

НА ВАШЕМ БЕСПЛАТНОМ DVD  
Mint 16 с рабочим столом Mate,  
Fedora 20 и сверх того!

# LINUX FORMAT

Главное в мире Linux

Март 2014 № 3 (181)



**Android — на волю!**

» Освободите свои телефоны и планшеты от кандалов Google **с. 26**

# ОСВОИМ ЗАМЕЧАТЕЛЬНЫЙ RASPBERRY PI

Из бесцветного существа станьте элитой, под руководством двадцати страниц советов и учебников

- » Программируем на Python
  - » Лучшие проекты с Pi
  - » ЖК-светодиоды под контролем GPIO
- И не только!



12 страниц знакомят с языками программирования  
**АКАДЕМИЯ КОДИНГА**

**Ada** Код для критических задач **с. 84**  
**Scratch** Освоимся и построим игру **с. 88**

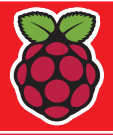


**Человек Pi**  
«Сдуру я сказал: «Оборудование у меня есть, пришлите чипы, а я спроектирую плату»  
**Пит Ломас** о создании Raspberry Pi **с. 44**

**ПЛЮС!**  
Видеокамеры и принципы СПО **с. 48**

Также в номере...

**Лучшие для Pi**  
Отставьте дистрибутив Raspbian ради мощных альтернатив **с. 52**



**Секреты SteamOS**  
Поиграйте в ОС от Valve, добавив оборудования и VM **с. 72**



**Внутри GCC**  
Компилятор, движущий мир Linux **с. 56**

Сисадминам  
**Сервер журнала**  
» Централизованное журналирование — это выход

Разработка  
**Строим Git-сервер**  
» Ответьтесь в свой личный репозиторий

Потоковое аудио  
**Subsonic**  
» Учредите собственный сервис музыкального вещания

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

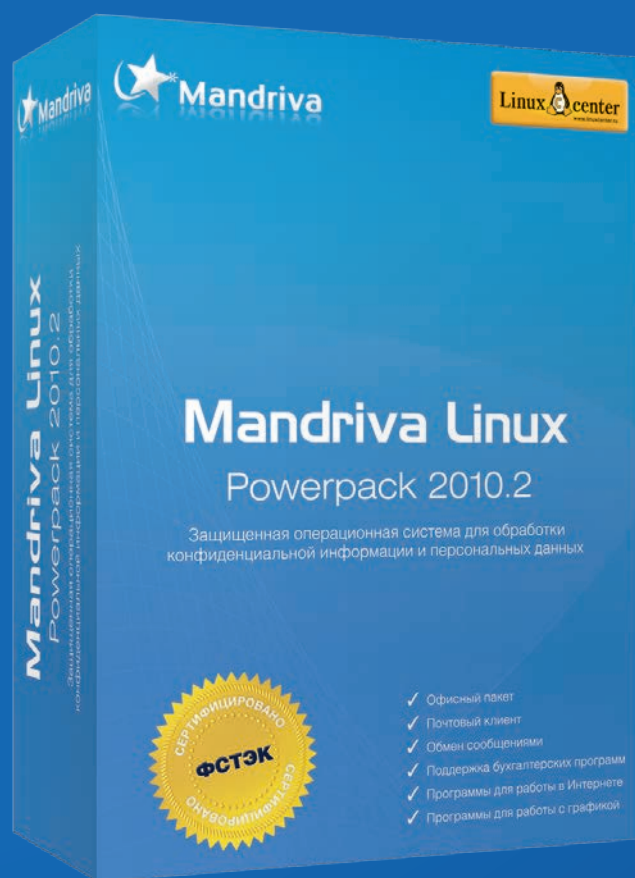
**Linux center**  
www.linuxcenter.ru



# Mandriva Linux

## Powerpack 2010.2

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



Простота установки, работы и администрирования

Легкость освоения новичками в GNU/Linux

Широчайший спектр поддерживаемого оборудования

Цена 4990 руб.

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



[www.linuxcenter.ru/shop/certified\\_fstek/mandriva\\_fstek/mandriva\\_2010-2\\_pp\\_certific\\_fstek/](http://www.linuxcenter.ru/shop/certified_fstek/mandriva_fstek/mandriva_2010-2_pp_certific_fstek/)

Москва  
+7 (499)

**271-49-54**

Санкт-Петербург  
+7 (812)

**309-06-86**

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)

## Что мы делаем

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



## Кто мы

В этом номере мы рассматриваем, как в Android отстроиться от Google. И мы спросили наших авторов: что бы они хотели обезгуглить?



**Крис Торнетт**  
Google хочет свить Nest, вот это я бы и убрал. Доступ к информации обо всех гаджетах у вас дома? Нет уж, спасибо.



**Мэтт Хэнсон**  
Я убежден, что внутри Google сидит офисный автомат, отслеживающий мой выбор горячих напитков.



**Эфраин Эрнандес-Мендоса**  
Когда-нибудь я засну этот Google в салсу и выужу его оттуда поварешкой.



**Лес Паундер**  
Изгнать Google из поиска и вернуться к поисковым движкам с именами из семейства собачьих, вроде Lycos и Dogpile.



**Маянк Шарма**  
Я бы убрал его из Очков Google — вышла бы полезная вещь для людей со слабым зрением. А я сколотил бы миллионы!



**Нейл Боткин**  
Я рад, что Google никогда не процветал в Мордоре. Кабы у Саурана был не Bing, он уж нашел бы то самое кольцо.



**Джон Лэйн**  
Мне бы датчик с запуском при загрузке, чтобы автоматически выбирал мне сорочку в тон к моему новому рабочему столу.



**Джульетта Кемп**  
Gmail был бы почти приличным почтовиком, если бы Google не шпионил за тем, что вы пишете.



**Шашанк Шарма**  
Отменить бы в Google рекламу, может, меньше стало бы всяких гнусных целевых объявлений.



**Дэвид Хейвард**  
Как-то побаиваюсь я этого Google: там небось целая армия роботов. И, чего доброго, все как Шварценеггер.



**Ричард Смедли**  
Ликвидировал бы гуглы-дудлы. До чего ж они затягивают! Я шесть часов уробил на один такой с Доктором Кто.



**Валентин Сеницын**  
Как насчет Интернета без Google? У меня все еще где-то лежит желтыйкий справочник 1997 года выпуска.



## Цели и средства

» Два месяца назад я высказал предположение, что причина успеха краудфандинговых проектов в том, что целью их авторов является не получение прибыли любой ценой, а создание чего-то полезного — устройства или программы — с заранее обещанными характеристиками. Возьму на себя смелость распространить это утверждение и на другие проекты свободного ПО и «железа».

На страницах данного номера вы найдете подтверждение этой мысли, высказанное участниками трех совершенно не похожих друг на друга проектов.

Обратите внимание, что ни основатели компании Cugpus, ни автор «железа» Raspberry Pi Пит Ломас, ни президент компании Elphel Андрей Филиппов не оценивают свои успехи в традиционных «уолл-стритовских» метриках — занятой доле рынка или месте в Gartner Magic Quadrant. Они говорят о реальных достижениях в реальной технологии.

Цели были определены в технических терминах, а важнейшим средством их достижения стало использование свободных лицензий. Далее «хорошая вещь начинала продавать сама себя», принося и деньги, и признание. Не могу судить о камерах Elphel, уж слишком специфична область их применения, но GCC и Raspberry Pi стали несомненными лидерами в своем классе. Количество их пользователей продолжает расти, и эта оценка успеха много точнее оценочек бизнес-аналитиков.

**Кирилл Степанов**

Главный редактор

» [info@linuxformat.ru](mailto:info@linuxformat.ru)

## Как с нами связаться

Письма для публикации: [letters@linuxformat.ru](mailto:letters@linuxformat.ru)

Подписка и предыдущие номера: [subscribe@linuxformat.ru](mailto:subscribe@linuxformat.ru)

Техническая поддержка: [answers@linuxformat.ru](mailto:answers@linuxformat.ru)

Общие вопросы: [info@linuxformat.ru](mailto:info@linuxformat.ru)

Проблемы с дисками: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

Вопросы распространения: [sales@linuxformat.ru](mailto:sales@linuxformat.ru)

Сайт: [www.linuxformat.ru](http://www.linuxformat.ru), группа «ВКонтакте»: [vk.com/linuxform](https://vk.com/linuxform)

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

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

9-10 АПРЕЛЯ | САНКТ-ПЕТЕРБУРГ  
В РАМКАХ «ПРЕМИИ HR-БРЕНД 2013»

# X ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ ПО HR-БРЕНДИНГУ

**СПЕШИТЕ ЗАБРОНИРОВАТЬ МЕСТА:**

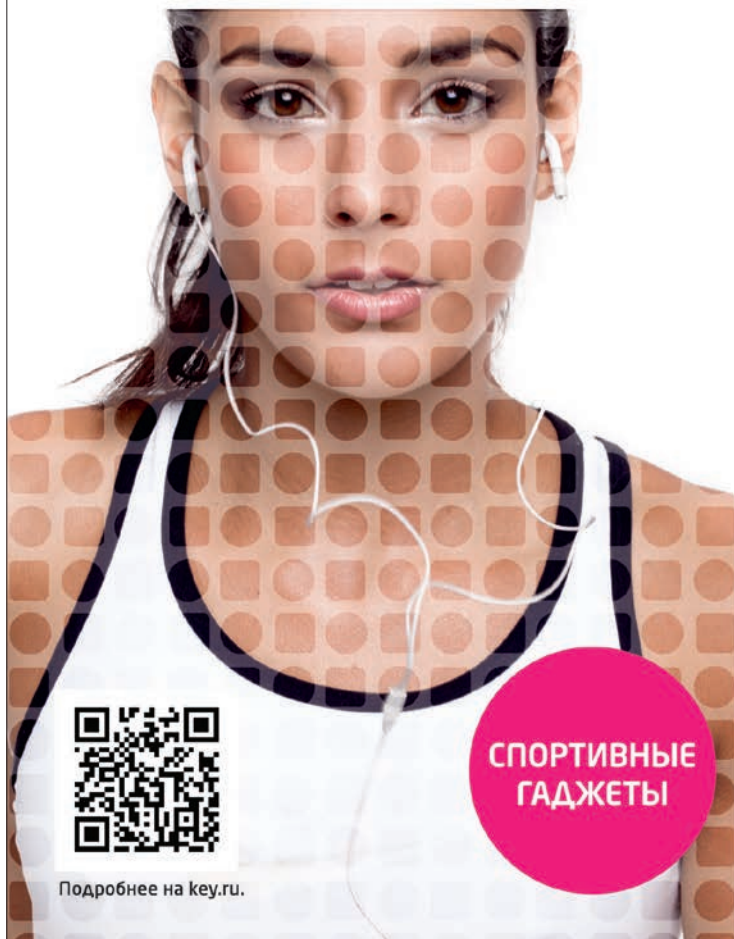
звоните своему персональному менеджеру НН

**(+7 812 607 77 29)**

или пишите по адресу [hrconf@hh.ru](mailto:hrconf@hh.ru)



# ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Подробнее на [key.ru](http://key.ru).

СПОРТИВНЫЕ  
ГАДЖЕТЫ

Экшн - камера



Экшн-камера Sony HDR-AS30

Smart - часы

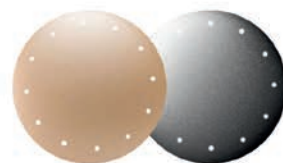


Smart-часы Cookoo Watch

Smart - здоровье



Браслет Jawbone UP



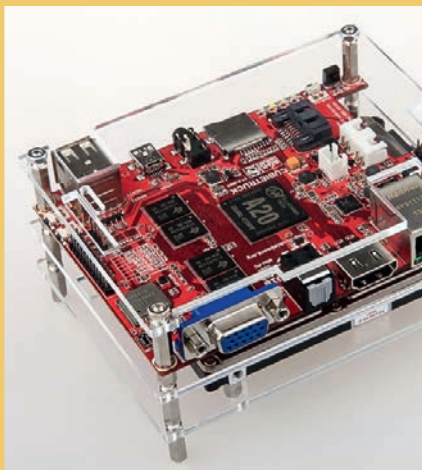
Шаромер Misfit Shine

Товар сертифицирован. Количество товара ограничено.



Одноплатный компьютер Cubieboard3

# Cubietruck



## ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

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

литиевого элемента питания

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

## КОМПЛЕКТАЦИЯ

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

3800  
руб.

[www.linuxcenter.ru/shop/ops\\_hard/cubieboard/cubietruck/](http://www.linuxcenter.ru/shop/ops_hard/cubieboard/cubietruck/)

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



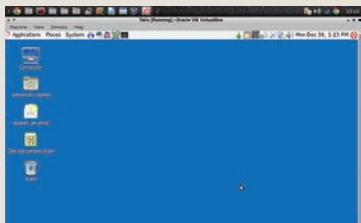
# Содержание

Даже самый крошечный компьютер способен изменить курс в будущее...

## Обзоры

**Ultimate Edition 3.8** ..... 16  
“Ultimate” значит «дальше ехать незачем». Заявка серьезная, но оправдана ли она? Удастся ли дистрибутиву ей соответствовать?

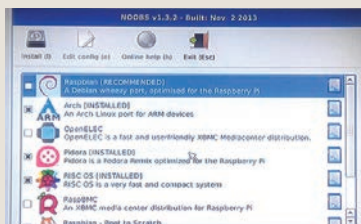
**Tails 0.22** ..... 17  
С этим дистрибутивом, заточенным на приватность, вы будете чувствовать себя в безопасности в Сети. И интерфейс у него очень прост.



