнаружите, что на данный момент сделать вы можете очень немного.

Рабочие столы и дистрибутивы

Во-первых, у нас есть *Maynard*, рабочий стол для *Weston*, который умеет добавлять панели, виджеты рабочего стола и помощников. Этот проект предназначен для Raspberry Pi, но, как видно на его странице GitHub, у него нет зависимостей. Более известные инструменты также стабильны: например, у *Enlightenment* есть опция компиляции для *Wayland* как в сочетании с *X*, так и в виде экспериментального пакета только *Wayland*. Во всей документации вы увидите, что поддержка *Wayland* исключительно экспериментальная, но при этом подчеркивается, что он вполне готов для тестирования и применения в специализированных системах.

GTK+ уже компилирует большую часть своих приложений для работы в *X* или *Wayland*, при этом по умолчанию настроен *X*. Это приводит к проблеме при использовании виртуального терминала и запуске приложения *Gnome* в сессии *Wayland*, поскольку нередко это оказывается сессия *X*, если специально не указать



➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

Разделы, доступные для weston.ini

В файле настройки доступны следующие разделы:

- **core** Используется для выбора опций композитора и загрузки модулей. В Ubuntu 16.10 мы обнаружили, что модули, упоминаемые в справочнике, не совпадают с модулями в нашей системе. Здесь вы также можете выбрать свой сервер, но обычно этого не требуется, потому что правильный уже выбран для вашей системы.
- **libinput** Настройка устройства ввода. На момент написания доступна только `enable_tap`.
- **shell** Индивидуальная настройка рабочего стола. Здесь настраиваются фон и поведение окон,

например, затухание при открытии и закрытии и желаемое число рабочих областей. Можно также аннулировать опцию по умолчанию `weston-desktop-shell`; для этого создайте собственную или найдите в сети.

- **launcher** Добавьте программу запуска в панель. Она будет запускать ваши приложения.
- **screensaver** Здесь можно настроить свой скринсейвер. Если раздел пуст, значит, скринсейвер отключен.
- **output** Используется для настройки экрана, что особенно полезно для ноутбуков.

- **input-method** Используется исключительно для настройки пути к исполняемому файлу экранной клавиатуры.
- **keyboard** Выберите раскладку клавиатуры, скорость повторения нажатия клавиш и состояние NumLock по умолчанию.
- **terminal** Задаёт шрифты для терминалов.
- **xwayland** Задаёт путь к `xwayland`, обычно используется по умолчанию.
- **screen-share** Опции совместного использования экрана, настраивает удаленный сервер для совместного использования экрана.

ее при запуске. Например, при запуске `gnome-session` произошел сбой. Мы советуем запускать ее под `X`, чтобы увидеть, что работает, и только потом запускать `Wayland`, когда решите пойти дальше. Если вы хотите использовать `Clutter`, то он опирается на `GDK`, то есть подкреплен пакетами `GTK+`.

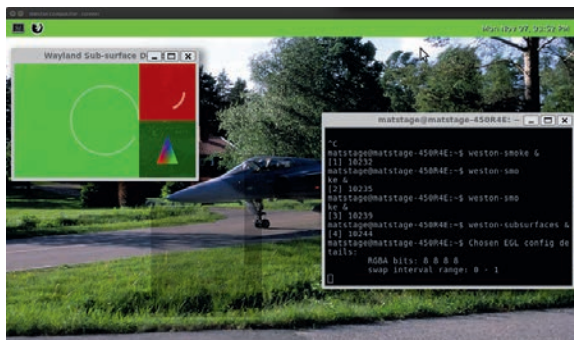
`Qt5` также имеет очень активное сообщество разработки, и дистрибутив `Hawaii` настроен на `Wayland` по умолчанию и имеет рабочий стол на базе `Qt`, разработанный для быстроты и простоты и внешне очень похожий на `Cinnamon`. Менеджер окон `Papirus` использует набор инструментов `Qt5`. Основная цель этого проекта — заставить оболочку адаптироваться к текущему форм-фактору. Это означает, что вы должны быть в состоянии использовать его на любом типе устройств, и рабочий стол сможет адаптироваться. `Papirus` использует `Qtquick` и среду `Green Island`, разработанную для дистрибутива `HawaiiOS`. Помните: `SDL` имеет экспериментальную поддержку, которая отключена по умолчанию.

Несколько дистрибутивов с самого начала строились в расчете на полноценный рабочий стол `Wayland`. Как только что упоминалось, один из них — `Hawaii`. Это компактный дистрибутив с несложным интерфейсом, который использует в качестве менеджера пакетов сменщика `Yum, DNF`.

Если ваш обычный рабочий стол — `KDE`, то вам может подойти `Plasma`: с версии 5.8 этот пакет считается пригодным к использованию, хотя и не лишен проблем. `Plasma` также доступна в виде пакетов для `Ubuntu` и других основных дистрибутивов.

Потрясающий проект — `Motorcar`, разработанный для отображения рабочего стола 3D: он выглядит, как настоящая комната для ваших приложений, а не плоский экран. Например, окно терминала можно «повесить на «стену» дальше на экране.

У вас также имеется возможность использовать `sway`, замену мозаичного менеджера окон `i3`. Подобный менеджер не предусматривает фона, поскольку окна заполняют весь экран целиком.



➤ Здесь строка заголовка сменила цвет посредством настройки атрибута `panel-color`, а для иллюстрации запущено несколько демо-приложений.



Принцип состоит в том, что первое окно закрывает всё, а когда пользователь открывает второе окно, экран делится пополам. Пользователь использует горячие клавиши на клавиатуре, чтобы решить, в каком направлении будет происходить разбиение экрана. `Sway` изначально разработан для `Wayland`.

Менеджеры окон

`Velox` очень на него похож, поскольку это тоже мозаичный менеджер окон. Однако у него есть система тегов для перемещения окон на определенные экраны, виртуальные или реальные. При запуске `Velox` пользователь помещает на окно тег, и присваивает такой же тег или теги экрану. Этот экран заполняется окнами с соответствующими тегами.

На нашем уроке мы представили вам `Wayland`, замену всех видов `X`, и ведущая в этой области разработка достойна восхищения. Однако на данный момент мы бы не советовали использовать `Wayland` на постоянной основе. Памятуя об этом, многие опытные пользователи, скорее всего, предпочтут перейти на него, используя функции `xwayland`, если для определенных приложений не существует иных решений. Причиной для подобного перехода, помимо чистого любопытства, является улучшение производительности благодаря более прямому соединению между клиентами и графическим оборудованием. Результатом нового подхода стало то, что для рендеринга экрана требуется меньше памяти и что для той же задачи создается меньше нагрузки на CPU и GPU.

Основные дистрибутивы уже включили `Wayland` и активно работают над тем, чтобы все их приложения были доступны для пользователя, желательно используя `Wayland` по умолчанию или через уровень совместимости `xwayland`. За 2017 г. большая часть компьютерных технологий откажется от `X11`, и закат самой первой попытки создания системы размещения окон для систем Unix будет неизбежен. **LXF**

➤ **Wayland**, на котором запущено несколько тестовых программ, для демонстрации разработчикам `Wayland` возможностей и тестовых функций.

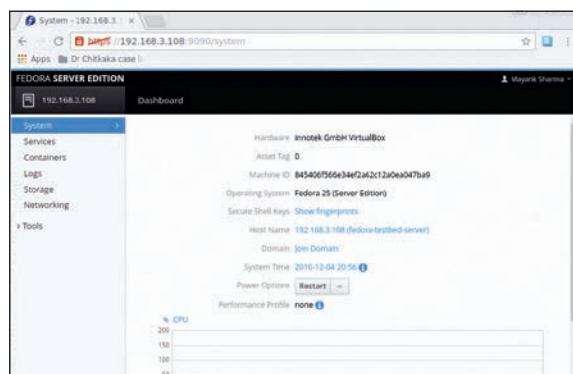
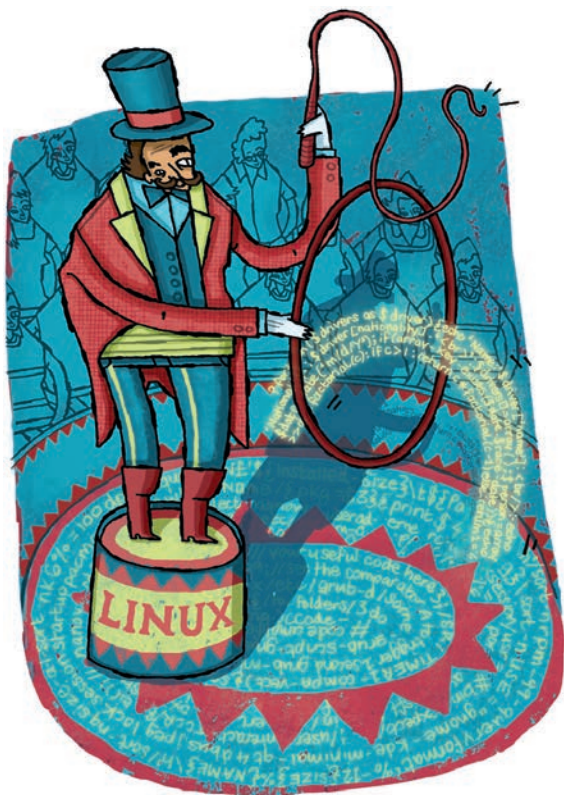
Cockpit: АДМИН для сервера

Шашанк Шарма надевает свою щегольскую шляпу сисадмина и рассматривает *Cockpit*, инструмент для управления серверами.



Наш эксперт

Шашанк Шарма пишет о программах с открытым кодом с 2004 г. И проводит свои дни, практикуя искусство юриспруденции в качестве судебного адвоката в Дели.



» *Cockpit* просто предоставляет интерфейс для управления существующими системами Linux. Для установки компонентов сервера, таких, как web- или файловый сервер, *Cockpit* не годится.

на сервер, то же самое должно быть доступно для работы из интерфейса *Cockpit* немедленно.

В порядке дополнительного бонуса, *Cockpit* не требует создавать новых учетных записей пользователя: после установки он интегрируется с вашим сервером, и пользователи могут входить в систему с помощью своей обычной аутентификации. Пользователи также сохраняют те же самые разрешения, и вам незачем беспокоиться об их повторном переопределении.

Установка и настройка

Cockpit доступен в установке по умолчанию Red Hat Enterprise Linux и даже Fedora Server. В Fedora Server по завершении установки можно указать любой системе на сеть на <http://server-ip-address:9090> и получить доступ к интерфейсу *Cockpit* для управления сервером.

Поскольку инструмент предназначен для содействия в администрировании серверов, в программных репозиториях его предлагают не все дистрибутивы. Те, кто работает на Ubuntu, Mint и производных дистрибутивах, могут установить его, запустив

```
$ sudo add-apt-repository ppa:cockpit-project/cockpit
$ sudo apt-get update
$ sudo apt-get install cockpit
$ sudo systemctl enable cockpit.socket
$ sudo systemctl start cockpit
```

Первая команда в приведенном коде добавит официальный PPA (Personal Package Archive), а затем вы должны обновить репозитории второй командой. Далее мы установим *Cockpit* с помощью команды `apt-get install` и, наконец, включим его с помощью `systemctl enable`. В других дистрибутивах, например, CentOS или даже Arch Linux, *Cockpit* устанавливается аналогично.

Cockpit предоставляет администраторам интерфейс на базе Web для мониторинга и управления несколькими серверами Linux. С его помощью вы можете запустить и остановить сервисы, инспектировать логи и управлять хранилищем, и т.д. Хотя *Cockpit* предназначен для администраторов, легковесный и интуитивный интерфейс делают его идеальным для домашних серверов и небольших предприятий.

Однако, в отличие от других инструментов администрирования, *Cockpit* не имеет собственного обширного списка функций. Фактически, всё, что он делает — обеспечивает вас средствами доступа к вашему серверу и изменению настроек. Вот почему *Cockpit* не нужна конфигурация или инфраструктура: он готов к использованию, как только будет установлен.

Более того, *Cockpit* не берет на себя и не блокирует обязанностей администратора. Иными словами, если вы подправили сетевые настройки через web-интерфейс, вы в любое время можете отменить их через командную строку вашего сервера, и наоборот. Кроме того, *Cockpit* динамически обновляется, поэтому изменения, внесенные на сервер, отражаются в интерфейсе браузера через несколько секунд. Например, если вы добавили новый LVM

Скорая помощь

Хотя большинство дистрибутивов Linux перешли на *Systemd*, некоторые по-прежнему используют старую систему *SysVinit*. На такие дистрибутивы установить *Cockpit* нельзя, поскольку ему необходим *Systemd* для выполнения многих своих функций, таких как управление сервисами и демонами.

Настройки *Cockpit* не требует, но вы всё же можете подправить некоторые из его основных параметров, отредактировав файл `/etc/cockpit/cockpit.conf`. Это включает несколько попыток входа в систему. Опция `AllowUnencrypted` позволяет принять в *Cockpit* нешифрованный HTTP-трафик. Если порт *Cockpit* по умолчанию (9090) занят другим вашим сервисом, можете настроить *Cockpit* на использование другого порта. Для этого служит файл `/etc/systemd/system/cockpit.socket.listen.conf`.

Возможно, вам придется создать `listen.conf` и, в зависимости от вашего дистрибутива, предшествующие директории:

```
$ sudo mkdir /etc/systemd/system/cockpit.socket.d/
$ sudo touch /etc/systemd/system/cockpit.socket.d/listen.conf
```

Когда все директории и файлы будут на месте, откройте файл `etc/systemd/system/cockpit.socket.d/listen.conf` в своем любимом редакторе и добавьте следующее:

```
[Socket]
ListenStream=
ListenStream=192.168.3.113:8989
```

Убедитесь, что вы добавили полный адрес и номер порта. Поскольку *Systemd* позволяет описывать несколько директив `Listen` в одном сегменте сокетов, вы можете настроить *Cockpit*, чтобы он слушал несколько портов, добавив следующие строки:

```
[Socket]
ListenStream=
ListenStream=9090
ListenStream=8989
```

При каждой правке файла *Systemd* надо запускать следующие команды, чтобы изменения вступили в силу:

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart cockpit.socket
```

Если ваш сервер настроен с SELinux, остается выполнить еще один шаг. Нам надо велеть SELinux разрешить соединения с этим портом с помощью команды `semanage`:

```
$ sudo semanage port -a -t websm_port_t -p tcp 8989
```

Приступаем

Установив и запустив *Cockpit* на своем сервере, запустите браузер с любого компьютера в вашей сети и направьте его на `http://192.168.3.150:9090`. Если вы получите предупреждение безопасности от *Firefox* при доступе к web-интерфейсу, надо будет добавить это соединение в исключения системы безопасности. Нажмите `Advanced > add Exception > Confirm Security Exception` [Расширенные > добавить Исключение > Подтвердить Исключение Системы Безопасности]; вы окажетесь на экране приглашения. Используйте информацию для входа на свой сервер, чтобы войти в *Cockpit*. Пользователи *Chrome* при получении подобного предупреждения могут таким же образом нажать на `Advanced`, а затем на ссылку `Proceed to 192.168.3.108 unsafe` [Перейти на 192.168.3.108 в небезопасном режиме] внизу.

Один из принципов проектирования *Cockpit* заключается в том, что пользователи не должны искать инструкции, чтобы разобраться с интерфейсом. Этот приоритетный принцип дизайна

структуры привел к появлению весьма простого и легкого в навигации интерфейса.

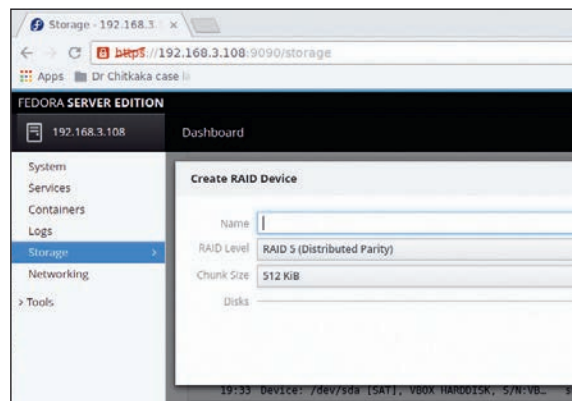
Интерфейс разделен на две части. Вверху находится вкладка с IP-адресом вашего сервера. Боковая панель слева применяется для мониторинга и настройки разных компонентов, таких как `Logs` [Логи], `Services` [Сервисы], `Storage` [Хранение] и `Networking` [Сеть], и т.д. Еще одна вкладка вверху, `Dashboard`, обеспечивает графическое отображение потребления CPU, памяти, ввода/вывода диска и сетевого трафика.

Если у вас несколько машин, которые надо отслеживать с помощью *Cockpit*, то первым шагом будет добавить все эти серверы. Нажмите на вкладку `Dashboard` вверху экрана. Все настроенные серверы перечислены в нижней части экрана. Нажмите кнопку `+ Add Server` справа. Вам предложат указать IP-адрес серверов, и вы также можете выбрать цвет для каждого вновь добавленного сервера. Доступ к каждому серверу осуществляется через SSH с первого сервера, через который вы соединились с *Cockpit*.

По добавлению в *Cockpit* нескольких серверов левая вкладка вместо IP-address изменится на `Machines`. Теперь при нажатии на эту вкладку она перечислит все добавленные серверы и выделит сервер, который вы хотите настроить.

Например, если вы хотите сменить имя хоста своего удаленного компьютера, выберите сервер на вкладке `Machines` и нажмите `System` на боковой панели. Вы увидите солидный массив информации, такой как `Hardware` [Оборудование], `Machine ID` [ID компьютера], `Operating System` [Операционная система], `Hostname` и `Performance` [Производительность], и т.д. Нажмите на `Hostname`, во всплывающем диалоговом окне введите новое имя хоста в поле `Real Hostname` и нажмите `Change`. Благодаря компоненту `cockpit-bridge` изменения записываются в соответствующие файлы настройки и отражаются мгновенно.

Интерфейс *Cockpit* также предлагает полнофункциональный терминал, пригодный для изучения или изменения файлов настройки или выполнения задач, которые не предлагаются в интерфейсе на базе Web. Чтобы запустить терминал, нажмите на левой боковой панели `Tools`, а затем `Terminal`. Немедленно появится окно терминала с находящимся в нем в данный момент пользователем.



Скорая помощь

Порт 9090, который слушает *Cockpit*, нельзя изменить посредством редактирования файла `/etc/cockpit/cockpit.conf`. Чтобы изменить этот порт, надо редактировать файл `/etc/systemd/system/cockpit.socket.d/listen.conf`.

»

» *Cockpit* использует хранилище для настройки и мониторинга хранилища, дисков и монтированных устройств.

За сценой

В *Cockpit* объединено несколько разных компонентов. Сервис `cockpit-ws`, запускаемый *Systemd*, обеспечивает web-сервер — `'ws'` означает именно это. Сервис `cockpit-ws` остается сидеть в фоне и автоматически активируется, если *Systemd* определяет, что *Cockpit* пытается установить соединение с портом по умолчанию, 9090. Сессия *Cockpit* добавок запускает программы `cockpit-ws` и `cockpit-bridge`,

которые тоже работают в фоновом режиме. Программа `cockpit-ws` — это компонент web-сервиса *Cockpit*, и она управляет коммуникацией между браузером и другими компонентами — такими как `cockpit-bridge`.

Программа `cockpit-bridge` используется подобным же образом для передачи сообщений между сетевым интерфейсом и сервером.

При добавлении к вашей сессии *Cockpit* еще одного сервера `cockpit-ws` устанавливает контакт с ним через SSH. Затем он берет управление программой `cockpit-bridge`, которая работает на вновь добавленном сервере с использованием SSH. Вот почему на каждом компьютере, за который вы хотите наблюдать через *Cockpit*, надо установить *Cockpit*, а также настроить демон SSH.

Вертикальной полосы прокрутки нет, но вы можете воспользоваться колесиком мыши, чтобы прочитать содержимое основанной на браузере сессии терминала.

Терминал на базе Web также содержит полную историю выполненных команд, как обычная сессия терминала *Bash*. Полнофункциональный терминал предлагает автодополнение по Tab и даже расширение истории. Если вы запускаете команды с помощью инструмента терминала на базе Web, вы сможете найти его в истории терминала на вашем удаленном сервере, и наоборот.

Пользователь, вошедший в *Cockpit*, имеет те же привилегии, что и пользователь, подключившийся к удаленной машине через SSH. В стандартной установке *Cockpit*, если у пользователя есть разрешение на `sudo` для расширения привилегий, то оно распространяется и на *Cockpit* для выполнения привилегированных задач.

Cockpit не допускает групповых операций, затрагивающих несколько серверов. То есть, любая ваша настройка ограничена только выбранным в данный момент сервером. Желая внести такие же изменения для каждого сервера, добавленного в *Cockpit*, вы должны вручную повторять процесс, сначала выбрав сервер во вкладке *Machines* и выполнив каждую задачу.

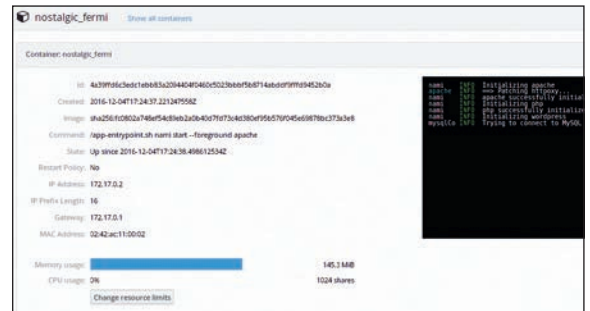
Например, если вы хотите добавить нового пользователя техобслуживания на роль Администратора для каждого из добавленных серверов, сначала выберите сервер на вкладке *Machines*. Затем нажмите *Tools > Accounts > Create New Account*. Введите всю информацию в диалоговое окно — Full Name [ФИО], User Name [Имя Пользователя] и Password [Пароль]. После этого вы окажетесь на экране *Accounts*, где перечислены все пользователи. Нажмите на вновь созданного пользователя, и теперь вы можете определить Роль [Role] для этого пользователя, например, *Server Administrator* [Администратор Сервера], и даже заблокировать учетную запись пользователя, если заподозрите нарушение правил. Затем повторите весь процесс для каждого сервера.

За сценой *Cockpit* использует для создания и изменения учетных записей пользователей старые добрые `useradd`, `usermod` и `passwd`. Интерфейс не позволяет специально приписывать группы учетным записям пользователя, но для подобных действий всегда можно призвать терминал.

Управление сервисами

Systemd относится ко всем ресурсам, с которыми система может работать и которыми может управлять как одним блоком. Каждый блок управляется демонами, а работают с ним при помощи разных инструментов системы. Блок сервисов описывает, как управлять сервисом или приложением на сервере. Сюда входит запуск или остановка сервера, обстоятельства, при которых он будет автоматически запущен, и зависимости для связанных с ним программ.

С другой стороны, блок сокетов не запускает собственных демонов; вместо этого он слушает на определенном IP-адресе или порте. Например, файл `cockpit.socket`, который мы обсуждали



➤ На экране управления контейнером можно изменить приписанные к контейнеру ресурсы.

в разделе *Настройка* для настройки альтернативного порта вместо порта по умолчанию 9090.

Для группировки и упорядочивания блоков *systemd* использует цели [targets]. Грубо говоря, это уровни запуска, в которых разные сервисы, сокет и блоки запускаются с разными целями.

Сервисы, работающие на удаленном сервере, управляются из вкладки *Services* на левой боковой панели. Экран *Services* разделен на вкладки *Targets* [Цели], *System Services* [Системные Сервисы], *Sockets* [Сокеты], *Timers* [Таймеры] и *Paths* [Пути].

Вы можете просматривать все сервисы, включенные и активные в данный момент, нажав на вкладку *System Services*. Прокрутите до конца страницы, чтобы просмотреть список всех Отключенных [Disabled] и Статических [Static] сервисов. Со включенными/выключенными сервисами разобраться легко. В зависимости от вашего случая, вы можете включить отключенный сервис или отключить включенный, если он выполнил свою обязанность. Статический сервис — это тот, который нельзя включить. Причина в том, что эти сервисы выполняют только одну задачу или функцию либо используются в качестве зависимости другого сервиса и не могут запускаться отдельно.

Вы можете нажать на имя каждого сервиса во вкладке *System Services*, чтобы просмотреть информацию о нем — например, как долго он работает. На том же экране можно через выпадающее меню *Start* [Запустить], *Stop* [Остановить], *Reload* [Перезагрузить], *Restart* [Перезапустить] любой сервис. Точно так же можно отключить (Disable) или включить (Enable) сервис из другого выпадающего меню.

Для сбора и хранения регистрационных данных *systemd* использует сервис `systemd-journald`. Эти данные готовятся посредством сбора информации из разных источников, таких как сообщения журнала ядра, сообщения системного журнала, стандартный вывод и стандартные сообщения об ошибках от различных системных сервисов, и т.д.

Доступ ко всем этим собранным данным можно получить из интерфейса *Cockpit*, нажав на вкладку *Logs* на левой боковой панели, которая группирует записи журнала в виде вкладок под разными подзаголовками, а именно: *Errors* [Ошибки], *Warnings* [Предупреждения] и *Notices* [Уведомления]:

➤ **Notices** Как и предполагает ее название, вкладка *Notices* содержит простые уведомления — например, когда система монтирует директорию или запускает сервис, выполняет процедуры `cron` или аналогичные действия.

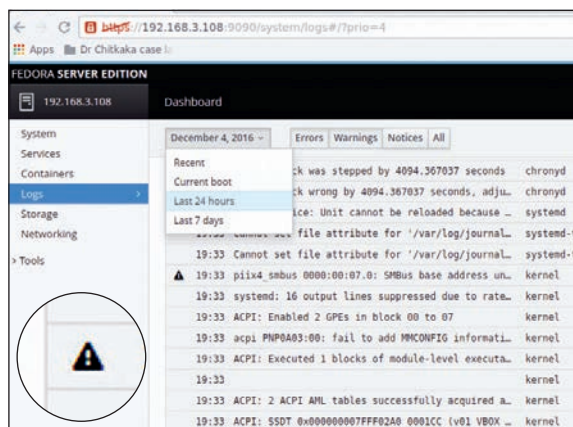
➤ **Warnings** Вкладка *Warnings*, опять же с «говорящим» названием, содержит предупреждения, например, когда система не может запустить команду из-за недостаточных разрешений или когда *Cockpit* не может соединиться с удаленным сервером из-за проблем соединения сетью, и т.д. Обычно подобные предупреждения и уведомления не содержат никакой критически важной информации.

➤ **Error** Вкладка *Error* содержит список ошибок, собранных по всей системе. Вы можете использовать выпадающее меню `calendar`

Скорая помощь

Вы всегда должны блокировать учетные записи пользователей на рабочем компьютере, если знаете, что данный пользователь не сможет получить доступ к серверу во время отпуска.

➤ Следует уделять особое внимание записям журнала с пометкой '!' в крайнем левом столбце.



Systemd против SysVinit

Очень долго дистрибутивы Linux использовали для запуска сервисов и демонов скрипт инициализации *SysVinit*. *Init* — это процесс демона, запускаемого, как только вы начинаете загружать дистрибутив Linux. Это превращает *init* в родительский процесс всех остальных процессов. Если в силу неких причин процесс *init* не запустился, не запустится ни один из остальных процессов, что приводит к тому, что называется «паникой ядра».

Унаследованный недостаток *SysVinit* — процессы *init* запускаются последовательно, то есть следующий процесс запускается только после успешного запуска предшествующего, и это увеличивает время загрузки. Для замены *init* было разработано несколько альтернатив, например, система *Mudar*, разработанная для Pardus Linux, и *Upstart*, который изначально был разработан для популярного дистрибутива Ubuntu.

Systemd точно так же был разработан, чтобы обойти неудачи *init*, и сейчас является системой *init* по умолчанию в разных дистрибутивах, включая Red Hat Enterprise Linux, Fedora, Ubuntu и многие другие современные дистрибутивы.

Systemd — процесс фоновый, разработанный для параллельного запуска процессов и сокращающий таким образом время загрузки дистрибутива. Процессы и демоны в системе *systemd* управляются с помощью команды *systemctl*. Следующая таблица иллюстрирует ряд основных команд:

Задача	Команды <i>systemctl</i>
Запуск сервиса	<i>systemctl start name.service</i>
Остановка сервиса	<i>systemctl stop name.service</i>
Перезапуск сервиса	<i>systemctl restart name.service</i>
Включение сервиса при запуске	<i>systemctl enable name.service</i>
Отключение сервиса при запуске	<i>systemctl disable name.service</i>
Проверка, включен ли сервис при запуске	<i>systemctl is-enabled name.service</i>

После установки *Cockpit* вы должны использовать команды включения и запуска, описанные выше.

вверху слева, чтобы ограничить количество отображаемых *Cockpit* записей журнала. Вы можете выбрать просматривать записи журнала, относящиеся к Current Boot [Текущей Загрузке] или логи за последние 24 часа или даже за прошлую неделю.

Каждая запись в журнале также упоминает в крайнем правом поле систему, где запись была создана. Например, запись, созданная *Network Manager*, покажет *Network Manager*. Нажав на вкладку *All*, вы можете просмотреть все созданные записи журнала.

Вкладка *Networking* на боковой панели точно так же содержит информацию по каждому настроенному интерфейсу, например, *eth0* и т. д., в том числе объем переданных или полученных данных в аккуратном графическом виде. Вы, кроме того, можете получить доступ ко всем записям журнала, созданным *Network Manager*, из вкладки *Networking*.

В отличие от вкладки *Logs*, нажатие на запись журнала во вкладке *Networking* не выдает никакой дополнительной информации. Подобно вкладке *Networking*, вкладка *Storage* на боковой панели, помимо логов, предоставляет информацию о диске. Все доступные устройства перечислены внизу страницы, и вы даже можете настроить RAID, нажав на кнопку '+' в разделе *RAID Devices*.

Работа с контейнерами

Cockpit настроен на работу с *Docker* для управления контейнерами. Но сначала вам надо установить на свой сервер пакет *cockpit-docker*. Сервер Fedora 25 предлагает этот пакет в своих репозиториях, хотя по умолчанию он не установлен; однако вы можете это сделать командой

```
$ sudo dnf install cockpit-docker
```

После установки вам придется выйти из сессии *Cockpit* и войти снова. Войдя, вы увидите, что боковая панель слева теперь содержит дополнительную вкладку под названием *Container*.

Нажмите на вкладку *Container*, и *Cockpit* сообщит вам, что сервис *Docker* еще не запущен. Чтобы запустить сервис, нажмите на кнопку *Start Docker*.

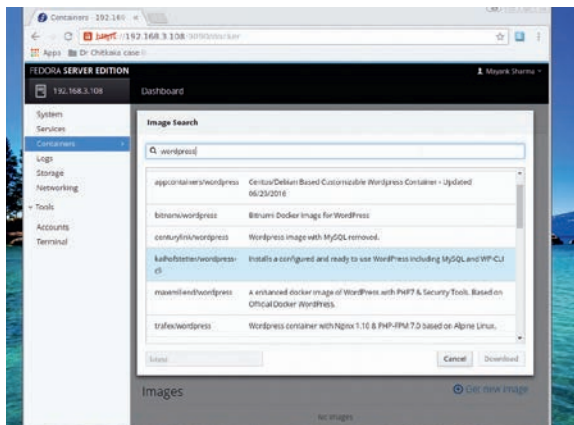
Вверху страницы *Container* отображается общее использование CPU и памяти для всех работающих контейнеров. Если у вас пока нет контейнеров, прокрутите страницу вниз, и под разделом *Image* нажмите на кнопку *Get new image* [Получить новый образ]. В диалоговом окне поиска образа введите имя требуемого образа, например, *WordPress*, и *Cockpit* будет искать указанные ключевые слова по всем опубликованным контейнерам в *Docker Hub*.

С помощью этого механизма можно очень быстро настроить практически любой контейнер. Например, для настройки *MariaDB* найдите его с помощью диалогового окна поиска и затем установите. После установки вы найдете все образы внизу страницы *Containers*. Нажмите на кнопку *Start* в нижней правой части экрана. Во всплывающем диалоговом окне введите соответствующую информацию, как то: пользователь, имя базы данных и т. д. Когда всё будет готово, вы сможете запустить контейнер с настроенными значениями.

Следует позаботиться об установке всех необходимых для образа *Docker* компонентов перед его запуском. Например, вам не удастся установить *WordPress*, если у вас еще нет настроенной базы данных. Поэтому перед запуском образа *WordPress* требуется сначала скачать и настроить образ *MariaDB*. Если вы запустите образ *WordPress*, не создав базу данных, *Cockpit* не сообщит вам об ошибках, но образ так и застрянет на стадии инициализации базы данных.

Чтобы запустить окно управления контейнером, можете нажать на любой работающий образ. Это даст вам опции *Stop* [Остановить], *Restart* [Перезапустить] и *Delete* [Удалить] контейнер, а заодно и обеспечения графиков, отображающих потребление CPU и памяти. Если образ *Docker* запущен в терминале, вы также получаете опцию запускать команды терминала через этот экран управления.

С помощью *Cockpit* даже неопытные администраторы могут легко управлять несколькими удаленными серверами без клавиатуры и монитора. Он предлагает незагроможденный и интуитивный интерфейс для выполнения обычных основных задач по управлению системой. Для расширенных задач, например, оформления пропускной способности канала, вы должны перейти в терминал и программировать файлы настройки. **LXF**



➤ Будьте осторожнее с образами, которые скачиваете. Основная информация не всегда предоставляет все необходимые детали.



Git: Вводка к контролю версий

Присоединяйтесь к **Шашанку Шарме** в его исследованиях популярнейших систем управления версиями и внедрении их в повседневное использование.



Наш эксперт

Шашанк Шарма — выступающий в суде адвокат из Дели и страстный фанат Arch Linux. Он постоянно охотится за ценными для технарей реликвиями.

Скорая помощь



Работая с папкой `.git`, вы должны быть предельно внимательны. Если вы случайно удалите эту папку, она удалит ваш репозиторий `git`, и, следовательно, всю историю вашего проекта.

Git — популярная система управления версиями (versioning control system, VCS), разработанная для выполнения двух основных задач. Во-первых, она дает пользователям возможность отслеживать изменения в файле, что позволяет определять и исправлять ошибки. Во-вторых, распределенная природа Git означает, что одновременно редактировать эти файлы могут несколько пользователей. Более того, система способна отслеживать все изменения, независимо сделанные несметным числом пользователей. В экосистеме Linux репозитории Git часто используются разработчиками программ, чтобы дать обычным пользователям доступ к самым свежим своим релизам, а также обеспечить совместную работу над большими проектами.

Для разработчиков добавочным преимуществом использования VCS является ее способность объединять разные независимо разработанные ветви в единую программу. Но Git не ограничивается только обслуживанием разработчиков. Поскольку VCS предназначена больше всего для работы с простыми текстовыми файлами, ее легко применить для слежения за важными документами. Она предоставляет простой механизм быстрого отката, чтобы отменить изменения в файле.

Прежде чем приступить к более интересным возможностям повседневного использования Git, давайте рассмотрим, как использовать Git для поиска программ. Самый популярный программный репозиторий — это GitHub, его используют более 14 миллионов совместно работающих над проектами разработчиков, и он размещает более 35 миллионов репозиториях.

Скорее всего, в вашем дистрибутиве Linux есть клиент *Git*, но его легко и установить, через менеджер пакетов вашего дистрибутива. После установки вы сможете дублировать файлы для любого проекта, размещенного на удаленном сервере, например, GitHub, запустив команду `$ git clone project-url`.

Команда скопирует все файлы в новую локальную папку с тем же именем, что и репозиторий в данном URL. Затем вы сможете компилировать программы, исследовать код и даже вносить изменения, уже не нуждаясь в доступе к серверу.

Если вы хотите добавить к своему проекту поддержку Git, вам сначала надо создать репозиторий Git этого проекта. Это делается из терминала, запуском команды `git init` в директории проекта. Внутри этой директории создается скрытая папка `.git`, где будет храниться вся информация о файлах и архивированные копии прошлых версий.

Git берется за дело

Создав репозиторий, вам надо заполнить его файлами, а следующий шаг — определить файлы, которые вы намерены отслеживать с помощью *git*. Команда `git add <filename>` допускает использование регулярных выражений и спецсимволов. Например, желая отследить в своем репозитории все файлы, имя которых начинается с буквы 'n', это можно сделать с помощью команды `git add n*`. Она добавляет указанные файлы на текущую сцену.

Подготовив репозиторий Git, вы сможете создавать моментальные снимки своего проекта перед каждым значительным изменением. Если вас не устроит новая модификация, вы легко вернетесь к предыдущему состоянию и начнете заново.

Для создания моментального снимка всех файлов, которые вы уже добавили на текущую сцену, надо запустить `git commit -m 'Осмысленный комментарий'`. А если вы хотите только обновить изменения, внесенные в одиночный файл, укажите его имя: `git commit -m 'Осмысленный комментарий' имяфайла`

Тщательно отбирайте слова для своего комментария, поскольку он будет использоваться для определения навскидку изменений в файле на момент каждого обновления, при просмотре истории

Основная терминология Git

Проект Git использует обширную и уникальную терминологию для описания множества своих функций. Вот основные термины, необходимые, чтобы понять смысл операций контроля версий Git.

- » **clone [клон]** Обозначает локальную копию удаленного репозитория.
- » **branch [ветвь]** Обозначает измененную версию репозитория.
- » **commit [обновление]** Определяет процесс сохранения изменений в файл в вашей локальной копии.
- » **merge [слияние]** Объединяет изменения из одного репозитория (удаленного) с другим (локальным).

- » **fetch** Вы можете использовать эту команду для поиска и возврата любых изменений в удаленном репозитории без их слияния.
- » **pull [вытягивание]** Берет изменения из удаленного репозитория и соединяет их с локальной копией.
- » **push [проталкивание]** Отправка изменений в файл в вашей локальной копии в соответствующий файл в удаленном репозитории.
- » **tree [дерево]** Похоже на директорию. Каждое дерево ссылается на другие деревья и/или блобы.
- » **blob [блób]** Объект Git, обычно файл или диск, хранящий данные.

- » **staging area [текущая сцена]** Место хранения незавершенной работы, пока не готовой для конечного обновления.
- » **working area [рабочая область]** Состоит из файлов, присутствующих в данный момент в директории `git-tracked`.

Рабочий процесс Git использует файлы в вашей рабочей директории, которые по мере выполнения работы переносятся в область текущей сцены. Следующий шаг включает обновление изменений; и, наконец, вы отправляете готовый файл в удаленный репозиторий.

Настраиваем Git

Инструмент *git config*, который является частью клиента Git, используется для определения переменных настройки, таких как имя, адрес электронной почты и т. д. Эти переменные служат для отслеживания личности любого пользователя, который вносит изменения в файлы, работая над в совместным проектом

Общие настройки, применимые для всех репозитиев Git, хранятся в файле `~/.gitconfig`, размещенном в вашей директории `/home`.