➤ Tails 0.22 костями ляжет, чтобы защитить вас и ваши данные онлайн.

**Perzo.com** ..... 18  
Мы рассмотрели онлайн-сервис, прикрывающий от глядаеав ваши чаты. Жаль, что к нему не подключались Фродо с Гэндальфом.

**NOOBS 1.3.2** ..... 19  
Благодаря *NOOBS* вы запросто установите и настроите на Raspberry Pi любимый дистрибутив, сэкономив массу времени для ценных изысканий.



➤ Легко и быстро устанавливается целый букет дистрибутивов, дружелюбных к Pi.

**Penguin Wee 4<sup>th</sup> Gen Desktop** ... 20  
Компактный и мощный настольный ПК запустит вам Linux вашей мечты прямо из коробки.

**Devolvo LiveCam** ..... 22  
Беспроводная web-камера на основе IP, которая претендует на работу в Linux.

**Broken Age** ..... 23  
Классический квест сохранил почитателей. Стоило выставить проект на Kickstarter, как ему накидали на развитие 3 миллиона.

**Извлеките максимум из чудесной машинки-крошки и создайте свой первый проект** с. 36

## Осваиваем Raspberry Pi

**Сравнение: Клиенты VoIP** с. 30



## Люди говорят



Эбен Айттон задумал создать макетную плату для обучения программированию

Пит Ломас Происхождение Raspberry Pi с. 44

# На вашем бесплатном DVD



## Linux Mint 16

» Версия с Mate

## Fedora 20

» С рабочими столами KDE и Gnome

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



### Ищите в этом номере



## Освободим Android ..... 26

Скинем оковы проприетарности.

## Дистрибутивы для Raspberry Pi ..... 52

Они уже плодятся, как кролики.

## GCC: Привет всем ..... 56

Без него, может, ничего бы и не было.



### Академия кодига

#### Ada, дочь поэта ..... 84

Изучаем сверхнадежный язык для разработки VIP-приложений, разработанный для Министерства обороны США и названный по имени первого в мире программиста Ады Лавлейс.

#### Мультяшный Scratch ..... 88

Scratch — язык программирования для самых маленьких. Он по плечу даже шестилеткам! Освоим его и создадим на нем незатейливую (для начала) игру-аркаду.

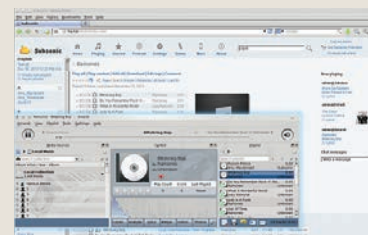
```

juliet@inspiral: ~/coding/ada
File Edit View Search Preferences Tabs Help
┌── juliet@inspiral:~/coding/ada ──
└─ juliet@inspiral:~/coding/ada$ ./habits
Write code more often
Floss teeth
What habit would you like to add?
Stare out of window extensively
juliet@inspiral:~/coding/ada$ ./habits
Write code more often
Floss teeth
Stare out of window extensively
What habit would you like to add?
    
```

### Учебники

#### Вещание музыки ..... 68

**Subsonic** Создадим собственный музыкальный сервер, чтобы покончить с надоевшим бардаком в фонотеке.



» **Subsonic:** ваша музыка повсюду.

#### SteamOS Знакомимся ..... 70

Дистрибутив с упором на игры: что это такое и с чем его едят. А также — как записать его в виртуальную коробку (по-нашему, VirtualBox).

#### RSS-автоматика ..... 74

**Новости идут к вам** Настроим ленты RSS, чтобы всегда находиться в курсе текущих событий: события будут притекать сами.

#### Системные журналы Мне видно все ..... 76

Подальше положишь — поближе возьмешь: вынесем лог-файлы на сервер. И всегда будем знать, что стряслось даже с удаленным ПК.

#### Git Стрим сервер ..... 80

Нечего взваливать всю работу на Github: вы и сами в состоянии справиться с ведением проекта.

### Постоянные рубрики

#### Новости ..... 6

Великая Британия бортанула MS Office, SUSE берется за тестирование, Черная Утка выставляет рейтинг, РАСПО меняет президента, Google наградил ScratchDuino, выпущены Mageia и Moonlight, и прошла FOSDEM.

#### Новости Android ..... 24

Компания HTC покушается на Nexus, Firefox двинулся на Android, карта 2GIS засекает абонента, а LG внедряет многооконность.

#### Сравнение ..... 30

Бесплатно звоните и любуйтесь лицами своих друзей со всего мира при помощи голосовой и видеосвязи VoIP. Кому нынче нужен телефон?

#### Интервью LXF: Опять двойка ..... 44, 48

**Пит Ломас** считает Манчестер центром компьютерной Вселенной,

а Андрей и Ольга Филипповы не поспевают принципами свободного ПО в видеокамерах.

#### Что за штука... ARM ..... 60

Процессоры ARM — реальные кандидаты на импорт революции в мобильные устройства. Этак смартфоны обгонят персональные компьютеры!

#### Рубрика сисадмина ..... 62

Доктор приходит вовремя — и раньше всегда вовремя приходил. Поскольку живет по Network Time Protocol.

#### Ответы ..... 96

##### ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

**Нейл Ботвик** — про установку HotPicks, проблемы с драйверами сканера Epson, BT Sport в Linux, запуск сервера MythTV, использование ZFS на главном диске, повреждение жесткого диска ноутбука, разбиение диска.

#### HotPicks ..... 100

Отдайте горяченького! Лучшие в мире новинки свободного ПО: Drawpile, Emacspeak, ExMplayer, Git-info-bar, Intricacy, JaxEdit, PCGen, PMW, RedNotebook, Warzone 2100.

#### Диск Linux Format ..... 106

Содержимое двустороннего DVD этого месяца.

#### Пропустили номер? ... 108

Нет ли в вашей коллекции дыры, глядящей на вас глазами обиженного ребенка? Возместите недостачу — и прощай это проклятое чувство вины при повороте головы к книжной полке.

#### Через месяц ..... 112

Заблокируем АНБ, закрепим безопасность своего гаджета с Android, погуляем по заводу-изготовителю Raspberry Pi... и не только!

## БЛАГИЕ НАМЕРЕНИЯ

# Британия выбирает ODT

Правительство Великобритании планирует отказаться от MS Office.

Издание The Guardian сообщило, что не исключает возможности отказа от использования знаменитого офисного пакета от Microsoft. Причина достаточно типична — только в 2010 году на оплату лицензий было потрачено около £200 млн из бюджета.

Позиция британского правительства интересна тем, что главным «источником зла» они видят вовсе не проприетарную офисную программу, а проприетарные офисные форматы и отсутствие жестких стандартов в этой области. Именно поэтому направление миграции в заявлении не указывается. В частности, министр Франсис Мод [Francis Maude] считает, что роль преемника *MS Office* может играть любое ПО, позволяющее работать с файлами формата ODT — *OpenOffice* или Google Docs.

Идея заключается в том, чтобы не только ввести в государственный документооборот свободные форматы, но и стандартизировать саму процедуру обмена информацией между государственными учреждениями. По мнению правительства, это позволит сэкономить значительные бюджетные средства.

Подобный подход, действительно, представляется оптимальным. Он позволяет не только минимизировать затраты, но и предоставить пользователю возможность выбирать инструментарий в соответствии с отраслевой или региональной спецификой. Очевидно, что в масштабах страны иные концепции попросту не будут работать.

Впрочем, может оказаться, что вся история затеяна исключительно для того, чтобы мотивировать компанию Microsoft

» Проприетарные форматы слишком дорого обходятся Британии.



предложить Великобритании более выгодные условия. Известны случаи, когда этот прием приводит к желаемому результату.

В России стандарт ГОСТ Р ИСО/МЭК 26300-2010 «Информационная технология. Формат Open Document для офисных приложений (OpenDocument) v1.0» принят в 2010 году. Но какого-то заметного положительного влияния на темпы и масштабы внедрения СПО в органы государственной власти это не оказало.

## СМЕНА ПРИОРИТЕТОВ

# Перемены в openSUSE

SUSE займется качеством свободного дистрибутива.

Компания SUSE сообщила о изменениях концепции участия ее штатных сотрудников в работе над свободным дистрибутивом openSUSE. Теперь непосредственная подготовка релизов будет осуществляться исключительно сообществом. Бизнес берет на себя заботу о качестве продукта: сотрудники компании займутся тестированием, выпуском обновлений с исправлением уязвимостей и поддержкой.

Очевидно, что это значительно увеличит срок подготовки новых выпусков, чему вряд ли обрадуются энтузиасты, устанавливающие ПО большей частью ради

изучения тенденций развития дистрибутива. Однако, судя по этому решению, бизнес значительно больше заинтересован в другой группе пользователей.

Не исключено, что SUSE сочла проблемой наличие дистрибутива от сообщества, явно и неоднозначно ассоциируемого с продукцией компании. Весьма вероятно, что потенциальный клиент начнет знакомство с системой, взяв более доступный продукт, и сделает для себя некие выводы. Менеджерам корпорации эти выводы могут не очень понравиться.

Разумеется, проблема возникла отнюдь не вчера, и она носит глобальный

» Рабочий стол системы openSUSE.





РЕЙТИНГИ

# Десятка лучших

Лучшие проекты по версии Black Duck Software.

Компания Black Duck Software опубликовала список наиболее интересных открытых проектов 2013 года. Победителями стали:

- » *Appium* — фреймворк для автоматизации тестирования мобильных приложений;
- » *Docker* — система управления изолированными Linux-контейнерами, которая

реализует полный цикл создания, обслуживания и сопровождения;

- » *Exercism* — облачная система, помогающая разработке кода посредством совместного решения возникающих проблем;
- » *Ghost* — платформа для организации работы блога;
- » *InfluxDB* — распределенная БД на основе *NoSQL*, позволяющая получать срезы

значений параметров через заданные промежутки времени;

- » *OpenDaylight* — платформа для организации программно-конфигурируемых сетей и их виртуальных компонентов;
- » *OpenIoT* — облачная платформа для работы с приложениями в рамках концепции Интернет вещей;
- » *Serverspec* — инструментарий для разработки RSpec-тестов, позволяющих контролировать правильность настройки серверных систем;
- » *Tox* — защищенная система для обмена информацией, использующая криптографические методы идентификации пользователя и защиты трафика;
- » *XPrivacy* — инструментарий управления доступом к конфиденциальной информации для платформы Android.

Отдельно был отмечен JavaScript-фреймворк Twitter Flight — несмотря на то, что в общий рейтинг его решили не включать.

Также были зарегистрированы следующие тенденции прошлого года, вызвавшие максимальный интерес со стороны разработчиков:

- » облака и виртуализация;
- » обеспечение приватности;
- » социальные медиа;
- » Интернет вещей.



» **Docker** — один из лидеров рейтинга.

РАСПО

# Выбран новый президент

Новым президентом РАСПО стала Любовь Орлова.

Президентом Российской ассоциации свободного программного обеспечения (РАСПО) избрана Любовь Орлова — директор по развитию бизнеса одной из дочерних компаний группы «АйТи». Юлия Овчинникова, возглавлявшая организацию в 2012–2013 годах, покинула этот пост по обстоятельствам личного характера.

На протяжении практически всего периода активной деятельности РАСПО основным направлением ее работы являлось продвижение Национальной программной платформы (НПП) как части государственной программы «Информационное общество» и распоряжения 2299-р, предусматривающего внедрение СПО в органы государственной власти. Не исключается, что подобная «узкая специализация» создала для ассоциации некоторые проблемы.

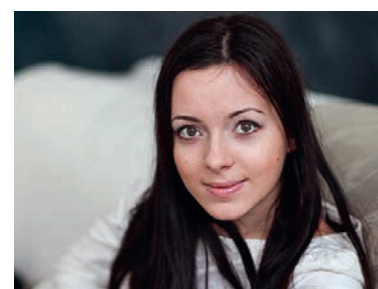
Начало активной деятельности было крайне успешным: в конце 2011 года конкурс на создание прототипов базовых

компонентов НПП выиграла компания «ПингВин Софтвр», а участие в выполнении госзаказа приняли входящие в ассоциацию предприятия. Однако после смены руководства Минкомсвязи работы в этом направлении были фактически свернуты. Активная критика подобной политики со стороны РАСПО в течение двух лет не дала ожидаемых результатов.

Резонно предположить, что новому президенту придется провести радикальную реформу деятельности РАСПО — практика показала, что существующий состав профильного министерства попросту не в состоянии реализовать сколько-нибудь масштабный проект, которым является НПП. Так что любая критика Минкомсвязи будет абсолютно правильной, но с прагматичной точки зрения совершенно бесполезной.

Таким образом, сегодня РАСПО поставлена перед выбором. Ассоциация может продолжать пытаться «образумить» чиновников, а может заняться другими,

» **Новый президент РАСПО Любовь Орлова.**



не менее полезными делами: поддержкой независимых разработчиков, популяризацией свободного ПО, созданием некоторых элементов инфраструктуры.

Очевидно, что сообщество более заинтересовано во втором пути. Проблем с развитием СПО в России достаточно много, поэтому работы для РАСПО наверняка хватит.

Впрочем, пока загадывать рано. Новый президент только приступил к выполнению своих обязанностей, так что лучше просто пожелать ему всяческих успехов.

РОБОТЫ

# Премия российскому проекту

Российский конструктор ScratchDuino получил премию Google RISE Awards 2014.