```
$ git config --global user.name "Shashank Sharma"
$ git config --global user.email "linuxlala@gmail.com"
```

Две этих команды задают соответственно имя пользователя и адрес электронной почты для всех репозитиев текущего пользователя. А если вы намереваетесь сотрудничать с разными группами, используя определенные для каждого репозитория адрес электронной почты или имя пользователя, вы просто запускаете эти команды из директории `.git` ваших репозитиев без опции `--global`.

Настройки для каждого репозитория хранятся в файле `.gitconfig` в директории `.git` репозитория.

Другим важным параметром, который вам надо настроить, является ваш текстовый редактор. Если он не определен, Git запустит редактор по умолчанию, когда вам понадобится написать сообщение, например, при обновлениях. Но вы можете запустить следующую команду, чтобы настроить редактор по своему выбору:

```
$ git config --global core.editor vim
```

файла. В идеале комментарий должен быть сжатым, но предоставлять достаточную информацию. После каждого успешного обновления Git будет выдавать уникальный ID.

Для каждого файла со значимыми изменениями надо запускать команду `git add` и `git commit` или команду `commit -m 'Осмысленный и информативный комментарий' -a`. Опция `-a` велит *git* обновить все файлы, уже содержащиеся в индексе.

Команды `git log` и `git show <commit ID>` пригодны для быстрого удостоверения в разных операциях обновления, которые вы время от времени выполняете в своем проекте. Если вы вдруг решили, что вас не устраивают недавние обновления, можно вернуться к ранее сохраненному состоянию, запустив команду `git checkout <commit ID>`. Вот почему важно пометить разные обновления соответствующими комментариями.

При совместной работе с другими пользователями обычная безопасность прошлых обновлений может показаться неадекватной. К счастью, Git позволяет разделить ваш проект на разные ветви. Каждая ветвь является копией вашего основного репозитория, именуемого главной ветвью, в том виде, в каком он существовал на момент определенного обновления. Вы можете решить собрать ветви в некую целостность или внести изменения, сделанные в ветви, в свой основной репозиторий.

Чтобы создать новую ветвь, надо запустить `git branch имяветви`. При запуске без указания имени ветви команда предоставит список всех ветвей. Теперь вы можете начать работу над вновь созданной ветвью без всякого риска повлиять на основную ветвь, запустив команду `git checkout имяветви`.

С этого момента вы работаете с файлами обычным способом, внося изменения в своем любимом редакторе и постоянно добавляя файлы и обновляя изменения. Решив соединить ветвь со своим основным репозиторием, вернитесь в него по команде `git`

`checkout` и затем запустите команду `git merge имяветви`. Фактически вы можете объединить две или более ветви и объединить завершенную ветвь с основным репозиторием.

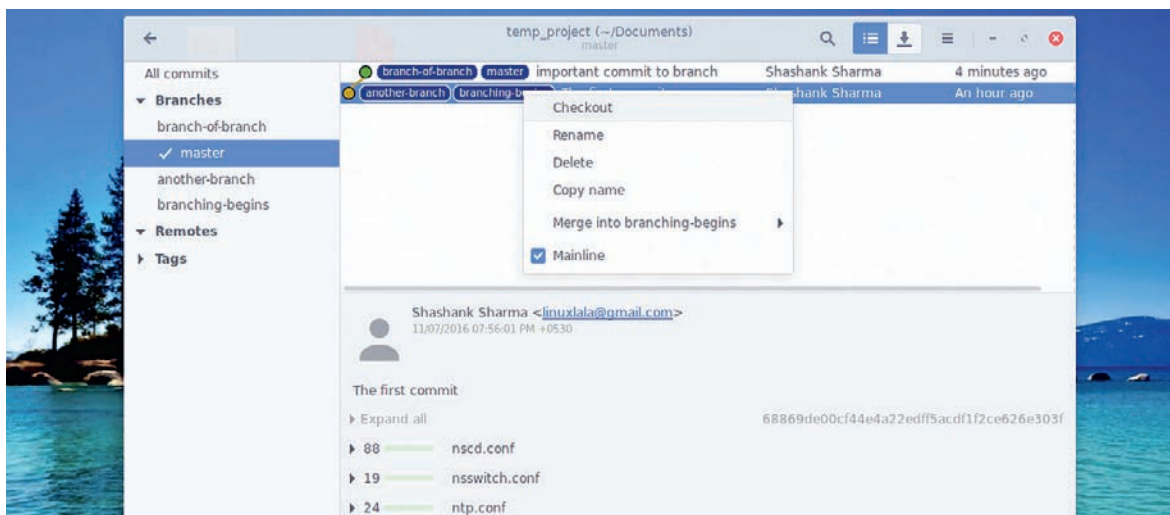
Git предлагает собственный набор графических утилит в помощь пользователям, которым неудобно работать в командной строке. Вы можете использовать утилиту *Gitk* для просмотра всех обновлений, сделанных в вашем репозитории, разных файлов в каждом дереве, и т. д. Инструмент *git-gui* может также использоваться для внесения изменений в репозиторий посредством добавления новых обновлений, редактирования существующих, создания ветвей, выполнения объединений, и т. д.

Графический Git

Помимо этого, есть несколько свободных приложений с открытым кодом, а также коммерческих приложений, позволяющих выполнять все эти функции. Однако из-за сложной природы Git эти инструменты могут показаться не слишком интуитивными.

Хотя изначально Git предназначен для управления открытым кодом, вы можете использовать Git для отслеживания любого файла, будь то первые наброски глав книги, отредактированные изображения семейного отдыха, PDF, ODT, и т. д. Единственный недостаток работы с файлами, которые не являются простым текстом — вы не можете использовать утилиту *diff* для просмотра изменений в файлах.

Если вам показалось, что мы чересчур быстро пробежались по важным концепциям Git — так оно и есть. Проект Git предлагает широкий выбор функций, и мы лишь коснулись поверхности того, что он умеет делать. Больше ресурсов и информации вы найдете на <https://git-scm.com/doc> — это официальное руководство, книга, а также видео и прочие руководства, объясняющие его многочисленные функции. **135**



» Git предлагает разные опции контекстного меню в зависимости от того, где вы щелкнули.

ЧАСТЬ 3

IoT: Дом, который послушен нам

Лучше перебдеть, чем недобдеть! На всякий случай **Максим Черепанов** дублирует канал управления, используя мобильную связь.



Наш эксперт

Максим Черепанов — заместитель директора в небольшой коммерческой фирме. По образованию железнодорожник-управленец. Linux занимается с 2008 г. Чем дольше это делает, тем дальше от Windows. За эти годы научился чистить реестр и забыл слово «антивирус».



Привет всем программистам своего послушного дома. Продолжим свой нелегкий, но интересный проект по управлению домашним хозяйством. Дорисуем блок-схему — теперь в ней появятся канал управления по электронной почте, физический канал контроля за нагрузкой и обратная связь от контроллера к пользователю. Канал управления по SSH показан одной

стрелкой, а почтовое управление — двумя. Это сделано потому, что за управление и контроль по почте отвечают разные службы.

Давайте сегодня немного оторвемся от грешной земли (я имею в виду локальную сеть и Интернет). Это очень полезно с точки зрения резервирования управления: мало ли что может произойти? Не вовремя произведенная оплата за Интернет, выход из строя роутера, перебои у провайдера — и вот вы уже не хозяин над своим ПД. Поэтому жизненно необходим второй канал, не связанный с Интернетом.

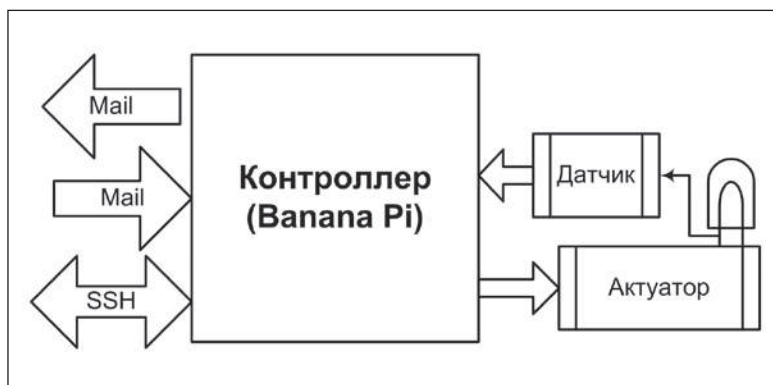
После непродолжительного перебора вариантов я остановился на сервере IP-телефонии *Asterisk*. Имея опыт применения его для организации интеллектуальной мини-АТС в офисе, я знаю, что *Asterisk* надежен, легко настраивается и существует для большинства популярных архитектур, в том числе и для ARM. О применении его для офиса вы сможете прочитать в серии статей, опубликованных в **LXF** в прошлом году. Для своих нынешних нужд мы установим его из репозитория и не будем ставить графическую среду *Free-PBX*, а управимся редактированием конфигурационных файлов.

Техническое задание

Сначала вкратце набросаем, что должен делать наш ПД:

- 1 При звонке на модем ПД должен запросить пароль — первый рубеж защиты от посторонних. После третьего неверного набора должен произойти сброс.
- 2 При правильно набранном пароле должен произойти контроль по списку допущенных к управлению номеров — второй рубеж защиты.
- 3 Как сервис, ПД должен проговорить вам текущую дату и время. Возможно, вам это не понадобится; вы сможете отключить эту опцию.
- 4 ПД должен кратко проговорить подсказку-помощь по управлению. При управлении одной нагрузкой это не обязательно, и будет лишней тратой времени. Но если у вас их более двух десятков, то вспомнить, как включается вентиляция на чердаке, получится не сразу.
- 5 Теперь ПД должен ожидать от вас команду. Ее будем подавать нажатием кнопок на телефоне, *Asterisk* будет декодировать сигналы DTMF, и в его конфигурационном файле достаточно будет указывать те числа, которые вы будете набирать на клавиатуре.
- 6 Приняв команду, ПД должен ее выполнить и доложить нам о выполнении.
- 7 По окончании сеанса ПД должен положить трубку.

Все вышеуказанное можно потом скорректировать под свои нужды. Мы в примере будем опираться только на эти пункты.



» Рис. 1. Блок-схема ПД с управлением по e-mail.

Установка Asterisk и канала связи

Сервер *Asterisk* ставим простой командой

```
# aptitude install asterisk asterisk-dev
```

Конфигурировать его будем позже, а сейчас займемся каналом GSM-связи. Почему именно сотовая связь? Потому что она не зависит от проводов, и если у вас дома есть электричество, то именно этот вид связи даст возможность управлять ПД при отсутствии Интернета.

Начать стоит с поиска или покупки GSM-модема. Повсеместно операторы переходят со стандарта 3G на 4G, поэтому модемы для компьютеров сетей 3G сейчас стоят недорого. Мало того, на досках объявлений по продаже б/у вещей вы можете купить такой модем очень дешево. Вполне возможно, что ненужный модем найдется у друзей или знакомых.

Теперь пару слов о разумности и бережливости (читатели моих статей эту фразу в том или ином виде видели неоднократно). Надо понимать, что управление ПД по каналу GSM будет производиться с вашего мобильного телефона. Неплохо будет, если SIM-карта, предназначенная для модема, будет от того же сотового оператора, что и ваша основная. Резоны: экономия (трафик внутри оператора обычно намного дешевле, чем между операторами), качество связи, скорость соединения. А потому и модем надо подбирать такой, чтобы он работал именно с вашей симкой. Часто в руки попадают устройства, предназначенные для работы в определенной сотовой сети. Это не беда, большинство из них можно разлочить (разблокировать) для работы с любой SIM-картой (инструкций в Интернете полно). Но если есть возможность выбора, то берите оба устройства (SIM-карту и модем) изначально такого же оператора, как и ваш.

Итак, вы имеете модем с симкой. Напоминаю о хорошем питании для каждого устройства: самый минимум для модема — 1,5 ампера. Если пренебрежете этим правилом, сбои будут вам гарантированы.

К сожалению, шина питания Banana Pi оставляет желать лучшего. Даже если вы примените блок питания на 5 А, USB-разъем может не выдать нужный ток на модем. Как вариант, можно воспользоваться USB-коммутатором с внешним питанием, типа D-Link DUB-N7. Для настройки я именно так и поступил.

Вставляем оттестированный модем (я вставил в коммутатор). Ждем, пока он начнет моргать светодиодом, и выполняем

```
$ lsusb
Bus 004 Device 004: ID 12d1:1446 Huawei Technologies Co., Ltd. E1552/E1800/E173 (HSPA modem)
```

Ага, устройство определилось. Но фраза HSPA modem заключена в скобки, а это означает, что в данный момент устройство не работает в режиме модема.

Теперь необходимо переключить модем в режим «только модем», для чего используем утилиту *usb_modeswitch*. Эта утилита имеется в репозиториях большинства популярных дистрибутивов, для ARM она тоже в наличии:

```
# apt-get install usb-modeswitch
```

Устанавливаем, создаем конфиг */etc/usb_modeswitch.conf* следующего содержания (или открываем имеющийся, всё стираем и вставляем свои строки):

```
DefaultVendor= 0x12d1
DefaultProduct= 0x1446
MessageContent=»55534243123456780000000000
000011062000000100000000000000000000»
```

Как видите, данные в первые две строки надо брать из вывода команды *lsusb*, где упоминается ваш модем.

Третья строка — управляющее сообщение для конфигурирования модема. Внимание, это строка для модема Huawei E173! Его

«одноклассников» (в строке вывода *lsusb* они указаны) для своего устройства вам придется искать самим.

После этого выполняем

```
# usb_modeswitch -c /etc/usb_modeswitch.conf
```

Если в консоль посыпалась куча строк, с последними двумя

```
Device is gone, skipping any further commands
-> Run lsusb to note any changes. Bye.
```

— значит, все произошло успешно. Надо вынуть и вставить модем, после этого проверим:

```
$ lsusb
Bus 004 Device 005: ID 12d1:1436 Huawei Technologies Co., Ltd.
```

И еще:

```
$ ls /dev/ | grep USB
```

```
ttyUSB0
```

```
ttyUSB1
```

```
ttyUSB2
```

Модем стал модемом. Теперь нам нужен драйвер канала для USB-модема. Качаем пропатченный под версию 11 *Asterisk*'а *chan_dongle* (подойдет только эта версия). Сделаем, как обычно, несколько действий одной командой:

```
# cd /usr/src && wget https://github.com/jstasiak/asterisk-chan-dongle/archive/asterisk11.zip && unzip asterisk11.zip && cd asterisk-chan-dongle-asterisk11 && aclocal && autoconf && automake -a && DESTDIR="/usr/lib/asterisk/modules"
```

Не обращаем внимания на этот вывод в консоль:

```
automake: no `Makefile.am` found for any configure output
```

Возникает всегда, но на работу не влияет.

Продолжаем:

```
./configure && make && make install && cp /etc/dongle.conf /etc/asterisk/
```

Подгрузим собранный модуль и посмотрим, увидел ли его *Asterisk*:

```
asterisk -rx 'module load chan_dongle.so' && asterisk -rx 'dongle show devices'
```

Вывод, указанный ниже, появляется не всегда сразу. У меня бывало и с первого раза, и приходилось перезапускать еще раз после 30-секундной паузы.

```
ID Group State RSSI Mode Submode Provider Name Model
Firmware IMEI IMSI Number
dongle0 0 Free 17 0 0 Beeline E173 11.126.85.00.209
35221604455xxx 250994091146429 Unknown
```

Всё в порядке; а теперь запоминаем имя (*dongle0*) и IMEI (*35221604455xxx*), и редактируем конфигурационный файл:

```
# nano /etc/asterisk/dongle.conf
```

комментируем секцию *[default]*, и в самом низу, где описывается устройство *[dongle0]*, комментируем всё, кроме строки *imei*, куда вписываем IMEI вашего модема из строки, которую показал *dongle show devices*, и дописываем

```
[dongle0]
imei= 35221604455xxx
context=from-gsm; контекст, куда будут передаваться звонки на модем
group=0
rxgain=3; увеличение громкости
txgain=3; увеличение громкости
resetdongle=yes; сбрасывать карту при инициализации
u2diag=-1
usecallingpres=yes
callingpres=allowed_passed_screen
```

Теперь добавим наш модуль к модулям, которые загружаются по умолчанию:

```
# nano /etc/asterisk/modules.conf
```

»

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

Добавим строку

```
load = chan_dongle.so
```

Перезапустим Asterisk:

```
# service asterisk restart
```

и проверим наличие своего модема, сейчас он должен быть сразу и без задержек:

```
$ asterisk -rx 'dongle show devices'
```

Собственно установка на этом заканчивается, далее будет настройка и написание диал-плана (плана набора).

Программируем план набора

Конфигурация плана набора содержится в файле настройки Asterisk — `extensions.conf`. Это один из самых важных конфигурационных файлов. Содержание файла `extensions.conf` разбито на секции, в которых могут быть определены или статические настройки и определения, или выполняемые команды плана набора — в этом случае они называются контекстами.

Контексты в файле `extensions.conf` начинаются со строки с именем контекста, заключенного в квадратные скобки. Далее следуют строки (экстеншены, обратите внимание на название самого файла), каждая из которых содержит определение или команду. Т.е. это язык программирования Asterisk, с помощью которого мы и зададим действия для своего ПД.

Вообще-то эта тема очень обширная — программированию для Asterisk посвящены целые книги и сотни сайтов в Интернете. Но нам для нашего пробного проекта ПД достаточно будет и одного контекста.

Если вы делали тестовый проект так же, как и я, то вам нужно будет просто сделать резервную копию файла `/etc/asterisk/extensions.conf`, затем стереть из него всё и вставить следующие строки:

```
[from-gsm]; From GSM
exten => s,1,Answer
exten => s,n,Playback(hello)
exten => s,n,Set(CHANNEL(language)=ru)
exten => s,n,SayUnixTime(,\'time\'kMQdb)
exten => s,n,Wait(1)
exten => s,n,Authenticate(8888)
exten => s,n,GotoIf("${CALLERID(num)}" =
"+79034444444")?999,1:)
exten => s,n,GotoIf("${CALLERID(num)}" =
"+79604444444")?999,1:)
exten => s,n,Wait(1)
exten => s,n,Hangup()
exten => 111,1,Set(RESULT=${SHELL(/home/user/on.sh 0)})
exten => 111,n,Playback(status/"${RESULT}")
exten => 111,n,Wait(2)
exten => 111,n,Hangup()
exten => 222,1,Set(RESULT=${SHELL(/home/user/off.sh 0)})
exten => 222,n,Playback(status/"${RESULT}")
exten => 222,n,Wait(2)
exten => 222,n,Hangup()
exten => 333,1,Set(RESULT=${SHELL(/home/user/st.sh 0)})
exten => 333,n,Playback(status/"${RESULT}")
exten => 333,n,Wait(1)
exten => 333,n,Hangup()
exten => 999,1,Background(toggle_channel)
exten => 999,n,WaitExten(5)
exten => 999,n,Hangup()
```

Прежде чем мы примемся разбирать этот файл построчно, вам необходимо записать 3 звуковых файла, перекодировать их в формат, пригодный для воспроизведения Asterisk'ом,

и «закинуть» в определенные места файловой системы вашего ПД. Текст этих файлов будет приблизительно таким: «Включить нагрузку — нажмите 111; выключить нагрузку — нажмите 222; узнать статус нагрузки — нажмите 333» (файл `toggle_channel`), «Нагрузка выключена» (файл `0`), «Нагрузка включена» (файл `1`). Записать их можно с микрофона с помощью *Audacity*, перевести в формат PCM 16 бит с частотой дискретизации 8000 Гц и сконвертировать в GSM-формат при помощи *sox* (это пакет, исполняющий роль «швейцарского ножа» для аудиофайлов) примерно так:

```
$ sox -V toggle_channel.wav -r 8000 -c 1 -t gsm toggle_channel.gsm
```

Наверное, не стоит упоминать, что пакет надо сначала установить, и файлы должны лежать в текущей директории?

Разложить их нужно таким образом.

В директорию `/var/lib/asterisk/sounds/en/` скопируйте файл `toggle_channel.gsm`, в этой же директории создайте поддиректорию `status` и уложите туда файлы `0.gsm` и `1.gsm`.

И еще — в домашней директории создайте файл `st.sh`, сделайте его исполняемым и наполните:

```
#!/bin/sh
#
echo -n `usr/local/bin/gpio read $1`
exit 0
```

Разбор плана набора

Бывалые программисты Asterisk нас обсмеют: их обычные файлы длиннее, чем лимузин Билла Гейтса (вообще-то эту фразу принято говорить про конфиг *sendmail*), а наш умещается на трети экрана. Но начинать надо с малого. И если вам понравится повелевать лампочками у себя дома с телефона, вы допишете свой план набора сами.

Любой экстеншен начинается со слова `exten =>`, затем следует его название/имя, затем приоритет. Первый экстеншен в контексте имеет приоритет 1, следующий может иметь приоритет `n` (next) или явное число (2,3,4...). Ставить `n` удобно: можно потом внутрь вставить еще один экстеншен, и нумерация не сообразится. В этом случае они выполняются в порядке написания. Если вы применяете явную нумерацию, то порядок расположения не важен, выполнение будет строго по номерам. Но пропускать номера нельзя: Asterisk будет упорно искать после 4-й строки 5-ю и никогда не выполнит 6-ю (если 5-я ошибочно пропущена). Уверен, что всё понятно; давайте разбираться.

Наш диалплан имеет только один контекст `[from-gsm]`, сюда будут приходиться все звонки с USB-модема. Asterisk использует некоторые имена экстеншенов в специальных целях, имя `s` — это start, начало обработки. Поэтому первым в контексте всегда исполняется экстеншен `exten => s,1,...`

Давайте уже разберем построчно то, что записали ранее, контекст `s`:

- 1 Отвечаем на звонок.
 - 2 Приветствуем звонящего.
 - 3 Устанавливаем для канала русский язык.
 - 4 Проговариваем время и дату.
 - 5 Ждем одну секунду.
 - 6 Просим ввести пароль, он задан в строке (8888).
 - 7 Если номер, с которого звонят, +79034444444, переходим на экстеншен 999.
 - 8 То же самое для номера +79604444444.
 - 9 Ждем одну секунду; сюда попадают, если звонят с чужого телефона или не прошли проверку паролем.
 - 10 Кладем трубку.
- Далее разберем контекст 999:

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

1 Фоном воспроизводим файл `toggle_channel`, в это время пользователь уже может производить набор.

2 Ждем еще 5 секунд по окончании звукового файла, вдруг не успели набрать команду? При наборе существующего экстеншена (111,222,333) переход будет выполнен на них.

3 Кладем трубку.

И разберем одну из командных групп экстеншенов, например, 111. Строго говоря, группы экстеншенов 111, 222 и 333 — это разные информационные группы, но построение их одинаковое.

1 Включаем первую нагрузку.

2 Информлируем пользователя, что мы включили нагрузку.

3 Ждем 2 секунды.

4 Отключаемся.

Остановимся подробнее на строке

```
exten => 111,1,Set(RESET=${SHELL(/home/user/on.sh 0)})
```

— это самый важный фрагмент. Данная команда сопрягает собственно *Asterisk* и скрипт операционной системы, т.е. мы исполняем команду прямо из телефонного сервера. А то, что возвращает скрипт, передается в переменную `RESULT` и может быть дальше использовано в работе *Asterisk*.

Дальше вам осталось только перезагрузить *Asterisk* —

```
# service asterisk restart
```

и зайти в консоль управления *Asterisk*:

```
# asterisk -rvvvvvvvv
```

Проверьте свой диалплан:

```
bananapi*CLI> dialplan show
```

Данная команда выдаст вам содержимое плана набора, и если в нем есть ошибки — укажет на них. Осталось только набрать номер вашего ПД и протестировать включение и выключение нагрузки. Консоль *Asterisk* не закрывайте, тогда вы увидите последовательное исполнение всех команд диалплана. Это поможет «выловить» ваши ошибки или усовершенствовать написанное.

Управление с помощью SMS

На удивление, это очень просто. Такая возможность уже предусмотрена в *Asterisk*. Для обработки SMS достаточно в контексте, отвечающем за прием звонков с USB-модема, поместить группу экстеншенов с именем “sms”. После чего все действия с SMS будут проходить именно в этих экстеншенах. Добавьте в свой диалплан две группы экстеншенов. Первая — вход для SMS:

```
exten => sms,1,Verbose(Incoming SMS from ${CALLERID(num)}
${SMS})
```

```
exten => sms,n,System(echo ${STRFTIME(${EPOCH},,%Y-%m-%d
%H:%M:%S)} - ${DONGLENAME} - ${CALLERID(num)}: ${SMS}' >>
/var/log/asterisk/sms.txt)
```

```
exten => sms,n,GotoIf("${CALLERID(num)}" =
"+7903444444")?888,1:)
```

```
exten => sms,n,Hangup()
```

Вторая — исполнительная часть:

```
exten => 888,1,GotoIf("${SMS}" = "ch0 on")?888,4:)
```

```
exten => 888,2,GotoIf("${SMS}" = "ch0 off")?888,6:)
```

```
exten => 888,3,Hangup()
```

```
exten => 888,4,Set(RESET=${SHELL(/home/setupini/on.sh 0)})
```

```
exten => 888,5,Hangup()
```

```
exten => 888,6,Set(RESET=${SHELL(/home/setupini/off.sh 0)})
```

```
exten => 888,7,Hangup()
```

Можно было всё разместить в одной группе, но я сделал для наглядности две. Разберем обе части, сначала с именем “sms”:

1 Выводим в консоль *Asterisk* сообщение, что принято неизвестное SMS от определенного номера, и показываем его содержимое. Это нужно только для отладки, но убирать не советую: времени не занимает, места тоже, а для контроля работы — очень важно.

2 Записываем в файл `/var/log/asterisk/sms.txt` содержимое SMS с указанием даты, времени и номера написавшего. Много места не займет, а для анализа пригодится.

3 Далее выполняем проверку: если SMS пришла от номера +79034444444, то принимаем его к анализу в экстеншене 888,1, если нет, то

4 Кладем трубку, т.е. прекращаем обработку.

Теперь экстеншены с именем 888 (понятно, что имя может быть любым — главное, чтобы переход на них был выполнен из экстеншена с анализом номера):

1 Если содержание SMS — “ch0 on” (включить канал с номером...), то переходим на 4-й экстеншен.

2 Аналогично проверяем, и при соответствии содержания — переходим на 6-й экстеншен, отключаем нагрузку.

3 При содержании SMS, не соответствующем шаблонам, прекращаем обработку.

4 Выполняем скрипт операционной системы по включению нагрузки.

5 Прекращаем обработку.

6 Выполняем скрипт по выключению нагрузки.

7 Прекращаем обработку.

Перезапустите *Asterisk* или только диалплан, проверьте его и попробуйте отправить SMS нужного содержания.

У меня в консоли *Asterisk* появилось следующее:

```
[dongle0] Got SMS from +79039999999: 'ch0 on'
```

```
-- Executing [sms@from-gsm:1] Verbose("Local/sms@from-gsm-00000000;1", "Incoming SMS from +79039999999 ch0 on")
in new stack
```

```
Incoming SMS from +79034354446 ch0 on
```

```
-- Executing [sms@from-gsm:2] System("Local/sms@from-gsm-00000000;1", "echo '2017-01-29 15:57:07 - dongle0 -
+79039999999: ch0 on' >> /var/log/asterisk/sms.txt") in new stack
```

```
-- Executing [sms@from-gsm:3] GotoIf("Local/sms@from-gsm-00000000;1", "1?888,1:") in new stack
```

```
-- Goto (from-gsm,888,1)
```

```
-- Executing [888@from-gsm:1] GotoIf("Local/sms@from-gsm-00000000;1", "1?888,4:") in new stack
```

```
-- Goto (from-gsm,888,4)
```

```
-- Executing [888@from-gsm:4] Set("Local/sms@from-gsm-00000000;1", "RESULT=4 -- 11") in new stack
```

```
-- Executing [888@from-gsm:5] Hangup("Local/sms@from-gsm-00000000;1", "") in new stack
```

```
== Spawn extension (from-gsm, 888, 5) exited non-zero on 'Local/sms@from-gsm-00000000;1'
```

Стоит упомянуть, что если на SIM-карте не хватает места для SMS, то модем уведомит *Asterisk* об этом, и никакой реакции на ваше сообщение не будет. Почистить сообщения можно прямо из *Asterisk'a*, командой

```
bananapi*CLI> dongle cmd dongle0 AT+CMGD=1,4
```

Разумно будет чистить сообщения из *cron*, по расписанию. Если вы планируете большой поток SMS, лучше чистить их прямо из диалплана командой `SHELL` (посмотрите, как мы управляем нагрузкой); оставляю написание такой команды для вас как домашнее задание.

И напоследок — еще одно замечание. Поскольку для Интернета модем сейчас не требуется, разумно его переключить в режим 2G — тогда он будет реже зависать и меньше потреблять тока (опять же экономия):

```
bananapi*CLI> dongle cmd dongle0 AT^SYSCFG=13,1,3FFFFFFF,2,4
```

Приведу еще несколько команд по теме — они вам пригодятся, если вы будете после работы с *Asterisk* использовать модем также и для выхода в Сеть:

» Только 3G/WCDMA — `AT^SYSCFG=14,2,3FFFFFFF,2,4`

» Предпочитать GPRS/EDGE — `AT^SYSCFG=2,1,3FFFFFFF,2,4`

» Предпочитать 3G/WCDMA — `AT^SYSCFG=2,2,3FFFFFFF,2,4`

Экспериментируйте; возможно, в каком-то варианте он будет работать у вас стабильнее.

На этом на сегодня всё, желаю вам стабильной связи! **LXF**

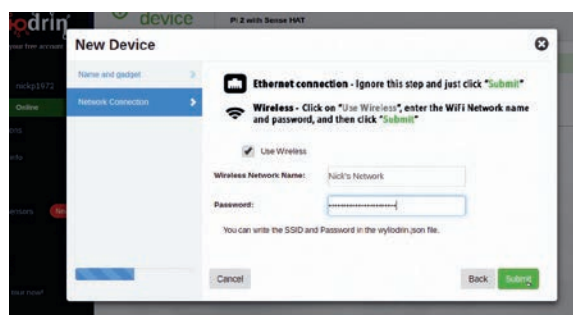
Wyliodrin: IoT-программы

Захотелось связать единой системой управления все свои гаджеты? Ник Пирс смотрит, как *Wyliodrin* поможет вам начать.



Наш
эксперт

Ник Пирс строит грандиозные планы насчет своей быстрорастущей коллекции Raspberry Pi. Он надеется, что *Wyliodrin* поможет ему пробиться через последний рубеж технологии умного дома.



» Перед началом программирования надо связать свое подключенное устройство с учеткой *Wyliodrin*.

то обнаружите, что по числу проектов, настраиваемых в оффлайн-режиме, ограничений нет. Вот это мы вам и покажем.

Вначале зарегистрируйте бесплатную учетную запись *Wyliodrin* на своем основном ПК (дальнейшие инструкции подразумевают, что у вас Ubuntu 16.04). Если *Chromium* у вас нет, установите его, затем откройте, перейдите на www.wyliodrin.com и нажмите на Sign up/in. Можно войти через Facebook, Google или GitHub — просто нажмите на соответствующую ссылку. На появившемся основном экране нажмите Add a device [Добавить устройство], чтобы начать.

Настройка на Pi

Затем подтвердите, что ваш Pi выбран как тип Gadget [Устройство/гаджет], дайте ему подходящее и понятное имя и нажмите Next. Если вы подключаетесь через Ethernet, нажмите Submit; в ином случае отметьте галочкой Use Wireless [Использовать беспроводное соединение], чтобы ввести SSID своей сети и пароль и продолжить. Здесь вам понадобится нажать на ссылку, чтобы скачать образ (в формате ZIP) в свою папку Downloads. Желая добавить *Wyliodrin* в уже готовую установку Raspbian, загляните во врезку Установка *Wyliodrin* вручную сверху стр. 75.

В процессе ожидания прокрутите немного вниз и нажмите на ссылку following file для скачивания файла с именем **wyliodrin.json** — пока оставьте его в Downloads. После скачивания извлеките его из архива, вставьте пустую отформатированную карту microSD и откройте Disks. Найдите в списке свою карту microSD и обратите внимание на имя устройства, например, /dev/sdb. Теперь откройте окно терминала и введите следующие две команды:

```
$ cd Downloads  
$ sudo dd if=wyliodrin_raspberry_pi_2016_03_24.img of=/dev/sdb
```

Замените **wyliodrin_raspberry_pi_2016_03_24** на имя образа, если оно отличается, а /dev/sdb — на имя своей карты. Подождите, пока образ Raspbian скопируется на карту. После этого наведите *Nautilus* на свою папку Downloads, скопируйте файл **wyliodrin.json** в том **boot**, который должен появиться на панели слева, и нажмите

Эх, старый добрый Интернет Вещей [IoT]. Он постепенно превращается в нечто менее навороченное и более полезное, но как интегрировать его в ваши собственные проекты? Если вы планируете использовать свой Raspberry Pi, ответ, скорее всего, включит различные виды кодига, и в большинстве случаев это будет кодиг на Python.

Все это хорошо и мило, но если вы новичок в программировании или же ваш Pi работает без монитора и клавиатуры, то вы нуждаетесь во взаимодействии с более дружелюбной к пользователю средой; здесь-то и выходит на сцену *Wyliodrin*. Это web-платформа с открытым кодом, разработанная для программирования и наблюдения за подключенными устройствами вроде Pi, Arduino и т. д. *Wyliodrin* приятен визуально и поставляется с собственным языком визуального программирования, через визуальный интерфейс drag-and-drop, который переключается со Scratch. На этом уроке мы увидим, как настроить *Wyliodrin*. Мы изучим, как подключить его к вашему Raspberry Pi, попрограммируем на Python и языке визуального программирования, ну и вообще проложим вам путь.

У *Wyliodrin* открытый код, но гибкий план ценообразования. При работе через его web-интерфейс бесплатная учетная запись ограничивает вас двумя устройствами и тремя проектами; но если вы обратитесь к *Wyliodrin STUDIO*, приложению на базе *Chromium*,

Скорая
помощь

Пока вы ждете, чтобы измененный образ Raspbian скопировался на вашу карту microSD, откройте другое окно терминала и введите `sudo kill -USR1 $(pgrep ^dd)`. Это не приостановит прогресс, но обеспечит вам разовое обновление. Повторите через минуту, чтобы убедиться, что оно всё еще работает.

Установка Wylidrin вручную

Быстрейший способ запустить *Wylidrin* на Raspberry Pi — это скачать готовый образ Raspbian со включенным *Wylidrin*. Тем не менее, продвинутые пользователи могут добавить его в уже существующую установку Raspbian, но имейте в виду: это займет несколько часов, и мы настоятельно рекомендуем предварительно сделать резервную копию вашей установки.

Вначале загрузитесь или войдите в свой Pi через SSH и скачайте скрипт установки Raspberry Pi:

```
$ wget "https://raw.githubusercontent.com/Wylidrin/wylidrin-server/master/scripts/install_raspberrypi.sh"
```

После этого откройте *raspi-config* (`sudo raspi-config`) и убедитесь, что файловая система была расширена. Вам также нужно выбрать Advanced Options [Дополнительно], а затем A6 SPI — Enable/Disable automatic loading [Включить/Отключить автозагрузку]. Повторите то же самое для A7 I2C — Enable/Disable automatic loading и A8 Serial — Disable serial shell [Отключить последовательную оболочку]. Затем введите `sudo nano /boot/config.txt` и добавьте строку `dtparam=i2c1=on`.

Сохраните изменения и выйдите, выполните `sudo nano /etc/modules` и добавьте следующие две строки:

```
i2c-dev
i2c-bcm2708
```

Сохраните изменения, выйдите и запустите скрипт:

```
$ chmod +x ./install_raspberrypi.sh
$ ./install_raspberrypi.sh
```

После установки добавьте свой Pi в *Wylidrin*, как описано в основном тексте, затем скачайте созданный файл *wylidrin.json*, передав его в раздел загрузки своего Pi. Перезагрузите Pi, и если всё хорошо — вы должны обнаружить, что он виден в *Wylidrin* и готов к программированию.

на кнопку Eject рядом с загрузкой для безопасного извлечения карты microSD. Затем вставьте карту в свой Pi и включите. Минуты две подождите, затем проверьте свою учетку в *Chromium* — если всё хорошо, вы увидите, что Pi значится в списке как находящийся в Сети. Далее надо добавить все требуемые библиотеки в свой Pi, что можно сделать удаленно из вашего браузера. Нажмите кнопку Settings [Настройки] рядом с индикатором Online и выберите Local Shell [Локальная оболочка]. Вы увидите кнопку, отмеченную eth0 (192.168.x.y), где 192.168.x.y — IP-адрес вашего Pi. Нажмите на нее, и откроется новое окно с введенными аутентификационными данными вашего пользователя Pi.

Эта оболочка работает так же, как и любая другая, и если вы планируете добавить библиотеки, например, для поддержки своего Sense HAT, введите

```
$ sudo apt-get update
$ sudo apt-get install sense-hat
$ sudo reboot
```

После перезагрузки вы готовы снова заняться своим основным ПК. Вы можете управлять проектами и запускать их прямо из этого web-браузера — нажмите на краткий экскурс, когда он появится, для быстрого знакомства с интерфейсом. Однако для целей нашего урока мы собираемся переключить внимание на приложение *Wylidrin STUDIO*. Это расширение *Chromium*, позволяющее управлять вашими проектами оффлайн — перейдите на <http://wylidrin.github.io/WylidrinSTUDIO> и установите его через магазин Chrome. Значок приложения появится под <chrome://apps/>; нажмите на него для запуска *Wylidrin*. Значок быстрого доступа появится в программе запуска, и его можно закрепить для упрощения доступа в будущем. Перед вами возникнет заставка — нажмите Exit [Выход], а затем кнопку Connect [Соединить] в верхнем правом углу окна *Wylidrin*. Ваш Pi должен показываться как опция, так что щелкните по нему для создания соединения. Введите имя пользователя своего Pi (*pi*) и пароль (*raspberry*), и должен открыться экран Pin Layout, показывающий раскладку соединений для вашего 40-контактного головного элемента.

Вы увидите основной экран, которые разделен на пять вкладок: Application [Приложение], Dashboard [Панель управления], Schematics [Схема], Pin Layout [Раскладка контактов] и Shell [Оболочка]. Вкладка Shell дает удаленный доступ к вашему Pi. Вкладка Application показывает код вашего проекта, а Dashboard содержит элементы, которые пригодятся для удаленного наблюдения за Pi.

Schematics работает в связке с приложением под названием *Fritzing* (<http://fritzing.org/home>), для тех, кто разрабатывает и собирает основанные на оборудовании проекты. Вы можете увидеть его в действии, запустив несколько демонстрационных скриптов: например, LED Blink показывает вам Pi, макетную плату, LED, резистор и два провода-переключателя типа «мама-папа», необходимых для завершения проекта. Выше вы увидите ряды кнопок. Здесь

есть кнопка play/stop [воспроизведение/стоп], запускающая ваш скрипт через Pi, рядом с ней — кнопка File Explorer [Файловый Менеджер], позволяющая удаленно просматривать хранилище вашего Pi. Вы можете создавать новые папки и загружать файлы с вашего ПК сюда. Затем идет Package Manager [Менеджер Пакетов], который, в основном, отображает список всех установленных модулей JavaScript и Python и обеспечивает легкий доступ к выбору еще не установленных библиотек — каждая из которых идет с описанием и примерами, включая *Django*, *MySQLdb* и *NLTK* (Natural Language Toolkit). Здесь также есть Task Manager [Диспетчер задач] для удаленного завершения процессов, сетевой менеджер для просмотра текущих настроек сети и удобная кнопка отключения.

Ваш первый проект

В этом проекте мы собираемся использовать Sense HAT вместе с модулем камеры Pi для создания скрипта, которые делает фото, когда вы перемещаете джойстик Sense HAT. Это не самое практичное применение HAT и камеры, но демонстрирует ваши возможности.

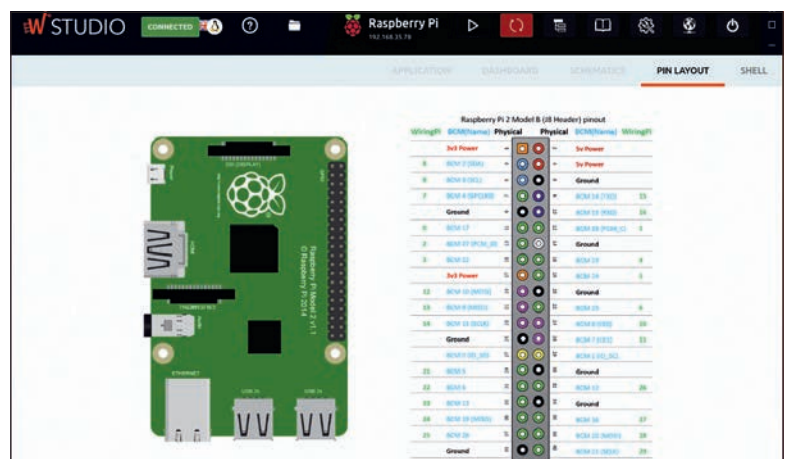
Библиотеки Sense HAT мы уже добавили; теперь надо открыть оболочку *Wylidrin STUDIO* для установки библиотек Python для Pi Camera, а также чтобы включить саму камеру:

```
$ sudo apt-get update
$ sudo apt-get install python-picamera
$ sudo raspi-config
```

Прокрутите вниз и выберите 6. Enable camera [Активировать камеру], выберите Enable [Активировать], а затем Finish [Завершить], и перезагрузитесь, когда вам это предложат. По завершении надо будет переподключить Pi.

Скорая помощь

С целью более подробного разбора языка визуального программирования посетите <https://github.com/google/blockly/wiki> для порции кода Google Blockly и <http://bit.ly/WylidrinVP> для эксклюзивных элементов *Wylidrin*.



➤ Основной интерфейс *Wylidrin STUDIO* содержит ряд удобных руководств, таких как эта полезная схема соединений.



https://www.raspberrypi.org/magpi-issues/Projects_Book_v1.pdf (а также v2) — онлайн-книга, включающая массу проектов для Pi. Совместите ее с Попутным Набором Популярных Компонентов [Sidekick Popular Component Kit] (£ 18, <http://bit.ly/SidekickKitv2>) для лучшего эффекта.

Мы собираемся создавать код этого проекта в Python: в Visual Programming нет прямой поддержки Sense HAT, так что вместо этого мы натащим по частям код из других источников. Вначале нажмите на кнопку папки [folder] в верхней части окна *Wyliodrin*. Нажмите Create new Application [Создать новое приложение], дайте ему подходящее имя, оставьте в качестве выбранного языка Python и нажмите Create. В окне Projects появится проект — нажмите на него, чтобы открыть. Вы увидите уже представленный демонстрационный скрипт. Нажмите на кнопку воспроизведения [play], и вы увидите, как в консоли появляются три строки:

```
python.main.py
Hello
Project exit with error 0
```

(Последняя строка показывает, что проект завершился успешно). Выделите весь код в окне проекта и нажмите Delete [Удалить]. Теперь создайте свой собственный проект, вводя следующие строки кода для вызова необходимых библиотек Python:

```
from sense_hat import SenseHat
from time import sleep
import picamera
```

Следующие строки создают переменные, которые мы будем использовать:

```
sense = SenseHat()
event = sense.stick.wait_for_event()
camera = picamera.PiCamera()
```

Очередная строка ждет обнаружения перемещения вашего джойстика:

```
event = sense.stick.wait_for_event()
```

А эти последние строки кода делают фотографию:

```
camera.start_preview()
sleep(2)
camera.capture('/home/pi/test.jpg')
```

По мере ввода вы заметите, что *Wyliodrin* предлагает автодополнение для ускорения ввода данных — нажмите Tab при появлении подходящей подсказки, чтобы выбрать ее. После этого убедитесь, что ваш Pi подключен, и нажмите клавишу воспроизведения. Вы должны увидеть, как **main.py** Python появляется в консоли, показывая, что теперь скрипт запущен. По нажатию или

перемещению джойстика позади вашего Sense HAT запустится камера — если ваш Pi подключен к экрану, вы увидите предпросмотр камеры, а спустя две секунды будет сделан снимок. В этот момент в консоли *Wyliodrin* должно появиться **Project exit with error 0**, показывая, что скрипт был успешно завершен.

Хотите удостовериться в этом? Нажмите на кнопку File Manager [Менеджер файлов] и перейдите в папку **home/pi**, где вы должны увидеть представленный файл **test.jpg**. Щелкните по нему, и вам предложат скачать его на ваш ПК, где вы сможете посмотреть его. Поздравляем, вы завершили свой первый проект!

Мы не кодировали с нуля, а позаимствовали соответствующие API (<https://pythonhosted.org/sense-hat/api> и <http://picamera.readthedocs.io>) и затем адаптировали код. В том и заключается красота Python: его код достаточно логичен, чтобы легко начать программировать, даже если вы раньше этим не занимались. Перечитав его как следует, вы поймете, как улучшить или адаптировать скрипт дальше — например, установить определенное разрешение для изображения или снимать видео вместо фото.

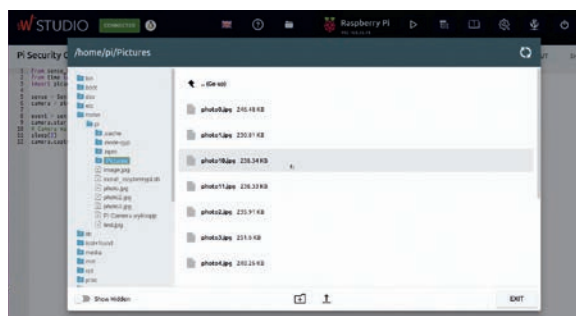
Визуальное программирование

Wyliodrin делает программирование в Python настолько лобовым, насколько возможно, но есть способ программирования и того проще, под названием визуальное программирование, который работает посредством перетаскивания ряда блоков кода в определенное место, и поднастройки переменных, чтобы совместить их в завершенном куске Python, не зная ни единой команды Python. К сожалению, поддержка Pi ограничена прямым взаимодействием с контактами GPIO (это не работает с Sense HAT, поскольку он взаимодействует через неподдерживаемый протокол I2C), а также основными командами для камеры.

Тем не менее, пошаговое руководство на стр. 77 показывает, как работает визуальное программирование, посредством создания очень простого проекта, использующего камеру Pi для получения серий фото, идущих подряд. Для начала нажмите на папку **Projects** и нажмите Create new application. Дайте вашему проекту название, затем щелкните по выпадающему меню Language [Язык]. Прокрутите его, пока не увидите Visual сверху. Выберите его, чтобы использовать визуальное программирование для создания вашего нового проекта, и нажмите Create [Создать]. Он появится в окне Projects, так что нажмите на значок, чтобы открыть его во вкладке Application в *Wyliodrin STUDIO*.

Вы увидите, что пара блоков находятся на месте — нажмите Show code [Показать код] справа, и появится код Python, который представляют собой эти блоки. Поочередно щелкните на каждом блоке и нажмите Delete, чтобы удалить их. Теперь, когда экран чист, следуйте пошаговому руководству (опять же, см. стр. 77), чтобы вставить создаваемые для вашего проекта блоки на место. После этого запустите проект и улыбнитесь для предстоящих фотографий — используйте Менеджер Файлов *Wyliodrin STUDIO*, чтобы скачать их со своего Pi на ПК для просмотра. **LXF**

➤ В *Wyliodrin STUDIO* встроен инструмент File Explorer, позволяющий просматривать и перемещать файлы с вашего Pi и на него.



Экспортируем ваш код

По умолчанию код хранится на вашем ПК, а значит, в *Wyliodrin STUDIO* вам надо запускать его вручную. В процессе тестирования это нормально; ну, а если вы захотите запустить его прямо со своего Pi? Одним из вариантов является открыть окно Projects, затем нажать Export рядом с проектом, который вы хотите экспортировать. Это позволит вам сохранить его как самостоятельный файл с расширением **.wylioapp**, который впоследствии можно переместить на ваш Pi через File Explorer — проблема в том,

что для его запуска вам понадобится установить *Wyliodrin STUDIO* на свой Pi (см. <http://wyliodrin.github.io/WyliodrinSTUDIO>); для этого при подключении выберите Chrome (Local Computer).

Куда более простой вариант — особенно если вы уже завершили свой проект — просто скопировать и вставить код Python в свой текстовый редактор, а потом сохранить его с расширением **.py**. Затем его можно перенести на ваш Pi и запустить через Python, примерно так:

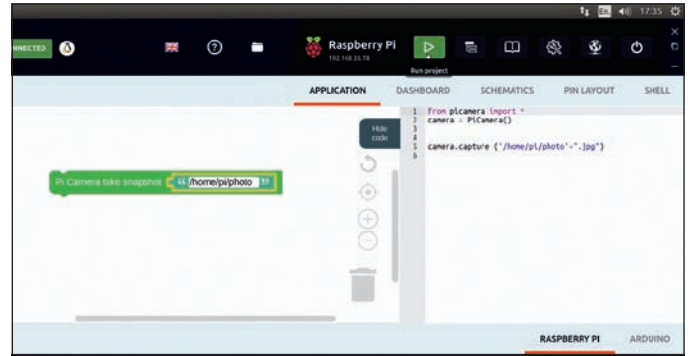
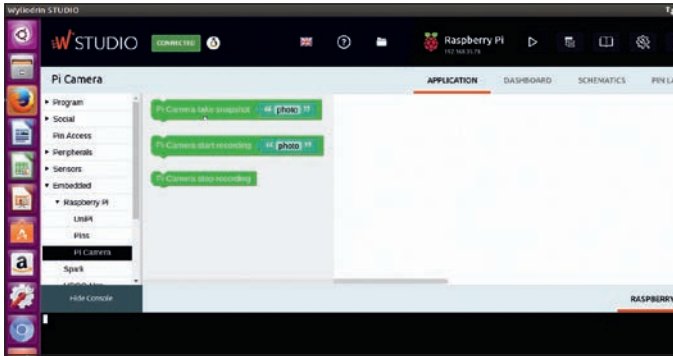
```
$ python script.py
```

Или можете сделать файл исполняемым и запустить его напрямую:

```
$ chmod +x script.py
$ ./script.py
```

Учтите: между *Wyliodrin* в вашем браузере и *Wyliodrin STUDIO* нет прямой связи — если вы хотите передать скрипт Python от одного к другому, простейшим решением будет, опять же, скопировать и вставить код между ними.

Создаем проект с визуальным программированием

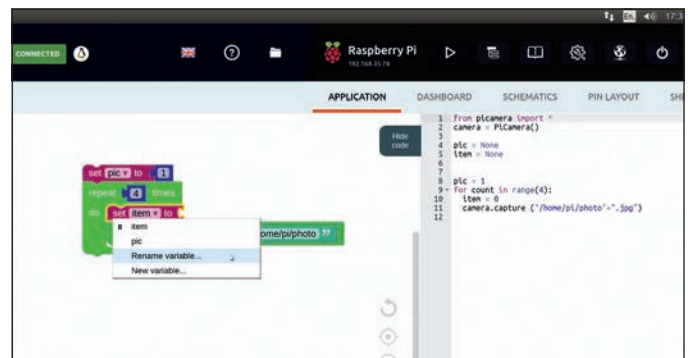
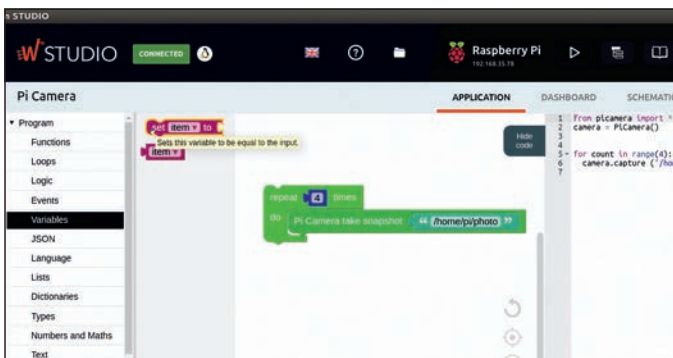


1 Добавим блок камеры

Строительные блоки визуального программирования разделены на секции слева: нажмите Expand Embedded [Раскрыть встроенное], Raspberry Pi и Pi Camera. Появится всплывающее меню, показывающее три доступных строительных блока. Нам нужен первый блок, так что перетащите его в середину экрана. Нажмите на вкладку Show code [Показать код], чтобы просмотреть лежащий в основе код Python.

2 Изменим блок кода

Вы увидите, что блок состоит из двух элементов: команды и фрагментов кода. Вас будет одолевать искушение нажать Play, но код в его текущем виде не работает; вам нужно щелкнуть внутрь строковой переменной, отмеченной как 'photo', и изменить ее на путь — например, /home/pi/photo. После этого нажмите на значок воспроизведения, а затем воспользуйтесь File Explorer, с целью удостовериться, что фото было сделано.

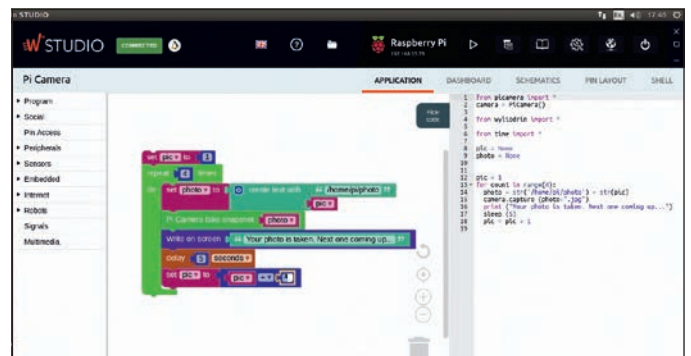
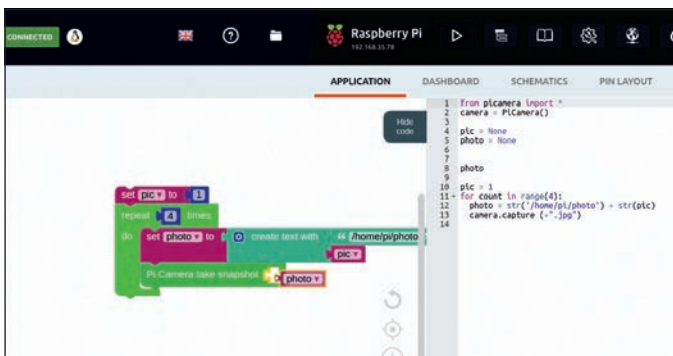


3 Добавим цикл

Добавим в скрипт цикл, чтобы он делал четыре фото (photo1.jpg, photo2.jpg и т.д.) с интервалом в 5 секунд. Начните, выбрав Program > Loops [Программа > Циклы], и перетащите блок Repeat 10 times [Повторить 10 раз], окружив исходный блок. Пощелкайте по 10, чтобы уменьшить это число. Выберите Program > Variables [Программа > Переменные] и перетащите set item to [установить значение в] поверх блока Repeat.

4 Зададим переменные

Нажмите item [объект] и выберите Rename variable [Переименовать переменную]. Смените ее имя на 'pic'. Выберите Program > Numbers and Maths [... > Числа и математика] и перетащите верхнее число так, чтобы оно встало справа от set pic to [установить картинку в]. Измените его на 1. Теперь перетащите еще один блок переменной set item to, чтобы он шел поверх Pi Camera takes snapshot [Pi Camera делает фото] внутри переменной повторяющегося цикла. Переименуйте переменную в 'photo'.



5 Завершим переменную имени файла [filename]

Разверните Program > Text [... > Текст] и перетащите create text with [создать текст с], чтобы прикрепить его к блоку set photo to [установить фото в]. У него есть два входа — перетащите строковую переменную из Pi Camera takes snapshot в верхний блок, затем перейдите в Program > Variables и перетащите переменную 'pic' под строковую переменную. Наконец, перетащите переменную 'photo' из Program > Variables в Pi Camera takes snapshot.

6 Последние штрихи

Используйте Program > Screen and Keyboard [... > Экран и клавиатура] для обеспечения обратной связи при создании фото, затем введите пятисекундную задержку в Program > Timing [... > Синхронизация]. Теперь добавьте set photo to из Program > Variables и прикрепите блок '+' из Program > Numbers and Maths. Перетащите переменную 'pic' во вход слева и перетащите блок номеров [number] в другой, задав его как 1.

CRIU: Тестируем

Павел Емельянов привык не доверять, а проверять, и обрушивает на CRIU серию микротестов.



Наш эксперт

Павел Емельянов — архитектор в команде Виртуозо, идейный вдохновитель проекта CRIU. В компании с 2004 г., начал в группе разработки ядра Linux, которую затем возглавил. Сейчас занимается архитектурой продукта и открытиями разработками компании.

В заключительном уроке мы рассмотрим своеобразный переход от простого использования CRIU к его разработке, а именно — подсистему тестирования. Частично мы эту тему уже затрагивали, теперь же рассмотрим вопрос подробнее.

Ядро тестовой подсистемы

Поскольку основной задачей CRIU является сохранение и восстановление состояния процессов, то как бы мы CRIU ни тестировали, в тестах обязательно будет присутствовать компонент, который запускает процесс, сохраняет его состояние и восстанавливает его состояние. Такой компонент появился в проекте со дня его основания и по историческим причинам носит аббревиатурное название ZDTM (что означает Zero Down-Time Migration, поскольку создавался он главным образом для тестирования живой миграции). Располагаются все тесты в поддиректории `test/`, а «ядро» представлено скриптом `test/zdtm.py`.

Вместе с ядром идет набор (чуть меньше 300 штук) миниатюрных программ, каждая из которых при запуске осуществляет одно небольшое действие, которое может выполнить процесс, работающий на Linux: например, открывает файл или сокет, создает кусок памяти, отправляет само себе сигнал, создает дочерний процесс и т. п. Этот набор уже стал неотъемлемой частью тестовой подсистемы и уже тоже может считаться частью ее ядра.

Базовое тестирование

В простейшем варианте тестируется самая базовая вещь — запуск микротеста, сохранение его состояния и восстановление. После восстановления необходимо как-то проверить, что состояние восстановлено корректно. Эта задача возложена на сами микротесты — после восстановления процессу с микротестом отправляется сигнал, после чего он должен проверить свое состояние и выйти с соответствующим кодом возврата. Запуск тестов в таком варианте осуществляется командой `# test/zdtm.py run --all`.

Схематически последовательность шагов для одного микротеста показана на рис. 1.



Рис. 1. Микротест.

Дополнительные тесты

Помимо этого, на `zdtm.py` возложена задача проведения гораздо большего количества проверок.

Тестирование тестов

Например, с помощью `zdtm.py` можно проверить работоспособность самого микротеста. Для этого тест запускается, после чего ему сразу отправляется сигнал для проверки своего состояния. При этом рабочий тест должен выйти без ошибок, ведь в его состоянии никто не вмешивался. Запускается этот режим командой

```
# test/zdtm.py run --all --nocr
```

Неразрушающее сохранение

Одним из базовых требований, которые мы предъявляли к проекту, было требование о том, что в процессе сохранения состояния

исследуемый процесс не должен «разрушаться», а его состояние не должно никаким образом изменяться или, тем более, «портиться». Для проверки того, что CRIU действительно ведет себя таким образом, `zdtm.py` запускает микротесты, сохраняет их состояние, но велит CRIU после сохранения процессы не убивать, а разрешить им исполняться дальше. После чего микротест останавливается, проверяя свое состояние. Для запуска тестов в таком режиме используется команда `# test/zdtm.py run --all --norst`.

Повторное сохранение состояний

Поскольку CRIU восстанавливает процесс так, что он *выглядит* для самого процесса таким же образом, каким он выглядел до этой экзекуции; может сложиться ситуация, что в восстановленный процесс находится в таком состоянии, которое невозможно сохранить повторно. Либо же, сохраненное и восстановленное второй (третьей, четвертой или больше) раз, состояние уже начинает изменяться. Для проверки многократных сохранений и восстановлений `zdtm.py` может запустить команды `dump` и `restore` несколько раз, прежде чем останавливать тест. Делается это командой

```
# test/zdtm.py run --all --iter N[:delay]
```

где `N` — количество требуемых итераций, а необязательный параметр `delay` указывает, какую паузу надо выдержать перед повторным сохранением состояния. Паузу имеет смысл делать для тестов, которые меняют свое состояние в процессе работы — среди микротестов есть и такие.

Тестирование живой миграции

Непосредственно миграцию `zdtm.py` не осуществляет, но дает возможность проверить, что требуемая функциональность в CRIU работает.

Пред-сохранение состояния

Основной функциональностью для поддержки живой миграции является возможность предварительного снятия состояний с процессов без их остановки, что реализовано в команде CRIU под названием `pre-dump`. ZDTM умеет тестировать и эту подсистему — после запуска микротеста с него снимается несколько пре-дампов, после чего делается финальный дамп, потом восстановление и останов теста. При этом ZDTM сам осуществляет управление директориями с образцами. Запуск такого режима выглядит следующим образом

```
# test/zdtm.py run --all --pre N[:delay]
```

Аргумент опции `pre` работает так же, как и с опцией `iters`, то есть указывается количество пре-дампов и задержка между ними.

Пред-сохранение полных состояний

В показанном выше примере CRIU осуществляет несколько последовательных пре-дампов, то есть пред-снимает часть состояний процессов (конкретно — только содержимое памяти). Но CRIU умеет также пред-снимать и полные состояния, делая полноценные снимки, или `snapshot`'ы. Для этого надо всего лишь пользоваться командой `dump`, вместо `pre-dump`, а параметры отслеживания изменений памяти (`--track-mem` и `--prev-images-dir`) оставить такими же. Система ZDTM тестирует и такое поведение.

```
# test/zdtm.py run --all --pre N[:delay] --snaps
```

Кроме того, указанные выше опции можно комбинировать, тестируя, например, сохранность процессов после снятия двух последовательных состояний и т. д.

» Тестирование page-server

Как было показано в одном из предыдущих уроков, при живой миграции память процессов можно сразу пересылать на принимающий сервер. Для этого надо на принимающей стороне запустить команду `page-server`, а командам `dump` или `pre-dump` указать его адрес. Система `ZDTM` тестирует такое поведение, запуская `page-server` и «отправляя» память процессов на адрес `127.0.0.1`. Такой режим прогона тестов запускается командой

```
# test/zdtm.py run --all --page-server
```

и тоже может быть скомбинирован с опциями выше. Например, чтобы протестировать почти настоящую живую миграцию, нужно указать следующий набор опций:

```
# test/zdtm.py run --all --pre N --page-server
```

В этом случае с тестов снимут несколько пре-дампов подряд (без задержки) и перешлют их сразу на `page-server`. Так же можно протестировать «отказ» миграции и корректное восстановление после этого:

```
# test/zdtm.py run --all --pre N --page-server --norst
```

Тестирование окружений

Некоторые состояния выглядят, снимаются и восстанавливаются по-разному, в зависимости от того, в каком окружении находится процесс. `CRIU` различает четыре таких типа окружений:

» Контекст не-суперпользователя

» Окружение хост-системы

» Окружение контейнера

» Окружение безопасного контейнера

Во-первых, основной сценарий использования `CRIU` — это сценарий, когда `CRIU` запущен с правами супер-пользователя (`root`). При этом `CRIU` доступны без ограничений любые данные о процессах (при условии, что в ядре вообще есть соответствующий интерфейс). По умолчанию `ZDTM` проводит тестирование в этом предположении. Но в `CRIU` также есть ограниченная поддержка работы без прав `root`. В этом случае на снимаемые состояния накладывается ряд ограничений (например, процесс-жертва должен быть запущен от имени того же пользователя). Протестировать данный режим работы можно так:

```
# test/zdtm.py run --all --user
```

Если `CRIU` работает с полным набором прав, то различаются три возможных окружения, в которых может работать процесс-жертва. В простейшем случае этот процесс запущен *не* в контейнере, то есть живет в тех же пространствах имен (`namespaces`), что и процесс `CRIU`. Во втором случае процесс запущен в своих собственных пространствах `PID`, `NET` и `MOUNT`, а в третьем случае — еще и в отдельном пространстве имен пользователя (`USER namespace`).

В последних двух случаях снятие состояний и их восстановление работает с особенностями: например, для других `MOUNT namespace` по-разному вычисляются пути открытых файлов; или, в случае `USER namespace`, в процессе восстановления часть привилегированных операций выполняются процессом `CRIU`, а не восстанавливаемым процессом.

Все три сценария тестируются отдельно, причем каждый микротест может дополнительно сообщать системе, в каком из окружений он может или не может быть выполнен. Выбор окружения осуществляется так:

```
# test/zdtm.py run --all --flavor {h|ns|uns}
```

где `h`, `ns` и `uns` соответственно означают окружения хоста, контейнера и безопасного контейнера.

Что еще можно протестировать

Помимо вышеупомянутого, `ZDTM` тестирует еще несколько сценариев использования `CRIU`.

» Восстановление «брата»

В одном из предыдущих уроков было показано, что восстановить поддерево процессов можно либо как дочернее дерево `CRIU`, либо

как дочернее дерево вызывающего процесса. Последний вариант активируется опцией `CRIU --restore-sibling` и тестируется так:

```
# test/zdtm.py run --all --sibling
```

» Восстановление в остановленном состоянии

После восстановления процессы сразу запускаются и продолжают исполняться. `ZDTM` может проверить, как `CRIU` восстанавливает процессы остановленными, и вот как это делается:

```
# test/zdtm.py run --all --stop
```

» Заморозка `cgroup`

Для заморозки процессов `CRIU` использует два механизма — остановка «отладчиком» или заморозка с помощью штатного ядерного «холодильника» (`freeze cgroup`). По умолчанию `ZDTM` тестирует первый вариант, второй же тестируется следующим образом:

```
# test/zdtm.py run --all --freezecg PATH:STATE
```

При этом возможны два варианта теста — с предварительно замороженной или предварительно размороженной `cgroup`.

» Проверка `RPC`

По умолчанию `ZDTM` запускает `CRIU` как утилиту командной строки. Но в `CRIU` присутствует и другой API, а именно `RPC`, при котором команды и опции задаются `CRIU` в виде сообщений, посылаемых через сокет. Такой режим работы тестируется командой

```
# test/zdtm.py run --all --rpc
```

Прочие опции

Все упомянутые выше опции используются в системе автоматического тестирования. Для удобства использования «руками» в `zdtm.py` есть еще ряд опций. Рассмотрим их.

» Опция `--parallel N` задает максимальный размер параллельно запущенных микротестов.

» Опция `--keep-img {mode}` говорит, в каких случаях сохранять промежуточные файлы с образцами: всегда (`always`) или никогда (`never`). По умолчанию образцы успешных прогонов удаляются сразу, а неуспешных — сохраняются.

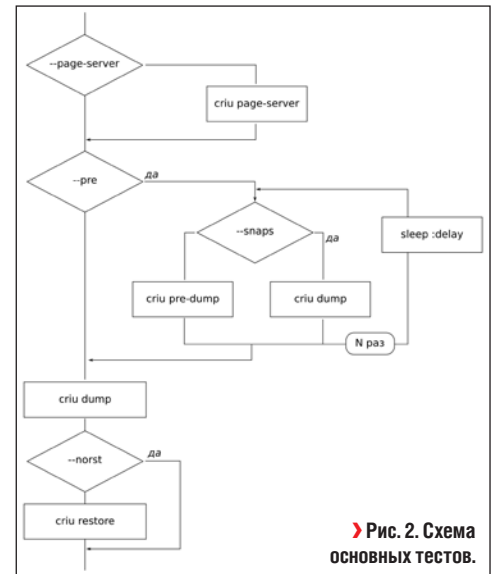
» Опция `--keep-going` говорит `ZDTM`, что после неуспешного теста надо продолжать тестирование и в конце запуска показать сводку. По умолчанию `ZDTM` останавливается после первой неудачи.

» Опция `--fault N` тестирует поведение `CRIU` при искусственном внедрении ошибок в работу.

Заключение

Помимо упомянутой подсистемы `ZDTM`, в директории `test/` присутствует ряд других тестов; но, как показала практика, любое тестирование, не основанное на микротестах, не дает уверенности в том, что протестированная подсистема останется работоспособной и в дальнейшем.

Некоторое время назад в тестовой подсистеме появился набор т. н. `exhaustive` [исчерпывающих] тестов. Это набор скриптов, которые *генерируют* все возможные состояния определенной подсистемы полным перебором возможных действий, вместо того чтобы полагаться на несколько микротестов, написанных вручную. В дальнейшем эта подсистема будет также интегрирована в `ZDTM`, чтобы иметь возможность не только генерировать состояния, но и проверять их поддержку всеми указанными выше способами. **LXF**



» Рис. 2. Схема основных тестов.

Python: Делаем БД SQLite 3

Михалис Цукалос показывает, как настроить базу данных *SQLite 3* для работы с серьезными данными на языке программирования Python.



Наш эксперт

Михалис Цукалос (@mactsouk) имеет степень магистра в информационных технологиях от Университетского колледжа Лондона и степень бакалавра по математике. Он администрирует базы данных, пишет программы и работает в Unix как сущая математическая машина. Вы можете связаться с ним через сайт www.mtsoukalos.eu.

На этом уроке используются Python 3.4.x и Python 3.5.x, но представленный код будет работать и в более старой версии Python 2.7.x. Полный путь к исполняемому файлу Python 3, который используется на уроке — `/usr/local/bin/python3` или `/usr/bin/python3`. Если в вашей системе Linux другой путь, измените скрипты соответствующим образом. СУБД *SQLite* имеет пару основных преимуществ: во-первых, для ее работы не нужен серверный процесс, а во-вторых, вся база данных содержится в одном файле, который включает все необходимые данные. *SQLite* также отличается высокой производительностью, надежностью, мобильностью, простотой и низким потреблением памяти, и поэтому может использоваться на многих устройствах, включая мобильные телефоны. Доступ к ней — вы удивитесь — можно получить с помощью языка SQL.

SQLite не содержит средств для аутентификации или авторизации. Для определения трех поддерживаемых *SQLite* уровней доступа (на чтение и запись, на чтение и на отсутствие доступа) используются обычные права доступа к файлам в Unix, определяемые командой `chmod`. Кроме того, *SQLite* не подходит для очень больших наборов данных, даже несмотря на то, что современные файловые системы поддерживают файлы размером более 1 ТБ. И последнее: *SQLite* не поддерживает репликацию — резервную копию базы данных можно создать, просто скопировав файл базы данных!

Для запуска *SQLite 3* на своем компьютере введите `sqlite3` в командной строке. Если *SQLite* не установлена, возможно, потребуется установить ее через менеджер пакетов.

Чтобы получить более подробную информацию об активной базе данных *SQLite 3*, можно заглянуть в таблицу `sqlite_master`, которая содержит ключевую информацию о таблицах базы данных:

```
sqlite> .schema sqlite_master
CREATE TABLE sqlite_master (
  type text,
```

```
name text,
tbl_name text,
rootpage integer,
sql text
);
sqlite> PRAGMA TABLE_INFO(sqlite_master);
0|type|text|0|1|0
1|name|text|0|1|0
2|tbl_name|text|0|1|0
3|rootpage|integer|0|1|0
4|sql|text|0|1|0
```

Первая команда выводит схему таблицы `sqlite_master`, а вторая команда отображает эту же схему в другом формате. Схема таблицы определяет столбцы таблицы и их атрибуты. Схема базы данных представляет собой способ визуализации таблиц и их связей.

Подключение к БД SQLite 3

Следующий код Python представляет собой автономный пример, который показывает, как подключиться к *SQLite 3* и выполнить запрос `SELECT`:

```
#!/usr/local/bin/python3
import sqlite3
conn = sqlite3.connect('LXF.sqlite')
c = conn.execute("SELECT count(*) from sqlite_master")
for row in c:
    print(row)
conn.close()
```

Как видно из запроса к базе данных, мы обращаемся к таблице `sqlite_master`. Вывод будет выглядеть примерно так:

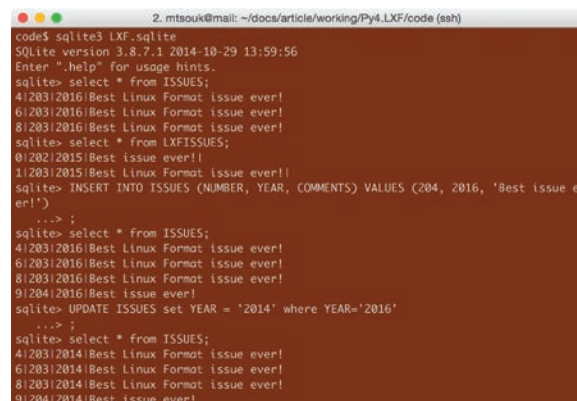
```
$. /connect.py
(0,)
```

Модуль `sqlite3` используется для связи с базой данных *SQLite 3*. Затем нужно определить базу данных, к которой вы хотите подключиться, указав имя файла базы данных. Для выполнения SQL-запросов вам надо будет создать курсор — представленный запрос считывает таблицу базы данных и возвращает количество записей в таблице. Обратите внимание, что если файла базы данных `LXF.sqlite` не существует, он создается автоматически; и если вы допустите орфографическую ошибку, результаты могут оказаться странными. Кроме того, если требуемый файл базы данных находится не в текущем рабочем каталоге, следует указать полный путь к нему.

Вышеуказанный код будет широко использоваться на нашем уроке, поэтому выполните его, прежде чем продолжить, и убедитесь, что всё работает.

Кроме того, в следующем выводе показано взаимодействие с оболочкой Python, в которой команды выводят сообщения об ошибках:

➤ На этом скриншоте приведены примеры базовых операций CRUD в базе данных `LXF.sqlite`.



Поддерживаемые типы данных

SQLite 3 поддерживает следующие типы данных:

- » **INTEGER** Этот тип представляет целые числа длиной до 8 байт в зависимости от значения.
- » **REAL** Этот тип представляет числа с плавающей точкой длиной 8 байт.
- » **TEXT** Тип данных TEXT представляет текстовые строки, которые хранятся в кодировке базы данных. В большинстве случаев это UTF-8.

» **Большой двоичный объект (BLOB)** Этот тип данных используется для хранения двоичных данных, таких как изображения и сжатые файлы.

» **NULL** Значение NULL используется для представления отсутствующих значений или просто пустых ячеек.

Логический тип данных в SQLite 3 отсутствует, но вместо него можно обойтись типом INTEGER.

Кроме того, SQLite 3 использует типы данных INTEGER, REAL и TEXT для хранения даты и времени: например, время Unix хранится с помощью типа INTEGER.

Встроенные функции, используемые для работы с датой и временем, автоматически выполняют преобразование между различными форматами данных.

```
>>> c = conn.execute("SELECT count(*) from sqlite")
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
sqlite3.OperationalError: no such table: sqlite [Таблицы не существует]
>>> conn.close()
>>> c = conn.execute("SELECT count(*) from sqlite_master")
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
sqlite3.ProgrammingError: Cannot operate on a closed database.
[Нельзя работать с неоткрытой таблицей]
```

Первая команда показывает, что если таблицы, к которой вы попытались обратиться, не существует, программа завершится с ошибкой. Последняя команда показывает, что если вы случайно закроете соединение с базой данных и попытаетесь обратиться к нему снова, скрипт завершится с ошибкой.

Создание таблицы

Так как самое первое выполняемое вами действие — это создание новой таблицы, в данном разделе будут показаны различные способы создания таблиц в SQLite 3:

```
$ sqlite3 LXF.sqlite
SQLite version 3.8.7.1 2014-10-29 13:59:56
Enter ".help" for usage hints. [Введите Help для подсказок]
sqlite> CREATE TABLE LXFISSUES (
ID INT PRIMARY KEY NOT NULL,
NUMBER INT NOT NULL,
YEAR INT NOT NULL,
COMMENTS CHAR(50)
);
sqlite> CREATE TABLE ISSUES (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
NUMBER INT NOT NULL,
YEAR INT NOT NULL,
COMMENTS CHAR(50)
);
```

Создание первичного ключа очень важно по соображениям производительности, поэтому выбирайте его осмотрительно. Ключевое слово **AUTOINCREMENT** бывает очень удобным, особенно если вы хотите создать уникальные поля идентификаторов — как вы догадываетесь, при использовании этого ключевого слова значение поля указывать не нужно, так как оно заполняется SQLite 3. Использование ключевого слова **AUTOINCREMENT** допускается только для полей **INTEGER PRIMARY KEY**.

Основное различие между таблицами LXFISSUES и ISSUES заключается в том, что при вставке записи в LXFISSUES потребуются указать четыре значения, тогда как для ISSUES требуются всего три, потому что поле ID автоматически заполняется SQLite 3.

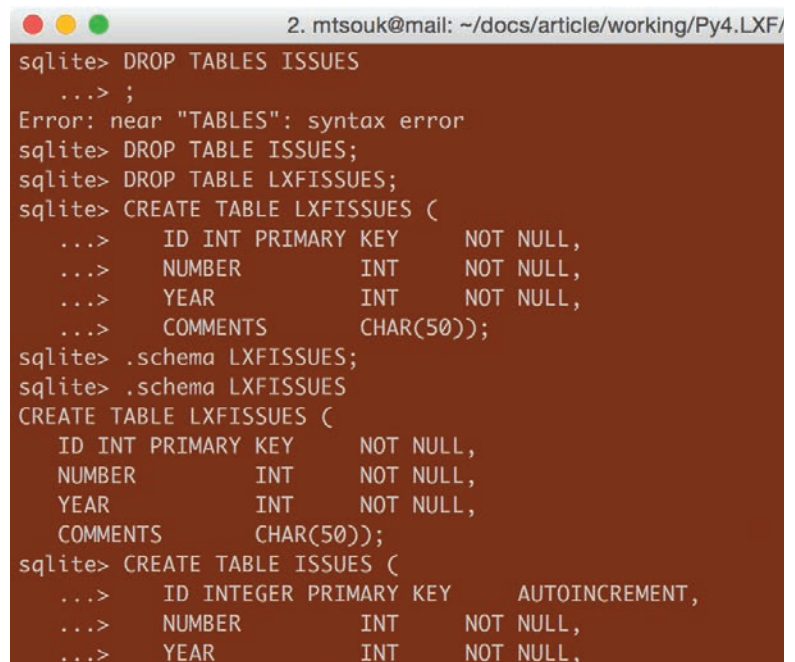
Еще одна очень удобная команда SQL — ALTER TABLE. Она позволяет добавлять новые столбцы в существующую таблицу базы данных:

```
sqlite> ALTER TABLE LXFISSUES ADD COLUMN EDITOR
CHAR(100);
sqlite> .schema LXFISSUES
CREATE TABLE LXFISSUES (
ID INT PRIMARY KEY NOT NULL,
NUMBER INT NOT NULL,
YEAR INT NOT NULL,
COMMENTS CHAR(50),
EDITOR CHAR(100));
```

В скрипте **alter.py** показано, как выполнить те же действия с помощью Python и модуля sqlite3:

```
conn.execute("""DROP TABLE LXFISSUES;""")
conn.execute("""DROP TABLE ISSUES;""")
conn.execute("""CREATE TABLE LXFISSUES (
ID INT PRIMARY KEY NOT NULL,
NUMBER INT NOT NULL,
YEAR INT NOT NULL,
COMMENTS CHAR(50));""")
conn.execute("""CREATE TABLE ISSUES (
ID INTEGER PRIMARY KEY AUTOINCREMENT,
```

»



» Здесь показано, как создать таблицу в SQLite 3, а также как изменить существующую таблицу, добавив в нее новое поле.

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



```
NUMBER INT NOT NULL,
YEAR INT NOT NULL,
COMMENTS CHAR(50);""
conn.execute("""ALTER TABLE LXFISSUES ADD COLUMN EDITOR
CHAR(100);""")
```

Как видите, используются те же команды, что и раньше, встроенные в функцию `conn.execute()`. Функции `execute()`, используемой для создания и изменения таблицы, не требуется курсор, поскольку она не возвращает выходных данных. Перед попыткой создания таблицы часто используется команда `DROP` для удаления таблицы. Однако если таблицы не существует, команда `DROP` завершится с ошибкой. (Скриншот с различными операциями по работе с таблицами в оболочке *SQLite 3* вы можете видеть на стр. 80 внизу слева.)

Использование драйвера Python

Теперь вы знаете, как подключиться к базе данных *SQLite 3* и создать таблицу с помощью Python; пора научиться выполнять другие основные операции — вставку, выбор, обновление и удаление данных — с помощью модуля `sqlite3`. Как вы видели в файле `connect.py`, для чтения результатов запроса `SELECT` надо выполнять итерацию курсора.

Код Python в файле `insert.py` таков:

```
conn.execute("INSERT INTO ISSUES (NUMBER, YEAR,
COMMENTS) VALUES (202, 2015, 'Самый лучший выпуск!') ")
conn.execute("INSERT INTO ISSUES (NUMBER, YEAR,
COMMENTS) VALUES (203, 2015, 'Самый лучший выпуск
Linux Format!') ")
conn.commit()
```

```
conn.execute("INSERT INTO LXFISSUES (ID, NUMBER, YEAR,
COMMENTS) VALUES (0, 202, 2015, 'Самый лучший выпуск!') ")
conn.execute("INSERT INTO LXFISSUES (ID, NUMBER, YEAR,
COMMENTS) VALUES (1, 203, 2015, 'Самый лучший выпуск
Linux Format!') ")
conn.commit()
```

Если попробовать запустить файл `insert.py` еще раз, появится сообщение об ошибке из-за ограничений таблицы. Запись с уже существующим полем идентификатора вставить нельзя, потому что идентификатор является первичным ключом и должен быть уникальным. К таблице `ISSUES` это не относится, поскольку в ней используется поле `AUTOINCREMENT`. Для фактической отправки записей в таблицу используется команда `conn.commit()`.

Код Python в файле `fetch.py`, демонстрирующем использование команды `SELECT`, таков:

```
c = conn.execute("SELECT * from ISSUES;")
for record in c:
    print(record)
```

Как вы понимаете, количество записей, которое возвращает оператор `SELECT`, заранее неизвестно, поэтому для получения всех результатов нужно выполнять итерацию курсора.

Код Python в файле `delete.py` выглядит так:

```
conn.execute("DELETE FROM ISSUES where ID='2'")
conn.commit()
print ("Общее число удаленных строк :", conn.total_changes)
```

Опять же, команда `conn.commit()` является обязательной. Если команда `DELETE` не обнаружит ни одной записи, вы не получите ошибок или предупреждений. Однако с помощью свойства `conn.total_changes` вы сможете понять, что произошло.

В следующем коде, который сохранен в файле `update.py`, вы увидите, как выполнять обновление в *SQLite 3* с помощью Python:

```
conn.execute("UPDATE ISSUES set YEAR = '2016' where
YEAR='2015'")
conn.commit()
print ("Общее число измененных строк :", conn.total_changes)
```

(Примечание: Четыре скрипта Python в действии показаны на рис. слева.)

В следующем коде, который сохранен в файле `tableNames.py`, показано, как получить имена столбцов для заданной таблицы:

```
import sys
# Первый аргумент - имя базы данных
database = sys.argv[1]
# Второй аргумент - имя таблицы
tableName = sys.argv[2]
conn = sqlite3.connect(database)
c = conn.cursor()
query = "select * from " + tableName + " where 1=0"
c.execute(query)
```

```
rMacBook:code mtsouk$ ./insert.py
Traceback (most recent call last):
  File "./insert.py", line 10, in <module>
    conn.execute("INSERT INTO LXFISSUES (ID, NUMBER, YEAR, COMMENTS) VAL
2015, 'Best issue ever!') ")
sqlite3.IntegrityError: UNIQUE constraint failed: LXFISSUES.ID
rMacBook:code mtsouk$ ./fetch.py
(1, 202, 2015, 'Best issue ever!')
(2, 203, 2015, 'Best Linux Format issue ever!')
(3, 202, 2015, 'Best issue ever!')
(4, 203, 2015, 'Best Linux Format issue ever!')
rMacBook:code mtsouk$ ./update.py
Total number of rows updated : 4
rMacBook:code mtsouk$ ./fetch.py
(1, 202, 2016, 'Best issue ever!')
(2, 203, 2016, 'Best Linux Format issue ever!')
(3, 202, 2016, 'Best issue ever!')
(4, 203, 2016, 'Best Linux Format issue ever!')
rMacBook:code mtsouk$ ./delete.py
Total number of rows deleted : 2
```

➤ Здесь показаны в действии различные скрипты Python, разработанные для этого урока. Некоторые из них могут выдавать ошибки при повторном запуске.

Язык SQL

CRUD — сокращение от "Create, Read, Update and Delete" (создание, чтение, обновление и удаление). Это основные операции, которые допускается выполнять над базами данных, в том числе и базами данных *SQLite 3*.

Читая это руководство, следует держать в голове базовые операции CRUD, чтобы вы могли проверить, что делают скрипты Python.

Следующие команды SQL иллюстрируют четыре базовые операции:

```
sqlite> INSERT INTO ISSUES (NUMBER, YEAR,
COMMENTS) VALUES (204, 2016, 'Самый
лучший выпуск!');
sqlite> SELECT * FROM ISSUES;
sqlite> UPDATE ISSUES set YEAR = '2014' where
YEAR='2016';
```

```
sqlite> DELETE FROM ISSUES where ID='8';
```

На рисунке (на стр. 81) показан вывод предыдущих операций для примера базы данных. Чтобы найти все таблицы в базе данных *SQLite 3*, выполните команду `.tables` из оболочки *SQLite 3*. Если всё нормально, результат будет примерно таким:

```
sqlite> .tables
ISSUES LXFISSUES
```

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

```
# Будет пустым из-за запроса
rs = c.fetchall()
field_names = [r[0] for r in c.description]
for f in field_names:
    print("**", f)
```

Возьмите этот код на заметку — похожий код скорее всего будет использоваться в графическом приложении, которое вы приметесь разрабатывать.

Название таблицы, столбцы для которой нужно получить, передается в качестве аргумента командной строки. Выполнение скрипта `tableNames.py` даст следующий результат:

```
$. /tableNames.py LXF.sqlite ISSUES
* ID
* NUMBER
* YEAR
* COMMENTS
```

Показанному приложению понадобится меню для отображения доступных таблиц базы данных. Для создания этого меню стоит выбрать класс `OptionMenu`, который создает всплывающее меню и кнопку для его отображения. Список доступных таблиц надо поместить в словарную структуру данных и передать его в качестве параметра класса `OptionMenu`. Чтобы получить текущее значение `OptionMenu`, можно использовать метод `get()` с переменной, которая использовалась при определении класса `OptionMenu`.

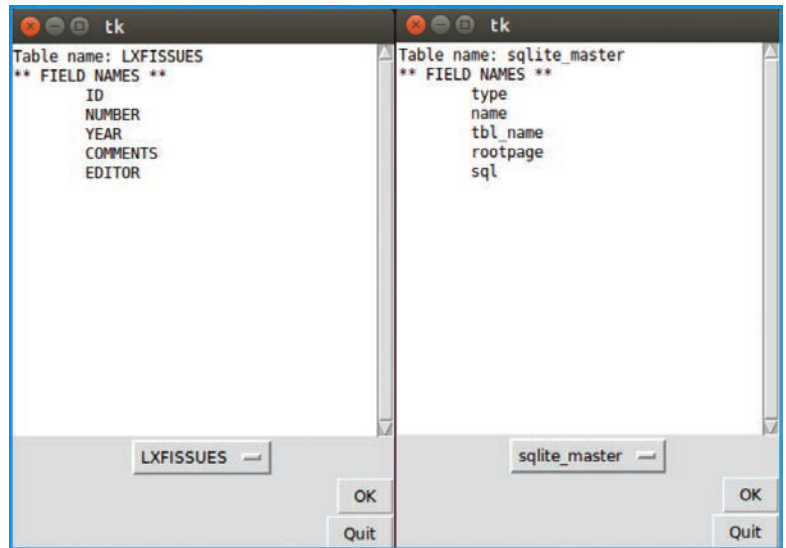
Про Tkinter

Для отображения необходимого текста можно использовать удобный класс `ScrolledText`. Для удаления текущих данных класса `ScrolledText` нужно использовать метод `delete()`, а для печати новых данных — метод `insert()`. Как вы, вероятно, согласитесь, логика приложения очень проста, но очень эффективна.

Представленное приложение прочтет файл базы данных и названия всех таблиц для этого файла. После этого вы сможете выбрать таблицу из списка для отображения ее полей. Файл базы данных передается в качестве аргумента командной строки для скрипта Python. Первоначальная версия приложения сохранена в файле `gui.py`:

```
#!/usr/bin/python3
import sqlite3
import tkinter
from tkinter import *
import tkinter.scrolledtext as ST
import sys

def callBack():
    print ("Значение равно", variable.get())
    text.delete('1.0', END)
    myText = "Имя таблицы: " + variable.get()
    text.insert('вставка', myText)
# Первый аргумент - имя базы данных
database = sys.argv[1]
conn = sqlite3.connect(database)
c = conn.cursor()
root=Tk()
canvas=Canvas(root,width=810,height=600)
canvas.pack()
text = ST.ScrolledText(canvas, width=35, height=20,
borderwidth=0)
text.pack()
# Это учебный список таблиц
listOfTables = {"sqlite_master",
"one",
"two",
"three"}
variable = StringVar(root)
variable.set("sqlite_master") # Значение по умолчанию
```



```
w = OptionMenu(root, variable, *listOfTables)
w.pack()
Button(root, text='Quit', command=root.quit).pack(side=BOTTOM,
anchor=SE)
Button(root, text='OK', command=callBack).pack(side=TOP,
anchor=SE)
mainloop()
conn.close()
```

➤ Здесь показано приложение, использующее базу LXF.sqlite, в действии.

Окончательная версия

Рабочая версия приложения называется `finalGUI.py`. В следующем коде показана реализация функции `callBack()`, так как именно в ней происходит основное действие:

```
def callBack():
    print ("Значение равно", variable.get())
    text.delete('1.0', END)
    tableName = variable.get()
    myText = "Имя таблицы: " + tableName
    text.insert('вставка', myText + '\n')
    text.insert('вставка', "*** FIELD NAMES **\n")
    # Получаем требуемые данные!
    query = "select * from " + tableName + " where 1=0"
    c.execute(query)
    # Будет пустым из-за запроса
    rs = c.fetchall()
    field_names = [r[0] for r in c.description]
    for f in field_names:
        text.insert('insert', '\t' + f + '\n')
```

Для доступа к названиям таблиц базы данных *SQLite 3* из таблицы `sqlite_master` надо скомандовать `SELECT name FROM sqlite_master WHERE type='table'`;

В следующем коде Python показано, как переменная `listOfTables` заполняется с помощью предыдущей команды `SELECT`:

```
listOfTables = {}
listOfTables['sqlite_master'] = 0
c.execute("SELECT name FROM sqlite_master WHERE
type='table';")
for record in c:
    # print(record[0])
    listOfTables[record[0]] = 0
```

(На скриншоте вверху страницы показан скрипт `finalGUI.py` в действии.)

Желая узнать о модуле `sqlite3` подробнее, зайдите на страницу <https://docs.python.org/3/library/sqlite3.html>. Официальный сайт *SQLite 3* — www.sqlite.org. **LXF**

Python: Обработка текста

«Слова, кругом слова!» — восклицает **Михалис Цукалос**, знакомя вас с тем, что необходимо для обработки текста в Python.



Наш эксперт

Михалис Цукалос (@mactsouk) имеет степень магистра в информационных технологиях от Университетского колледжа Лондона и степень бакалавра по математике. Он администрирует базы данных, пишет программы и работает в Unix как сущая математическая машина. Вы можете связаться с ним через сайт www.mtsoukalos.eu.



Автоматическая обработка текста может сэкономить время и усилия. Поэтому давайте научимся эффективно работать с текстовыми файлами, освоив основы обработки текста в Python, включая поиск и замену с помощью регулярных выражений и преобразование формата даты в другой формат, и разработаем графический интерфейс, чтобы упростить себе жизнь.

Как вы, возможно, знаете, сейчас имеют хождение две версии Python. В этом руководстве используется более «старая» версия (Python 2.7.x), но если вы решите воспользоваться версией 3, никаких проблем не будет.

В следующем коде Python, который сохранен в файле **lbl.py**, показано, как обработать текстовый файл строка за строкой, что составляет основу обработки текста:

```
try:
    f = open(filename, 'r')
except IOError:
    print "Файл %s не открывается!" % filename
    raise SystemExit
for line in f:
```

```
2. mtsouk@mail:~/docs/article/working/PyLXF/code (ssh)
code$ ll characters.py lines.py words.py wcPython.py
-rwxr-xr-x 1 mtsouk mtsouk 534 Mar  2 22:33 characters.py
-rwxr-xr-x 1 mtsouk mtsouk 522 Mar  2 22:29 lines.py
-rwxr-xr-x 1 mtsouk mtsouk 852 Mar  2 22:33 wcPython.py
-rwxr-xr-x 1 mtsouk mtsouk 556 Mar  2 22:29 words.py
code$ ./characters.py characters.py
Number of Characters: 534
code$ ./lines.py characters.py
Number of Lines: 29
code$ ./words.py characters.py
Number of Words: 75
code$ ./wcPython.py characters.py
 29  75  534
code$ wc characters.py
 28  75 534 characters.py
code$ wc wcPython.py
 46 116 852 wcPython.py
code$ ./l1.py wcPython.py
 46 116 852
code$ ./l1.py words.py
#!/usr/bin/python
# Programmer: Michalis Tsoukalos
#
# Description: Count number of words on a text file
from sys import exit
if __name__ == '__main__':
    import sys
```

➤ Здесь мы видим все скрипты Python (**l1.py**, **characters.py**, **words.py**, **lines.py** и **wcPython.py**) в действии. Их результаты сравниваются с выводом команды **wc**.

```
print line.rstrip()
f.close()
В следующем коде Python, сохраненном в файле lines.py, под-
считывается количество строк в текстовом файле. Для этого изме-
няется предыдущий цикл for и перед ним добавляется новая
переменная:
numberOfLines = 0
for line in f:
    numberOfLines = numberOfLines + 1
print "Число строк: %d" % numberOfLines
```

Простой пример

Следующий код Python, сохраненный в файле **words.py**, считывает текстовый файл строка за строкой и подсчитывает общее число строк в текстовом файле:

```
numberOfWords = 0
for line in f:
    words = len(line.split())
    numberOfWords = numberOfWords + words
print "Число слов: %d" % numberOfWords
```

Опять же, надо изменить только команды для цикла **for**. Подсчет общего количества слов, возможно, сложнее, чем подсчет количества строк, но реализовать его всё равно просто. Хитрость здесь состоит в том, как отделить одно слово от другого. Единственное, что требует объяснения — функция `line.split()`, которая позволяет определять символы, отделяющие одно слово от другого. Если не указать аргументы, будет использован разделитель слов по умолчанию.

Разделив слова в каждой строке и поместив их в список, можно подсчитать элементы списка с помощью функции `len()`, чтобы определить число слов.

Наконец, вы узнаете, как подсчитать число символов в текстовом файле. Это делается несколько иначе, так как вам придется считать текстовый файл символ за символом. В соответствующем коде Python, который сохранен в файле `characters.py`, используется цикл `while` вместо цикла `for`:

```
numberOfChars = 0
while f.read(1):
    numberOfChars = numberOfChars + 1
print "Число символов: %d" % numberOfChars
```

Хотя длину строки можно определить с помощью функции `len()`, здесь мы обрабатываем каждую строку символ за символом, чтобы использовать как можно более общую схему, так как это позволит нам выполнять изменения посимвольно.

Все три программы имеют похожую основу и отличаются только в некоторой функциональности, что совершенно логично, поскольку они реализуют разные функции. В окончательной версии под названием `wcPython.py` объединены все три программы — она может подсчитывать строки, слова и символы.

Поздравляю, вы только что разработали упрощенную версию утилиты командной строки Linux `wc!` (Скрипты `characters.py`, `words.py`, `lines.py` и `wcPython.py` в действии показаны на рис. вверху стр. 84). Как вы также видите, скрипт `IBL.py` реализует базовые функции утилиты `cat`.

Пока вы узнали, как обрабатывать простые текстовые файлы строка за строкой, слово за словом и символ за символом. В следующих разделах вы увидите, как обрабатывать текст с помощью регулярных выражений, а также выполнять поиск и замену с помощью очень полезного модуля Python под названием `re`.

Модуль `re` Python

Python использует модуль `re` для поддержки регулярных выражений. При определении регулярного выражения могут применяться некоторые символы со специальным значением:

- » `'.'` соответствует любому одиночному символу, кроме перевода строки.
- » `'^'` соответствует началу строки.
- » `'$'` соответствует концу строки.
- » `'*'` указывает, что надо проверить соответствие 0 или более вхождений регулярного выражения.
- » `'+'` указывает, что надо проверить соответствие как минимум одному вхождению регулярного выражения.
- » `'?'` указывает, что надо проверить соответствие 0 или 1 вхождению регулярного выражения.
- » `'[]'` определяет набор символов, соответствие которому надо определить.

Существуют и другие специальные символы, но эти — самые главные.

Чтобы такой символ утратил специальное назначение, следует указать перед ним символ `'\'`. Так, если вы захотите найти точку (`'.'`) в тексте, надо написать `'\.'`. В следующем коде показано несколько простых примеров использования `re`:

```
>>> import re
>>> text = "12343"
>>> m = re.search("3", text)
>>> print m.group(0)
3
>>> m = re.search("7", text)
>>> print m.group(0)
Traceback (most recent call last):
```

Регулярные выражения

Регулярные выражения позволяют выполнять сложный поиск, который без них был бы очень сложен или вообще невозможен. Соответствие шаблонов — это метод, который позволяет искать в строке набор символов на основе указанного шаблона поиска, который определяется регулярным выражением. Если

соответствие шаблона успешно, можно извлечь желаемые данные из строки и воспользоваться ими, заменить их и т. д. Однако регулярными выражениями нужно пользоваться с большой осторожностью, так как они способны привести к появлению трудных в выявлении и исправлении ошибок!

```
File "<stdin>", line 1, in <module>
AttributeError: 'NoneType' object has no attribute 'group'
```

Команда `import re` загружает модуль `re`. Затем выполняется поиск текста с помощью функции `re.search()`. Также существует функция `re.match()`, которая проверяет соответствие только в начале строки; функция `re.search()`, которой мы пользовались, проверяет соответствие в любой части строки. Обычно именно это и нужно.

При обнаружении соответствия функция `re.search()` возвращает его. Так как вы используете статическое регулярное выражение, то соответствием будет именно то, что вы искали, в данном случае, символ 3. Если соответствия не найдено, функция `re.search()` не возвращает ничего. Позже на нашем уроке вы увидите, что делать, если для регулярного выражения, которым вы воспользовались, было обнаружено несколько соответствий. Функция `re.group()` возвращает подстроку, которой соответствует регулярное выражение.

В следующем коде Python показано, как выполнить проверку соответствия для пустой строки:

```
>>> print re.match(r'^$', 'a')
None
>>> print re.match(r'^$', '')
<_sre.SRE_Match object at 0x10a8faa58>
```

Пустая строка — это строка, которая начинается с `^` и заканчивается на `$` и между этими двумя символами не содержит ничего. Почти во всех языках программирования есть похожий способ обнаружения пустых строк.

Скорая помощь

Arch Linux — один из нескольких дистрибутивов, где Python 3 используется по умолчанию. При этом Python 3 может успешно сосуществовать со своим предшественником (который доступен в пакете `python2`).

```
mtsouk@mail:~$ python
Python 2.7.9 (default, Mar 1 2015, 12:57:24)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import re
>>> text = "Linux Format is a magazine about Linux!"
>>> re.search("Linux", text)
<_sre.SRE_Match object at 0x7f8e3b8c74a8>
>>> print re.search("Linux", text)
<_sre.SRE_Match object at 0x7f8e3b8c7510>
>>> print re.search("Linux", text).group(0)
Linux
>>> print re.search("Linux", text).group(1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: no such group
>>> newText = re.sub("Linux", "LINUX", text)
>>> print newText
LINUX Format is a magazine about LINUX!
>>> noSpace = re.sub(" ", "", text)
>>> print noSpace
LinuxFormatisamagazineaboutLinux!
>>> uxUX = re.sub(r'ux\b', 'UX', text)
>>> print uxUX
LinUX Format is a magazine about LinUX!
>>> uxUXspace = re.sub(r'ux ', 'UX ', text)
>>> print uxUXspace
LinUXFormat is a magazine about Linux!
>>> another = re.sub(r'ux\s', 'UX ', text)
>>> print another
LinUX Format is a magazine about Linux!
>>> withCase = re.sub('Linux', 'Linux ', newText)
```

» На рисунке введены некоторые операции поиска и замены, выполняемые в оболочке Python. Оболочка — прекрасное место для экспериментов с регулярными выражениями.

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



Более подробную информацию о модуле `re` можно найти на странице <https://docs.python.org/2/library/re.html>.

В этом разделе мы продолжим там, где остановились в прошлом разделе, и научим вас искать заданную строку в текстовом файле. Ключевой раздел кода в файле `basicSearch.py` таков:

```
numberOfLines = 0
for line in f:
    if re.search("Linux Format", line):
        numberOfLines = numberOfLines + 1
    print line.rstrip()
```

Общая идея состоит в том, что вы производите поиск по текстовому файлу строка за строкой и сравниваете каждую строку с искомой строкой. При обнаружении соответствия вы выводите содержащую его строку и продолжаете поиск в оставшейся части файла, пока не достигнете конца файла.

Поиск и замена текста

Функции `re.search()` достаточно для этого примера, так как одного вхождения желаемой статической строки достаточно для печати строки, которая его содержит. Примечание: функция `re.findall()` может найти все вхождения шаблона, определенного регулярным выражением, и поэтому позволяет выполнять глобальный поиск.

Теперь выясним, как заменить строку, которая соответствует искомому выражению. Опять же, общая идея здесь в том, что вы выполняете поиск в файле строка за строкой и пытаетесь сопоставить каждую строку с шаблоном, который вы хотите найти, требуемое количество раз. Функция `re.sub()` помогает выполнять глобальный поиск и замену с помощью регулярных выражений.

В следующем коде Python показано взаимодействие с оболочкой Python, в которой происходят две операции глобального поиска и замены:

```
>>> text = ""
>>> out = re.sub("^\$", "EMPTY LINE", text)
>>> print out
EMPTY LINE
>>> names = "Mihalis Mike Michael Mikel"
>>> newNames = re.sub(r"^(MikelMichael)\b", "Mihalis", names)
>>> print newNames
Mihalis Mihalis Mihalis Mikel
```

Первая операция заменяет пустую строку на строку "EMPTY LINE", а вторая операция заменяет слова Mike и Michael на Mihalis в любом месте строки. Символ `|` означает OR («или»).

Символ `\b` соответствует пустой строке, но только в начале или в конце слова — это позволяет заменять только целые слова! При объявлении регулярного выражения означает, что регулярное выражение нужно обрабатывать с помощью обозначения «сырой строки». Как вы увидите, `r` используется довольно часто. Функция `re.sub()` находит все соответствия и заменяет весь соответствующий текст.

Скрипт `sAndR.py` меняет строку "Linux Format" на "LINUX Format". В скрипте `sAndR.py` важен следующий код:

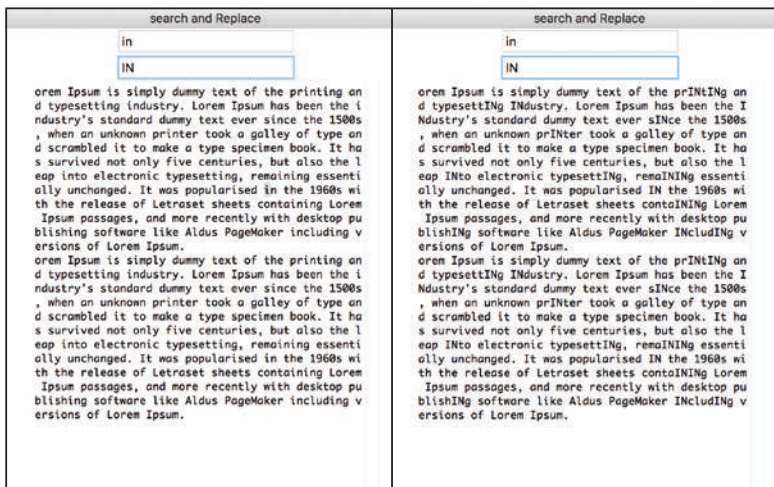
```
for line in f:
    if re.search("Linux Format", line):
        newLine = re.sub("Linux Format", "LINUX Format", line)
        print newLine.rstrip()
```

Код довольно прост, и вы должны легко в нем разобраться — как обычно, он обрабатывает текстовый файл строка за строкой. Ключевой момент здесь в том, что замена выполняется только при наличии соответствия, которое проверяется выражением `if`. На экране отображаются только строки, которые были изменены. (Дополнительные операции поиска и замены в оболочке Python см. на стр. 85 внизу.) Советуем немного поэкспериментировать с `re`, прежде чем продолжить с остальной частью урока. Учтите, что регулярные выражения часто являются источником неприятных ошибок, поэтому всегда проверяйте свои выражения в оболочке Python, прежде чем использовать их в скриптах Python.

Изменение формата даты

Приведенный код Python, который сохранен в файле `dateFormat.py` и основан на файле `sAndR.py`, считывает текстовый файл строка за строкой, выполняет поиск дат в указанном формате с помощью регулярного выражения и изменяет формат даты на другой:

```
numberOfLines = 0
for line in f:
    if re.search(r"(\d{2})/(\d{2})/(\d{4})", line):
        newline = re.sub(r"(\d{2})/(\d{2})/(\d{4})", r"%2-%1-%3", line)
```



➤ Приложение `tkinter` во всей красе. Слева — введенные пользователем данные, справа — то, что происходит, когда пользователь нажимает кнопку Go [Вперед].

Выполнение кода Python

Существует два способа выполнения кода Python: с помощью автономного скрипта или с помощью указания исполняемого файла Python вручную. Оба способа работают, но создание автономных скриптов, которые выглядят, как скрипты оболочки, является более удобным способом программирования.

Также можно выполнить команду `python`, которая откроет оболочку Python. В ней вы сможете

выполнять команды интерактивно. Именно этот вариант рекомендуется при проверке кода и изучении языка.

При запуске оболочки Python мы получим следующий результат:

```
$ ls -l /usr/bin/python
lrwxrwxrwx 1 root root 9 Mar 17 2015 /usr/bin/python
-> python2.7
$ python
```

```
Python 2.7.9 (default, Mar 1 2015, 12:57:24)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for
more information.
>>>
```

Как видите, на этом компьютере Linux используется версия Python 2.7.9. (Пример выполнения операции поиска и замены в оболочке Python см. справа сверху на стр. 87.)

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.