Второй год подряд ЗАО «Тырнет» побеждает в программе Google RISE Award и получает грант на поддержку образовательных инициатив. Эта премия является одной из самых престижных в отрасли — награду получают менее 5% от всех подавших заявки проектов.

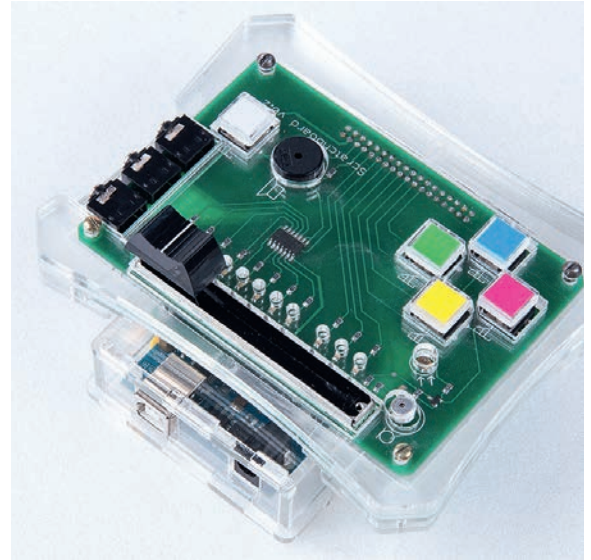
Например, в 2014 году было подано около 1000 заявок, а премии получили всего 49 компаний из 19 различных стран. Это позволяет говорить, что разработки ЗАО «Тырнет» выполнены на весьма высоком уровне и полностью соответствуют мировым стандартам образовательных инициатив в научно-технической сфере.

В текущем году очередной полученный грант поможет ЗАО «Тырнет» запустить образовательную программу для школьников и преподавателей, где они смогут получить практические навыки работы на комплексах оборудования ScratchDuino.

А специально для российских педагогов будут проводиться семинары и вебинары, к участию в которых допускаются все желающие.

Также летом будет запущен специальный проект по самостоятельной сборке конструкторов ScratchDuino на базе Центров молодежного инновационного творчества. И в первом полугодии состоится конкурс по свободной робототехнике.

Руководитель проекта Ия Хрусталева подчеркнула важность финансовой поддержки открытых образовательных проектов, ведь именно благодаря компании Google школы получили возможность повысить уровень обучения. Проект ScratchDuino будет и дальше стремиться максимально ликвидировать барьеры для инициации обучения робототехнике в школах, предлагая преподавателям все необходимое для начала работы.



Один из компонентов конструктора ScratchDuino.

ДИСТРИБУТИВЫ

# Выход Mageia 4

Наследник Mandriva выдал новый релиз.

Вышла очередная версия дистрибутива Mageia, создаваемого сообществом. По уже сложившейся традиции для загрузки предлагается несколько образов: полная DVD-сборка, Live-сборки на базе Gnome и KDE, а также минимальный набор для установки по Сети. Основными рабочими столами системы являются KDE 4 и Gnome 3, но желающие смогут использовать, кроме того, Cinnamon, MATE, Razor-qt, Enlightenment, LXDE и Xfce.

Программа установки, вероятнее всего, практически полностью унаследована у прародителя Mageia — дистрибутива Mandriva. Она чрезвычайно дружелюбна, зато результат порой бывает совершенно непредсказуемым. В частности, известный всем российским пользователям популяризатор Linux Алексей Федорчук сообщил, что основанная на рабочем столе Cinnamon система с набором ПО по умолчанию заняла на его диске 3,2 Гб, в то время как аналогичная конфигурация, но без офисных приложений, потребовала уже 3,7 Гб.

Одно из самых заметных новшеств — приложение *MageiaWelcome*, автоматически запускаемое при старте любого рабочего окружения. Начинающему пользователю предлагается доступ к нескольким информационным страницам и типовым операциям, которые большинство пользователей проводят сразу же после инсталляции — установке кодеков и некоторых проприетарных приложений.

Второе новшество менее бросается в глаза: реализована новая схема именования сетевых интерфейсов в соответствии с *systemd*. Команда *ifconfig* поведет себя любознательному пользователю, что *eth0* теперь будет *enp19s0*, а *wlan0* — *wlp18s0*. Разумеется, на работу интерфейсов это никак не повлияло.

Судя по уже появившимся в Интернете немногочисленным обзорам, Mageia 4 предназначена прежде всего для энтузиастов, которые следят за развитием именно этого дистрибутива. О его неготовности к повседневному использованию говорит хотя бы тот факт, что привычные всем

клавиатурные комбинации (например, Ctrl+C/V/X) работают только при включенной английской раскладке.

Впрочем, не исключено, что разработчики и не ставили перед собой задачу угодить обычным пользователям. Тем более, что для этой группы выпускается достаточно дистрибутивов, с которыми непросто конкурировать.

Рабочий стол системы Mageia 4.



# softline®



Services

Software

Cloud

## ИТ-архитектура вашего бизнеса



ПЕРСПЕКТИВЫ

# Нам светит Moonlight

Стартовала разработка нового рабочего окружения.

**Р**абочих окружений для системы Linux создано великое множество. Для мощных рабочих станций есть KDE и Gnome, для компьютеров послабее — *Xfce* и *LXDE*. Однако некоторым этого мало. И, судя по описанию нового пользовательского окружения Moonlight, о начале разработки которого было объявлено в феврале текущего года, они абсолютно правы. Рабочий стол Moonlight ориентирован на маломощные машины — например, Raspberry Pi, и будет разрабатываться на базе Qt 5. Используемая лицензия — GPLv3.

Инициаторы проекта видят Moonlight классическим рабочим столом по типу KDE и Gnome 2: лучше не удивлять пользователя новым концептом, интересным только гикам.

Архитектура Moonlight будет модульной. Работу окружения обеспечит всего одно приложение, функциональность которого можно наращивать плагинами. Компоненты среды не будут поставляться по отдельности, что существенно снизит

расход памяти и, соответственно, увеличит производительность системы. Отдельные процессы понадобятся только файловому менеджеру и конфигуратору. А панель, рабочий стол и система индикаторов будут функционировать как один процесс.

Moonlight не планируют разрабатывать с нуля: используется код других окружений — *Razor-qt* и *LXQT*. При обнаружении ошибки и ее исправлении модифицированный код возвращается в апстрим.

Вопрос дизайна пока остается открытым. Окончательное решение разработчики должны были принять на следующий день после завершения Олимпиады в Сочи (не исключено, что это просто совпадение).



➤ Один из вариантов оформления рабочего окружения Moonlight.

КОНФЕРЕНЦИИ

# HackDay в Москве

От идеи до прототипа — за 48 часов.

**С** 21 по 23 февраля в Москве проходил 30-й Хакатон [Hack+Marathon] — ставший традиционным марафон хакеров. Участники создают работающие прототипы своих проектов и через 48 часов представляют их коллегам, экспертам и инвесторам. Организатор — Студия Михаила Кечинова, которая проводит хакатоны по городам России с 2009 г.

На HackDay записались программисты, менеджеры, дизайнеры, проектировщики интерфейсов, маркетологи, PR-специалисты, а также все те, у кого есть идея интернет-проекта, но нет команды для ее воплощения. Можно прийти с уже готовой идеей, на реализацию которой никак не находилось времени, или собрать команду непосредственно на HackDay, или влиться в готовую команду. А для людей, одержимых технологиями, HackDay — еще и уникальный опыт общения с единомышленниками и специалистами, которые занимаются

созданием и развитием собственных продуктов. Здесь каждый может ощутить на себе, что означает войти в свой персональный God Mode. Тематика заявленных проектов на сей раз охватывала диапазон от голосового управления домашними устройствами до геймификации работы в компаниях и принятия решений на основании человеческих биоритмов.

Отчет о хакатоне можно просмотреть на сайте [hackday.ru/hackday-30/report](http://hackday.ru/hackday-30/report). **LXF**



➤ HackDay прошел в помещении DI Telegraph на Тверской.

Новости короткой строкой

➤ 14 марта в Санкт-Петербурге для разработчиков ПО мобильных устройств пройдет конференция Mobius. Доклады будут сугубо техническими. Источник: [mobius.codefreeze.ru](http://mobius.codefreeze.ru)

➤ Разработчик ядра Linux Теодор Цо [Teodor Tso] и Марк Мерлин [Mark Merlin] из Google объявили, что из-за ошибок в UEFI-прошивке ряд ноутбуков Thinkpad при попытке поставить Linux ломается аж до замены материнской платы. Источник: [www.heise.de/open/](http://www.heise.de/open/)

➤ В очередной версии офисного пакета *LibreOffice 4.2* табличный процессор *Calc* стал поддерживать параллельные вычисления формул. Источник: [blog.documentfoundation.org](http://blog.documentfoundation.org)

➤ Опубликован 10-й релиз настольной системы PC-BSD, с KDE 4.11, MATE, Cinnamon, *LXDE* и *Xfce*. Источник: [blog.pcbsd.org/2014/01/](http://blog.pcbsd.org/2014/01/)

➤ Google и Samsung условились о взаимном использовании технологий — и по текущим, и по будущим патентам. Источник: [techcrunch.com/2014/01/](http://techcrunch.com/2014/01/)

➤ Новое ядро Linux, с интеграцией фильтра пакетов *Nftables* и более производительной *Squashfs*, поддерживает автоматическое переключение GPU в драйвере *Radeon* и архитектуру *Intel MIC*. Источник: [lkml.org/lkml/2014/1/19/148](http://lkml.org/lkml/2014/1/19/148)

➤ При поддержке государства в Китае вышла новая ОС — *COS* (China Operating System). Ее цель — разрушить существующую монополию на рынке. Источник: [www.engadget.com](http://www.engadget.com)

➤ В апреле 2014 состоится СТАЧКА — крупнейшая IT-конференция в России, по всем аспектам IT-сферы. Источник: [nastachku.ru](http://nastachku.ru)

➤ Один из основателей Open Source Initiative Эрик Реймонд хочет разрешить в *GCC* проприетарные плагины, что добавит преимуществ перед *Clang*. Источник: [gcc.gnu.org](http://gcc.gnu.org)

➤ Глава группы по настольным системам в Red Hat Кристиан Шаллер [Christian Schaller] предложил смягчить политику Fedora по несвободному ПО, расширив круг пользователей системы. Источник: [lists.fedoraproject.org](http://lists.fedoraproject.org)

➤ ПингВин Софтвел интегрировала ПО IP-телефонии *Asterisk* с системой работы с клиентами *SugarCRM*. Источник: [www.pingwinsoft.ru](http://www.pingwinsoft.ru)

Весна. Интернет. Конференция.

# РИФ+КИБ 2014

**23–25 апреля**  
**пансионат «Поляны»**

[www.rif.ru](http://www.rif.ru)



**Подарок для подписчиков журнала —  
видео-трансляция РИФ+КИБ 2013.**

Для активации необходимо перейти по  
QR-коду и ввести промо-код: RIF13VIDEO

# FOSDEM глазами очевидца



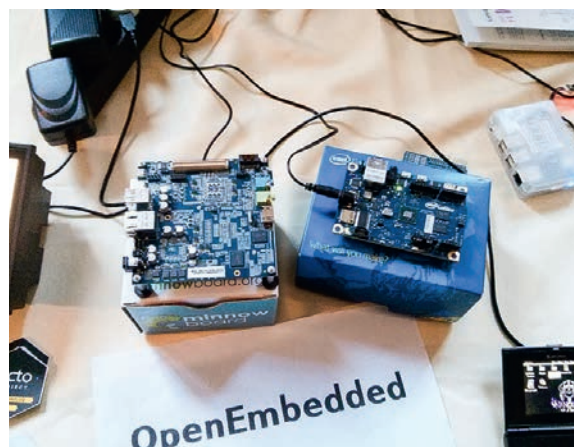
**Александр Хрюкин** делится впечатлениями от конференции по свободному ПО и дает ценный совет.

**В** городе Брюсселе 1 и 2 февраля 2014 года состоялась крупнейшее в Европе ежегодное мероприятие, целиком и полностью посвященное свободному программному обеспечению, под названием FOSDEM (<https://fosdem.org/2014/>). В этом году на конференции было 487 докладчиков, в том числе и автор этой статьи, 512 событий (сюда включены любые обсуждения, презентации и тому подобное) и 44 трека (это что-то наподобие мастер-классов по различным дисциплинам, начиная от того, как спаять робота из нескольких деталей, и заканчивая построением графов на языке Go).

Проходила конференция в Université Libre de Bruxelles [Свободный университет Брюсселя] — по всей видимости, это единственное место в Брюсселе, которое может предоставить такое количество помещений для одновременного выступления докладчиков и принять огромное количество посетителей, сохранив при этом атмосферу неформального общения между участниками и посетителями.

За те два дня, что проходила конференция, ее успело посетить больше пяти тысяч человек, о чем рапортуют организаторы, но мне кажется, что людей было значительно больше, так как посещение не требовало предварительной регистрации и было целиком бесплатно; ну и ко всему прочему, аудитории были набиты под завязку. Люди сидели даже на ступенях лестниц.

Говоря об общении, отдельно стоит сказать о беспроводном доступе к сети Интернет. Организаторы развернули на всей



➤ Слева — Minnow Board, справа — просто плата x86.

территории университета свободный и бесплатный Wi-Fi; правда, адреса сперва выдавались только IPv6, и, как оказалось, мир еще не готов к тому, чтобы полностью отказаться от IPv4. Но спустя пару часов после начала развернули еще одну сеть, уже с IPv4; кроме того, каждое устройство в сети умудрилось получить внешний/белый IP-адрес. Каким образом это реализовали для 5000+ человек, для меня остается загадкой.

Докладов же было просто неподъемное количество. Я посетил всего несколько, потому что у меня, как у активного участника и разработчика дистрибутива OpenMandriva, была необходимость присутствовать на нашем стенде и отвечать на вопросы интересующихся.

Всего стендов было 45 штук — их выставили такие дистрибутивы, как Fedora, CentOS, Mageia, OpenMandriva, openSUSE и Debian, а также дистрибутив для детей — DoudouLinux. Удивило то, что нигде не было стенда или какого-то доклада от Ubuntu. Дорогого и корпоративного RedHat тоже не было, хотя RedHat заявлен как один из спонсоров конференции. Впрочем, это вполне компенсировалось большим количеством докладов сотрудников этой компании о технологиях и продуктах, развиваемых крупнейшим IT-вендором под флагом свободного ПО. Они также поделились подробностями процесса разработки, в частности, сообщив, что используют *Gerrit* для инспекции исходного кода, *Nitrate* для управления тестами и *Dogtail* для автоматизации тестирования приложений с графическим интерфейсом. Предсказуемо прозвучали их слова об особой важности автоматизации процесса тестирования. Увы, практика показывает, что чересчур надеяться на помощь сообщества — вариант для серьезных проектов не самый подходящий.

Замечательной была секция, посвященная свободным текстовым процессорам, где рассказывали про *OpenOffice/LibreOffice*, преимущества их использования, скорости работы и тому подобное, а в дополнение рассказали, как улучшается отладчик GDB



➤ Пройти мимо такой вывески было невозможно, поэтому аудитории помелись от публики.



➤ Аппарат по приему биткойнов.

(Gnu Debugger) при взаимодействии с разработчиками свободных офисных пакетов. Подобные отношения прослеживаются и между средой разработки Eclipse и *OpenOffice*: в Eclipse был специально отлажен CDT для работы с офисным пакетом.

Меня больше всего, как embedded-разработчика, интересовали доклады про всяческие интересные железки. Один из таких докладов назывался "ARM: Allwinner Sunxi SoCs and the Community behind It. The Most OpenSource (Friendly) SoC!"

Allwinner — это такая система-на-чипе, придуманная китайским брендом на базе процессоров Cortex A-8. Как ни удивительно, разработка девайсов на этих процессорах ведется под контролем и при участии достаточно крупного сообщества. Кроме того, этот чип очень дешев, и практически все китайские девайсы, типа планшетов, медиа-серверов и прочих

## «На FOSDEM были представлены не только технические доклады.»

курс биткойна. Чтобы отправить биткойны, нужно приложить телефон к светящейся панели на автомате. А дальше — то ли камера считает адрес кошелька, то ли NFC; но так или иначе, все работает и вы-



➤ Pandora крупным планом.

USB-PC построены либо на Allwinner, либо на Rockchip. Доклад же по большей части состоял из отчета о взаимодействии инженеров компании и энтузиастов Open Source в деле портирования и реверс-инжиниринга драйверов для графических процессоров (lima/freedreno) и их взаимодействием с чипами Allwinner. Достаточно познавательно, а главное — очень перспективно: ведь меня, как и многих других, не радует перспектива использования проприетарного костыля вместо полноценного X11-модуля.

Отдельно порадовал стенд с девайсами OpenEmbedded, причем демонстрировались не только всяческие ARM-железки, но даже x86-«пекарни», по габаритам сравнимые с ARM-железками. Почему «пекарни»? Дело в том, что они лишены крупных охлаждающих элементов, что товарищами из Intel продвигается как суперская идея и чуть ли не прорыв в x86-embedded. Хотя процессор Minnow Board от Intel — буквально переносная печь по конвертации электричества в тепловую энергию.

Невдалеке демонстрировалась работа аппарата по приему биткойнов — на фото показано, как происходит пожертвование небольшой суммы, во имя криптоанархии и скорейшего роста курса биткойна. Чтобы отправить биткойны, нужно приложить телефон к светящейся панели на автомате. А дальше — то ли камера считает адрес кошелька, то ли NFC; но так или иначе, все работает и вы-

глядят классно и очень футуристично.

Интересно было взглянуть на OpenPandora — это свободная платформа для запуска старых игр с Денди, SEGA, Game Boy, для тех, кто хочет знатно поиграть в Чип-и-Дейла, утиные истории и прочие чудесные игры из прошлого.

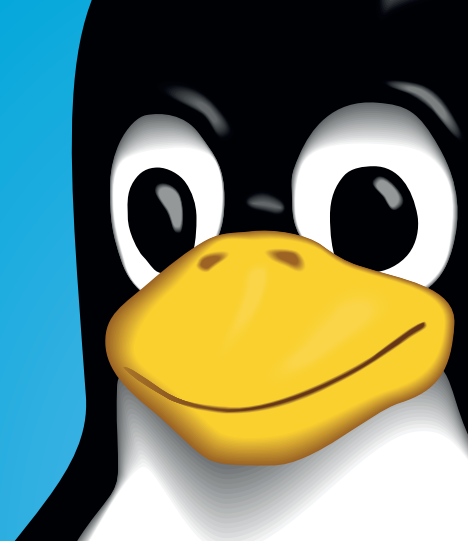
Кстати, на FOSDEM были представлены не только технические доклады, но и некоторое количество юридических. В основном они были на темы «Как использовать свободное ПО, чтобы вам за это ничего не было», «Тролли живут не только под мостом, но и в патентной системе» и «Как получить инвестиции в свой проект». Народа в аудитории набилось столько, что мне влезть не удалось — пришлось слушать из коридора; а потом и из коридора пришлось ретироваться, так как число желающих послушать лекцию в коридоре превысило объем необходимого для жизнеподдержания воздуха.

Ну и в завершение хочу высказать небольшое замечание о том, что нужно брать на конференцию. Как показал мой опыт, нужно обязательно прихватить большую бутылку воды, и желательно из страны, где цена воды не дороже 30 рублей за литр, потому что в Брюсселе наблюдается удивительный парадокс: вода стоит 80–100 евроцентов за малюсенькую бутылку, тогда как пиво такого же объема — от 40 центов. Лично для меня выбор очевиден, а вот для трезвенников и язвенников все очень плохо. LXF



➤ Геймеров отнюдь не обошли вниманием.

# Используйте свободное ПО — сэкономьте годовой бюджет!



Операционная система GNU/Linux поможет вам **с наименьшими затратами** решить проблему лицензирования программного обеспечения, навсегда избавиться от компьютерных вирусов и повысить надежность вашей компьютерной сети.



**С нашей  
помощью  
вы сможете**

**Сконцентрироваться  
на своем бизнесе,**  
не отвлекаясь на вопросы  
поддержки своей  
ИТ-инфраструктуры



**Забывать о вирусах,  
угрозах безопасности**  
и необходимости  
лицензирования  
программного обеспечения



**Оптимизировать  
затраты**  
на лицензирование ПО  
за счет максимально  
возможного использования  
свободного ПО

## ГНУ/Линуксцентр предлагает:

- Лицензирование во ФСТЭК.
- Внедрение наиболее дружественных вариантов ОС GNU/Linux и прикладных решений на базе свободного ПО.
- Абонентскую поддержку вашей сети.
- Обучение сотрудников вашей компании.

Наш опыт внедрения свободного программного обеспечения в организациях различного профиля поможет выбрать **оптимальное сочетание свободного и коммерческого программного обеспечения**, подходящее именно для вашей компании, а также поможет избежать технических и организационных проблем при внедрении свободного ПО.

## Решите проблемы лицензирования ПО и поддержки компьютерной сети с помощью профессионалов!

Москва  
+7 (499)

**271-49-54**

Санкт-Петербург  
+7 (812)

**309-06-86**

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)

Linux  center



Новинки программного и аппаратного обеспечения в описании наших экспертов



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

## Эпохальная init-система

Пока космические корабли бороздят просторы Вселенной, а разработчики Linux'а удумывают все более сложные системы инициализации, некоторые идут своим путем: сочиняют init-системы простые и маленькие. В числе этих некоторых — Бен Дибелл [Ben Dibell], aka Subsentient, сайт его <http://universe2.us/>: в один прекрасный июльский день минувшего года он решил сочинить аналог *Upstart* или *Systemd*, но без их наворотов, показавшихся ему лишними. Вроде бы соответствующую его требованиям init-систему *Busybox* Бен считал недостаточно функциональной.

Пацан решил — пацан сделал. И новая init-система, названная автором «эпохальной (*Epoch*)», к сентябрю обрела рабочий вид. А в январе года нынешнего под именем *Sage* вышла ее версия 1.0. Зависимости *Epoch* сводятся к библиотеке *glibc* или аналогичной — ну, и ядро Linux. И для ее установки желательно иметь какую-либо sh-совместимую командную оболочку; но можно и обойтись. Работу обеспечивает единственный файл настройки, никакие shell-сценарии в процессе загрузки не используются. Распространяется *Epoch* как Public Domain в смысле <http://unlicense.org/>.

Пока прикрутить *Epoch* к какому-либо «настоящему» дистрибутиву Linux все же проблематично — разве что в *Tiny Core* она могла бы заменить *Busybox*. А что из нее выйдет дальше? Увидим. Но будет забавно, если такая игрушечная init-система окажется не менее жизнеспособной, чем активно внедряемые ныне монстры. [alv@posix.ru](mailto:alv@posix.ru)

## Сегодня мы рассматриваем:

**Ultimate Edition 3.8** ..... 16  
Дистрибутив со столь звучным именем просто обязан быть впечатляющим. Оправдает ли он сделанную заявку, став всем для всех подряд пользователей?

**Tails 0.22** ..... 17  
Вы ставите свою анонимность онлайн превыше всех прочих соображений? Вы отнюдь не обязаны чем-то поступиться в остальных областях. Тестируем Tails.

**Perzo.com** ..... 18  
Непрекращающийся поток разоблачений про всякие там АНБ заставил нас задуматься: а кто слышит, о чем мы чирикаем в Сети? Пришлось

рассмотреть Perzo.com — онлайн-сервис, который обещает скрыть от посторонних глаз нашу почту, мгновенные сообщения и другие формы коммуникаций.

**NOOBS 1.3.2** ..... 19  
Свежая версия! Владельцы Raspberry Pi, вы наверняка захотите обратиться к *NOOBS*: установить и настроить дистрибутив (и не один) с этим ПО невероятно легко.

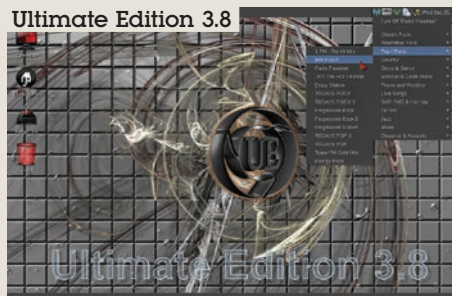
**Penguin Wee 4th Gen Desktop** ..... 20  
Компактный — и мощный настольный ПК с Linux-дистрибутивом по вашему выбору; на современных комплектующих — и по разумной цене. ПРЕДОСТЕРЕГАЕМ: Нам понравился.



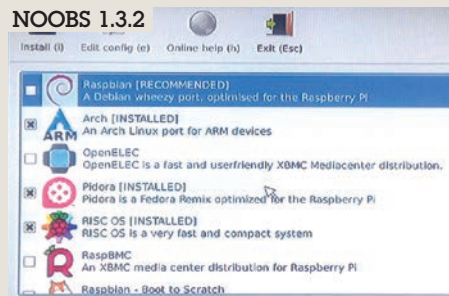
► **Penguin Wee**: в этом небольшом ящичке таится приличная мощь.

**Devolvo LiveCam** ..... 22  
Мы соблазнились обещанием готовой поддержки Linux в беспроводной камере. Оказалось неоднозначно.

**Broken Age** ..... 23  
Игра из двух историй: надоел интерьер космического корабля — пожайлуйте на природу, и наоборот.



► Требуется известная борзость, чтобы включить слово “ultimate” в название дистрибутива.

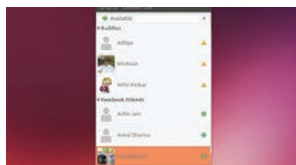


► Благодаря *NOOBS* дистрибутивы на Raspberry Pi запросто устанавливаются через GUI.

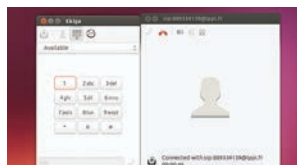
## Сравнение: Клиенты VoIP

с. 30

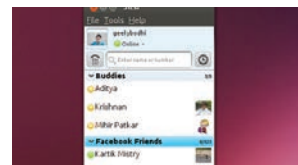
### Empathy



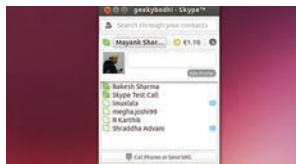
### Ekiga



### Jitsi



### Skype



### Yate



То, что вы живете с любимым существом на разных полушариях, не мешает вам общаться лицом к лицу — спасибо VoIP-клиентам! Взяли микрофон и web-камеру — и разговаривайте вволю.

# Ultimate Edition 3.8

Поиски дистрибутива Linux типа «все включено» навели Маянка Шарму на дистрибутив, который определенно метит в таковые.

## Вкратце

» Дистрибутив Linux, позиционируемый как наиболее удобный сразу же из коробки и доверху наполненный приложениями. См. также Knoppix.