```

numberOfLines = numberOfLines + 1
print newline.rstrip()

Существующий формат даты — MM/DD/YYYY, а новый формат — DD-MM-YYYY. Строка \d{4} означает, что вы ищете четыре (4) цифры (\d). Также с помощью скобок можно обратиться к предыдущему соответствию в той части команды re.sub(), которая касается замены. При запуске файла dateFormat.py мы получим следующий результат:
$ cat variousDates
12/13/1960 01/02/2000
Today is 03/04/2016 or is it 04/03/2016
12/21/10
$ ./dateFormat.py variousDates
13-12-1960 02-01-2000
Today is 04-03-2016 or is it 03-04-2016
Number of Lines matched: 2

```

Создание графического интерфейса

В этом разделе вы узнаете, как разработать графический интерфейс, который упростит вам жизнь. В окне графического интерфейса будет основная область, где вы сможете набирать текст, и еще две области для определения двух строк, которые будут использоваться для операций поиска и замены. Стандартное средство для разработки графического интерфейса в Python — *Tkinter*. Это интерфейс для инструментария *Tk*. Чтобы воспользоваться *Tkinter*, нужно включить в ваш скрипт Python команду `import Tkinter` или `from Tkinter import *`. Обе команды импортируют два наиболее важных модуля *Tkinter*: *Tkinter* и *Tkconstants* — обратите внимание, что модуль *Tkinter* автоматически импортирует *Tkconstants*.

Следующий код Python, который сохранен в файле `simple.py`, представляет собой простой пример использования модуля *Tkinter* — выполните его, чтобы убедиться, что всё работает так, как предполагается:

```

#!/usr/bin/python
from Tkinter import *
root = Tk()
message = Label(root, text="Hello World!")
message.pack()
root.mainloop()

```

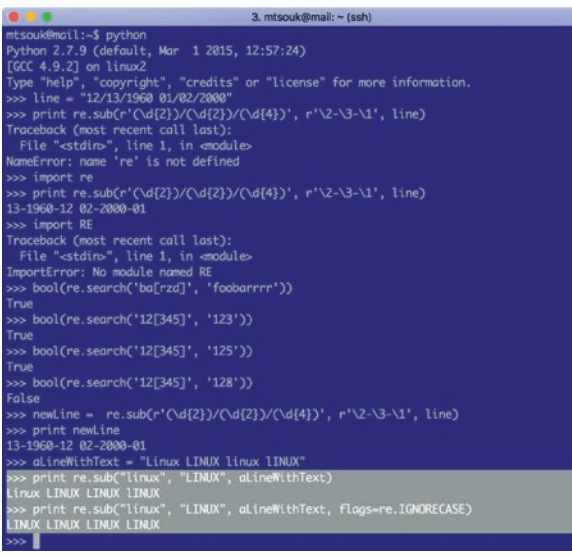
Корневой виджет *Tk* инициализирует *Tkinter* — у каждого объекта *Tkinter* должен быть один корневой виджет, который должен быть создан перед всеми остальными виджетами. Виджет `Label()` — дочерний виджет для корневого виджета, и он содержит сообщение, которое вы хотите отобразить. Метод `pack()` изменяет размер виджета `Label` для его правильного отображения. Виджет не будет отображаться, пока вы не войдете в цикл событий *Tkinter* с помощью метода `root.mainloop()` — до этого момента вы ничего на экране не увидите.

Итак, вы ознакомились с основами *Tkinter*, и пора создать пользовательский интерфейс для приложения. Чтобы добавить необходимые элементы на экран, потребуется выполнить следующий код Python (`emptyGUI.py`):

```

#!/usr/bin/python
from Tkinter import *
from ScrolledText import *
# Виджет Text для ввода и вывода текста
root = Tk(className="Search and Replace")
# Два виджета Entry для поиска и замены
search = Entry(root, text="search")
search.pack()
replace = Entry(root, text="replace")
replace.pack()
text = ScrolledText(root, width=50, height=40, borderwidth=1)
text.pack()

```



```

text.insert('insert', "...")
# Кнопка Go
def callback():
    print "Нажата кнопка Go!"
b = Button(root, text="Go", command=callback)
b.pack()
root.mainloop()

```

Первая версия интерфейса представляет собой пустышку: два поля для ввода, область ввода, в которой вы набираете текст, и кнопка `Go` [Вперед], по нажатию на которую ничего не происходит! В следующем разделе мы реализуем функции кнопки.

Еще о графическом интерфейсе

Теперь пора добавить в приложение необходимые функции. Приложение будет считывать значения двух полей, а также области ввода и выполнять необходимые действия при нажатии кнопки `Go`. Все необходимые действия можно найти в функции `callback()`, которая вызывается при нажатии кнопки `Go`. Остальной код аналогичен коду из файла `emptyGUI.py`.

```

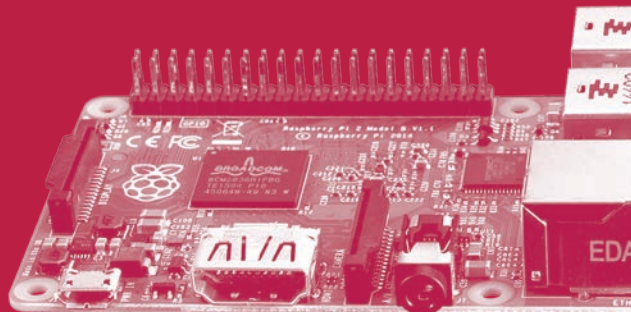
# Кнопка Go
def callback():
    mySearch = search.get() or "null"
    myReplace = replace.get() or "null"
    myText = text.get('1.0', END)
    text.delete('1.0', END)
    # После поиска и замены, печатать новый текст
    text.insert('insert', re.sub(mySearch, myReplace, myText))

```

Чтобы получить текст для виджета `Entry()`, надо использовать метод `get()`. Он показан в коде функции обратного вызова `[callback]` для кнопки `Go`. Аналогично, текст виджета `ScrolledText()` можно получить с помощью метода `get()` и удалить с помощью метода `delete()`. Несмотря на то, что `gui.py` поддерживает только поиск статичного текста, приложение полностью функциональное и довольно удобное. (Скрипт `gui.py` в действии показан на стр. 86.) По нажатию кнопки `Go` программа вызывает функцию `callback()` и делает нужную работу!

Есть много книг, которые помогут вам в изучении Python. Две лучших — «Кулинарная книга Python, 3-е издание» [Python Cookbook, 3rd Edition] Дэвида Бизли [David Beazley] и Брайана К. Джонса [Brian K. Jones] и «Изучаем Python, 5-е издание» [Learning Python, 5th Edition] Марка Лутца [Mark Lutz]. Также существует сайт www.diveintopython.net, который содержит бесплатную книгу по Python для опытных программистов. Подробную информацию о *Tkinter* можно найти на страницах www.pythonware.com/library и <https://docs.python.org/2/library/tkinter.html>. **LXF**

► Простые операции поиска и замены с помощью модуля *re* Python. Чем больше вы будете экспериментировать с регулярными выражениями, тем лучше вы будете их понимать и тем полезнее для вас они будут.



Ваша порция смачных новостей, обзоров и учебников от Raspberry Pi

ЛЕС ПАУНДЕР
путешествует
по стране, обучая
преподавателей
вместе с Picademy



Привет...

На заре своей истории цели Raspberry Pi высоко не лезли, но потом к команде присоединился Дейв Хонесс [Dave Honeess], и Pi взмыли до заоблачных высот! Дейв — глава проекта Astro Pi, в рамках которого два компьютера Raspberry Pi в комплекте с платами Sense HAT будут размещены на Международной космической станции до 2020 г. Детей по всей Великобритании попросили написать свои собственные проекты на основе кода для этих устройств. Избранные проекты должны были отправиться на МКС, чтобы астронавт ЕКА Тим Пик [Tim Peake] запустил этот код в космосе! Ну и не секрет, что этот проект был безумно популярен и породил новый всплеск интереса к космосу и науке. Такой богатый и трудный для исследования материал, как космос, может стать удивительным источником вдохновения для изучения многих предметов, и благодаря Astro Pi и Sense HAT мы можем интегрировать это в обучение программированию. Проект Astro Pi перерос в конкурс общеевропейского масштаба: теперь в нем может участвовать любая школа из стран — членов ЕКА. По возвращении Тима Пика на твердую землю задача выполнения этих экспериментов передается французскому астронавту ЕКА, Томá Прескэ [Thomas Presquet]. Если вы учитель или волонтер Code Club, зайдите на сайт Astro Pi <https://astro-pi.org> и узнайте больше об этом замечательном проекте. Если у вас нет Sense HAT, но вы хотели бы испытать, что он умеет, есть онлайн-эмулятор, который можно попробовать на www.raspberrypi.org/blog/sense-hat-emulator.

Д-р Эбен Аптон, CBE

Пока продажи Pi продолжают бить миллионные рекорды, Pi Foundation наносит визит в Бэкингемский дворец.

Новость о том, что д-р Эбен Аптон в 2016 г. вошел в Почетный список королевы, была не так уж удивительна. Он стал кавалером ордена Британской империи, то есть CBE, за вклад в развитие бизнеса и образования.

25 ноября он получил награду и прокомментировал продолжающееся повышение продаж Pi: «В сентябре мы отпраздновали отметку в 10 миллионов, а сейчас уже преодолели 11 — так что мы продолжаем идти по нарастающей, и это приятно».

Отвечая на вопрос, что должно стать следующей важной вехой, он высказался так: «Хорошо, если мы сделаем еще пару миллионов, это было бы потрясающе, но если обойдем Commodore 64, который находится на третьем месте после PC и Mac, мы стали бы третьей по популярности компьютерной архитектурой в истории».

При нынешних показателях, когда Raspberry Pi Foundation продает около 100000 единиц в неделю, Pi обгонит Commodore 64 уже в начале 2017 г.



SUSE 64-битный

Pi корпоративного уровня.

До сих пор не существовало ни одного официального релиза 64-битных ОС для 64-битного Raspberry Pi 3 и его системы-на-чипе Cortex-A53. Первым восполнить этот пробел решили в SUSE Linux Enterprise Server (SLES) — и создали версию на базе проекта SLES для ARM. А с распространением uboot и новых беспроводных технологий вскорости следует ожидать релизов и от других дистрибутивов.

<https://www.suse.com/communities/blog/suse-linux-enterprise-server-raspberry-pi>



➤ Занимайтесь бизнесом на Pi.

Pi'рожки

В помощь offline-образованию.

Сomputer Aid создан для того, чтобы предоставить автономный доступ к образовательным сайтам на территориях с ограниченным доступом к Интернету. Прямо сейчас, проект занимается переработкой старых Raspberry Pi в портативные точки доступа. В результате, по словам Николя Гэмпелла [Nicola Gampell] из Computer Aid: «Уже готовы двадцать штук для проекта в Эфиопии и один — в Мавритании. Так что вскоре мы надеемся увидеть их в действии». Можете пожертвовать им свой.

www.computeraid.org/donate.html



➤ Распространяя знания по миру.



ZeroSeg

Лес Паундер рассматривает комплект от сообщества, где с помощью старых технологий воссоздан классический, в стиле 1980-х, дисплей к вашему Pi.

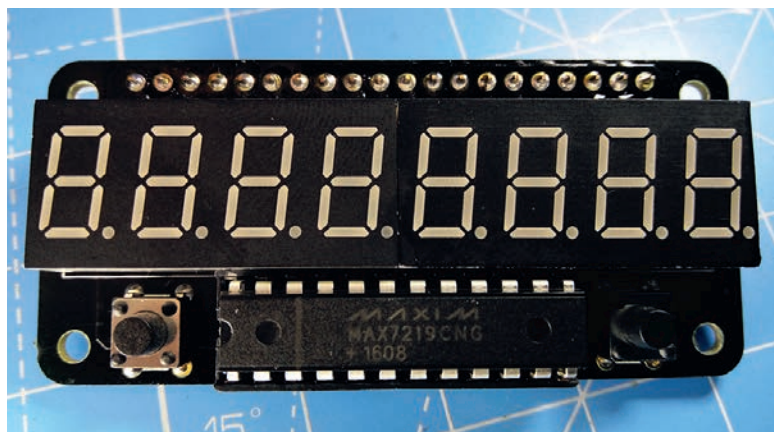
Вкратце

» Двойной семисегментный LED-дисплей управляется чипом MAX7219 SPI и простой библиотекой Python — скромная, но мощная платформа для экспериментов. Как комплект, требующий пайки, эта плата представляет бюджетный вариант для тех, кто хочет попробовать использовать дисплей, и два тактильных переключателя для управления проектом Raspberry Pi без интерфейса.

Т е из нас, кто вырос в 1980-х, помнят примитивный семисегментный дисплей, показывавший '12:00' на наших первых семейных видеомагнитофонах. Этот простой метод вывода, всего семь отдельных светодиодов (LED), давал достаточно информации для того, чтобы пользоваться устройством. В XXI веке у нас есть OLED-дисплеи, есть простые голографические, но и для этого, из семи сегментов, тоже осталось место.

ZeroSeg — это плата расширения, доступная для всех 40-контактных моделей Raspberry Pi, но в первую очередь предназначенная для Raspberry Pi Zero. Инициатором ее разработки один из членов сообщества, AverageManVsPi, создавший несколько плат, ориентированных на новичков. ZeroSeg поставляется в виде комплекта для самостоятельной сборки, который требует пайки. Здесь всё достаточно стереотипно, но есть некоторые нюансы: например, дисплеи надо припаивать рядом с пластиковым корпусом разъема GPIO, что может быть весьма непросто. Нужны твердая рука и хорошее зрение.

Плата имеет два четырехзначных дисплея из семи сегментов и два тактильных переключателя. Дисплеи управляются с помощью чипа MAX7219CNG, предназначенного для контроля 8-битовыми светодиодами дисплеями через интерфейс SPI, который в Raspberry Pi можно использовать для связи. Этот общий чип также используется для управления сеткой светодиодов 8×8 для цифровых табло, и приятно видеть, что чип размещен в держателе, ради замены его в случае необходимости. Чтобы исполь-



» ZeroSeg — это комплект, с которым приятно работать. Конечно, есть кое-какие хитрости, связанные с пайкой, но в этом-то всё удовольствие/дерзание.

зовать плату, надо установить ряд драйверов и библиотек — все они перечислены в небольшом PDF-файле для загрузки, который также содержит инструкции по сборке.

Просто хороший дисплей

Завершив настройку, можно начать работать с ZeroSeg. Библиотека Python для этой платы является ответвлением библиотеки MAX7219 Ричарда Халла [Richard Hull], созданным специально под ZeroSeg. Библиотека, к сожалению, доступна только в Python 2; было бы лучше иметь в качестве стандарта Python 3.

Тактильные переключатели используются с библиотеками Python RPi.GPIO и GPIO; также до нас дошли слухи, что поддержка этого семисегментного дисплея была добавлена в GPIO Zero, тогда экспериментировать будет еще проще. Скрипты примеров обеспечивают достаточно вводной информации, чтобы вы смогли разобраться, как использовать плату. В них показаны такие распространенные варианты, как бегущий текст, простые даты, время и данные о температуре.

Через несколько минут у нас на дисплее уже был бегущий текст, и мы обнаружили маленький подвох, а именно: экраны устроены так, что на них не отображаются знаки '@' или 'W'. Это не вина AverageManVsPi, а скорее проектные ограничения семисегментных дисплеев. Но для проектов, где требуется простой вывод, скажем, таймер камеры, системный монитор или монитор к IoT-кофейнику, эта плата подходит идеально, и стоит всего £10 — вот отличный повод стряхнуть пыль со своего паяльника

и собрать свой собственный проект. Эта забавная плата возвращает нас к старым технологиям и делает их вновь актуальными и доступными сегодняшним мастерам. Также приятно видеть, что имеются проекты, обращенные к сообществу, направленные на удовлетворение его потребностей, и по весьма доступной цене.

Эта плата не для всех, в ней нет ничего нового или уникального, и да, мы могли бы просто создать свой собственный семисегментный дисплей на макетной плате; но ZeroSeg — это аккуратная в плане дизайна и простая в сборке вещь, позволяющая без лишних проводов и возни легко построить простую печатную плату, как раз для вашего Pi Zero. Если вы никогда раньше не использовали семисегментный дисплей, попробуйте — и узнаете, как с нами общался наш старый видеомагнитофон. LXF

Свойства навскидку



MAX7219

Этот стандартный чип служит для управления двойным семисегментным дисплеем и для подключения его к Pi с помощью интерфейса SPI.



Дизайн

Хорошо продумано, что малые компоненты расположены сзади, а их соединения прикрыты более крупными частями, расположенными спереди.

LINUX FORMAT Вердикт

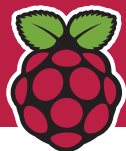
ZeroSeg

Разработчик: AverageManVsPi & Pi Hut
Сайт: <http://bit.ly/ZeroSeg>
Цена: £10

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

» Простой, но функциональный метод вывода, пригодный для разнообразных проектов.

Рейтинг 8/10



Whiptail: Меню как система

Лес Паундер знакомит нас с миром *Whiptail*, системой меню, которая сделает ваши приключения с Raspberry Pi еще увлекательнее.



Наш эксперт

Лес Паундер — изобретатель, которому нравится возиться с техникой. Вооружившись Raspberry Pi, он обучает педагогов и ведет блог, где полно всяких задумок, на <http://bigl.es>.

Возьмите в руки свой Raspberry Pi: перед вами — компьютер Linux за \$35, на котором вы можете учиться, писать код и экспериментировать. Во многих учебниках в качестве основного языка мы брали Python, но здесь воспользуемся *Bash*, языком терминала.

На нашем уроке мы создадим собственную систему меню, используя интерфейс под названием *Whiptail*, который добавляет к ряду основных команд терминала графический слой. Для создания нашего проекта мы возьмем *Geany*, редактор, поставляемый с предустановленной последней версией рабочего стола Raspbian PIXEL. Проект *Geany* разрабатывался много лет, и теперь обрел новый дом на Raspberry Pi. *Geany* годится для кодирования на любом языке, и его функциональность расширяется за счет системы плагинов.

Geany можно найти в разделе Программирование главного меню. Открыв пустой документ, сразу выберите в меню Файл > Сохранить и сохраните свою работу как **system-tools.sh**. Не забудьте сохраняться почаще. Наш новый проект представляет собой скрипт оболочки, на что указывает расширение файла SH. Скрипты командной оболочки служат для автоматизации задач и издавна применяются администраторами Unix. Чтобы начать наш проект, мы должны сообщить Raspbian, какая программа будет запускать наш скрипт. Поскольку это будет *Bash*, надо добавить строку

```
#!/bin/bash
```

После этого создадим большой многострочный графический логотип **LXF**. Это обязательный шаг, но выглядит довольно круто. Мы используем команду `echo` “здесь идет текст ASCII”. Для создания текста мы употребили онлайн-генератор ASCII Art, а затем скопировали и вставили текст в проект. Мы также добавили команду `sleep 2`, чтобы дать пользователю время разглядеть наше творение.

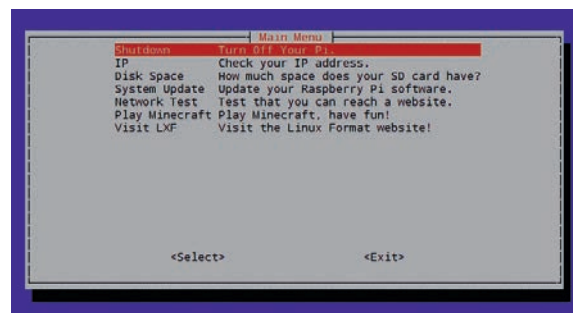
Теперь перейдем к созданию первой строки нашего кода *Whiptail*.

```
whiptail --title "Системные утилиты для Raspberry Pi от LXF"
--msgbox "Управление вашим Pi" \ "C 2016 г." 10 40
```

Попросту, мы создаем окно сообщения, которое будет приветствовать пользователя. Его синтаксис будет включать заголовок и само сообщение. Также там будет стандартная кнопка OK, для закрытия диалога. В конце строки вы заметите два числа: 10 40. Первое — это высота окна в строках, а второе — его ширина в столбцах.

Наш следующий фрагмент кода будет включен в состав цикла, который будет выполняться до тех пор, пока пользователь не захочет выйти:

```
until [ "$answer" == "Exit" ]; do
    answer=$(whiptail --title "Главное меню" --menu "Выберите
опцию" --ok-button "Выбор" --cancel-button "Выход" 20 78 14 \
"Выключение" "Выключите свой Pi." \
"IP" "Проверьте свой IP-адрес." \
...)
```



Наше меню простое и не перегруженное. Но содержит целый ряд функций, подстраиваемых под нужды пользователя.

```
"Посетите LXF" "Заходите на сайт Linux Format!" ${value}
3>&1 1>&2 2>&3)
```

Этот выбор записывается в переменную с именем `answer`. Для захвата пользовательского ввода мы используем меню. Пользователь выбирает некую опцию — например, хочет узнать свой IP-адрес. Пункт меню содержит тег, IP и краткое описание. Тег сохраняется в переменной `answer` с помощью процесса, который фиксирует выход команды *whiptail*, записывает его в `standard error`, журнал ошибок, и переадресуется так, как прописано в нашем коде.

Далее мы вставили строку, которая выглядит довольно невинно: `exitstatus=$?`. Сначала мы зафиксировали статус выхода нашего меню в переменной. Если пользователь хочет выйти, устанавливается статус “1”, в противном случае — “0”. Без этой переменной пользователь не сможет выйти из меню и вернуться в терминал.

Наши следующие несколько строк кода являются частью условия `if`.

```
if [ $exitstatus = 0 ];
then
    echo "Вы выбрали опцию:" $answer
else
    echo "Вы выбрали отмену."
    answer="Выход"
fi
```

Если пользователь желает выйти из приложения, цикл установит переменную `answer` в “Выход”, что остановит созданный нами цикл `until`. При любом другом ответе просто выведется результат в оболочку для последующей отладки. Обратите внимание, что условие `if...else` мы замыкаем `fi`, как этого требует *Bash*.

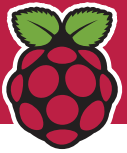
Далее, мы входим в большое условие `if...elif...else`, которое будет обрабатывать пользовательский выбор меню и действовать соответствующим образом:

```
if [ "$answer" == "Выключение" ]
then
    if (whiptail --title "Выключение системы" --yesno
"Вы уверены?" 10 60) then
```



Скорая помощь

Графических интерфейсов для Linux очень много. Один из самых популярных — *Zenity*, <https://help.gnome.org/users/zenity/stable>, где есть окна и диалоги, которые удобно использовать на рабочем столе.



Превращаем меню в команду

Наше меню прекрасно, но оно заперто внутри директории и является довольно громоздким, поэтому давайте сделаем из него команду, которая будет доступна во всей системе. Во-первых, надо изменить разрешения для нашего скрипта, чтобы это был исполняемый файл.

Откройте терминал и перейдите туда, где размещен ваш скрипт — как правило, это папка `/home/pi` — и скомаундите `$ sudo chmod +x system-tools.sh`.

Теперь наш скрипт стал исполняемым, и мы должны урезать имя файла, чтобы удалить `‘.sh’`. Чтобы изменить имя файла, воспользуемся в терминале командой `mv`:

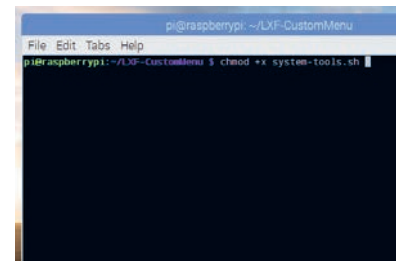
```
$ mv system-tools.sh system-tools
```

Команда `mv` в основном служит для перемещения файлов, и мы воспользуемся ею с `sudo`, чтобы переместить нашу команду в каталог, откуда любой пользователь сможет получить к ней доступ. В терминале надо ввести

```
$ sudo mv system-tools /usr/bin
```

Теперь команда размещена где полагается, и мы можем ввести ее имя в терминале независимо от того, в каком каталоге мы находимся. В терминале вернитесь в свой домашний каталог (`$ cd ~`), а затем введите `system-tools`

Поздравляем, вы превратили свое меню в системную команду!



» Чтобы сделать из меню исполняемую команду, воспользуйтесь командой `chmod` (сокращение от `‘change mode’`).

```
sudo sync && sudo poweroff
else
whiptail --title "Выключение отменяется" --msgbox
"Выключение системы отклонено!" 10 60
fi
```

Синтаксис здесь сравнивает выбор пользователя с рядом заданных кодовых значений. Мы начинаем тест с опции `Выключение`, чтобы выключить наш Pi. При условии `True` откроется еще одно диалоговое окно *Whiptail*, на этот раз окно `Да/нет`, которое просит пользователя подтвердить его выбор. Если выбор подтвержден, активируется команда синхронизации файловой системы и питание Raspberry Pi отключается. Если нет, то окно *Whiptail* сообщает пользователю, что отключение было отменено, и возвращает его в главное меню.

Наше следующее условие — `else if`, которое сокращается до `elif`. Если предыдущее условие было ложно, то проверяется это новое, и так далее, столькими циклами `elif`, сколько требуется:

```
elif [ "$answer" == "IP" ]
then
ip=$(hostname -l)
echo $ip
whiptail --title "Ваш IP-адрес..." --msgbox "$ip" 10 30
```

Для этого условия `elif` мы будем запускать команду получения нашего IP-адреса: `hostname -l`. Вывод этой команды хранится в виде переменной с именем `$ip`, которая затем используется в окне с сообщением пользователю.

Наши следующие два условия отвечают за выбор пользователем проверки места на диске или желания обновить свое программное обеспечение. Структура их аналогична `elif`. Код вы найдете в в материалах для загрузки к этому проекту.

Теперь перейдем к опции `“Network Test”` — она будет пинговать IP-адрес, указанный пользователем. Это отличный способ диагностики сетевых проблем. Легко запоминаемый IP-адрес для этой цели — DNS-сервер компании Google `8.8.8.8`:

```
elif [ "$answer" == "Network Test" ]
then
address=$(whiptail --inputbox "Какой IP-адрес проверять?"
--title "IP-адрес?" 19 90 3>&1 1>&2 2>&3)
test=$(ping $address -c5)
whiptail --title "Результаты теста" --msgbox "${test}" 19 90
```

В данном случае мы используем окно ввода, чтобы зафиксировать IP-адрес, который хотим пропинговать. Ввод сохраняется в переменной с именем `address`, затем мы отправляем команду `ping` на этот IP-адрес пять раз и потом используем `--msgbox` для

отображения результатов. Последние два наших условия `elif` отвечают за запуск *Minecraft* и браузера *Chromium*. Они просто осуществляют запуск соответствующего приложения в терминале:

```
elif [ "$answer" == "Играть в Minecraft" ]
then
Minecraft-pi &
elif [ "$answer" == "Visit LXF" ]
then
chromium-browser http://linuxformat.com &
```

В случае с *Chromium* мы также передаем `web`-адрес, куда будет перенаправлен пользователь. Оба эти приложения запускаются со знаком `&` в конце команды. Это трюк терминала, чтобы выполнять команду в фоновом режиме, освободившись для других команд пользователя.

Наш последний раздел кода — это условие `else`, которое активируется, если все остальные условия являются ложными:

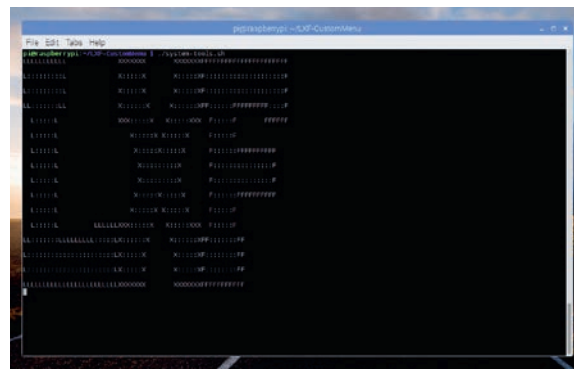
```
else
echo "Выход"
fi
done
```

Это просто выводит в оболочке «Выход». Затем мы замыкаем условие `if`, которое мы создали в начале кода, и замыкаем цикл `until`, строкой `done`.

Завершив код, сохраните свою работу. Теперь откройте терминал и перейдите к папке, где код хранится. Введите следующую команду, чтобы его запустить:

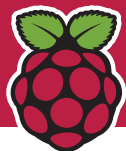
```
bash system-tools.sh
```

Теперь у вас есть свое собственное пользовательское меню. Во врезке вверху, *Превращаем меню в команду*, мы показали, как сделать из него системную команду. **LXF**



» Не удержавшись, мы добавили нашему экрану загрузки меню скромное украшение от ASCII Art.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



TalkiePi: Рация-переноска

Нейт Дрейк дразнит вашу ностальгию прекрасным проектом Дэниела Чоте — TalkiePi, позволяющим создать портативную Wi-Fi рацию в стиле ретро.



Наш эксперт

Нейт Дрейк — журналист-фрилансер, пишущий о технологиях. Его девушка вежливо отказала ему в возможности позаимствовать ее Raspberry Pi для данного проекта, и ему пришлось купить собственный.

Те, кто смотрел потрясающий сериал «Очень странные дела» [Stranger Things — американский научно-фантастический сериал, — прим. пер.], возможно, испытали укол ностальгии, глядя на ретро-рации, которыми пользовались дети. К счастью, новозеландец Дэниел Чоте [Daniel Chote] решил вывести это наследие 1980-х на шаг вперед. Его детище TalkiePi — это проект, превращающий ваш Pi в обычную рацию, работающую через Wi-Fi по нажатию кнопки.

Чоте разработал свой проект, думая о своих детях. По умолчанию, как только устанавливается ПО, оно автоматически соединяется с его собственным сервером через протокола Mumble, позволяя тут же начать разговаривать. ПО на его странице GitHub объединено с подробными инструкциями для компонентов и даже для дизайна корпуса, который можно напечатать на 3D-принтере, чтобы в полной мере испытать ностальгические чувства, которые тронут сердца тех, кому за тридцать.

Интерфейс TalkiePi — проще некуда. При нажатой кнопке загорается индикатор, и можно говорить. При отжатии канал очищается. Есть два других ЖК-индикатора состояния, которые показывают соединение с беспроводной сетью и то, есть ли в вашем канале кто-нибудь еще.

Конечно, этот проект — плод отцовской любви, но полезен он не только детям. Многие геймеры используют протокол Mumble для частных переговоров во время игры. Две рации TalkiePi можно также использовать как удобный интерком для входной двери.

Приступим

Для начала вам нужен Raspberry Pi с Raspbian, установленным на SD-карту для каждого TalkiePi, который вы хотите создать. Для данного проекта идеален Raspberry Pi 3, поскольку Wi-Fi в него встроены. Микрофон, изначально используемый в проекте — US Robotics USB Speakerphone (USR9610), его можно вытащить из корпуса и подключить к Pi. За пределами США его найти трудно, поэтому поищите на eBay такую же модель или постарайтесь найти микрофон USB. Будьте внимательны — выбирайте микрофон, работающий через USB-кабель, а не Bluetooth.

Для питания Pi также нужен короткий 90-градусный USB-кабель. (Например, Amazon UK предлагает 10-сантиметровую версию за £ 1,84 от продавца SWUK.) Если вы планируете печатать корпус TalkiePi, вам также понадобится пять полиамидных шурупов M3 и одна полиамидная изоляционная прокладка M3×20. Вы можете купить наборы этих приспособлений соответственно за £ 2 и £ 3 на родственном сайте Pi Hut, Makersify (<https://makersify.com>). Для корпуса также понадобится два болта M3 15 мм и два болта M3 25 мм. Например, продавец с Amazon UK, Sourcingmap, продает упаковки по 50 штук того и другого из них за £ 3,70 и £ 3,86. Для динамика нужно два болта M3 10 мм и гайки; и, например, продавец с Amazon, Falcon Workshop Supplies Ltd, предлагает упаковки по четыре каждого из них за £ 1,40 каждая.



➤ Идеальный завершённый проект. Зеленый LED показывает, что устройство онлайн; нажмите голубую кнопку LED, чтобы говорить.

Чтобы подключить кабели и ЖК-диоды (LED) к вашему Pi, понадобится соединитель GPIO Header. Подойдет практически любая соединительная головка. Например, Pi Hut продает две гнездовых соединительных головки «мама» 20 за £ 3.

Оба LED «состояния» (см. ниже) должны быть 5 мм, каждый с собственным держателем. Makersify продает упаковку по 50 LED в качестве разновидности держателей Adafruit LED за £ 4–5 каждый. Makersify также продает LED для кнопки; один из них понадобится для кнопки «Разговор». Имеются также пластмассовые кнопки с подсветкой голубого, зеленого, красного, белого и желтого цвета, по £ 3 каждая, а также интересные металлические кнопки голубого, зеленого, красного и белого цвета, по £ 5 каждая.

Исходный проект TalkiePi был создан с пластиковой кнопкой, но вы можете спокойно заменить ее, если хотите. TalkiePi также понадобится три резистора 3,30 Ом. Они должны быть у вас в магазине электротоваров, но если рядом с вами нет такого магазина, то всегда готовый помочь Makersify продает «набор базовых компонентов» за £ 4, и помимо прочих полезных вещей, туда входят 10 резисторов.

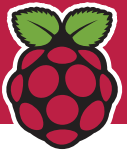
Если вы планируете использовать TalkiePi в качестве домашнего интеркома, то, скорее всего, подключите его к электросети, но если нужен действительно портативный передатчик, стоит рассмотреть возможность приобретения набора батарей, например, PiBorg (о котором мы расскажем на стр. 93).

Понятно, что если вы серьезно намерены собрать это устройство, то понадобятся провода и доступ к паяльнику. Например, Pi Hut продает удобный набор Breadboarding Wire Bundle из 75 гибких витых шнуров всего за £ 3. Ту же сумму можно вложить в макетную плату половинного размера, для экспериментов. Помните, что если вы хотите разговаривать с другими, то надо удваивать количество необходимых частей. Однако, как вы увидите, большая часть компонентов входит в наборы, так что вы сможете создать сколько угодно TalkiePi.



Скорая помощь

Более подробные инструкции по получению надежного сертификата вы найдете на <https://wiki.mumble.info/wiki/MumbleCertificates>.



Сертифицированный Mumbler

По Mumble, как и программа TalkiePi, может для подключения к серверам Mumble использовать сертификаты. Тогда вы избавитесь от ввода паролей, и это позволит вам легко регистрировать учетные записи на новых серверах. По умолчанию данная функция в TalkiePi выключена, поскольку собственный сервер Дэниела Чоте, который он настроил для своих детей, сертификатов не требует. А если вы хотите использовать другие серверы Mumble, подключите свой Raspberry Pi через SSH и запустите `su mumble`, чтобы переключиться на пользователя `mumble`, и `cd ~`, чтобы перебраться в директорию `home`.

Вам потребуется создать контейнер для вашего сертификата, командой

```
openssl genrsa -aes256 -out key.pem
```

Вас также попросят ввести пароль. Далее создайте сам сертификат, командой

```
openssl req -new -x509 -key key.pem -out cert.pem -days 1095
```

Введите тот же пароль, что и раньше, и нажмите возврат, когда вас попросят выбрать опции по умолчанию,

такие как местоположение, адрес электронной почты и т.д. Потом запустите

```
openssl rsa -in key.pem -out nopasskey.pem
```

и введите пароль в последний раз. Наконец, запустите

```
cat nopasskey.pem cert.pem > mumble.pem
```

Эта команда объединит ваши файлы в один сертификат. Запустите `sudo -i`, чтобы снова вернуться к пользователю `root`, затем скомандуйте