Часто подчеркивается: выбор — это лучшее, что есть в свободном ПО. Можно взять стандартный дистрибутив Linux и вдоволь насладиться его настраиванием — напичкать его приложениями на раздутом рабочем столе, или раздеть его до голого железа сервера. Но бывают и времена, когда вам не хочется вылизывать дистрибутив, чтобы установить или запустить модули, или раскошегивать менеджер пакетов, чтобы добавить одно-два приложения. Вы просто хотите достать его из коробки и начать работать. Специально для таких нетерпеливцев создан дистрибутив Ultimate Edition.

Дистрибутив заботливо предоставляет пользователям лучшие свободные приложения и послужит хорошей стартовой точкой для начинающих линуксоидов, которые могут не подозревать о мирадах свободно доступных приложений. Кроме обычных приложений вроде веб-браузера Firefox и пакета LibreOffice, как в большинстве настольных дистрибутивов, в Ultimate есть целый арсенал, для пользователей всех уровней. Это внешние интерфейсы для эмуляторов Qemu и Wine, для запуска приложений Windows и игр на Linux, и масса менее известных (но очень толковых) приложений: пакетный фоторедактор Phatch, Gmail Notifier, редактор тэгов Entagged, инструмент сервиса для скринкастинга RecordMyDesktop и Avidemux для редактирования видео. В 2,5-ГБ ISO-образ также вошло множество мультимедиа-приложений, включая VLC, Amarok, Clementine, Totem, а также приложение XBMC — медиа-центр. Более того, веб-браузер Firefox обвешан всякими



» Ultimate Edition принял за основу Ubuntu 13.04.

модулями для работы с разнообразными видами проприетарного контента.

Хотя этот релиз Ultimate Edition использует рабочий стол Mate 1.6, в нем также содержится несколько приложений KDE, а еще и клиент KVIrc для IRC, читалка электронных книг Calibre и K3b для записи оптических дисков. Также есть инструменты IDE для разработчиков Gnome и KDE. Не забыты и системные администраторы. С такими инструментами, как FSlint, UNetbootin, BootUp-Manager, Boot Repair, Grub Customizer и GParted можно модернизировать, восстанавливать и исправлять установку Ultimate Edition, а также применять Live media как спасательный дистрибутив.

## И красота, и мозги

По списку приложений дистрибутива понятно: он воспринимает свое название всерьез. Изюминка данного релиза — пять модулей конфиденциальности Firefox (по умолчанию они отключены). Модули предотвращают слежение сайтов за вами по Сети, а также помогают безопасно удалить cookies и объекты flash. Другое уникальное предложение релиза — облачное приложение для синхронизации Cory. Можно войти в систему или зарегистрироваться с помощью сервиса изнутри дистрибутива (для этого предоставляют 15 Гб в хранилище). Еще у приложения милый значок-индикатор, который по правому щелчку мыши не отображается в контекстном меню дистрибутива. Прекрасным дополнением к контекстному меню стали скрипты Caja, позволяющие копировать/

переместить файлы, конвертировать изображения в другой формат и записать ISO-образ одним щелчком мыши.

Раздражающих моментов только два. Первый — приложения ненадлежащим образом настроены по умолчанию. Например, изображения открываются в Firefox, а не во встроенном просмотрщике. Второй — тема по умолчанию: на наш взгляд, она слишком темная и способна сделать непознаваемыми некоторые пункты меню.

Во внешнем мире у дистрибутива есть сообщества очень активных пользователей и хорошо отлажена работа форумов. Достаточно документации для помощи начинающим. В целом, дистрибутив отлично демонстрирует лучшее, что есть в Linux, и сгодится как настольный на каждый день. LXF



## Свойства навскидку



**Примочки приватности**  
Примочки приватности Firefox являются изюминкой этого релиза.



**Надо, не надо — есть все**  
В дистрибутиве огромный выбор приложений для линуксоидов всех уровней.

## LINUX FORMAT Вердикт

### Ultimate Edition 3.8

Разработчик: Three Man  
Сайт: <http://ultimateedition.info/>  
Лицензия: GPL

Функциональность	9/10
Производительность	8/10
Удобство использования	8/10
Документация	7/10

» Прекрасный рабочий дистрибутив, переполненный приложениями. Если и не предел мечтаний, то все равно отличный.

**Рейтинг 8/10**

# Tails 0.22

**Лес Паундер** задумался: можно ли сохранять приватность и при этом пользоваться Интернетом в свое удовольствие? Разбирается... аноним.

## Вкратце

» Безопасный, ориентированный на приватность дистрибутив для тех, кто ценит анонимность в Сети.

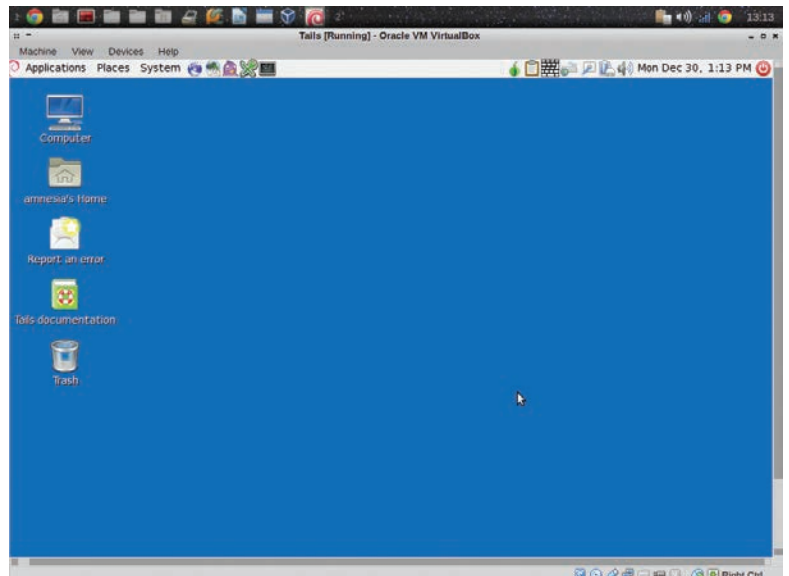
**В**аша безопасность в Сети находится под угрозой, и многочисленные заголовки прессы кричат о том, что АНБ ведет слежку за рядовыми гражданами и норовит сговориться с производителями ПО и вставить лазейки в различные технологии. Интернет открыт по самой своей природе, а следовательно, ваши данные не так уж сложно будет перехватить.

Тут-то и пригодится Tails и его инструменты обеспечения безопасности. Tails, или полностью The Amnesic Incognito Live System [Живая система с амнезией и инкогнито] — это дистрибутив live DVD/USB, специально созданный на тот случай, когда вы ставите свою конфиденциальность во главу угла.

Три главных функции Tails — анонимный выход в Интернет через сеть Tor; заметание следов от вашего сеанса; инструменты криптографии для шифрования ваших почтовых сообщений и файлов, отправленных через сети.

Чтобы использовать Tails, нужно просто записать ISO-образ на DVD, загрузочную флэшку или SD-карту с помощью *UNetbootin*. Затем загрузите свой ПК с DVD или USB, и Tails запустится в режиме live. Если вы используете USB или SD-карту, то сможете создать на ней перманентный раздел, где будут храниться ваши файлы в зашифрованной папке.

После загрузки Tails задаст вам пару вопросов, самый интригующий из которых — нужно ли замаскировать ваш рабочий стол Tails под Windows XP. Да, вы не ошиблись:



» На первый взгляд, рабочая среда Tails выглядит как обычный настольный Debian, но внутри таится когорта приложений для безопасности.

Tails умеет даже притворяться XP, чтобы вы затерялись в море XP-дронов, для вящей уверенности в том, что ваша приватность должным образом сохранена.

Tails создан на базе последнего Debian Wheezy и включает практически полный набор стандартных приложений — в том числе всеми любимые *GIMP*, *Inkscape* и *Audacity*, а также *OpenOffice*. Пользоваться Tails ничуть не сложнее, чем собственно Debian.

## Пробираясь по туннелям

Настройка сети Tor производится через средство удобного приложения, сидящего в системном лотке. Правый щелчок по значку приложения позволит вам задать свое Местонахождение, и из этой точки попасть в сеть Tor.

В роли встроенного браузера выступает *IceWeasel*, использующий настройки ПО Tor для сокрытия пути вашего просмотра. Tor благополучно прошел проверку в странах с правительственной цензурой и был de facto признан инструментом открытых коммуникаций для различных деликатных целей.

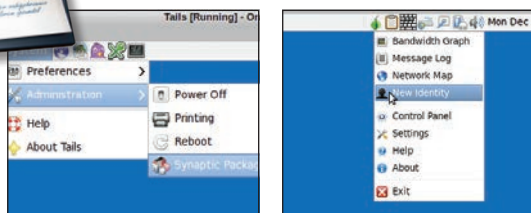
Дистрибутив использует криптографические технологии для шифрования ваших файлов, писем и даже мгновенных сообщений. Еще одна ценная функция безопасности Tails в том, что он не оставляет

и следа своего использования на ПК, тщательно стирая все после закрытия.

Для кого же предназначается Tails? Если вы много путешествуете и вынуждены использовать открытые точки доступа Wi-Fi, это самый подходящий дистрибутив. Tails включает уникальный инструментарий для защиты вашей анонимности в сети, в том числе от тех, кто пытается применить атаку типа “man-in-the-middle [Посредник]” для перехвата ваших данных. Другими словами, в Tails есть все, чтобы во Всемирной паутине вы чувствовали себя в безопасности. **LXF**



## Свойства навскидку



### На базе Debian

Дистрибутив Tails располагает доступом к богатейшим репозиториям своего прародителя Debian.

### Меняйте места

Tails предоставляет простой способ изменить свое местоположение, когда вы выходите из сети Tor.

## LINUX FORMAT Вердикт

### Tails 0.22

**Разработчик:** Boum  
**Сайт:** <https://tails.boum.org>  
**Лицензия:** GPL v3+

<b>Функциональность</b>	8/10
<b>Производительность</b>	8/10
<b>Удобство использования</b>	9/10
<b>Документация</b>	8/10

» Надежный live-дистрибутив с упором на безопасность, удобный, когда точка доступа в Интернет не вызывает доверия.

**Рейтинг 8/10**

# Perzo

Стремясь пресечь шпионов, **Маяк Шарма** открыл web-сервис, преобразующий онлайн-взаимодействие встроенной конфиденциальностью.

## Вкратце

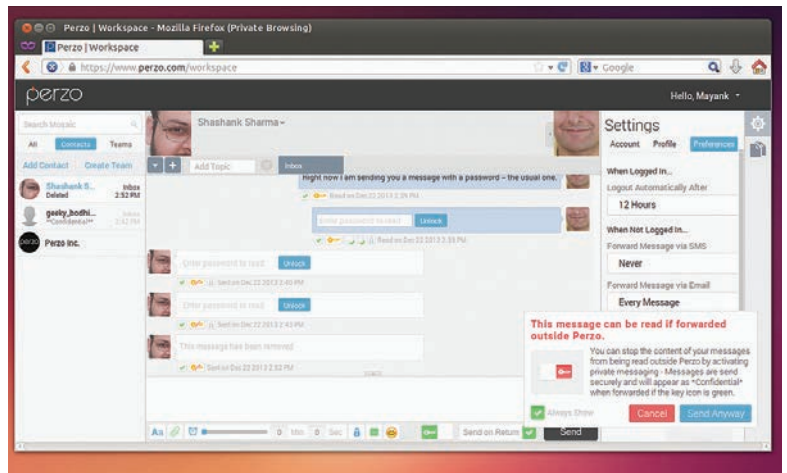
» Онлайн-сервис для безопасного общения. См. также: модуль расширения OTR для Pidgin, Enigmail для Thunderbird.

**В**зглянем правде в глаза. В пост-Сноуденовскую эру интернет-коммуникаций жаждущих зашифровать свои каналы связи параноиками уже не назовешь. Хорошая новость в том, что инструменты для шифрования электронных способов связи, включая почту и мгновенные сообщения, доступны пользователям настольного Linux достаточно давно. Однако их настройка требует определенных навыков.

Здесь в игру вступает Perzo. Бесплатный сервис представляет единый коммуникационный пакет, позволяющий контактировать через мгновенные сообщения, электронную почту или отправку текстового сообщения с телефона. Приложение дает вам прозрачный режим просмотра всех коммуникаций с контактом в этих трех вариантах, ограждая вас от проблем с настройкой множества устройств.

Сервис создан с учетом конфиденциальности и предлагает комплексное шифрование. По словам разработчиков, в Perzo используется трехкратное шифрование наряду с генератором рандомизированных ключей, и каждое сообщение шифруется своим отдельным ключом. Также предприняты меры для отражения атак типа «посредник (MITM)».

Компания Perzo пообещала выдать код под лицензией свободного ПО. В интервью *The Inquirer* основатель Perzo Дэвид Гурле [David Gurle], бывший вице-президент Skype, утверждает, что разрешить доступ к коду — единственный путь обрести доверие и убедить пользователей, что сервис не содержит троянов и лазеек.



» В web-приложении налицо все удобства, какие можно ожидать от приложения IM.

Пока Perzo доступен только как web-сервис; мобильное приложение для Android — в стадии разработки. Вам нужно будет сначала создать учетную запись в сервисе. После проверки вы попадете в рабочее пространство Perzo, с ясной и интуитивной раскладкой. Прежде чем вы сможете пользоваться сервисом, надо добавить контакты с указанием их адресов электронной почты. Как ни странно, импорта контактов нет, зато можно разбить их на группы. Возможность оперировать и личными, и профессиональными контактами с одного интерфейса — один из козырей Perzo. Легко перейти от разговоров с одним контактом к другому или к группе. Интерфейс позволяет переключаться между отправкой зашифрованных или незашифрованных сообщений, а перед отправкой незашифрованного сообщения всплывает окно-предупреждение.

не доступен в Perzo, сообщение можно зашифровать и отправить на его электронный адрес. При получении такого письма получателю для прочтения будет предложено зарегистрироваться на Perzo. Еще можно отправлять сообщения на мобильники, но у нас эта функция не работала.

Документации, кроме FAQ, пока немного, но Perzo находится в стадии активной разработки. Разработчики усердно трудятся над интеграцией голоса, видео и возможностью совместного использования рабочих столов, а также над развитием Perzo как полноценного клиента электронной почты. Perzo движется по верному пути, и если все пойдет как было обещано, наведет-таки шороху среди свободных онлайн-услуг связи. **LXF**

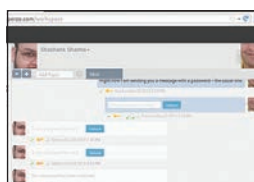


## Свойства навскидку



### Все виды связи

Контакты смогут отвечать вам через мгновенные сообщения, электронную почту, мобильные телефоны.



### Самоликвидация

Можно запрограммировать сообщения на самоликвидацию по истечении определенного срока.

## 3, 2, 1... самоликвидация

Помимо шифрования, можно прикрыть сообщение паролем. Получатель сможет его просмотреть, только введя пароль; и если оно отправлено, даже вы сами в своем списке сообщений его без пароля не увидите. Согласитесь, здорово! Также можно повторно заблокировать прочитанное сообщение. Другая интересная функция, повышающая безопасность — отправка сообщений, которые самоуничтожаются по истечении заданного времени. После уничтожения они исчезнут из чата — и вашего, и получателя. Если контакт

## LINUX FORMAT Вердикт

### Perzo.com

Разработчик: Perzo, Inc  
Сайт: www.perzo.com  
Лицензия: TBC

Функциональность	7/10
Производительность	8/10
Удобство использования	7/10
Документация	6/10

» Если Perzo выполнит свои обещания, придется традиционным онлайн-каналам связи побороться за свои деньги.

**Рейтинг 7/10**

# NOOBS 1.3.2

Перед нами новый установщик для Raspberry Pi, вызвавший у Маянка Шармы слезы умиления.

## Вкратце

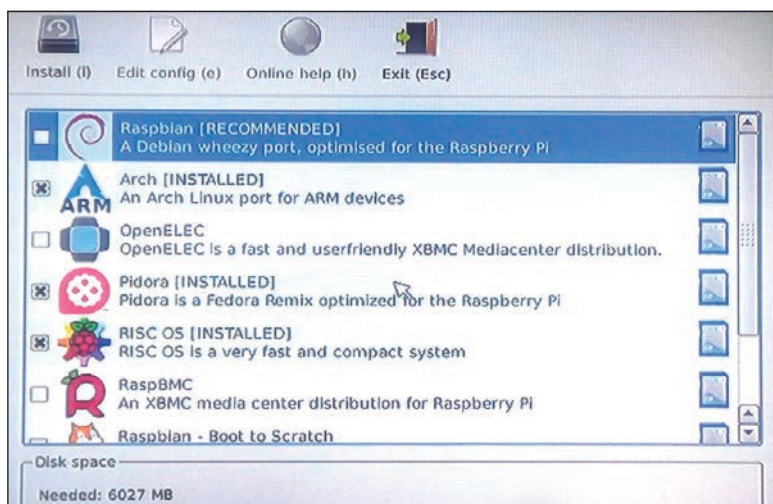
» Одно из дружелюбных приложений для установки нескольких дистрибутивов на Raspberry Pi. См. также команду dd.

**Д**истрибутивов для Raspberry Pi в мире хватает. И хотя установка их на компьютер-крошку была не настолько трудоемка, как установка Linux на ПК, но все же требовала специальных инструментов и вашего пристального внимания. Времена, однако, меняются! Теперь установщик *NOOBS*, он же *New Out Of Box Software*, для Raspberry Pi упрощает и стандартизирует установку дистрибутива на устройство. Все, что надо сделать — скачать ZIP-файл *NOOBS*, разархивировать его и копировать содержимое на SD-карту объемом 4+ ГБ, отформатированную в FAT. Делается это хоть в Linux, хоть в Windows или Mac OS X. Готово! Вставляете карту в Pi, и дело в шляпе!

Установщик *NOOBS* содержит несколько дистрибутивов — как общего назначения, например, Raspbian и Pidora, так и вариантов для продвинутых пользователей, типа Arch, или с уклоном в медиа-центр — OpenELEC, RaspBMC и т. п.

*NOOBS* дебютировал еще в июне 2013 года. Он предлагал самый удобный способ установки дистрибутивов, но оставался ряд серьезных ограничений: самое большое из них — неспособность установить одновременно несколько дистрибутивов на одну SD-карту. Последняя версия не только сняла это ограничение, но и добавила массу новых полезных функций.

При загрузке свежей версии *NOOBS* можно выбрать, какой дистрибутив установить. В отличие от предыдущей версии, из списка можно выбрать несколько дистрибутивов, при условии, что хватит места на вашей карте. Выбрав дистрибутивы для



» Выбираете, щелкаете — и всё! Вот бы так же просто было ставить Linux!

установки, нажимаете кнопку Install — и можно выпить чаю, в ожидании установки дистрибутивов на SD-карту.

## Не только для новичков

Если вы продвинутый пользователь Pi, можете также поиграть с настройками упомянутого дистрибутива, путем редактирования конфигурационного файла перед его установкой. Также предусмотрена опция получения онлайн-помощи.

После перезагрузки Pi вы получаете меню (аналогичное экрану загрузчика ОС Linux на ПК), где перечислены все установленные дистрибутивы, и вы выбираете тот, который хотите загрузить. Проекты вроде загрузчика ОС *BerryBoot* позволяют делать то же самое, но в отличие от них *NOOBS* позволяет каждому дистрибутиву использовать собственное ядро, позволяя загрузить ОС не-Linux, типа RiscOS.

Разработчики дистрибутива также предоставляют специально разработанные версии своих продуктов, предназначенные для конкретной цели или группы пользователей. В текущую версию включена опция *Boot to Scratch*, загружающая Raspbian прямо в программу *Scratch*. В стадии разработки — похожая опция, загружающая Pi прямо в платформу *Google Coder*. Если вы продвинутый пользователь Linux, можно также создать специальную версию одного из дистрибутивов, поддерживаемых *NOOBS*. Открывается ряд возможностей. Эта функция пригодится в учебной

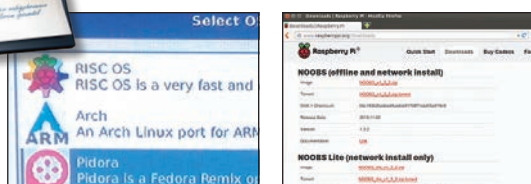
аудитории для формирования наборов Raspberry Pi с созданным вами дистрибутивом. А можно сделать резервное копирование измененной установки, для уверенности, что любая будущая установка не сотрет ваши модификации.

Установщик *NOOBS* доступен в виде сжатого образа объемом 1,2 ГБ и включает 6 различных дистрибутивов. «Облегченная» версия весит всего 20 МБ — во время установки Raspberry Pi она получает дистрибутивы из репозитория.

Неважно, используете ли вы Raspberry Pi как трамплин в мир бесплатного открытого ПО или вы уже опытный игрок команды Linux; скачав установщик *NOOBS*, вы доставите себе удовольствие. **LXF**



## Свойства навскидку



### Мультизагрузка

Можно установить несколько дистрибутивов на одну SD-карту и выбирать их во время загрузки.

### «Легкая» версия

При наличии безлимитного Интернета выбранные дистрибутивы загружаются уже в процессе установки.

## LINUX FORMAT Вердикт

### NOOBS 1.3.2

**Разработчик:** Raspberry Pi Foundation  
**Сайт:** [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads)  
**Лицензия:** MIT

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

» Идеально для начинающих. Скорость, функциональность и удобство налицо в этом отлично сделанном приложении.

**Рейтинг 9/10**

# Penguin Wee 4th Gen Desktop

Компактный, невероятно гибкий в настройке ПК с широким выбором предустановленных дистрибутивов. Разбирается **Мэтью Хэнсон**.

## Вкратце

» Компактный компьютер со сменяемыми компонентами и предустановленным дистрибутивом Linux на ваш вкус.

## Спецификация

- » Процессор Intel Core i5-4430, 3 ГГц
- » Видеокарта Intel HD Graphics 4600
- » ОЗУ DDR3, 16 ГБ
- » Жесткий диск 100 ГБ
- » Ethernet 10/100/1000
- » 7.1 audio
- » DVI, D-Sub и HDMI
- » 2×USB 2.0, 3×USB 3.0

**М**ало кто из производителей сегодня создает машины специально в расчете на пользователей Linux, и до чего ж приятно видеть эту специально скроенную для линуксоидов систему с использованием современных технологий, с предустановленной и готовой к работе Ubuntu 13.10. Конечно — в духе Би-Би-Си — приходится добавить, что и другие дистрибутивы тоже доступны. Конфигурацию Penguin Wee 4th Gen GNU/Linux Desktop можно полностью настроить еще до покупки, то есть сделать так, чтобы Penguin Wee полностью соответствовал вашим нуждам и вашему бюджету.

По умолчанию установлен Ubuntu 13.10, но на официальном сайте есть длинный список поддерживаемых дистрибутивов, с пометкой, что если даже вашего там нет, почти наверняка он тоже установится на Penguin Wee — разве что уж окончательно устарел.

Укомплектовать систему будет ничуть не сложнее: каждый сменяемый компонент снабжен выпадающим списком различных опций, и при необходимости указывается добавка к цене.

В плане стандартных компонентов, минимальный объем жесткого диска составляет 80 ГБ. Можно расширить его до 1 ТБ (приплатив \$144), или выбрать между сочетанием HDD/SSD или просто SSD (solid



» Дизайн выполнен в духе одной «фруктовой» калифорнийской компании.

state drive, твердотельный диск). Здесь разница более существенная — за SSD на 500 ГБ вы выложите уже на \$485 больше; но если финансы позволяют, стоит поразмыслить о приросте производительности, получаемом за счет отсутствия механических деталей и проворачивания дисков.

## Подбираем части

Предоставляется также выбор из ОЗУ DDR3 на 2, 4, 8 или 16 ГБ, и если вам нужна действительно передовая система, одновременно мощная и с запасом на будущее, позаботиться об этом в ваших силах. Вы можете также сами решить, нужны ли вам встроенный беспроводной адаптер

и оптический DVD-привод, а также сэкономить средства, отказавшись от монитора, клавиатуры и мыши, поставляемых в комплекте.

Сменяется большинство компонентов, но некоторые характеристики Penguin Wee 4 нанесены на каменные скрижали — ну или на белый пластик. Процессор указан однозначно, и хотя на сайте значится, что это Intel Core i3-413T, в полученном нами Penguin Wee 4 оказался Core i5-4430 с тактовой частотой 3 ГГц. Мы сообщили Think Penguin об этом несоответствии, и нам ответили, что уже после отправки нашей системы они перешли на Intel Core i3-4130T, потому что, как и другие процессоры 4-го поколения серии 'T', они лучше подходят для компактных систем, благодаря более низкому энергопотреблению и меньшему нагреву во включенном состоянии. Правда, это немного сказывается на производительности чипа по сравнению с нами полученным.

Кстати, у Intel есть и процессор Core i5, i5-4570T, тоже пригодный для компактных систем. Но его не приняли по причине применяемых в нем технологий от Intel, Trusted Execution Technology и vPro, из-за которых могли возникнуть проблемы совместимости с дистрибутивами Linux, а также проблемы с безопасностью, из-за функций удаленного доступа vPro.

Если вам нужна более мощная система, Think Penguin предлагает также модель с запредельными характеристиками

## Результаты тестирования

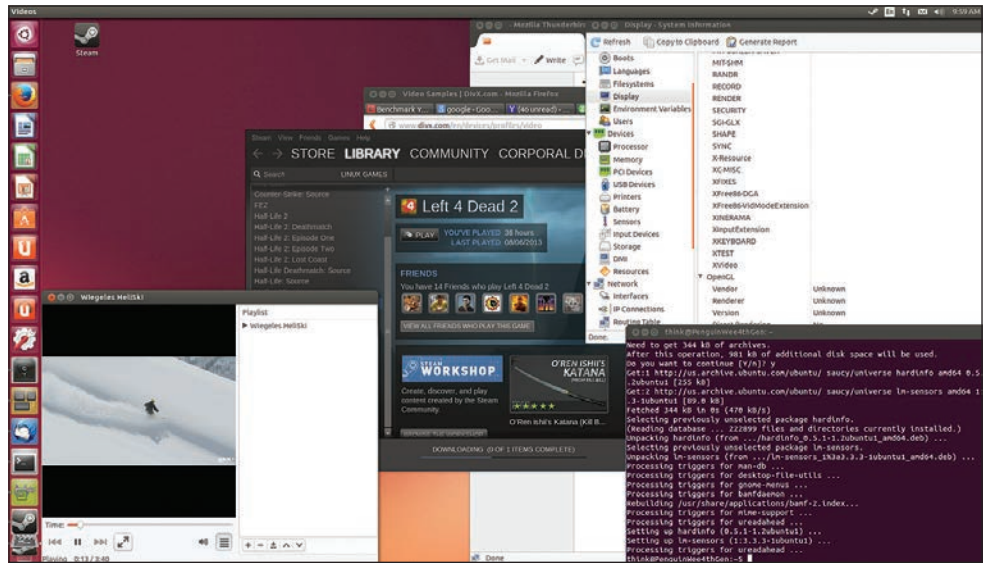
Тест	Penguin Wee 4th Gen GNU/Linux Desktop	Mirabox
Blowfish*	2,97	44,12
Cryptohash**	595,14	27,35
Fibonacci*	1,28	13,35
N-Queens*	3,77	142,11
FFT*	0,78	102,55
Raytracing*	4,96	506,57

\* Меньше — лучше. \*\* Больше — лучше.