```
nano etc/systemd/system/mumble.service
```

для редактирования настройки Mumble. Прокрутите до строки

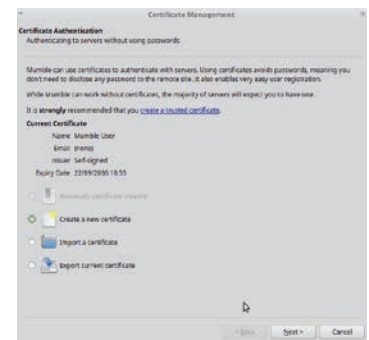
```
ExecStart = /home/mumble/bin/talkiepi
```

Введите пробел, а затем следующее:

```
-username yourusername -certificate /home/mumble/mumble.pem e.g -username rocketman
```

```
-certificate /home/mumble/mumble.pem
```

Нажмите `Ctrl+x`, затем `y`, затем `Enter`, чтобы сохранить и выйти. Перезагрузите Pi, чтобы изменения вошли в силу. Теперь Raspberry Pi при подключении к поддерживаемым серверам Mumble будет использовать TLS-сертификат.



» Сертификат можно создать из командной строки; или, если вы можете подключить свой Pi к HDMI-монитору, используйте простой мастер сертификатов клиента Mumble.

Кнопки и выводы

Основная раскладка выводов GPIO проста для понимания. В кнопку встроены LED на передней панели самого TalkiePi, и он загорается при передаче. Есть также два отдельных LED в качестве индикаторов состояния — «онлайн» и «участник».

Если это ваш первый проект, вам стоит пройти все описанные на сайте шаги на макетной плате, и только потом браться за паяльник. Обратите также внимание на диаграмму GPIO на сайте, прежде чем приступать к работе. Полные инструкции по сборке см. на странице проекта — <http://bit.ly/Wi-FiWalkieTalkie>. Если вы решите полностью повторить проект TalkiePi, не обязательно печатать корпус на 3D-принтере, однако спрятать провода в корпус определенно будет удобнее. 3D-дизайн корпуса и крышки динамика основан на ретро-рации, и был загружен Дэниелом Чоте на его страницу GitHub на <https://github.com/dchote/talkiepi/tree/master/stl>.

Формат файлов STL (StereoLithography) совместим с большинством 3D-принтеров. Проект TalkiePi был выполнен на 3D-принтере Monoprice Select Mini с использованием пластика PLA со 100% заполнением. PLA имеет преимущество — у него приятный запах сахара при работе, и он достаточно прочный.

Если у вас уже есть 3D-принтер, вы, вероятно, уже поднаторели в самостоятельном создании частей. Если у вас его нет, лучше заказать создание верхней и нижней части корпуса и крышки для динамика стороннему производителю.

Простой онлайн-поиск услуг 3D-печати в Великобритании покажет все компании, которые с радостью сделают для вас любое количество требуемых частей. Обычно надо заполнить форму запроса и загрузить STL-файлы на сайт компании, занимающейся 3D-печатью, и они свяжутся с вами, чтобы сообщить цену. Одна такая фирма, с которой мы связались, чтобы выяснить стоимость печати верха и низа корпуса для TalkiePi в двух экземплярах, выставила стоимость производства и отгрузки частей в £142,54! Учитывая, что стоимость 3D-принтеров DIY сейчас составляет всего £178, возможно, вы скорее решите приобрести его или попросить друга дать вам воспользоваться его принтером.

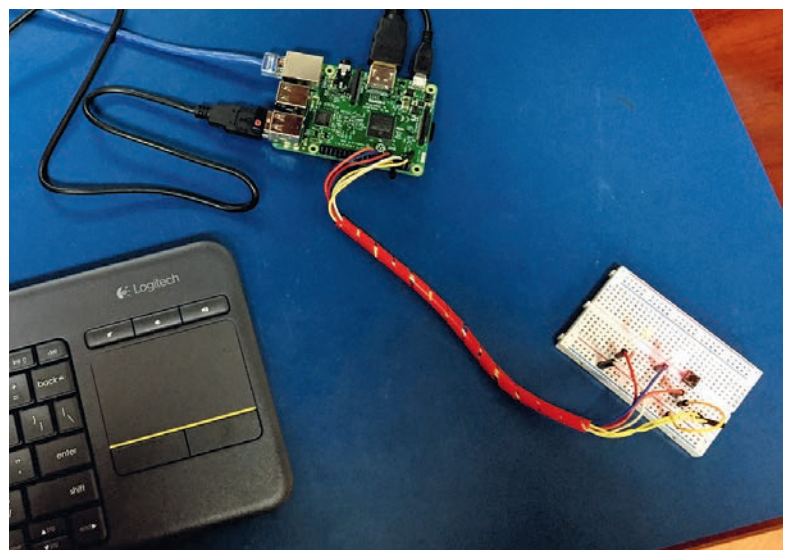
Что касается сервера Mumble, они очень хорошо настраиваются, и можно создать отдельные группы или каналы для всех,

с кем вы планируете разговаривать. Например, вы вряд ли захотите, чтобы ваши друзья, играющие с вами в *World of Warcraft*, слушали звонки в вашу дверь. Если вы настроите свой сервер Mumble (см. врезку *Управление сервером Mumble* на стр. 94), сгруппируйте беседы. Возможно, при каждом переключении каналов придется использовать команду `go run`.

Безопасность в Mumble

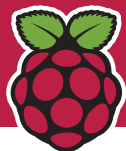
Чтобы за вашими разговорами никто не следил, стоит рассмотреть также создание SSL-сертификата для вашего клиента Mumble. (см. врезку *Сертифицированный Mumbler* вверху, чтобы узнать подробнее, как это делается). Стандартный клиент Mumble идет с собственным сертификатом, но вы можете решить создать еще один на свое имя, запустив мастера создания сертификатов. См. www.mumble.com/support/mumble-creating-a-certificate.php, где

»



» Прежде чем разбирать свой микрофон и что-то паять, попробуйте сначала собрать компоненты на макетной плате, чтобы проверить, работают ли они.

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



вы найдете более подробную информацию. Эти шаги позволят вам создать «самоподписанный» сертификат, идеальный для небольшой группы людей. Если вы планируете открыть свой сервер Mumble для публики, вы, возможно, решите получить надежный сертификат в надежном Центре сертификации.

Перекрестная наводка

По умолчанию ПО TalkiePi соединяется с сервером Mumble Дэниела Чоте и автоматически создает для вас имя пользователя. Это, конечно, всё сильно упрощает, но, помимо того, что это негуманно для пропускной способности канала Чоте, это еще и не самая безопасная настройка. (Опять же, читайте *Управление сервером Mumble*, там вы найдете информацию по настройке вашего TalkiePi для работы с вашим собственным сервером Mumble.)

Технически, вы можете запустить ПО сервера Mumble на том же Pi, на котором разговариваете, но это способно вызвать сетевые ошибки, поэтому лучше иметь отдельное устройство. Если вы планируете использовать Pi в том же здании, стоит рассмотреть вариант компьютера, подключенного к вашей сети специально для работы с серверным ПО Mumble, *Murmur*.

По умолчанию Pi не в состоянии использовать USB-микрофон с громкоговорителем, подключенным в качестве звукового устройства. Если вы можете подключить свой Pi к HDMI-монитору, это легко исправить, нажав правой кнопкой на кнопку громкости и настроив звуковые предпочтения. Если у вас нет монитора или Pi уже помещен в корпус, подключитесь через SSH и запустите команду `sudo -i`, а затем `aplay -l`, чтобы вывести список всех звуковых устройств.

Отметьте номер «карты» устройства USB (скорее всего это будет 'card 1'). Далее запустите `nano /usr/share/alsa/alsa.conf`.

Прокрутите вниз и замените `defaults.ctl.card 0` на соответствующий номер своего микрофона с громкоговорителем. Сделайте то же самое для `defaults.pcm.card 0`.

Если вы решили печатать корпус для TalkiePi на 3D-принтере, помните, что он был разработан специально для микрофона US Robotics, указанного на сайте, поэтому другие микрофоны могут подойти отнюдь не идеально. Обратите особое внимание на размер компонентов, чтобы зря не расходовать дорогой материал.

Настройка по умолчанию для TalkiePi предполагает, что для разговора надо нажать кнопку. Mumble не обязательно должен работать именно так, но это соответствует рациям старого стиля. Если вы хотите использовать Mumble по-другому, например, с горячими клавишами на клавиатуре или с гарнитурой, когда вы повышаете голос, простейший способ — запустить графический аудиомастер в официальном клиенте Mumble. Подключите свой Pi к монитору, откройте Terminal и запустите `sudo apt-get install mumble`. Далее запустите Mumble, и вас автоматически попросят настроить предпочтения микрофона и звука. Настроив собственный сервер Mumble, вы также получите удобные меню для ввода информации.

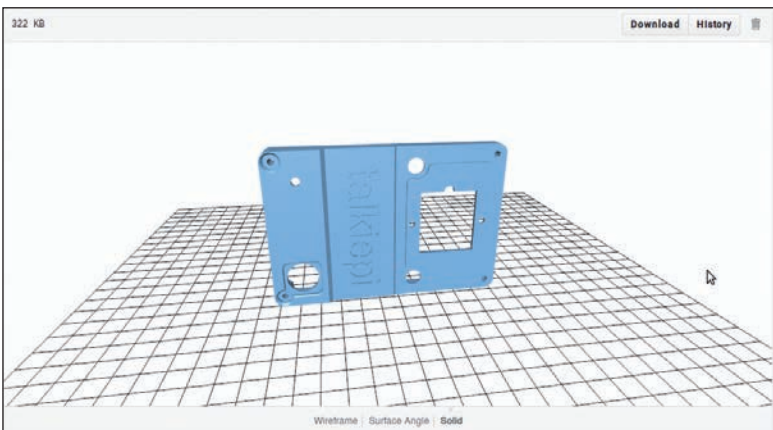
Если в порыве ностальгии вы решите, что вам нужна аутентичная рация, помните, что она будет работать только при подключении к той же беспроводной сети, где был настроен Pi. Это может стать проблемой, если вам понадобится куда-то переместить Pi, потому что нельзя подключить HDMI, когда он находится внутри корпуса TalkiePi. Вместо этого подключитесь через SSH и запустите команду `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`, чтобы отредактировать свои сетевые настройки. В самом низу файла вставьте информацию о новой беспроводной сети, такого содержания:

```
network={
  ssid="yourwifinetworkname"
  psk="yourwifipassword"
}
```

Нажмите Ctrl+x, затем у, затем Enter, чтобы сохранить и выйти. TalkiePi работает ровно столько, сколько установленные в него батареи. Вы можете использовать батареи USB, как предложено на сайте TalkiePi, или, если предпочитаете более аккуратное решение, рассмотрите вариант с PiBorg (<https://thePIhut.com/products/battborg?variant=1103793752>). Это индивидуальный преобразователь энергии для Pi, который работает с большей частью батарей и даже предлагает набор из восьми батарей AA. Сайт Pi Hut рекомендует использовать аккумуляторы.

Работа над проектом TalkiePi продолжается. Сейчас Чоте обдумывает, как запустить его на Pi Zero, сделав намного компактнее и менее прожорливым в плане питания. Спасибо Дэниелу за предоставленные иллюстрации (<http://chote.com>), а самые свежие обновления вы найдете на <https://github.com/dchote/talkiepi>. **LXF**

Скорая помощь
Руководство по настройке *Murmur* для Windows и Linux см. на <http://bit.ly/MurmurServSetup>.



➤ Страница проекта TalkiePi на GitHub предлагает схемы корпуса, который вы можете напечатать на 3D-принтере. Или создайте собственную схему.

Управление сервером Mumble

Чтобы убедиться в безопасности своей настройки, вам нужен собственный сервер Mumble. Вы можете сами настроить его на отдельном Raspberry Pi, если планируете разговаривать только внутри своей домашней беспроводной сети, например, в качестве интеркома. В порядке альтернативы можно использовать сервер Mumble онлайн. Желая настроить собственный сервер Mumble, можете получить бесплатный пробный период на один месяц на <http://www.mumble.com/free-mumbleserver.php>.

Настроив свой сервер, запишите имя домена и порт. Вы также должны придумать имя

пользователя, которое вы зарегистрируете, и построить канал для себя и своих друзей (по умолчанию это **Root**). Далее подключите свой Pi через SSH или откройте Terminal и запустите команду `sudo nano /home/mumble/gocode/src/github.com/dchote/talkiepi/cmd/talkiepi/main.go` для редактирования настройки TalkiePi.

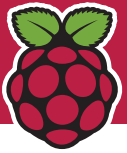
Прокрутите вниз до строки, начинающейся с `server`, и замените адрес сервера TalkiePi `talkiepi.projectable.me:64738` на свой собственный, например, `natedrake.mumble.com:64248`. В строку, начинающуюся с `c/username/cend/`, вы, вероятно,

решите вставить постоянное имя пользователя между кавычками.

Вводите пароль только тогда, когда ваш сервер Mumble его потребует. Если ранее вы создали сертификат, найдите строку, начинающуюся с `insecure`, и замените `true` на `false`. Нажмите Ctrl+x, затем у, затем Enter, чтобы сохранить изменения. В конце концов скомандуйте

```
go run /home/mumble/gocode/src/github.com/dchote/talkiepi/cmd/talkiepi/main.go
```

чтобы запустить ПО TalkiePi с вашими новыми настройками.



Настраиваем свой TalkiePi

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Sep 22 13:17:09 2016
pi@raspberrypi:~$ fish
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
pi@raspberrypi ~$ sudo -i
root@raspberrypi:~# adduser --disabled-password --disabled-login --gecos "" mumble
Adding user `mumble' ...
Adding new group `mumble' (1001) ...
Adding new user `mumble' (1001) with group `mumble' ...
Creating home directory `/home/mumble' ...
Copying files from `/etc/skel' ...
root@raspberrypi:~# usermod -a -G cdrom,audio,video,plugdev,users,dialout,dip,i
nput,opio mumble
root@raspberrypi:~# █
```

```
Preparing to unpack .../libopus-dev_1.1-2_armhf.deb ...
Unpacking libopus-dev:armhf (1.1-2) ...
Selecting previously unselected package golang-go.tools.
Preparing to unpack .../golang-go.tools_0.0-hg20140703-4_armhf.deb ...
Unpacking golang-go.tools (0.0-hg20140703-4) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up golang-src (2:1.3.3-1) ...
Setting up golang-go-linux-arm (2:1.3.3-1) ...
Setting up golang-go (2:1.3.3-1) ...
Setting up golang-doc (2:1.3.3-1) ...
Setting up golang (2:1.3.3-1) ...
Setting up libopenal-dev:armhf (1:1.15.1-5) ...
Setting up libopus-dev:armhf (1.1-2) ...
Setting up golang-go.tools (0.0-hg20140703-4) ...
root@raspberrypi:~# su mumble
mumble@raspberrypi:/root $ mkdir ~/gocode
mumble@raspberrypi:/root $ mkdir ~/bin
mumble@raspberrypi:/root $ export GOPATH=/home/mumble/gocode
mumble@raspberrypi:/root $ export GOBIN=/home/mumble/bin
mumble@raspberrypi:/root $ cd $GOPATH
mumble@raspberrypi:~/gocode $ █
```

1 Создайте пользователя Mumble

Откройте Terminal на своем Raspberry Pi или подключитесь через SSH и переключитесь на пользователя root с помощью команды `sudo -i` ()

Далее создайте себе пользователя, запустив

```
adduser --disabled-password --disabled-login --gecos "" mumble
```

и выдайте ему все необходимые разрешения, что проделывается следующей командой:

```
usermod -a -G cdrom,audio,video,plugdev,users,dialout,dip,input,GPIO mumble
```

```
Processing triggers for man-db (2.7.0.2-5) ...
Setting up golang-src (2:1.3.3-1) ...
Setting up golang-go-linux-arm (2:1.3.3-1) ...
Setting up golang-go (2:1.3.3-1) ...
Setting up golang-doc (2:1.3.3-1) ...
Setting up golang (2:1.3.3-1) ...
Setting up libopenal-dev:armhf (1:1.15.1-5) ...
Setting up libopus-dev:armhf (1.1-2) ...
Setting up golang-go.tools (0.0-hg20140703-4) ...
root@raspberrypi:~# su mumble
mumble@raspberrypi:/root $ mkdir ~/gocode
mumble@raspberrypi:/root $ mkdir ~/bin
mumble@raspberrypi:/root $ export GOPATH=/home/mumble/gocode
mumble@raspberrypi:/root $ export GOBIN=/home/mumble/bin
mumble@raspberrypi:/root $ cd $GOPATH
mumble@raspberrypi:~/gocode $ go get github.com/layeh/gopus
mumble@raspberrypi:~/gocode $ go get github.com/dchote/talkiepi
mumble@raspberrypi:~/gocode $ cd $GOPATH/src/github.com/dchote/talkiepi
mumble@raspberrypi:~/gocode/src/github.com/dchote/talkiepi $ go build -o /home/
mumble/bin/talkiepi cmd/talkiepi/main.go
mumble@raspberrypi:~/gocode/src/github.com/dchote/talkiepi $ █
```

3 Скачайте и скомпилируйте программы

Запустите `go get github.com/layeh/gopus` и `go get github.com/dchote/talkiepi` для скачивания всех программ. Перейдите в директорию **TalkiePi** с помощью `cd $GOPATH/src/github.com/dchote/talkiepi`. Теперь скомпилируйте TalkiePi: `go build -o /home/mumble/bin/talkiepi cmd/talkiepi/main.go`. Запустите `sudo -i` и заставьте **TalkiePi** загружаться каждый раз, когда перезагружается Pi:

```
cp /home/mumble/gocode/src/github.com/dchote/talkiepi/conf/systemd/mumble.
service /etc/systemd/system/mumble.service
```



5 Опциональные LED и корпус

Эта опциональная стадия опять-таки требует выполнения инструкций с сайта проекта (<http://bit.ly/Wi-FiWalkieTalkie>). Вам понадобится припаять плюсы и минусы проводов LED к контактам GPIO, добавить термоусадку, чтобы пресечь закорачивание, подключить соединитель кнопки GPIO и разместить и закрепить LEDы. Вы можете поместить Pi внутрь корпуса с кабелем питания или батареями, тогда с ним будет намного удобнее ходить.

2 Установите зависимости

Переключитесь на своего нового пользователя 'mumble' с помощью `su mumble`. Далее создайте директории установки с помощью `mkdir ~/gocode` и `mkdir ~/bin`. Вам надо будет создать программные ссылки на эти две директории, командой `export GOPATH=/home/mumble/gocode` за которой следует вторая команда экспорта:

```
export GOBIN=/home/mumble/bin
```

И, наконец, перейдите в директорию **install** по `cd $GOPATH`.



4 Подключите USB-микрофон для громкой связи

Подключите USB-микрофон для громкой связи к соответствующему порту на Pi. В этом пункте вам надо следовать инструкциям на сайте TalkiePi, <http://bit.ly/Wi-Fi-WalkieTalkie>, и проходить их по стадиям.

Печатаемый корпус для проекта специально разработан, чтобы подходить для Pi 3 и микрофона громкой связи US Robotics. Присоедините правый/левый угол кабеля USB к нижнему правому порту USB на Pi и прикрепите динамик к передней панели.

```
mumble@raspberrypi:/root
File Edit View Search Terminal Help
GNU nano 2.2.6 File: cmd/talkiepi/main.go
func main() {
// Command line flags
server := flag.String("server", "natedrake.mumble.com", "the server to
username := flag.String("username", "rocketman", "the username of the c
password := flag.String("password", "", "the password of the server")
insecure := flag.Bool("insecure", true, "skip server certificate verifi
certificate := flag.String("certificate", "", "PEM encoded certificate
channel := flag.String("channel", "talkiepi", "mumble channel to join b

flag.Parse()

// Initialize
b := talkiepi.Talkiepi{
```

6 Настройки Mumble Server

Рекомендуем сертифицировать ваш сервер Mumble, чтобы вам не пришлось использовать пароль для соединения, мы объяснили, как это сделать, во врезке *Сертифицированный Mumbler* на стр. 93. Также рекомендуем прочитать врезку *Управление сервером Mumble* на стр. 94, чтобы получить информацию по использованию собственного сервера Mumble и канала. А можете, если хотите, сразу использовать сервер TalkiePi Mumble.

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru, и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Медиа-проигрыватель на Linux
- 2 Задание интервалов Cron
- 3 Доступ к календарям Outlook
- 4 Замену дисков RAID
- 5 Использование эргономичной мыши
- 6 Поиск отсутствующих программ

1 Небольшой медиа-плеер

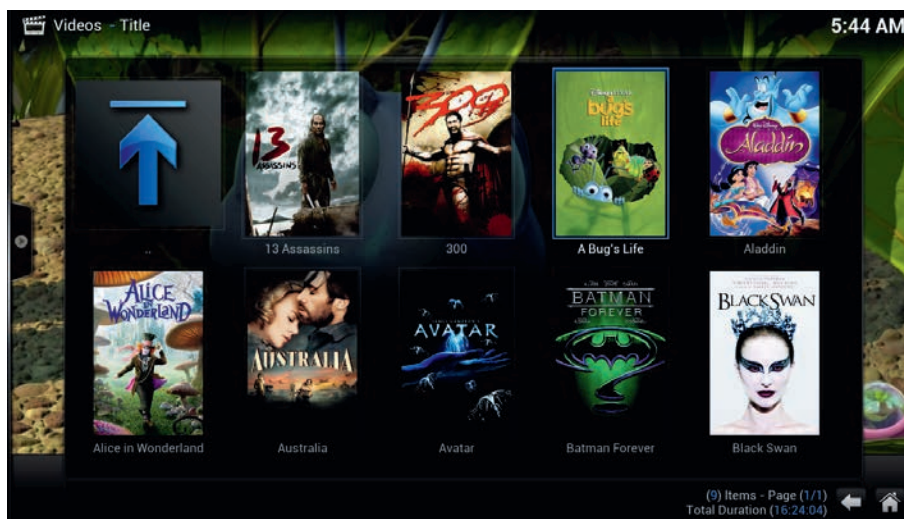
У меня есть старая версия WD TV Live Media Player Wi-Fi 1080p. В отличие от Roku 3, который нельзя подключить к компьютеру, WD TV подключается к моим компьютерам запросто. Я прочитал, что Western Digital больше не будет продавать эти устройства, и хотел бы узнать, нет ли в Linux чего-нибудь пригодного для создания собственного устройства. Мне нравится медиа-проигрыватель WD TV, так как его можно подключить к своему компьютеру с CentOS, на котором настроена Samba, и делиться фильмами. WD TV подключается к нему как «сервер Windows», и я могу воспроизводить видео на своем телевизоре. Можно ли сделать подобное устройство на основе Pi?

tpgbz4z

Я пользовался WD TV Live, но его оказалось довольно легко заменить. Если Вам нужно только воспроизвести фильм из папки Samba, можно использовать почти всё что угодно. Один из моих любимых открытых медиа-проигрывателей — Kodi (<https://kodi.tv>). Его можно запустить на Pi или на любом из многих небольших медиа-проигрывателей под управлением Android, а также на устройствах Amazon Fire TV. Небольшие устройства под управлением Android с предустановленным Kodi стоят гораздо дешевле, чем WD TV Live.

Другой вариант — запустить сервер мультимедиа на компьютере Linux, а соответствующий проигрыватель — на устройстве поменьше. Два основных кандидата — Plex и Emby. Мы воспользуемся Plex, но ранее публиковали обзор Emby [см. «Домашний медиа-рай», стр. 28 LXF204] и руководство по Plex [стр. 68 LXF190]. Преимущество устройств Android в том, что обычно к ним прилагается пульт дистанционного управления, но нам прекрасно подошел Rasplex (www.rasplex.com) на Pi с пультом от обычного центра мультимедиа USB.

А преимущество сервера мультимедиа в том, что он сам находит данные о сериалах и фильмах



» Kodi — одна из самых популярных открытых программ для домашнего кинотеатра/медиа-проигрывателя.

с изображениями. Он также может запоминать, где Вы остановились при просмотре серии или фильма и какие фильмы и серии Вы уже просмотрели. Kodi тоже может кое-что из этого — основное различие в том, что для Emby и Plex требуется сервер, который делает всю работу, и это позволяет разгрузить клиенты, тогда как Kodi представляет собой клиент без сервера и делает всю работу сам. Kodi проще в настройке, особенно если Вам нужен только один проигрыватель и у Вас уже есть настроенный сервер Samba. К тому же Kodi ближе к тому, что Вы делаете с WD TV. Plex и Emby дают большую гибкость, в том числе для мобильных устройств, но для этого придется чуть больше поработать в начале. Если Вам нужна прямая замена для Вашей текущей схемы, ближе всего к ней устройство Android или Pi с Kodi, но все варианты у Вас под рукой, и выбор остается за Вами.

2 Интервалы Cron

В Я загрузил скрипт и хотел бы настроить его запуск в Cron. Я хотел бы, чтобы он запускался с 19:00 до 05:00 каждые 10 минут — скажите, как это сделать?

Золи [Zoli]

О Cron умеет оперировать с самыми причудливыми сочетаниями времени. В первых пяти полях указано время запуска. Эти поля содержат минуту, час, день месяца, месяц и день недели.

Для выполнения команды каждое из полей должно соответствовать текущему времени, поэтому для полей, которые проверять не требуется,

мы используем '*'. Символ '*' также используется для указания частоты. Так, например, если поместить */10 в поле «минута», это будет означать «время, когда количество минут делится на 10», т.е. каждые 10 минут, как Вы и хотите. Поля также могут содержать несколько значений и/или диапазоны значений. Несколько значений разделяются запятыми — скажем, требуемый Вам вариант можно получить с помощью следующих пяти полей: */10 19,20,21,23,0,1,2,3,4 * * *. Работать это будет, но выглядит несколько неуклюже. Диапазоны указываются в виде двух значений, разделенных дефисом; так, например, 19-23 означает «с 7 до 11 часов вечера» (соответствует только номеру часа для всех используемых значений минут). Поскольку диапазон — это разновидность значения, можно указывать несколько диапазонов, разделяя их запятыми. Таким образом, в файл crontab нужно добавить строку */10 19-23,0-4 * * * пользователь команда — а в системном файле crontab (обычно /etc/crontab) пользователя можно не указывать. Скрипт будет запускаться каждые 10 минут с 7 часов вечера до 4:50 утра. Если Вы хотите запускать его и в 5 утра, проще всего будет добавить в файл дополнительную строку.

3 Календари Outlook

В Недавно я вернул Windows 7 на настольный компьютер, удалив Windows 8.1 и отказавшись подчиниться давлению Microsoft, принуждавшей меня установить Windows 10. Единственное, что мне нравится в Windows — календарь Outlook и удобная синхронизация с iPhone 6s Plus. Можете ли вы

Терминалы и суперпользователи

Мы часто предлагаем в качестве решения проблемы ввести те или иные команды в терминале. Хотя обычно то же самое можно сделать с помощью графических утилит дистрибутива, такие решения будут слишком конкретными (будут зависеть от дистрибутива). Команды в терминале более гибкие, и — самое главное — ими можно пользоваться во всех дистрибутивах. Команды настройки системы часто нужно выполнять от имени суперпользователя, называемого также `root`. Существует два основных способа это делать, в зависимости от используемого дистрибутива. Во многих дистрибутивах, особенно в Ubuntu и его производных, перед командой можно написать `sudo` — при этом будет запрошен пароль пользователя, и ему будут предоставлены привилегии `root` только на время выполнения команды. В других дистрибутивах применяется команда `su`, для использования которой требуется ввести пароль `root` и которая предоставляет полный доступ `root` до того момента, пока вы не наберете `logout`. Если в вашем дистрибутиве используется `su`, запустите ее один раз и выполняйте любые заданные команды без предшествующей `sudo`.

посоветовать аналогичные программы в Linux, которые позволят полностью отказаться от операционной системы от Microsoft?

Я очень хотел бы установить Mint на свой компьютер, но приходится оставаться в Windows из-за календаря Outlook. А было бы так приятно объявить свой компьютер «зоной, свободной от Microsoft»...

Джон Кинчингтон [John Kinchington]

Требуется ли Вам делиться календарем с другими, или он строго личный? В последнем случае можно экспортировать данные в другой сервис календарей, который хорошо работает с Linux. В календарях Outlook есть возможность сохранить календарь в файл CSV.

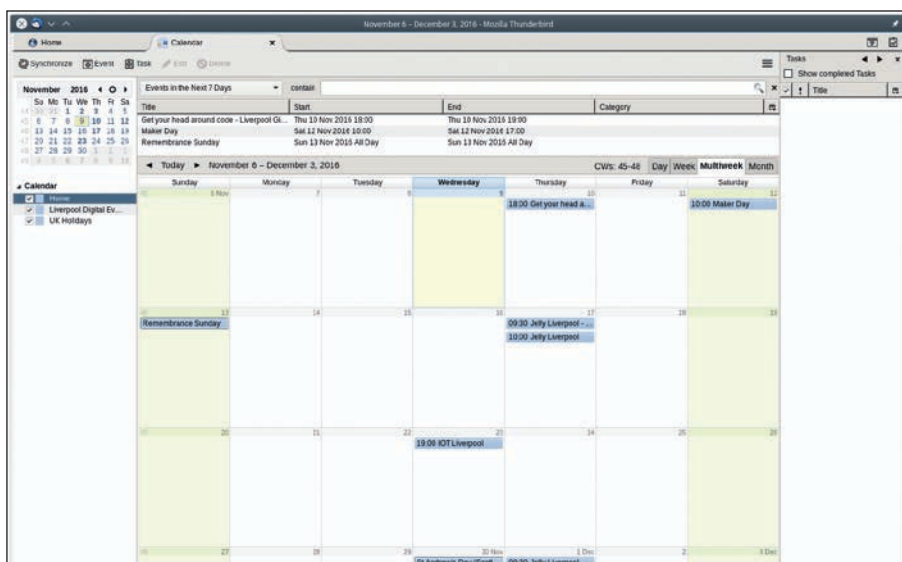
Затем можно импортировать эти данные в другой календарь, например, в календарь Google.

Если Вы хотите остаться с Outlook, есть несколько вариантов. Один из них — web-приложение, которое работает как в Firefox, так и в Chrome, так что календарь будет доступен на любой платформе. В более поздних версиях календаря Outlook, которые используют службу Office365, также можно включить автономный доступ — это может Вам пригодиться.

Вы не уточнили, который именно почтовый клиент стоит у Вас в Linux, но за календарь отвечает Lightning, плагин для Thunderbird (Lightning разрабатывается внутри Mozilla, это не сторонний плагин). Если этого плагина нет в менеджере пакетов Вашего дистрибутива, Вы можете загрузить его со страницы <https://www.mozilla.org/en-US/projects/calendar>. В последние версии Thunderbird Lightning уже встроено, так что этот шаг может быть

необязательным. После установки загрузите плагин `exchangecalendar` с <https://github.com/Ericsson/exchangecalendar/releases>. Загрузите файл XPI, перейдите в менеджер дополнений в Thunderbird и выберите установку дополнения из локального файла. Перезапустите Thunderbird, и Вы сможете добавить новый календарь в разделе Calendars [Календари]. Выберите сетевой календарь типа Microsoft Exchange и укажите параметры своей учетной записи Outlook. Теперь календари Outlook должны появиться в Thunderbird и синхронизироваться с ним.

Существуют и другие почтовые клиенты с поддержкой Exchange, например, Evolution; но поддержка календаря в Thunderbird выглядит самой лучшей. При этом Вы полностью не отказываетесь от Microsoft и продолжаете пользоваться их сервисами, но зато Вам не приходится пользоваться Windows.



С помощью плагина в Thunderbird можно отображать календари, в том числе из Outlook.



Коротко про...

Символы подстановки

Символы подстановки используются для выбора нескольких файлов в оболочке. Два основных символа подстановки — звездочка `*` и знак вопроса `?`. `*` соответствует любому количеству символов, поэтому `*.jpg` оболочка преобразует в список всех файлов с расширением `.jpg`. Важно понимать, что это преобразование (расширение) выполняется оболочкой перед передачей списка файлов вызванной команде. Если у вас есть два файла `in.txt` и `out.txt` и вы запускаете команду `$ ls *.txt`, оболочка заменит выражение подстановки списком соответствующих файлов и затем вызовет команду, поэтому оболочка

запустит команду `$ ls in.txt out.txt`. Команда не имеет понятия о том, что была использована подстановка.

Другой часто используемый символ подстановки — `?`, который соответствует любому одиночному символу: `a?c.txt` соответствует `abc.txt` или `aBc.txt`, но не соответствует `ac.txt` или `abbc.txt`. Существует один символ, которому эти символы подстановки не соответствуют. Это разделитель каталога `/`, поэтому `*.txt` соответствует `abc.txt`, но не `abc/def.txt`: для последнего нужно указать `*/*.txt`. В некоторых оболочках, таких как ZShell, для соответствия любым символам, включая разделитель каталога, используются

символы `**`, поэтому список файлов в текущем каталоге и вложенных каталогах можно получить командой `$ ls **/*.jpg`.

Символы подстановки и регулярные выражения — это не одно и то же, хотя кое в чем они похожи. Оболочка также принимает диапазоны символов в квадратных скобках:

```
$ ls [abcd]*
$ ls [a-d]*
```

Обе команды выводят списки всех файлов, имена которых начинаются с `a`, `b`, `c` или `d`. А что произойдет, если в имени файла встречаются `**`, `[` или `?`? Обычно это не слишком хорошая идея...

4 Замена дисков

В у нас есть сервер с двумя дисками 320 ГБ с RAID 1. В среде установлена Red Hat OS и настроен сервер. Жесткий диск надо расширить, но свободных слотов нет. Нам нужно заменить диски 320 ГБ на диски емкостью 2 ТБ, но мы не можем выполнять настройку повторно. Каково наилучшее решение в этой ситуации? Какое клонирование нужно использовать? Можно ли напрямую экспортировать клонированный образ на жесткий диск? Будет ли клонирована таблица разделов?

С форумов

Для любого клонирования диска нужно загрузить компьютер с Live CD, и поэтому компьютер не может работать в качестве сервера. Так как Вы используете RAID 1, можно использовать *mdadm* для замены одного жесткого диска другим в работающей системе с минимальным временем простоя. Перед этим нужно выполнить разбиение на разделы и загрузку.

Если в системе используется разбиение DOS с MBR (а не GPT), можно скопировать главную таблицу разделов и загрузочный сектор командой *dd*. Сначала удалите второй диск (*sdb*) из RAID командой *mdadm*:

```
$ mdadm /dev/mdX --fail /dev/sdbY --remove
```

Здесь X — номер устройства RAID, а Y — номер раздела для каждого устройства, являющегося частью RAID. Определить, какие устройства входят в какие массивы RAID, можно командой

```
$ cat /proc/mdstat
```

Если используется весь диск RAID, это будет просто */dev/sdb*. Тогда диск будет удален из массива RAID, поэтому теперь он будет работать в автономном режиме. Теперь снимите второй диск и замените его на диск объемом 2 ТБ (отключите питание, если Ваша система не поддерживает горячее подключение). Если Вы используете весь диск RAID, можете сразу перейти к следующему разделу, который посвящен разбиению на разделы и MBR. Скопируйте MBR командой

```
$ dd if=/dev/sda of=/dev/sdb bs=512 count=1
```

Эта команда копирует первые 512 байт старого диска на новый диск (убедитесь, что диски в аргументах команды указаны в правильном порядке). Затем с помощью подходящего редактора разделов Вы сможете изменить раздел на новом диске так, чтобы он занимал весь диск.

Если ядро не распознает новые размеры разделов незамедлительно, потребуется перезагрузка. Размеры всех разделов должны соответствовать размерам разделов старого диска, за исключением размеров главного RAID.

Теперь можно добавить новый диск в RAID-массив, командой

```
$ mdadm /dev/mdX --add /dev/sdbY
```

Эта команда создаст массив RAID на новом диске, и во время создания массива Вы сможете продолжить работу в системе. Если Вы используете RAID для разделов, нужно включить загрузку на обоих дисках, в противном случае *Grub* не сможет загрузить их. Red Hat обычно использует LVM, поэтому есть всего один основной раздел,

а также */boot*. Ход создания массива, как и прежде, можно контролировать в файле */proc/mdstat*. После окончания сборки установите *Grub* на второй диск командой

```
$ grub-install /dev/sdb
```

После этого необходимо повторить указанные действия для другого диска. После отключения для переключения диска и нескольких перезагрузок система запустится и будет полностью готова к использованию. Хотя обычно в таких случаях рекомендуется создать резервную копию данных, здесь диски останутся нетронутыми, поэтому Ваши данные будут в безопасности. Если что-то пойдет не так, Вы сможете просто подключить старые диски, разобраться в том, что не получилось, и попробовать снова.

5 Эргономичная мышь

В Я сравнительный новичок и пользуюсь 64-битной версией Linux Mint 17.1 Rebecca Mate. У меня есть вертикальная эргономичная мышь Evoluent с третьей (средней) кнопкой и двумя боковыми кнопками. Есть ли какая-то программа для настройки нестандартных кнопок мыши, чтобы ими можно было пользоваться в Mint? Я заглянул в Software Manager, но ничего не нашел.

HM

Для изменения параметров устройства ввода, включая назначение кнопок мыши, используется утилита *xinput*. Это команда оболочки, и приличных графических оболочек для нее, похоже, нет (есть много утилит для джойстиков, но не для обычных координатных устройств). Впрочем, *xinput* проста в настройке и позволяет легко экспериментировать с различными параметрами. Сперва нужно определить идентификатор своего устройства командой *\$ xinput list*. Он потребуется для последующих команд. Затем ознакомьтесь с функциями своей мыши, выполнив команду *\$ xinput list ID*, где ID — идентификатор, полученный из предыдущей команды. Эта команда должна вывести список поддерживаемых кнопок с названиями. Для большинства устройств первыми тремя кнопками будут “Button Left”, “Button Middle” и “Button Right”. Предположим, что Вы хотите поменять левую и правую кнопки. Сделайте следующее:

```
$ xinput --set-button-map ID 3 2 1 4 5 6 7 8 9 10
```

Числа здесь соответствуют номерам поддерживаемых кнопок, которые можно просмотреть с помощью команды *xinput list*, и применяются к кнопкам в их физическом порядке. Эта команда назначает правую кнопку (Button Right) на левую кнопку, левую кнопку (Button Left) — на третью, и оставляет всё остальное без изменений; таким образом, она меняет местами левую и правую кнопку. Указав 0, можно отключить конкретную кнопку. Если Вы не уверены в том, какая физическая кнопка соответствует какому числу, можете выполнить команду

```
$ sleep 3; xinput --querystate ID
```

Теперь нажмите и удерживайте кнопку, которую хотите проверить. Через три секунды *xinput*

выведет список кнопок и их состояние, и Вы увидите, какая кнопка нажата. Теперь можно связать эту кнопку с любым желаемым действием.

Все изменения будут временными. Это хорошо, так как можно поэкспериментировать с системой, не боясь вывести ее из строя. Даже если Вы умудритесь отключить все функции мыши, при нажатии Ctrl+Alt+Del система вернется в нормальное состояние. Выбрав конфигурацию кнопок, Вы сможете добавить ее в конфигурацию X. Сначала нужно получить идентификатор USB-устройства мыши (идентификатор X меняется после перезагрузки) командой *\$ lsusb*. Идентификатор представляет собой два четырехзначных числа, например, **0a5c:216f**. Теперь создайте файл */etc/X11/xorg.conf.d/10-mouse.conf* от имени пользователя *root* и добавьте в него следующие строки:

```
Section "InputClass"
    Identifier "Evoluent"
    MatchUSBID "1111:2222"
    Option "ButtonMapping" "3 2 1 4 5 6 7 8 9 10"
EndSection
```

Обратите внимание, что в строке *ButtonMapping* идентификатора нет. Приведите идентификатор USB в соответствие с Вашим. Теперь Ваша конфигурация кнопок будет применяться каждый раз при загрузке рабочего стола.

6 Отсутствующие программы

В У меня Linux Mint Cinnamon 17.3, и я установил *MuPDF* с помощью *Software Manager*. Судя по всему, *MuPDF* установлена, но в меню ее нет. В прошлом у меня была такая же проблема с другой программой — иногда после установки программы не появляются в меню. У меня есть четыре-пять программ, которые были установлены, но не появились в меню, поэтому я не могу ими воспользоваться.

Grumpyskeptic

MuPDF не появляется в меню потому, что эту программу следует запускать из командной строки, указав имя открываемого файла в качестве аргумента. Из меню ее запустить нельзя. Даже если запустить ее в терминале, но не указать имя файла, она завершится с ошибкой. Хотя программа запускается в окне, ее нужно вызывать с именем открываемого файла, потому и нельзя просто запустить ее из меню.

Другой способ воспользоваться этой программой — установить ее в качестве программы для просмотра файлов PDF. Щелкните правой кнопкой мыши на файле PDF, выберите *Open With > Other application* [Открыть с помощью > Другое приложение]. Выберите из списка *MuPDF* и установите флажок *Set as default* [Использовать по умолчанию]. После этого *MuPDF* будет использоваться по умолчанию.

Возможно, остальные программы, которые Вы не видите в меню, отсутствуют по той же причине. За пункты меню отвечают файлы рабочего стола. Если в пакете нет такого файла, то и пункт меню не появится. Если файл есть, можете просмотреть

Помогите нам помочь вам

Ежемесячно к нам поступает несколько писем, на которые мы не в состоянии ответить, поскольку проблема описана в них с недостаточной полнотой. Чтобы дать вам наилучший ответ, нам необходимо получить как можно больше информации о проблеме.

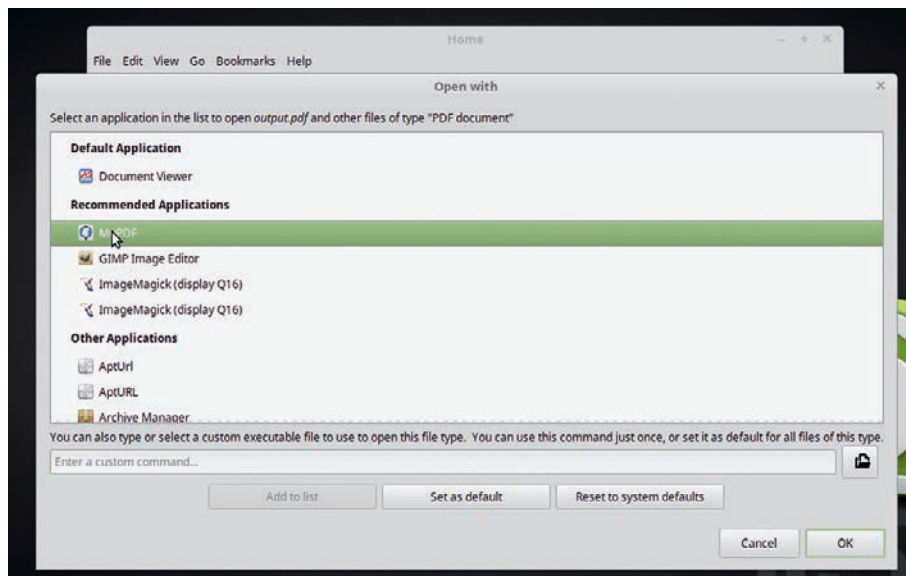
Если у вас появляется сообщение об ошибке, приведите его текст в точности и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все устройства, которые у вас установлены.

Если Linux в вашей системе запущен, вы сможете применить для этого превосходную программу *Hardinfo* (<http://sourceforge.net/hardinfo.berlios>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму, отправляемому нам.

Не уступающий в удобстве альтернативный вариант — *lshw* (<http://ezix.org/project/wiki/Hardware-LiSter>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда и обе).

Если у вас нет желания или возможности их установить, выполните в терминале от имени *root* приведенные ниже команды и приложите сгенерированный ими файл *system.txt* к письму. Это окажет неоценимую помощь в диагностике вашей проблемы.

```
uname -a > system.txt
lspci >> system.txt
lspci -vv >> system.txt
```



➤ Не все программы можно запустить из меню, но их все равно можно использовать в качестве средства просмотра по умолчанию.

его, чтобы отыскать, в каком именно месте в меню появится программа.

В *Software Manager* не отображается содержимое пакетов, поэтому установите *Synaptic*, если он не установлен. Найдите интересующий Вас пакет, щелкните на нем правой кнопкой мыши и выберите *Properties* [Свойства] в меню. Откроется окно с вкладкой, на которой будет находиться список установленных файлов. У *MuPDF* есть файл

рабочего стола, поскольку этот файл также содержит информацию о том, как запускать программу для выполнения действий с файлами, как описано выше. Но в этом файле рабочего стола также есть строка `NoDisplay=true`, так что данная программа не отображается в меню. Без этой строки программа появилась бы в меню в категории, которая определяется параметром `Category` в файле рабочего стола. **LXF**



Часто задаваемые вопросы

DNS

➤ Что такое DNS?

DNS — сокращение от “Domain Name Service [служба доменных имен]” и по сути представляет собой телефонную книгу Интернета.

➤ Как она работает?

Каждое подключенное устройство определяется уникальным IP-адресом. Это числовой адрес, который обычно выражается в качестве последовательности четырех чисел, разделенных точками, например, `80.244.178.150`. Эти адреса довольно-таки трудно запомнить, поэтому мы пользуемся доменными именами, например, www.linuxformat.com.

➤ То есть, система DNS связывает имена с адресами?

Именно. Когда вы набираете в адресной строке адрес сайта,

например, www.linuxformat.com, браузер обращается к системному разрешителю [DNS resolver], который преобразует имя сайта в IP-адрес.

➤ К системному... чему?

Системный разрешитель — механизм, с помощью которого операционная система преобразует доменные имена в IP-адреса. Обычно разрешитель запрашивает требуемый адрес у сервера доменных имен.

➤ Откуда он знает, где находится сервер доменных имен?

Сервер настраивается в рамках конфигурации сети. Если у вас используется DHCP для автоматической настройки сети, то DNS-адреса также задаются автоматически. Впрочем, их можно указывать и вручную.

➤ Их? Так сколько же адресов надо задать?

Обычно нужен только один, но лучше задать несколько на случай сбоя, так как без DNS возможности Интернета серьезно ограничены.

➤ Как получить адреса DNS-серверов?

Интернет-провайдеры обычно предоставляют собственные серверы DNS. Также есть общедоступные DNS-серверы, например, `4.4.4.4` и `8.8.8.8`, принадлежащие Google, ну и другие, платные и бесплатные.

➤ Означает ли это, что каждый DNS-сервер должен знать адрес каждого компьютера?

Нет, он должен знать только то, где их найти. Серверы DNS имеют свою иерархию, поэтому когда вы набираете linuxformat.com, сервер должен только знать, где найти информацию для `.com`. Затем этот сервер сообщит сведения об ответственном сервере для linuxformat.com. Каждое доменное имя зарегистрировано на двух ответственных

серверах. Надежность DNS имеет огромное значение, поэтому обязательно используются два отдельных сервера.

➤ Это связано с тем, что на серверах DNS часто возникают сбои?

Нет, сбои возникают нечасто, но если это происходит, последствия очень серьезные. Последняя атака на крупный DNS-шлюз в США показала серьезность этой проблемы, когда многие крупные сервисы оказались недоступными в затронутых регионах.

➤ Вы сказали, что системный разрешитель обычно использует DNS — какие еще методы он использует?

Адреса компьютеров в вашей локальной сети могут находиться в файле `/etc/hosts`. Разрешитель обычно проверяет его первым. Также возможны другие источники, например, базы данных и домены Windows.



LXF Hot Picks



Александр Толстой

бродит по репозиториям сотен проектов и дистрибутивов, осиливая тысячи кодовых баз, чтобы добыть для вас самые лучшие релизы.

Krusader » Digikam » Neural Enhance » Tuxguitar » Devilspie2 » Kup » Homebank » Featherpad » Spek » Mouse Boat » Freeciv

Менеджер файлов

Krusader

Версия: 2.5 Сайт: <https://krusader.org>

Двухпанельный стиль менеджеров файлов, похоже, не собирается исчезать, и это определенно верно в отношении *Krusader*, менеджера файлов с долгой историей.

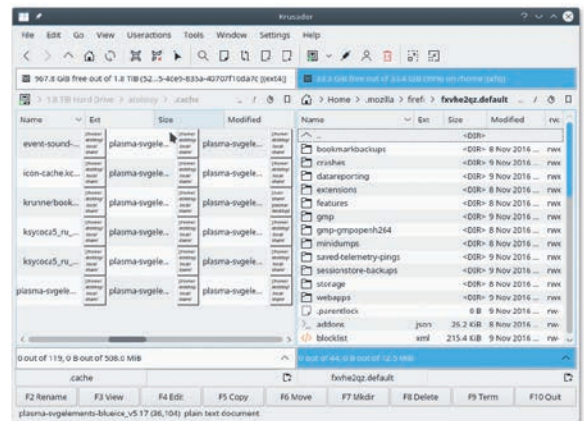
Krusader 2.5 — блестящий новый релиз менеджера после более четырех лет разработки, который включает реализацию многих новых функций и портирование приложений в библиотеки *Qt5* и *KF5*.

Krusader откроет перед вами всю глубину своих встроенных функций, когда вы начнете применять его в повседневной жизни. Среди самых замечательных — встроенная поддержка архивов, опции удаленного соединения (такие, как *SFTP* и *Samba*), круговая диаграмма статистики

потребления диска, внутренняя программа просмотра и редактор разных типов файлов, инструмент массового переименования, модуль поиска и калькулятор контрольной суммы, и т. д. По сути, *Krusader* работает как универсальный инструмент для тех, кому нужен менеджер файлов, чтобы делать всё, не переключаясь между окнами.

Графический интерфейс приложения должен показаться знакомым всем, кто

Чтобы делать всё, не переключаясь между окнами.



» *Krusader* не выглядит старомодным, и уж точно позволяет вам лучше контролировать свои файлы.

использовал *Midnight Commander* в Linux, *Total Commander* в Windows и, возможно, *Norton Commander* в DOS. Здесь есть две панели, которые могут отображать любые директории, так что можно копировать, перемещать и удалять файлы между источником и пунктом назначения. Нижняя панель *Krusader* также предусматривает клавиши быстрого запуска для популярных действий, например *F5* для копирования, *F6* для перемещения и *F3* для отображения. Эти привязки к клавишам *F* соответствуют исторически сложившимся правилам и отвечают за соответствующие действия, которые вы найдете почти во всех приложениях в *Commander*.

При самом первом запуске *Krusader* выполняет разовую проверку всех внешних приложений — и чем больше он обнаружит, тем функциональнее будет на вашей машине. Если приложения нет, *Krusader* не откажется работать, но вы увидите, что соответствующая функция будет отключена.

Поскольку *Krusader* — это приложение KDE, вам нужно быть готовыми установить некоторые относящиеся к KDE дополнительные утилиты, например, *Filelight* и *Kate*, чтобы сделать *Krusader* более мощным. Однако вы всегда можете изменить свои предпочтения в подразделе *Dependencies* раздела *Krusader*. Например, *Krusader* прекрасно интегрируется со стандартными утилитами Gnome, если вы укажете их вручную.

Исследуем интерфейс Krusader

Правая панель

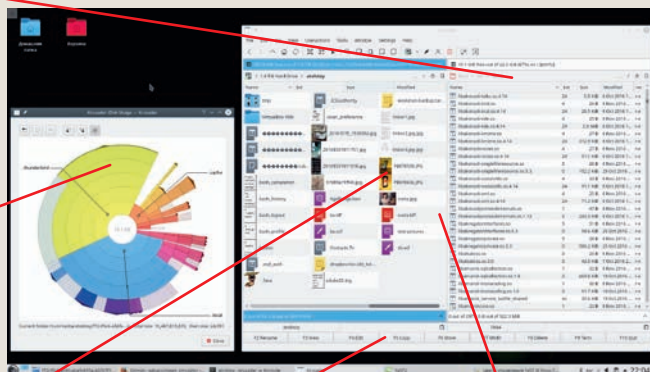
Между панелями можно переключаться, просто щелкая по ним. Активная будет выделена цветовой строкой наверху и внизу.

Интеграция с другими инструментами

Круговая диаграмма потребления диска обеспечивается *Filelight*, но запускается прямо из *Krusader*.

Просмотр файлов

Krusader имеет несколько режимов просмотра и может отображать ярлыки для большинства медиа файлов.



Подсказки внизу

По традиции, клавиши *F* привязаны к популярным действиям, таким, как копирование и перемещение для файлов и директорий.

Левая панель

Режимы просмотра зависят от панелей, так что можно задать разные настройки для источника и адреса.

Органайзер изображений

DigiKam

Версия: 5.3 Сайт: www.digikam.org

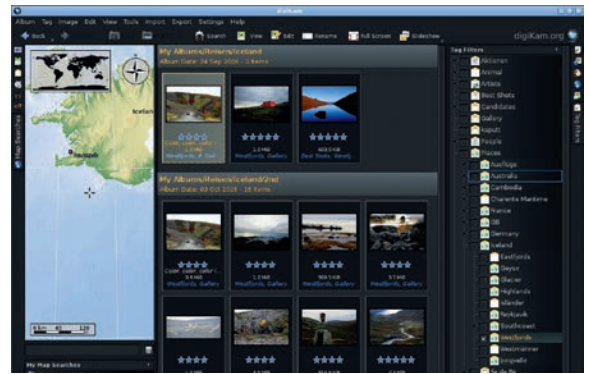
Немного найдется проектов с открытым кодом, которые доросли до положения *digikam* и стали программами профессионального уровня со множеством функций, оптимизированным интерфейсом, плагинами, обстоятельными переводами и т.д. LXF уже давно восхищается органайзером изображений: от хвалебных отзывов о *digikam 1.0* [См. Обзоры, стр. 8 LXF130] до заслуженных похвал в более поздних обзорах [см. Обзоры, стр. 14 LXF150 и HotPicks, стр. 100 LXF201].

DigiKam — это менеджер фото и довольно многофункциональная лаборатория для обработки, индексации, сортировки, редактирования и предоставления совместного доступа к изображениям. Программа является естественным эквивалентом органайзеру фото *Shotwell*, хотя *Shotwell* обычно считается куда более простым по сравнению с *digikam*. Итак, если

вам нужен полный контроль над своей библиотекой изображений и ваш компьютер способен потянуть работу с большими наборами данных, *digikam* станет отличным товарищем. Когда *digikam* запущен, он сканирует вашу директорию `~/Pictures` и получает метаданные изображений для их сохранения в базе данных *SQLite*. Миниатюры, контрольные суммы и теги тоже хранятся в базе данных, что позволяет выполнять множество расширенных задач с изображениями, например, обнаружение дублей или сортировка по изображенным людям (с применением распознавания лиц).

В серии 5.x *digikam* был портирован в среды *Qt5* и *KF5*, и если вам захочется

Лаборатория для обработки и предоставления доступа.



> DigiKam имеет плагины почти что для всех необходимых вам действий, ну вот только чай не заваривает.

скомпилировать *digikam* из исходника, понадобится много зависимостей. Однако в этом нет никакой сложности: как только у вас появятся все многочисленные пакеты `*-devel`, необходимые *digikam*, вы продолжите с помощью простых команд, таких как `$ cmake ../ && make && make install` внутри пустой поддиректории компиляции.

Поддерживается также технология *AppImage*, т.е. возможность скачивать независимые пакеты с готовым к использованию *digikam*. *AppImage* похож на *Flatpak* из мира *Gnome* и *GTK 3*, но еще проще в использовании: скачайте копию с <http://download.kde.org/stable/digikam>, сделайте файл исполняемым и щелкните по нему дважды!

Оптимизатор изображений

Neural Enhance

Версия: Git Сайт: <http://bit.ly/NeuralEnhance>

Мы уже знаем, как имитировать в Linux голливудскую технологию при помощи специального приложения (см. HotPicks, стр. 105 LXF195); ну, а если бы некоторые волшебные трюки из секретных лабораторий CSI стали реальностью, и мы смогли бы, например, оптимизировать снимок с камеры наблюдения, чтобы разглядеть номерной знак машины? Что ж, благодаря *Neural Enhance* мы подошли к этому весьма близко.

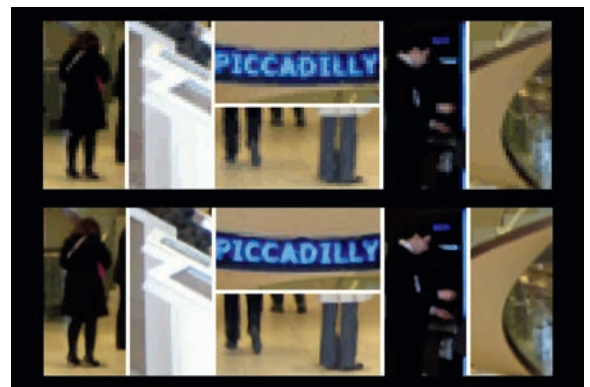
Судя по всему, наблюдается тенденция роста алгоритмов с открытым кодом, использующих нейронные сети для оптимизации изображений. Мы обзавелись GitHub и нашли немало подобных проектов, однако *Neural Enhance* является самым примечательным из них. И кроме того, он весьма прост в настройке.

Идея *Neural Enhance* проста: хотя изображения с низким разрешением нельзя обогатить более реальными деталями, мы можем сделать реконструкцию и додумать эти детали, используя модель данных, основанную на других изображениях.

В результате стало возможным увеличить масштаб изображения с лучшим качеством, чем с помощью привычных методов (таких, как `Image > Resize` в *GIMP*). Полученное в результате качество очень сильно зависит от этой самой модели. Но *Neural Enhance* очень дружелюбен к начинающим и по умолчанию предлагает вполне пригодную тестовую модель. Вся установка может быть сложновата для тех, кто не особо подкован технически в Linux, но, к счастью, автор предлагает более быстрый способ с применением *Docker*. Если вам хочется раскрыть возможности алгоритма вручную, просто установите *Docker* для своего дистрибутива и запустите образ *Neural Enhance*:

```
$ docker run alexjc/neural-enhance /bin/bash
```

Увеличьте масштаб изображения с лучшим качеством.



> Истина где-то рядом — за этими размытыми пикселями...

Более практичный способ — создать алиас, который автоматически отыскивает требуемый образ и передает команду:

```
$ alias enhance='function ne() { docker run --rm -v "$PWD"/:dirname "$@:$#" /ne/input -it alexjc/neuralenhance "${@:1:-1}" "input/" /basename "$@:$#" ; } ; ne'
```

Затем увеличьте масштаб изображения командой

```
$ enhance --zoom=2 --model=small images/example.jpg
```

Поварьюруйте опцию `-zoom`, чтобы настроить масштабный коэффициент.

Музыкальное приложение

Tuxguitar

Версия: 1.3.2 Сайт: <http://tuxguitar.herac.com.ar>

Н е каждый пользователь Linux является страстным разработчиком или администратором — Linux предназначен для всех, включая творческих людей. *Tuxguitar* — полнофункциональный нотный редактор, отчасти похожий на *Frescobaldi* [см. HotPicks, стр. 103 LXF184]. Те, кто много занимается композицией и аранжировкой музыки, определенно должны дать *Tuxguitar* шанс. В отличие от других музыкальных программ, где вы составляете крупные партитуры, *Tuxguitar*, как и следует ожидать, предназначен для гитарной музыки и визуально оптимизирован для написания табулатур для гитары.

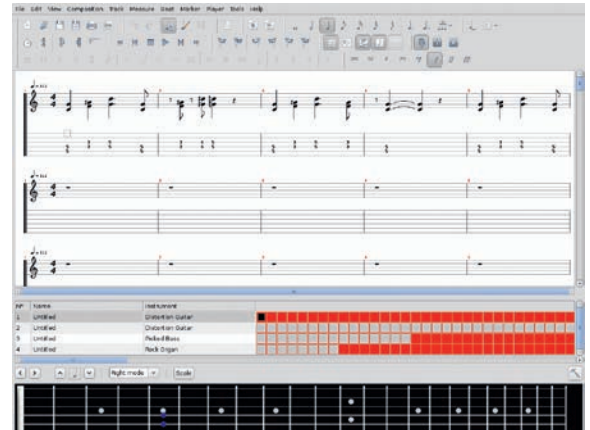
Вместо использования нот в классическом смысле *Tuxguitar* предлагает набор позиций на грифе. Кроме того, писать для гитары может быть сложно для тех, кто не умеет читать ноты — это способно осложнить им задачу. *Tuxguitar* как раз и создан для облегчения этого процесса. Начинаящие пользователи не ограничены

необходимостью ввода исключительно в виде табулатуры: для ввода музыки, которую вы хотите записать, можно также использовать внешнюю клавиатуру MIDI.

Если у вас нет под рукой клавиатуры, *Tuxguitar* предоставит вам виртуальную клавиатуру. Если это покажется вам сложным, вы всегда можете прибегнуть к старой доброй музыкальной нотации, которая мгновенно превращается в табулатуру для гитаристов. Чтобы вы не сбились, вы всегда можете прослушать свой шедевр по мере обретения им формы, и понять, какой звуковой хаос вы способны породить своими виртуозными пальцами, глядя, как ноты летают по грифу внизу экрана.

Серия *Tuxguitar 1.3.x* — хороший знак, поскольку приложение не обновлялось

Сочинителям музыки Tuxguitar будет прекрасным подспорьем.



➤ Полнофункциональная замена *Frescobaldi* для любителей гитары.

с 2009 г., когда вышел *Tuxguitar 1.2*. В этой версии множество новых функций, например, поддержка обратного отсчета, новые плагины импорта/экспорта, улучшенная печать и поддержка шаблонов песен, и т. д. Если вы жаждете стать сочинителем музыки, новый *Tuxguitar* будет вам прекрасным подспорьем.

Утилиты окон

Devilspie2

Версия: 0.41 Сайт: www.gusnan.se/devilspie2

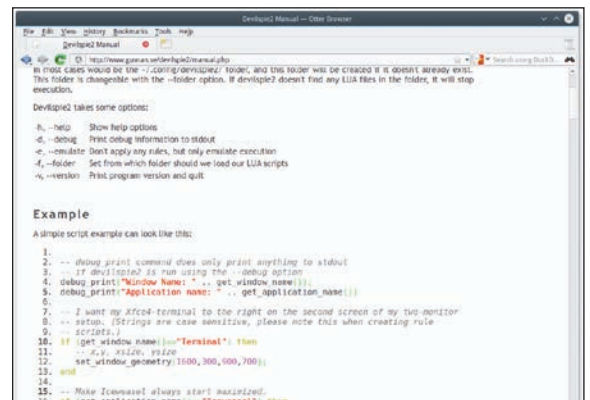
Э той крошечной программке более пяти лет, но недавно она привлекла к себе дополнительное внимание благодаря включению SUSE SLE 12 SP2 changelog. Подобно большинству других продуктов Linux уровня предприятия, SUSE предлагает рабочий стол Gnome 3, который очень удобен, но не лишен пробелов в функциональности, которые необходимо заполнять сторонними инструментами.

Devilspie2 — это инструмент, способный исполнять скрипт, когда окно открывается, закрывается, минимизируется, максимизируется... и при прочих манипуляциях. Эта функция очень удобна при стандартной рабочей нагрузке, когда вам, например, нужно, чтобы окно терминала каждый раз появлялось в определенном месте (возможно, справа при настройке с несколькими мониторами, или на определенной рабочей области). На рабочем столе Plasma это настраивается в *Kwin*, но в случае с другими менеджерами окон поможет *Devilspie2*.

При запуске приложение заглядывает в директорию `~/.config/devilspie2` в поиске скриптов Lua и исполняет всё, что находит. Да, это означает, что для работы с *Devilspie2* необходимо минимальное знание скриптов Lua, хотя на самом деле это проще, чем может показаться. Начинаящие могут взять готовые к употреблению примеры за шаблоны для собственных скриптов и просто изменить условия и действия в соответствии со своими нуждами. Например, чтобы заставить браузер *Chromium* появляться на правой стороне установки с двумя мониторами (2560×1440), следует сделать следующее:

```
if (get_application_name()=="Chromium"
and get_window_name() ~= "Print") then
```

Однажды вы вдруг обнаружите, что стали программистом Lua.



➤ Играть с размещением окон с помощью *Devilspie2* можно бесконечно.

```
set_window_geometry(1400,0,1150,1200);
maximize_vertically();
end
```

Этот код надо сохранить внутри файла Lua, после чего запустить *Devilspie2*. Программа будет тихо сидеть в фоне и ждать; но когда запустится *Chromium*, *Devilspie2* автоматически настроит правильное положение окна. С помощью *Devilspie2* вы можете размещать окна поверх других окон или под ними, минимизировать или разворачивать их, баловаться с геометрией окон и с условиями. И однажды, создав с помощью *Devilspie2* идеальный рабочий стол, вы вдруг обнаружите, что заодно стали достойным программистом Lua.

Инструмент резервного копирования

Kup

Версия: 0.6.1 Сайт: <https://github.com/spersson/Kup>

Хотя в одном из прошлых номеров мы провели исчерпывающее Сравнение инструментов резервного копирования [см. стр. 20 LXF207], в одну статью втиснуть все существующие инструменты резервного копирования не получается: их слишком много.

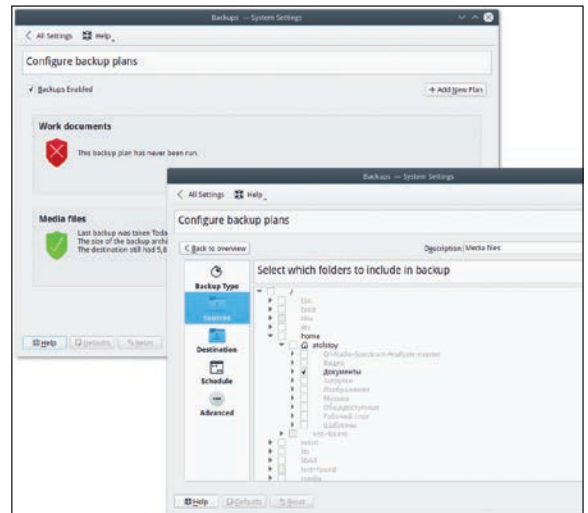
Kup Backup System — интересный проект, который помогает делать резервные копии. В отличие от многих других утилит подобного рода с открытым кодом, *Kup* не является интерфейсом для популярного инструмента синхронизации, *rsync*. *Kup* основан на *bup* (<https://github.com/bup/bup>), другом решении, имеющем особые преимущества перед другими конкурирующими инструментами. Например, *bup* использует формат packfile из Git и работает с ним очень эффективно, так что инкрементные резервные копии с очень большим количеством данных выполняются очень быстро. Если ваша резервная копия была прервана на середине, следующий запуск *bup* возобновит ее именно с этого места

без повторного индексирования и пустой траты времени.

Как можно смекнуть по его названию, *Kup* — это GUI и модуль интеграции *Vir* для рабочего стола Plasma. Вы можете найти его в System Settings в разделе Backups. Первым делом отметьте окошко, чтобы включить *Kup*, а затем нажмите на кнопку Add New Plan. *Kup* позволяет настроить исходную (что копировать) и целевую (где хранить) директорию и расписание.

Поддерживается два типа схем резервного копирования: одна сохраняет директорию **backup** при полной синхронизации с тем, что у вас есть на компьютере, а другая — инкрементная, т.е. хранит более старые версии ваших файлов в директории **backup**. Инкрементные резервные копии

Инкрементные резервные копии удобны с большими файлами.



Вы можете создать в *Kup* сколько угодно планов резервного копирования и запустить их одновременно.

особенно удобны при работе с большими файлами, причем вы получаете доступ к вашим файлам, как если бы они были полной резервной копией — каждая резервная копия содержит полную версию ваших директорий. А за сценой всё содержимое, которое не изменялось, сохраняется только однажды. Просматривая адресат резервной копии с помощью *Dolphin*, вы можете получить доступ к содержимому архивов *Kup/bup* благодаря Kioslave в *Kup*.

Финансовое приложение

Homebank

Версия: 5.1.1 Сайт: <http://homebank.free.fr>

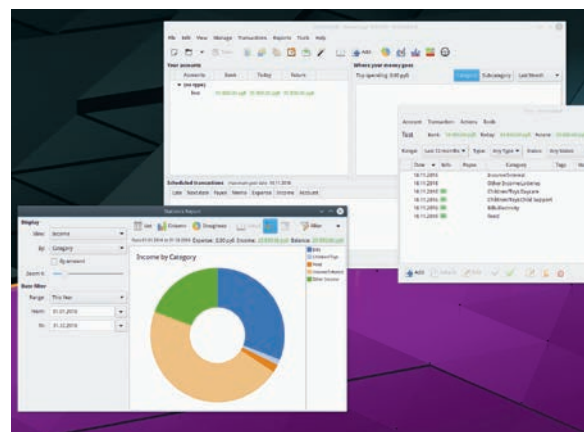
Все хотят еще хоть чуть-чуть растянуть свой персональный бюджет, и немало приложений Linux способны тут поспособствовать. *Homebank* поможет вам управлять своими деньгами, и основная идея заключается в точной записи всех доходов и расходов: чем кропотливее вы считаете свои денежки, тем легче их сохранить непросвисанными.

К интерфейсу *Homebank* очень легко привыкнуть, и в нем три основных области: верхняя левая для списка ваших счетов, верхняя правая для отображения ваших расходов и нижняя часть интерфейса для запланированных транзакций. После первого запуска следует создать новую учетную запись, задать имя, тип и прочую важную информацию, и после этого можете приступать к фиксации своих финансовых действий. Для этого обратитесь к меню Transactions > Add и заполните сумму, метод оплаты, категорию, получателя платежа и прочие детали. *Homebank*

автоматически предлагает широкий выбор возможных категорий, и мы должны признать, что они мудро отмечены значками '+' или '-', в помощь пониманию направления движения потока.

Конечно, вы можете добавить к каждой транзакции много добавочных деталей, например, дополнительные записи в чековую книжку при получении чеков, указание статуса транзакции и т.д. *Homebank* — очень мощный инструмент для персонального использования, способный выполнять такие действия, как импорт банковских выписок (OFX, QIF, CSV, QFX), подсчет месячного или годового бюджета, импорт файлов данных из *Quicken* и *Microsoft Money*, создание отчетов и графиков. Это самая милая

Предлагает массу цветных диаграмм за пару щелчков.



Homebank — мощная программа для персональных финансов и для всех, кто любит статистические отчеты.

графическая функция в бухгалтерском ПО, в остальном довольно скучном. *Homebank* предлагает массу цветных круговых диаграмм, создаваемых за пару щелчков, нажатием на значок соответствующего отчета в правой стороне главной панели инструментов *Homebank* и выбором графического вида. С версии 5.1 *Homebank* приобрел еще более полезные функции, из которых самая важная, видимо, бухучет в нескольких валютах. Не пропустите обновление!

HotGames Развлекательные приложения

Симулятор рыбалки

Mouse Boat

Версия: Git Сайт: <http://bit.ly/MouseBoat>

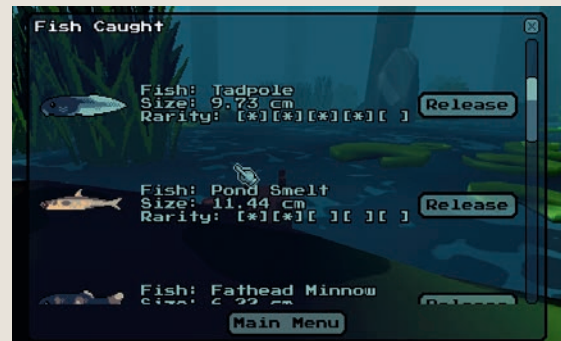
Отыскивая достойный симулятор для HotPicks, мы выбрали нечто отличное от обычных воздушных или гоночных. Как насчет рыбалки на пруду? А для пущей интриги — и странности — как насчет того, чтобы поиграть за мышку, которая сидит в лодке из крошечного листика и плывет по бурным водам в неизвестность мимо листов кувшинок, кривых пенечков [Ред.: — А что здесь симулируется-то?] и зарослей?