процессора i7-4770K. Серия Intel Core i5-4430 — i3-4130T и i7-4770K — представляют четвертое поколение (отсюда и '4' в названии Penguin Wee) процессоров Intel Core, известных под кодовым именем Haswell.

В этом четвертом релизе масса новшеств и улучшений, по сравнению с предыдущим поколением процессоров Ivy Bridge, в том числе по энергоэффективности. Более всего выиграют от этого мобильные устройства, поскольку более низкое энергопотребление увеличит срок жизни батареи, и — как мы уже отмечали — уменьшит нагрев, повысив надежность в таких маленьких корпусах.

И это очень кстати, ведь Penguin Wee 4th Gen впечатляет своим минимализмом: размер его корпуса — всего 8×8×2,5 дюйма. Он легко транспортабелен и везде поместится. Белый корпус чем-то напоминает фирменный минимализм Apple: спереди есть только кнопка включения и логотип Think Penguin. Сзади же расположено множество портов, в том числе Gigabit Ethernet, два USB 2.0, три USB 3.0, порт eSATA, оптический и прочие аудио-порты. Подключить Penguin Wee можно к монитору или телевизору через D-Sub, DVI или HDMI.



» Многозадачность — да легко, при многоядерном процессоре и щедрым объеме ОЗУ.

сравнили их с результатами Mirabox, маленькой машинки с Linux на базе ARM (обзор см. в **LXF174**, стр. 21). Хотя напрямую их сравнивать не очень корректно (Mirabox гораздо дешевле и вообще-то предназначен для начального обучения программированию), нам показалось интересным посмотреть, как мощные компоненты Penguin Wee скажутся на общей производительности.

### Как по маслу

Четвертое поколение процессоров Intel Core включает новый и более мощный графический процессор (для Intel Core i5-4430 это Intel HD Graphics 4600). Это означает, что редактировать видео и просматривать его в высоком качестве, да и играть в некоторые игры вы сможете даже без отдельной графической карты, что позволяет еще больше уменьшить как размеры корпуса Penguin Wee, так и его стоимость.

Мы протестировали файл MKV в полном 1080-пиксельном разрешении — качество проигрывания и картинки было прекрасным. Мы также попробовали *Left 4 Dead 2*, игру на выживание среди зомби — не самую требовательную к графике, но когда за вами гонится толпа зомби, более слабым машинам приходится туго. При разрешении в 1080 пикселей и среднем качестве графики, Penguin Wee продемонстрировал высокую и стабильную частоту кадров.

Столь впечатляющая производительность графики превращает Penguin Wee в прекрасный медиа-центр — с учетом его скромных размеров, способный прижаться у телевизора в вашей гостиной. Правда, его миниатюрность имеет свою цену, ведь при столь тесном соседстве всех компонентов важно следить за температурой.

Как следствие, Penguin Wee, а точнее, встроенные в него вентиляторы, охлаждающие атмосферу внутри, изрядно шумит — если вы попытаетесь применить его для просмотра фильма, это может раздражать. И все же лучше маленькая и шумная машинка, чем тихая, но хронически ломающаяся.

В целом, Penguin Wee нас очень порадовал. Правда, мы бы придумали ему какое-нибудь другое название. Когда мы обсуждали в офисе его преимущества, все наши соседи думали, что нас чрезвычайно заботит мочеиспускание перелетных птиц [wee — *англ.* «пи-пи»]...

Здорово, что можно максимально настроить систему «под себя» еще до покупки, и что она так проста в плане запуска и работы. Очень порадовали и дизайн, и производительность. Если вам нужен маленький настольный компьютер, чтобы работать, смотреть видео и даже иной раз играть, вам определенно стоит глянуть в сторону Penguin Wee 4th Gen GNU/Linux Desktop. **LXF**

## «Работает Penguin Wee из коробки просто великолепно.»

Работает Penguin Wee из коробки просто великолепно: мы распаковали его, воткнули мышью с клавиатурой, подцепили монитор и включили. Настраивать или устанавливать ничего не потребовалось. Мы получили готовенькую Ubuntu 13.10 с именем пользователя и паролем по умолчанию (указанными на приложенном листе). Как и ожидается от компьютера со столь мощными характеристиками, система работала быстро, плавно и отзывчиво.

Тестирование тоже доказало мощь Penguin Wee (см. таблицу на стр. 20). Мы



» Множество портов для подключения периферийных устройств, включая быстрый USB 3.0 и eSATA. Кроме того, на Think Penguin Wee есть аудиоконнекторы для объемного звучания.

**LINUX** **Вердикт**  
FORMAT

**Think Penguin Wee 4th Gen**

**Разработчик:** Think Penguin  
**Сайт:** www.thinkpenguin.com  
**Цена:** \$449 (без подключаемых устройств, НДС, доставки)

<b>Функциональность</b>	<b>8/10</b>
<b>Производительность</b>	<b>9/10</b>
<b>Удобство использования</b>	<b>10/10</b>
<b>Оправданность цены</b>	<b>8/10</b>

» Потрясающий миниатюрный компьютер Linux с внушительным арсеналом готовых дистрибутивов.

**Рейтинг 9/10**

# Devolo LiveCam

Грэм Моррисон в курсе, что АНБ ужасно обременен занятиями слежкой, и приходит к ним на помощь, пристроив web-камеру к IP в своем офисе.

## Вкратце

» Web-камера, сообщаемая с Интернетом через powerline Ethernet. См. также: множество беспроводных web-камер на IP.

У нас есть простое правило: все, что мы рассматриваем, обязано поддерживать Linux. Было раз, когда мы рассмотрели ПО для Windows для восстановления файлов в разделе Linux, да еще кое-какое кросс-платформенное ПО на Java. Но в принципе — нет Тукса, нет обзора.\* Камера Devolo LiveCam становится одним из наших исключений. Тукс есть во всех ее рекламах и сбоку на коробке, но для работы поставляемого с ней ПО придется запускать Windows, через Wine. Devolo тут честен; но такое несколько удивляет.

Вероятно, вы слышали о Devolo и его превосходных адаптерах Ethernet, которые несут в ваш дом Интернет по проводам. Чтобы представить комплектацию LiveCam, воображение особо напрягать не потребуется: модуль web-камеры, созданной для работы с электросетями.

Есть два способа купить этот продукт: один комплект поставляется с адаптером 200 AVduo и web-камерой (такой нам и прислали), а другой включает только web-камеру. Первый полезен, если у вас



» Выглядит LiveCam элегантно, и действительно plug-n-play, но здесь явно не хватает продвинутых функций, уже ставших необходимой принадлежностью.

к сети; но, в отличие от практически любой другой web-камеры, больше ничего нельзя сделать, пока вы не установите ПО.

## Красное, красное Wine

Тут и настает черед совместимости с Linux, потому что на конце ссылки Devolo на скачивание — вышеупомянутый исполняемый файл Windows. Чувства наши по этому поводу смешанные: для той же работы есть прекрасные, простые в установке приложения Android и iOS, и они ставятся там, где вы, скорее всего, и будете его использовать. Вдобавок поддержка Linux у Devolo сомнительная, несмотря на Тукса на упаковке. В защиту компании скажем, что установка в Wine прошла гладко (правда, поломались шрифты), хотя web-камера находилась дома, а клиентское приложение мы установили в офисе. Мы просто зашли в Интернет с именем устройства и паролем — куда уж проще.

Сквернее исполняемого файла для Windows, однако, то, что собственное приложение Devolo весьма убого. Максимальное предоставляемое разрешение — 640×480, и все, что можно сделать — это подключиться, немного понаблюдать, сделать экранный снимок и отключиться. Нет обнаружения движения, нечем предупредить вас о появлении нарушителя,

нет кадровой кино съемки или режима видеозаписи. Ничего такого прикольного, что умеет ZoneMinder или даже приличный NAS от Synology или Qnap. LiveCam сгодится для присмотра над спящим ребенком, особенно с ее очень эффективным кольцом инфракрасных светодиодов для освещения темной комнаты, и позволит по-быстрому соединиться с Интернетом; но и рядом не стоит с обычной web-камерой Wi-Fi, особенно если учесть потенциальные уязвимости, добавляемые протоколами Devolo для доступа в Интернет. Идеальная камера для хомячков. LXF

## «Ссылка Devolo на скачивание — исполняемый файл Windows...»

нет коммуникационной сети, ведь вам понадобится всего лишь воткнуть адаптер по соседству с вашим маршрутизатором и соединить их прилагаемым проводом Ethernet. Но поскольку беспроводные интернет-камеры бывают и дешевле, цена как-то отвращает. Второй вариант лучше, так как позволяет присоединить LiveCam к вашей сети в местах, куда Wi-Fi не дотягивает, например, гараж или сарай. Этому способствует прочная конструкция, наличие отверстий для крепежа и обещание, что все будет работать на открытом воздухе — хотя камера не водонепроницаемая.

LiveCam подключается к компьютеру достаточно просто: воткните ее в разъем и нажмите кнопку шифрования на боку одного из адаптеров, а затем на боковой стороне LiveCam. Загорится небольшой зеленый индикатор, показывая, что новое устройство принято. Вроде и все, что требуется для подключения устройства

\* Нам, однако, случилось и поступаться этим золотым принципом. Да!

## LINUX FORMAT Вердикт

### Devolo LiveCam

Разработчик: Devolo  
Сайт: [www.devolo.co.uk](http://www.devolo.co.uk)  
Цена: £ 89,99/£119,99 (с адаптером)

Функциональность	4/10
Производительность	6/10
Удобство использования	7/10
Оправданность цены	5/10

» Вот бы она работала, как обычная web-камера, по IP, а не через убогое ПО от Devolo...

Рейтинг 5/10



# Broken Age

Евгений Балдин не удержался и сделал обзор игры — хоть и не AAA, но с многомиллионным бюджетом и GNU/Linux прямо из коробки.

## Вкратце

» Рисованный квест! Самый настоящий, пронзительно-искренний и приятно раскрашенный. Пока только половина, но и половина стоит пристального внимания.

Говорят, классический квест умер: настоящее заполнено безумными стрелялками и казуальными фермами. Но к счастью, идеи не умирают, пока живы их носители. Тим Шафер [Tim Schafer], создатель *Full Throttle* и *Grim Fandango*, решил восстановить справедливость и вернуть жанр квеста его почитателям.

Сейчас никого не удивит играми под GNU/Linux. Не удивит не только свободными, сделанными «за интерес» и этим ценными; не только уже многочисленными инди-проектами, использующими Unity3D в качестве игрового движка; но даже проекты AAA не являются чем-то уж космически недостижимым.

А ведь начался этот «игропад» относительно недавно. Лично у меня ощущение, что «началось», возникло, когда в марте 2012 года на Kickstarter вышел Тим Шафер и попросил денег на квест... Просто взял и попросил. Он ничего не показывал:



» Дом, милый дом, где все, кроме деда, хотят кормить тебя чудовищу. Тяжело быть молодой и красивой надеждой деревни в спасении от внимания чудовища.

## «Даже часть игры оказалась изысканно законченной.»

никакого геймплея, никаких концептов, никаких историй. А ему эти деньги дали; а потом еще, и еще, и еще... И так восемь раз и треть от этого. Прошло два года, и вот наконец-то Тим Шафер сообщил, что те 3 миллиона с гаком почти кончились, и для пополнения ресурсов он официально выпускает 27 января 2014 года первую половину истории,

а вторая будет в этом же году, но попозже. О, как вскинулись любители теорий заговора с предсказаниями, что деньги были потрачены на пиво и что это провал! И как же они были не правы — даже часть игры оказалась изысканно законченной; кроме самой истории, конечно.

Получился классический квест. Со своей стилистикой, со своим миром, с двумя разными историями. В любой момент можно переключиться между двумя разными главными героями и разными «театральными площадками». Если надоел излишне навязчивый в своей попытке защитить от любых тревог и потенциальных опасностей искусственный интеллект корабля, на котором существует юноша Шей [Shay], то всегда можно переключиться на целеустремленную в своем желании не быть съеденной девушку Веллу [Vella].

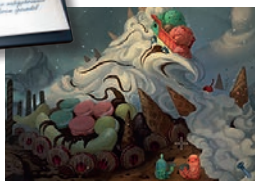
У вас есть классический инвентарь, вы собираете разного рода объекты, ходите и пытаетесь применить эти объекты самыми разными способами, разговариваете с окружающими и глазеете на декорации. «Пиксель хантинг», как и в каждом классическом квесте, присутствует, но из-за значительных размеров активных областей он не создает проблем. Почти любая попытка неверно использовать предмет на активной области комментируется. Поэтому, если не торопиться, то нет необходимости решать загадки — можно просто заниматься случайным перебором всех возможностей, благо реакция на неверные действия тоже может порадовать.