Игра движется неторопливо, но постоянно дразнит вас потрясающей перспективой нового улова. Действия в игре очень просты: вы перемещаете свою лодку по пруду с помощью клавиш WASD и внимательно следите за тенями рыб в глубине. Приближаться к ним надо медленно, чтобы не распугать рыб; потом нажмите на левую кнопку мыши и удерживайте ее, для выбора правильного угла. Если ваш

поплавок близок к рыбе, та клюнет на живку. Момент истины наступает, когда вам надо предельно точно всё рассчитать и не дать рыбе сорваться. В нужный момент следует выдернуть удочку и использовать подсказки на экране для расчета силы и направления, чтобы рыба оказалась на земле. Возможно, потребуется не одна попытка, прежде чем у вас что-то получится, но удовольствие от хорошего улова явно стоит трудов.

Mouse Boat ведет запись всех собранных вами видов, и это порой позволяет вам отпустить одну-другую рыбку обратно в пруд. Когда вы плывете, вы также

Приближаться надо медленно, чтобы не распугать рыб.



› Корюшка малоротая — не самый редкий вид. Может, и пусть себе плывет?

можете прикинуть возможный вид рыбы по размеру тени в воде, и это уже стратегия в игре.

Mouse Boat создана с помощью графического движка Godot и имеет простые, но симпатичные 3D-эффекты, которые будут хорошо работать даже на встроенных видеокартах.

Стратегия

Freeciv

Версия: 2.5.5 Сайт: www.freeciv.org

Freeciv — клон с открытым кодом игры *Civilization* от Сиды Мейера [Sid Meier], исключительно популярной стратегии 1996 г. Инициатива с открытым кодом появилась вскоре после и разрабатывается уже 21 год.

В начале игры вы можете выбрать локальный или сетевой многопользовательский режим. Неплохо было бы начать играть локально и привыкнуть к игре перед тем, как начинать соперничать с более зрелыми игроками — но даже и тогда вас, возможно, мигмом уничтожат.

Вы можете выбрать нацию и стиль зданий и, конечно, задать количество игроков ИИ, которые будут делить с вами карту. Затем вы начинаете свою игру на мозаичной карте 6000 г. до н. э. с группой начальных ресурсов: колонистов, рабочих, лучников или иных воинов. Колонисты могут строить поселения, которые со временем превратятся в города, рабочие могут

орошать поля, строить дороги, строить шахты и обогащать почвы.

По мере роста ваших городов вы можете приказывать создавать отряды колонистов и расширять свою территорию, закладывая новые города. *Freeciv* — пошаговая игра, и вы можете быть уверены, что с ней ничего не случится, если вы просто оставите игру работать, пока занимаетесь чем-то другим или просто обдумываете следующий ход.

Уровень сложности во *Freeciv* установлен по умолчанию как Легкий [Easy], и скорее всего, в первой игре вы выживете, и вас не завоюют другие цивилизации. Это также очень хорошо для мирного

Постройте империю, сбалансировав войны и рост экономики.



› Мы превратили нашу цивилизацию в твердыню и защитили ее от угроз с моря!

развития. Если вам нужно больше конфликтов, можете поднять уровень и сосредоточиться на создании воинов, ненадолго оставив гражданские объекты. Однако искусство построения успешной империи заключается в том, чтобы сбалансировать военные действия и рост экономики. Одно всегда влияет на другое — прямо как в нашем мире. *Freeciv* создана с применением подхода клиент-сервер. Вам не обязательно устанавливать сервер, если вы хотите попробовать *Freeciv*: используйте онлайн-сервер на <https://play.freeciv.org> и наслаждайтесь.

Текстовый редактор

Featherpad

Версия: Git Сайт: <http://bit.ly/FeatherPad>

Текстовый редактор — важный аксессуар любого рабочего стола. Простой редактор, чтобы вести записи или изменять настройки в файлах конфигурации, нужен каждому, а разработчикам хочется писать код с выделением синтаксиса и с прочими вспомогательными функциями. Нет нужды говорить о бессмысленности сравнения десятков текстовых редакторов в Linux, которые могут потратить любому вкусу.

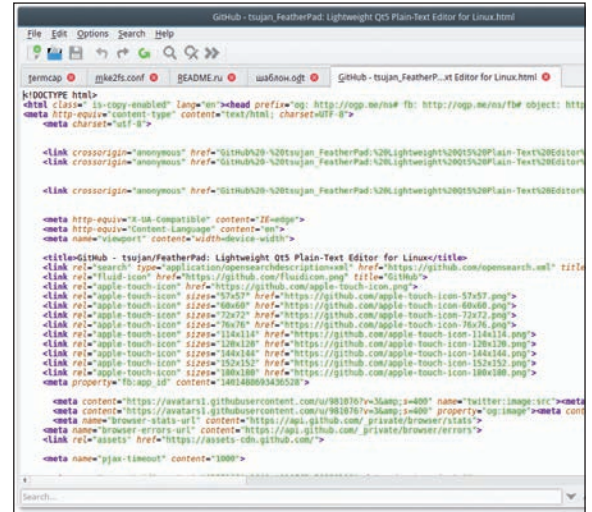
Featherpad — еще один текстовый редактор, обнаруженный в недрах GitHub. Нам очень понравилась идея иметь полнофункциональный редактор на базе *Qt5*, не требующий огромного количества зависимостей или, того хуже, особых компонентов рабочего стола. У нас установлено несколько пакетов *Qt5 devel* (например, *qt5core-devel*); мы запустили `$ make-qt5 && make`, и у нас быстро появился портативный и очень энергичный текстовый редактор. Давайте посмотрим, что он может для вас сделать.

Featherpad открывает каждый документ в отдельной вкладке в одном окне, хотя вы можете распределить вкладки по отдельным окнам.

Приложение поддерживает такие функции, как автоматическое выделение синтаксиса, масштабирование текста, нумерация строк, поиск и замена, автоопределение кодирования строковой переменной, печать и т.д.

Вы также можете перейти в *Options > Preferences* и настроить разные аспекты внешнего вида и работы *Featherpad*, например, темную тему, значки, шрифты и некоторые настройки *Qt*.

Конечно, в плане функций *Featherpad* не может конкурировать с такими редакторами, как *Kate*, но зато он гораздо легче



► **Текстовый редактор *Featherpad* хоть и легковесный, но успешно обрабатывает тонны текстовых файлов.**

и быстрее, а это отличные свойства для тех, кто во главу угла ставит минимализм.

Приложение не слишком хорошо поддерживается многими дистрибутивами Linux, так что вам нужно скачать tar-архив с сайта проекта и скомпилировать его самостоятельно. Правда, в случае *Featherpad* это очень просто. Вы можете использовать редактор в качестве крутого отдельного приложения, отлично дополняющего индивидуальный легковесный рабочий стол, населенный удобными рабочими приложениями *Qt5*.

Редактор на *Qt5*, не требующий огромного количества зависимостей.

Анализатор диапазона звуковых частот

Spek

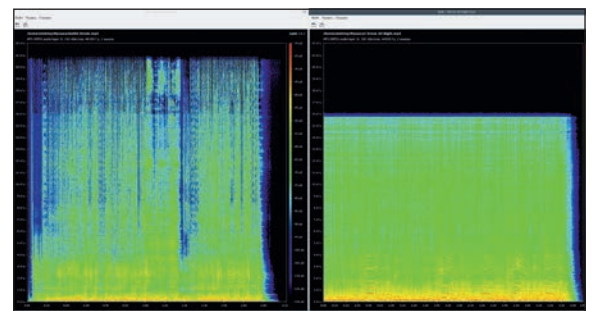
Версия: 0.8.3 Сайт: <https://github.com/alexkay/spek>

Эта небольшая кросс-платформенная утилита — нечто необычное, и потому у нее не слишком много аналогов в Linux. *Spek*, что по-голландски означает «бекон», является программой построения звуковой спектрограммы. Приложение очень простое и предлагает только одну функцию: вы загружаете аудиофайл и через пару секунд видите графическое представление его спектра.

Горизонтальная ось представляет время в секундах, а вертикальная ось — для частоты. На самом деле есть еще и третья измерение: цвет спектрограммы, который изменяется от теплого красного до холодного голубого и представляет амплитуду частоты. Вы всегда можете перестроить спектрограмму со сдвинутым диапазоном амплитуд, нажав `Ctrl + Up/Down`, или с индивидуальными границами амплитуды, нажав `Ctrl + Shift + Up/Down`. На панели инструментов *Spek* только две кнопки: одна — чтобы открыть ваш аудиофайл, а другая — чтобы

сохранить спектрограмму как изображение PNG.

Spek выглядит красочным и привлекательным, но способен быть еще и полезным, когда надо визуальным образом представить параметры вашей записи. Спектрограмма поспособствует в преодолении дефектов речи и поможет людям с ослабленным слухом, а также пригодится для определения и сравнения голосов животных и птиц. Для аудиофилов мира Linux она может показать качество аудиокомпрессии определенного кодека или продемонстрировать возможные дефекты записи с аналогового носителя (например, потрескивание грампластинок). Грубо говоря, эти синие точки и линии в верхней части спектрограммы обычно



► **Верите только глазам? Некачественная запись, многожды перекодированная (слева), выглядит хуже студийного сингла (справа).**

соответствуют высокочастотному шуму или слышимой утечке мощности, обычной для цифровых записей. *Spek* также показывает, что формат OGG звучит лучше, чем MP3, а FLAC лучше обоих.

Spek использует кросс-платформенный *Fftmpeg* для синтаксического анализа аудиофайлов, применяет Быстрое преобразование Фурье для создания графиков и отображает свой GUI посредством *wxWidgets*. В Сети очень мало бинарников *Spek*, так что приготовьтесь компилировать этот удобный инструмент из исходника. **LXF**

Может показать качество аудиокомпрессии определенного кодека.

На диске

Дистрибутивы, приложения, игры, книги и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



Дистрибутивы

Э то все пользователи Linux становятся консервативнее, или только я? Раньше нам хотелось новых программ, и немедленно! А сейчас, похоже, хочется избегать новых идей: семантического рабочего стола KDE, Gnome 3, Unity, и больше всего — по крайней мере, сейчас — *systemd*. Громогласное меньшинство орет и топает ногами при каждом появлении чего-то «не в стиле Unix», чем бы оно на самом деле ни было. Забавно, что всё продолжает работать, и подавляющее большинство пользователей всё равно довольны. Ubuntu по-прежнему популярнейший домашний дистрибутив, несмотря на весь гнев насчет перехода на Unity. Теперь Ubuntu перешел на *systemd*, но большинство пользователей, вероятно, и не заметят разницы.

Что случилось с сообществом Linux? Когда мы из авантюристов превратились в реакционеров? Загрузите Fedora 25 с DVD этого месяца, в нем полно того, чего люди, судя по всему, так боятся: Gnome 3, *systemd* и *Wayland*, да плюс *PulseAudio*, хотя он работает хорошо и делает то, чего от него хочет большинство. Разве не таким должен быть дистрибутив? Или нам надо организовать кампанию за возвращение *twm* в качестве рабочего стола по умолчанию?

Neil

» Важно ВНИМАНИЕ!

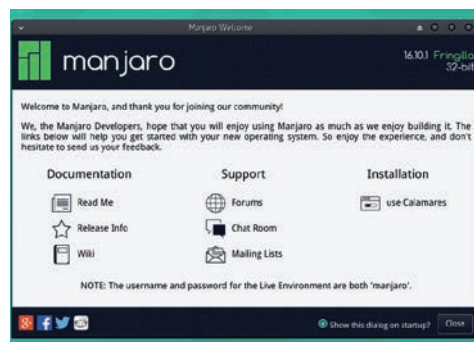
Порченые диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Дружелюбный дистрибутив на Arch

Manjaro 16.10.2

Многие дистрибутивы появились как ответвления от других. Это и неплохо; ну зачем изобретать велосипед? Даже SUSE начинал именно так. Однако сейчас уже столько ветвей Ubuntu, что они заполнили весь мир (или SourceForge), и приятно увидеть дистрибутив, использующий другую базу. Arch Linux известен своей гибкостью и легким весом, и оба этих фактора — неплохая отправная точка. Судя по всему, производители Manjaro с этим согласны, потому что Manjaro тоже такой. Есть версии с рабочими столами KDE и *Xfce*; и мы включили 32-битную версию *Xfce* для тех, у кого старое оборудование. Логин: manjaro, пароль: manjaro.



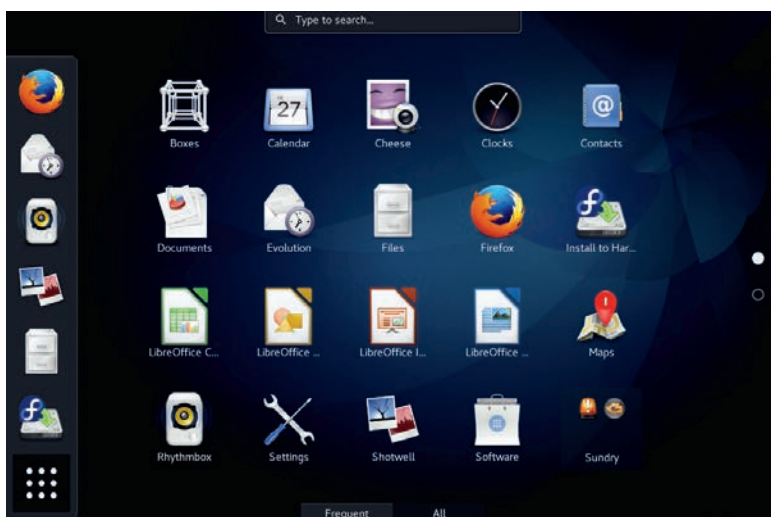
Еще один дружелюбный дистрибутив

Fedora 25

С трудом верится, что Fedora достигла версии 25 — мы помним первый релиз Fedora Core 1; ладно, номера версий *Chrome* и *Firefox* его удваивают, но мы здесь говорим о серьезной схеме номеров версий.

Fedora — это поддерживаемый сообществом дистрибутив от Red Hat; именно здесь и испытываются новые технологии, пока они не сделаются достаточно зрелыми для включения в корпоративный продукт. Именно здесь мы пробуем что-то новое. Fedora первым стал использовать *systemd* (выждем, пока утихнет

пламя гнева...), и они подталкивают разработку *Wayland*, нового графического движка. Хотя Fedora включил *Wayland* в недавние релизы, сейчас он используется по умолчанию, если ваш рабочий стол — Gnome. Возможно, вы уже использовали *Wayland*, если загрузили Fedora с LXF DVD. Не переживайте, если ваше оборудование не поддерживается: в этом случае Fedora откатывается к *X.org*, так что всё по-прежнему будет работать. Информация для входа в систему: имя пользователя — *liveuser*, а пароль пустой.



» Самый простой способ запустить лапы в *Wayland*.

LIVE-ДИСК: ЗАГРУЗИ > ОПРОБУЙ > УСТАНОВИ!

FEDORA 25 [64-битная]

Обновиться до самой свежей и популярной ОС Linux

» Графика нового поколения на Wayland » Новейшее ядро 4.8
» Интерфейс GNOME 3.22 » Поддержка жестких дисков по всей ОС

openSUSE 42.2 [64-битная]

Настольная система класса предприятия

» Включено 1000+ приложений
» Стабильное ядро 4.4 LTS » Рабочий стол KDE Plasma 5
» Хранилище нового поколения на Btrfs

Bodhi 4.0

Красивая сборка на базе Ubuntu
+ рабочий стол Enlightenment

antiX 16

Быстрый, легковесный
и идеальный для старых ПК

LINUX LIVE-ДИСК: ГОТОВ К РАБОТЕ
ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX

Новичок в Linux? Начните отсюда!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент *MS Office*?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

Дружелюбный дистрибутив без systemd

antiX 16

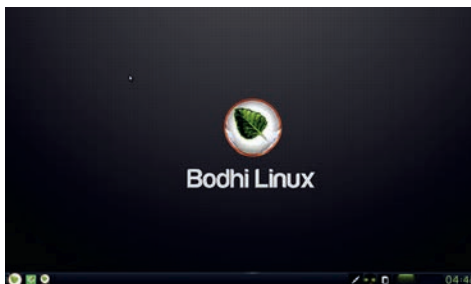
AntiX, немного похожий на Bodhi Linux — один из дистрибутивов, которые остаются вне зоны особого внимания. Он легковесный — у нас здесь 32-битная версия, в высшей степени подходящая для старого оборудования; кроме того, он хорошо работает в качестве live CD для администрирования. AntiX использует менеджер окон *IceWM*, который существует уже далеко не первый год и имеет преданных поклонников. *IceWM* — не полная среда рабочего стола, но очень добротный, и особенно удобен для слабых компьютеров. Информация для входа в систему: имя пользователя — demo, пароль — demo.



Легковесный настольный дистрибутив

Bodhi Linux 4.0

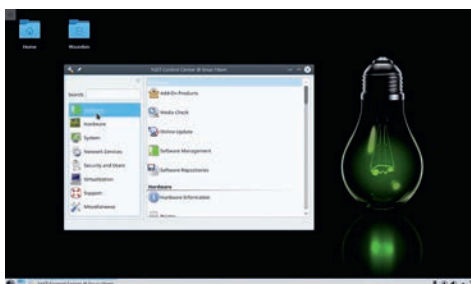
На сей раз это 64-битная версия, сменившая рабочий стол *Enlightenment 17* на его ответвление *Moksha*. Это легковесный рабочий стол, но отнюдь не серый или скучный. *Moksha* — хорошо настраиваемая среда, и вы можете изменить большинство ее аспектов. Хотя это может потребовать чуть больше работы, чем с KDE, вы получите намного более быстрый и отзывчивый рабочий стол, притом расширяемый добавочными модулями. Информация для входа в систему: имя пользователя — bodhi, пароль пустой.



Гибридный Linux уровня предприятия

openSUSE Leap 42.2

openSUSE больше не выпускает дистрибутивов live, предоставляя только сетевой установочный диск или полноценный загрузочный диск; этот второй и занял сторону нашего диска целиком. Пошаговое руководство по установке см. на другой стороне диска. Сначала создается базовая система, а затем устанавливается выбранный вами рабочий стол (или вообще никакой). Загрузив свой новый openSUSE, вы можете обратиться к *YaST* — универсальному пакету настройки и управления. Имя пользователя — root, пароль пустой (просто нажмите Enter). **LXF**



И еще!

Системные инструменты

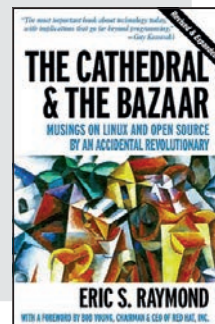
Главное

- Checkinstall** Установка tar-архива с помощью менеджера пакетов.
- GNU Core Utils** Основные утилиты, которые обязаны быть в каждой операционной системе.
- Hardinfo** Инструмент тестирования системы.
- Kernel** Исходный код самого последнего стабильного релиза ядра.
- Memtest86+** Проверьте ОЗУ на предмет неисправностей и сбоев.
- Plop** Простой менеджер загрузки для запуска ОС с CD, DVD и USB.
- RaWrite** Создавайте загрузочные диски в MS-DOS в Windows.
- SBM** Независимый от ОС менеджер загрузки с несложным интерфейсом.
- WvDial** Соединитесь с Интернетом через телефонный модем.

Чтение

Книжная полка

- Расширенное руководство по скриптам Bash** Изучите написание скриптов еще лучше.
- Руководство Bash для начинающих** Овладейте написанием скриптов *Bash*.
- Руководство по созданию скриптов Bourne Shell** Начните осваивать скрипты оболочки.
- The Cathedral and the Bazaar [Собор и Базар]** Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.
- Справочник администратора Debian** Базовое руководство для системных администраторов.
- Введение в Linux** Удобное руководство со множеством подсказок для начинающих пользователей Linux.
- Словарь Linux** Терминология Linux от А до Я.
- Linux Kernel in a Nutshell [Ядро Linux в Двух Словах]** Введение в ядро, написанное великим магистром ядра Греггом Кроа-Хартманом [Greg Kroah-Hartman].
- Руководство системного администратора Linux** Контролируйте свою систему.
- Обзор инструментов** Полный обзор инструментария GNU.



Пропустили номер?

LINUX
FORMAT
Главное в мире Linux

Закажите его на сайте www.linuxformat.ru в «ГНУ/Линуксцентре»!
Журналы доставляются и в печатной, и в электронной форме,
так что получение нужного вам выпуска LXF может занять всего
пару минут с момента открытия браузера!

LXF218 Январь 2017



- » Приручение Яка Он же — Ubuntu 16.10
- » Новые, хромовые Выбираем хромбук
- » Помирать нам рановато Шанс 32-битных машин
- » Телефон в терминале Управляем с Linux-ПК

LXFDVD: BunsenLabs, Calculate Linux, GeckoLinux, PeiScanPC, Ubuntu, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_218/
PDF-версия: shop.linuxformat.ru/elxf_218/

LXF217 Декабрь 2016



- » Сплошная малина Всё про Raspberry Pi
- » Болтунам Без находок для шпиона
- » Блогоедания Свежая WordPress
- » Интернет вещей Строим домашних, построим и дом

LXFDVD: Android-X86, Elementary OS, Korora, Mint KDE, Maui, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_217/
PDF-версия: shop.linuxformat.ru/elxf_217/

LXF216 Ноябрь 2016



- » Береженого и Бог бережет Взлому — не быть!
- » Какая штука... Чем печатать 3D-объекты
- » Кофеварка на Raspberry Pi Рулим температурой
- » Блог на Phoenix И пуская комментируют

LXFDVD: 4MLinux, Kubuntu, Lubuntu, Xubuntu, Mint 18 Xfce, OpenMandriva Lx 3.0, SELKS 3.0, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_216/
PDF-версия: shop.linuxformat.ru/elxf_216/

LXF215 Октябрь 2016



- » Четверть века с Туксом Внутри ядра
- » Управляем деньгами Вы как финансовый туз
- » Всё о дронах От сборки до калибровки
- » Музыка Linux Демо для гитаристов из подворотни

LXFDVD: Fedora, Neon, ParrotSecurityOS, Superb Mini Server, Ultimate Boot CD, Voyager, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_215/
PDF-версия: shop.linuxformat.ru/elxf_215/

LXF213/214 Сентябрь 2016



- » Сервер на дому Настройка проста как никогда
- » Web-серфинг со свистом Браузеры эконо-класса
- » Персональная соцсеть Минута Интернет
- » Селфи с эффектами Я ль на свете всех милее?

LXFDVD: antiX, Debian, ClearOS, Mint, NethServer, OwnCloud, Peppermint, Ubuntu Server, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_213-214/
PDF-версия: shop.linuxformat.ru/elxf_213-214/

LXF212 Август 2016



- » Много-много приложений А вы о них и не знали!
- » У меня совещание Менеджеры персональной информации
- » Ловушки SIP-телефонии Как в них не упасть
- » Docker на Pi Контейнеры — это модно и практично

LXFDVD: 4MLinux, Kubuntu, Lubuntu, Xubuntu, Mint 18 Xfce, OpenMandriva Lx 3.0, SELKS 3.0, 12 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_212/
PDF-версия: shop.linuxformat.ru/elxf_212/

Подпишитесь на печатную версию журнала на www.linuxformat.ru/subscribe
или www.linuxcenter.ru, и получите электронную версию в подарок!

Телефоны отдела подписки

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



Linux Format ВКонтакте:
vk.com/linuxform

Страница 1

» Содержание



ДИСТРИБУТИВЫ

Fedora 25 (64-битный)
Дистрибутив, разработанный сообществом при поддержке проекта Fedora и принадлежащим компаниям Red Hat. Содержит ПО, распространяемое под лицензией свободного и открытого исходного кода, и стремится быть на переднем крае современных технологий, делая упор на инновации, интеграцию новых технологий на ранней стадии и тесное сотрудничество с сообществами «преджелезачего» ПО Linux. Рабочий стол по умолчанию — Gnome, с интeрфейсом Gnome Shell.

Void 4.0 (64-битный)

Эlegantный и легкий дистрибутив на базе Ubuntu, с рабочим столом Enlightenment. Предлагает минималистский подход, применяя модульность и настраиваемость.

Manjaro 16.10.2 (32-битный)

Быстрая, удобная настольная ОС на базе Arch Linux с рабочим столом Xfce, с интуитивным процессом установки, автоматическим обнаружением оборудования и стабильным графиком обновлений.

antiX 16 (32-битный)

Быстрый, легкий и несложный в установке дистрибутив на базе стабильной ветки Debian, с легковесным рабочим столом IceWM.

НОТРИКС

Devilspine 0.47 Утилита окон
www.devilspine.org
Driftnet 5.3 Организатор изображений
www.driftnet.org

Featherpad (Gn) Текстовый редактор
<http://dillo.kde.org/featherpad/>

Freebsd 2.5 Иг-стратегия
www.freebsd.org

Hamdrank 5.1.1 Финансовое приложение
<http://hamdrank.free.fr/>

Kinsader 2.5 Менеджер файлов
<http://www.kinsader.org>

Klip 0.6.1 Инструмент резервного копирования
<http://www.klip.org/>

Mouse Vrat (Gn) Иг-ра, симулятор рыбалки
<http://dillo.kde.org/mousevrat/>

Neutral Epilator (Gn) Оптимизатор изображений
<http://dillo.kde.org/neutral-epilator/>

Spek 0.8.3 Анализатор диапазона звуковых частот
<http://dillo.kde.org/spek/>

Tuxguitar 1.3.2 Музыкальное приложение
<http://dillo.kde.org/tuxguitar.com.de/>

Обновление на обороте »

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

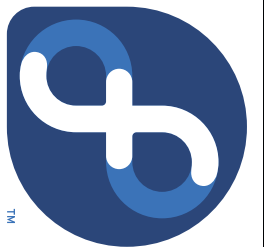
- » **имя_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: “tgz” — это сокращение от “tar.gz”;
- » **имя_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: dists@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

LIVE-ДИСК: ЗАГРУЗИ И ОПРОБУЙ И УСТАНОВИ!



ФЕДОРА 25
64-битный

Обновитесь до самой свежей и популярной ОС Linux

- » Графика нового поколения на Wayland
- » Новейшее ядро 4.8
- » Интeрфейс Gnome 3.22
- » Поддержка эмодзи по всей ОС

Настольная система класса предприятия

» Включено 1000+ приложений
» Стабильное ядро 4.4 LTS
» Рабочий стол KDE Plasma 5
» Хранилище нового поколения на Btrfs

openSUSE 42.2
64-битный

Void 4.0



antiX 16



Красивая сборка на базе Ubuntu
+ рабочий стол Enlightenment 64-битный

Быстрый, легковесный и идеальный для старых ПК 32-битный

Linux Format LIVE-ДИСК: ГОТОВ К РАБОТЕ ВСЕ НЕОБХОДИМОЕ ДЛЯ СТАРТА В LINUX



Содержание



Страна 2

ГЛАВНОЕ

- Checkinstall
- Coreutils
- Hardinfo
- Kernel
- Memtest86+
- Plop
- SBM
- WVDial

ПОМОЩЬ

- Руководство новичка
- Руководства
- Ответы
- ЧаВо (FAQ)

ДОКУМЕНТАЦИЯ: 12 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

- Linux Kernel in a Nutshell** Описание ядра Linux, созданное одним из его выдающихся разработчиков — Греггом Крау-Хартманом [Greg Koorh-Hartman]

- Advanced Bash Scripting Guide** Подробное руководство по программированию на Bash
- Bash Guide for Beginners** Руководство по Bash для начинающих
- Boome Shell Scripting** Начальное руководство по программированию на Bash
- The Cathedral and the Bazaar** Классический текст Эрика Реймонда [Eric S Raymond] «Собор и базар»
- The Debian Administrator's Handbook** Руководство администратора, написанное разработчиками Debian
- Dive Into Python** Учебник по программированию на Python
- Intro to Linux** Начальное руководство по Linux
- Linux Dictionary** Словарь Linux, объясняющий специфическую терминологию
- Rule** Знаменитый справочник
- System Administrators Guide** Руководство по базовому администрированию Linux
- GNU Tools Summary** Руководство по работе в командной строке и обзор основных утилит GNU

ДИСТРИБУТИВЫ

- openSUSE 42.2** (64-битный)
- Установочный диск openSUSE является собой программный репозиторий. Установка создает базовую систему, а затем устанавливает выбранную вами среду рабочего стола (или вообще никакой). Для установки любых выбранных вами программ применяется универсальный пакет настроек и управления YaST. Имя пользователя — root, пароль — пуст.

Все дистрибутивы представлены ISO-образами, который можно записать на отдельный носитель, и загрузить в live-режиме прямо с LXF DVD. У всех присутствует возможность установки на жесткий диск.

Пожалуйста, перед использованием Аджнотного Диска ознакомьтесь с инструкцией, опубликованной в журнале на стр. 109!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕЯТЕЛЬНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех этапах производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать аппаратный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, с сайта разработчика узнайте все о вашей конкретной ситуации.

Тираж издательства ООО «Марком», 188652, Россия, Ленинградская область, Всеволожский р-н, дер. Юрки, Школьная ул., 7-а. Лицензия ИМПТР ВАФ 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 размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Для записи ISO-образа вам осталось набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени root и выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажмите на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

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!



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж печатной версии 2000 экз., распространение электронной версии 30000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Даниил Кривошеин, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

Редактор диска

Александр Баракин

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Андрей Смирнов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатель

ООО «Линукс Формат»

Отпечатано в типографии ООО «ЛД-ПРИНТ»

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. 6/н, лит. Ф
Тел. (812) 462-8383, e-mail: office@ldprint.ru
Заказ 15482

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Нейл Мор [Neil Mohr] neil.mohr@futurenet.com

Научный редактор Джонни Бидвелл [Jonni Bidwell]

jonni.bidwell@futurenet.com

Выпускающий редактор Крис Торнетт [Chris Thornett]

chris.thornett@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоза

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Матс-Тэге Аксельссон [Mats Tage Axelsson], Джонни Бидвелл [Jonni Bidwell], Нейл Ботвик [Neil Botwick], Джолион Браун [Jolyon Brown], Шон Кэмерон [Sean Cameron], Нейт Дрейк [Nate Drake], Таджиндер Калси [Tajinder Kalsi], Джон Найт [John Knight], Кристофер Ливингстон [Christopher Livingston], Джон Мак-Канн [John McCann], Ник Пирс [Nick Peers], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Зак Стори [Zak Storey], Александр Толстой [Alexander Tolstoy], Михаил Цукалос [Mihalis Tsoukalos], Евгений Балдин, Андрей Гондаренков, Павел Емельянов, Алексей Федорчук, Максим Черепанов, Лада Шершова

Иллюстрации Шейн Коллиндж [Shane Collinge]

Иллюстрация с обложки www.magictorch.com

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 (0) 1604 251045, email: linuxformat@myfavouritemagazines.co.uk

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

Тел. +7 (812) 309-0686, e-mail: info@linuxformat.ru

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права защищены. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].
"GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт www.futureplc.com.

В мартовском номере

Настала пора... Удрать с Google!

Заберите свои облака обратно: мы зарядим лучшие открытые сервисы и серверы, которые позаботятся о ваших данных.

«Другое» ядро

Никак вы гнусный перебежчик? Сравниваем лучшие варианты BSD, чтобы показать вам, зеленее ли трава у соседей.

Ноутбук с Linux

Пусть ваш Linux протянет подольше, благодаря нашей статье о переделке ядра, которая подружит его с батареями.

Ваш друг Drupal

Даже не имея своего домена, вы можете создать сайт у себя на компьютере — и запросто администрировать его.

Содержание будущих выпусков может меняться — вдруг нас повяжут за вольнодумство...

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)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

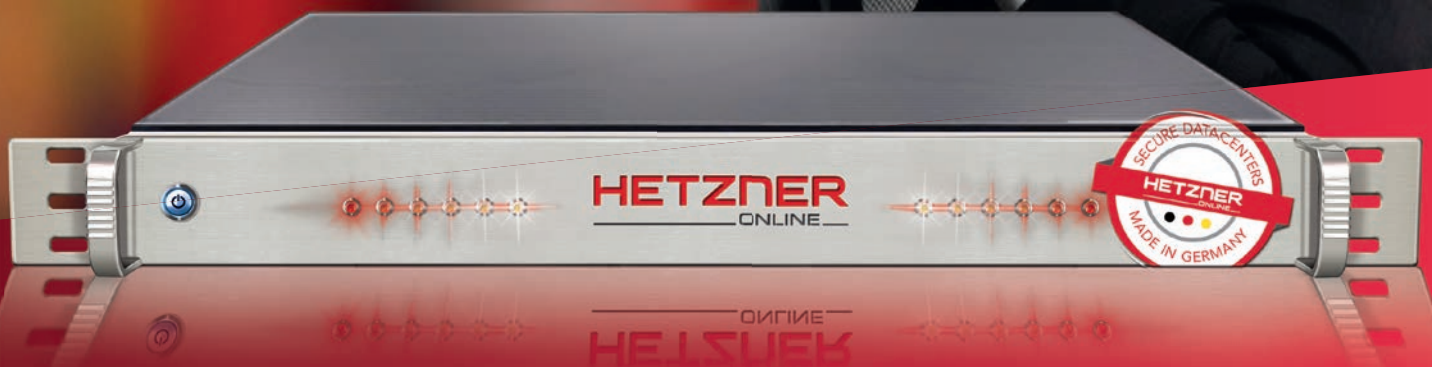
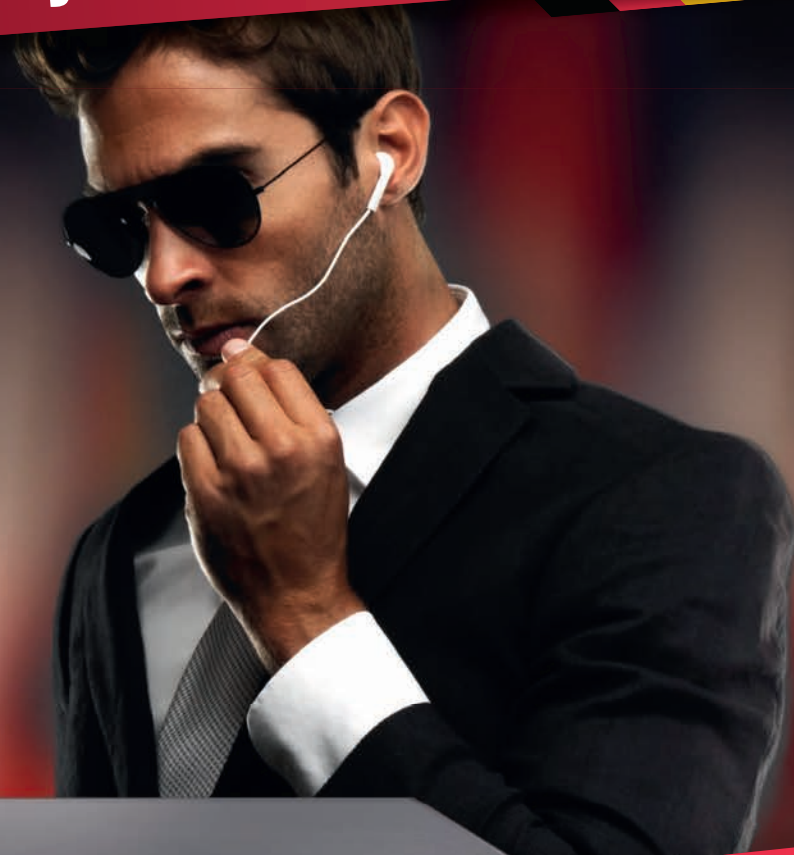
Linux  center

HETZNER
ONLINE

We're always on guard

Security made in Germany

Все цены указаны без учёта НДС и регулируются условиями предоставления услуг компанией Hetzner Online GmbH. Цены могут быть изменены. Все права защищены соответствующими производителями.



Выделенный сервер PX61-NVMe

Intel® Xeon® E3-1275 v5
Quad-Core Skylake Processor
64 ГБ DDR4 ECC RAM
2 x 512 ГБ NVMe Gen3 x4 SSD
1 Гбит/с гарантированно
100 ГБ место для резервных копий
30 ТБ трафик*
Без минимального контракта
Установка 8600 рублей

4300
рублей в месяц

Хранилище данных в Германии

Храните свои данные безопасно в дата-центрах Hetzner Online, находящихся в Германии и соответствующих высоким немецким стандартам по защите данных.

Расположенные в Nuremberg и в Falkenstein/Vogtland два дата центр парка задают стандарт будущего и предоставляют великолепную защищённую инфраструктуру для Интернет-бизнеса.

* 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 86 руб. за каждый дополнительный ТБ.

www.ru.hetzner.com