Новелла, точнее первая ее половина, не сильно протяженная. Если не развлекаться с перебором заведомо диких вариантов и не глазеть на открывающиеся виды слишком долго, то около четырех часов вполне достаточно для прохождения. Хотя посмотреть есть на что. Анимировано все, что можно и нужно. Нет совершенно статичных видов. Протагонисты не стоят на месте, а переминаются с ноги на ногу, меняют позы и оглядываются. Вид героев мультяшный, но они весьма по-взрослому ехидничают, хоть в них и нет брутальности, присущей предыдущим творениям мастера.

К сожалению, в данной новелле не предусмотрено русских субтитров, поэтому детям будет сложно; но их родители игра порадует. **LXF**

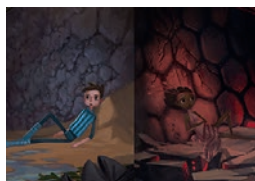


## Свойства навскидку



### Отрисовка

Замечательная мультипликация. Анимированы не только герои, но и все, что их окружает.



### Добрая история

Хоть и не в сильно добром окружении, но вполне логичная и с достойной концовкой.

## LINUX FORMAT Вердикт

### Broken Age

Разработчик: Double Fine Adventure  
Сайт: <http://www.brokenagegame.com/>  
Цена: 449 руб. на Steam

Сюжет	9/10
Графика	10/10
Увлекательность	8/10
Оправданность цены	8/10

» Не вышибает дверь ботинком, но заползает в души игроков доброй рисовкой, светлыми устремлениями и чарующей озвучкой.

Рейтинг **9/10**



## ПЛАНШЕТЫ

# Nexus от HTC

HTC может выпустить планшет Nexus.

Возможно, в текущем году линейку планшетов Nexus ждет пополнение — наряду с Asus и Samsung их выпуском займется компания HTC. У HTC уже есть опыт производства таких аппаратов, и вероятность расширения сотрудничества оценивается экспертами как достаточно высокая. Потребителю HTC известна в основном по смартфонам. Ее опыт выпуска планшетов был весьма невелик, да и удачным его назвать нельзя — мало кто сейчас помнит названия Flyer и Jetstream. Однако уже в этом году ситуация может радикально измениться.

Дело в том, что несмотря на некоторое смещение жанров, главная разница между смартфоном и планшетом заключается в том, что первый пока остается продвинутым телефоном, а второй — своеобразным мини-компьютером. И требования

к ним из-за этого радикально разные: если смартфон воспринимается как некий готовый программно-аппаратный комплекс, то планшет всегда требует некоторой индивидуальной настройки.

Отсюда напрашивается вывод: любое предустановленное ПО на планшете будет восприниматься значительно большим злом, чем на смартфоне. На практике это приводит к тому, что значительная часть опытных пользователей выбирает планшет линейки Nexus хотя бы потому, что на нем установлено значительно меньше лишних приложений.

Интрига заключается еще и в том, что обновленный 7-дюймовый планшет Nexus пользователи уже увидели, а вот судьба десятидюймового варианта пока остается под вопросом. Кстати, продажа Nexus 10 производства компании Samsung



» Flyer — не самый успешный опыт планшетостроения.

на Google Play прекратилась еще в январе текущего года.

Резонно предположить, что новая «десятка» Nexus будет принадлежать к классу Hi-End как по «начинке», так и по качеству экрана. Впрочем, рассуждать об этом пока можно только в режиме догадок и предположений. С прагматичной точки зрения очевидно одно: если вы решили приобрести десятидюймовый планшет, то покупку лучше отложить. А вот когда появится новый Nexus, тогда и идти в магазин.

## РАЗРАБОТКА

# Запускаем Firefox

Firefox продвигается на Android.

Компания Mozilla и стартап EverythingMe объявили о создании нового пускателя для платформы Android с вполне предсказуемым названием *Firefox Launcher*. Также нетрудно догадаться, что основан он будет на знаменитом свободном браузере *Firefox* в сочетании с возможностями контекстного поиска EverythingMe.

Одна из главных особенностей нового *Launcher'a* — показ не только установленных, но и доступных для загрузки приложений. Причем программа будет пытаться фильтровать результаты в соответствии с возможными потребностями пользователя в текущий момент времени.

Например, в начале дня *Firefox Launcher* будет предлагать установить приложения с прогнозом погоды, а ближе к обеду — инструменты для поиска ближайшего ресторана. Сама по себе эта идея достаточно

перспективна, хотя очень многое будет зависеть от реализации, поскольку в данном случае легко перешагнуть грань, отделяющую полезную информацию от навязчивой рекламы.

При этом поисковая система будет находить приложения по любым запросам и сразу же иллюстрировать их работоспособность в соответствии со смыслом запроса. Например, если пользователя интересует Android, то *Firefox Launcher* покажет все приложения для чтения технологических новостей, причем соответствующие дайджесты отобразятся на экране.

Удобство такого подхода сложно переоценить, поскольку он позволяет существенно сэкономить время, которое обычно тратится на загрузку кучи приложений и последующего их тестирования в контексте актуальной задачи. Разумеется, вряд ли *Firefox Launcher* сразу же



» Внешний вид нового *Launcher'a*.

предложит оптимальный вариант, но с его помощью станет возможно отказаться от заведомо непригодных.

На момент написания новости приложение проходило бета-тестирование и было недоступно для загрузки. Официальная информация о выходе первого релиза пока отсутствует.

ПРИЛОЖЕНИЯ

# Выход 2GIS Dialer

2GIS Dialer — очередной удар по анонимности.

**Н**аверняка практически всем пользователям системы Android знакомо приложение 2GIS — лучшая на сегодняшний день географическая карта с массой дополнительных возможностей. Она является максимально полным справочником, который может быть полезен не только путешественникам, но и местным жителям — благодаря исключительно подробной информации о расположенных в любом городе организациях.

Немаловажную роль в популярности 2GIS сыграл продуманный интерфейс, избавленный от практически всех лишних сущностей. Пользователю достаточно найти интересующую его организацию по названию, адресу или каким-то примерным ориентирам, и стукнув по зданию, он получит практически все требуемые ему сведения, включая контакты.

2GIS Dialer — логическое продолжение старой доброй 2GIS, но с расширенной

функциональностью, ориентированной не на планшеты, а на современные смартфоны. Его главная особенность — автоматическое определение звонков, поступающих от организаций, находящихся в базе данных 2GIS.

Пользу этой возможности легко понять на простом примере. Допустим, у вас зазвонил телефон и вы видите на экране просто номер, поскольку в ваших контактах этого абонента нет. А программа 2GIS Dialer может сообщить вам, что данный номер принадлежит какому-нибудь банку, сотрудник которого наверняка начнет предлагать вам ипотеку, автокредит или иное надувательство.

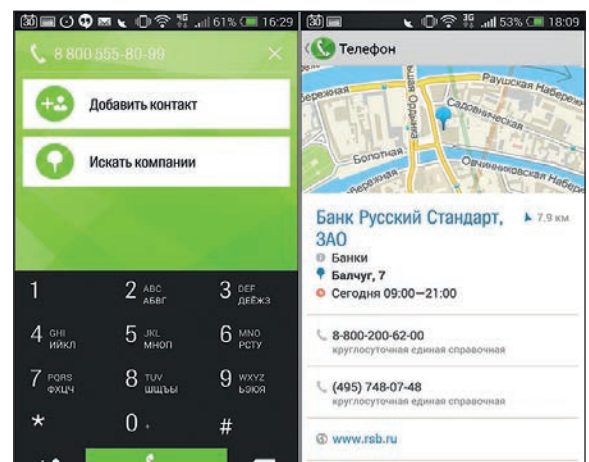
Поэтому лучше сразу нажать на отбой. В противном случае звонки будут повторяться, и вы рискуете стать неврастемником: ведь давно известно, что отвечать на спам — самая плохая идея.

Поиск осуществляется не только по внутренней базе системы 2GIS,

но и по учетным записям Google. Таким образом, вы сможете быстро деанонимизировать себя обычным добавлением своего телефонного номера в учетную запись Google. Все пользователи смартфонов, установившие 2GIS Dialer, будут видеть, кто именно им звонит.

Однако есть и ложка дегтя. Поскольку программа бесплатная, пользователи жалуются на то, что после нажатия на кнопку «звонить» они вынуждены созерцать рекламу. К сожалению, платной версии нет, и откупиться от этой функции, по всей видимости, пока невозможно.

► Приложение 2GIS Dialer.



ЮЗАБИЛИТИ

# Многооконный интерфейс

Компания LG сделала QSlide доступным для всех разработчиков.

**О**дним из основных недостатков системы Android считается фактическое отсутствие многооконного режима. Разве что некоторые плееры типа DicePlayer умеют показывать видео в отдельном окошке. Но, как правило, привычная всем настольным пользователям функция на планшетах пока недоступна...

За исключением смартфона G2 от компании LG, на которых реализована фирменная функция QSlide. По мнению экспертов, она стала наиболее полезным и востребованным дополнением, благодаря которому пользователь может открывать на одном экране несколько приложений. Случай, кстати, довольно редкий — чаще всего предустановленные программы от вендоров вызывают только раздражение.

До недавнего времени QSlide обслуживало ограниченное число приложений.

Теперь же компания сообщила о выпуске SDK, позволяющего внедрить эту возможность в любую программу. Инструментарий предусматривает несколько API, позволяющих встроить соответствующие управляющие кнопки в интерфейс программы и область уведомлений.

Увы, автоматически это не решает проблемы, поскольку возможность определяется не рабочей средой, а конкретным приложением. Вовсе не факт, что независимые разработчики добровольно согласятся внедрить в свои программы возможность делиться экраном с другими программами от других разработчиков.

Вероятнее всего, последнее слово останется за потребителем. Если он начнет явно предпочитать приложения с такой функциональностью, то игнорирующие его разработчики будут вынуждены вносить изменения в свои приложения. **LXF**



► На смартфонах G2 реализована функция QSlide.



# На волю, Android!

Не становится ли Android менее открытым? **Нейл Ботвик** утверждает, что да, и исследует варианты возможных встречных действий сообщества.

**М**ы любим воспевать Android как пример успеха открытого ПО в коммерческом мире, к тому же в довольно жесткой части этого мира. Все разговоры о том, «станет ли этот год годом Linux на рабочем столе», затихли, когда ОС на основе Linux, хотя и не того GNU/Linux, к которому мы привыкли, телефон за телефоном начала завоевывать мир.

Но мы гораздо меньше думаем о том, что хотя у Android открытое ядро, да и львиная доля остальной ОС тоже открыта, доля закрытого кода

в Android все растет и растет, как растет доля этого кода в сетевых сервисах, от которых так зависит Android.

**«Ваш телефон с открытым кодом зависит от серверов с закрытыми сервисами.»**

Смартфоны и планшеты — сетевые устройства: без подключения к Интернету их возможности серьезно ограничены, и телефоны Android по умолчанию используют сервисы

Google. Код этих сервисов не открыт — GPL не применяется к ним по тому типу, как к программам, поскольку распространяются не программы, а только результаты

их работы. Поэтому ваш телефон с открытым исходным кодом зависит от целого ряда серверов с закрытыми сервисами. Да, в качестве почты вы не обязаны брать Gmail, но есть ли открытая альтер-

натива навигации и картам Google Maps? OpenStreetMap неплохо развивается, и все же по части охвата или лоска навигации до Google Maps ей далеко.

# Открыв, закрыли



**В** открытости Android имелся немалый смысл для Google еще в 2007 году; тогда появился iPhone, и Google хотел избежать доминирования одной операционной системы в мобильной экосистеме, как произошло с настольными компьютерами. Поэтому Google выпустила Android открытым, и ОС начала быстро развиваться. Теперь Android доминирует, и Google хочет забрать свой контроль обратно. С ядром и другими открытыми компонентами, полученными извне, Google сделать ничего не может, но собственное ПО мало-помалу делает закрытым.

В проекте AOSP (Android Open Source Project — открытый проект Android) все еще доступны все программы для Android, выпущенные под открытыми лицензиями, но так как это код Google, компания может выпустить его и под другой лицензией, что и происходит. Каждый раз, когда Google переводит одно из приложений на закрытую лицензию, разработка AOSP-версии фактически прекращается, и не добавляется никаких новых функций. Возможно, вы заметили, что в нескольких последних обновлениях Android — различных версиях Jelly Bean и KitKat — новых функций было не слишком много. И не потому, что

Google делает меньше, а потому, что это делается иначе. Многие программы, входившие в основной релиз, перемещены в сервисы Google Play (это не то же самое, что хранилище Google Play). Преимущество для Google в том, что основные приложения больше не привязаны к релизу Android и могут обновляться в любое время (все они теперь есть в хранилище Google Play), чтобы пользователей не оставляли в неопределенности производители телефонов и операторы, которые не могут или не хотят предоставлять обновления ОС. Проводя аналогию со знакомым миром настольного GNU/Linux, Android стал ближе к скользящему релизу, такому как Arch, и ушел дальше от модели, принятой другими дистрибутивами, когда ошибки в пакетах исправляются только с выходом новой версии ОС.

Возможность установки и обновления пакетов в Play Store также означает, что их можно удалить, тогда как в прежнем варианте можно было удалить только обновления основных программ Google, откатив их до версий, которые были в релизе Android.

## Свободные варианты

Но это не полная блокировка Android со стороны Google. Большая часть Android все-таки открыта — закрыты лишь программы Google и сервисы, которыми они пользуются. Значит, как всегда, есть варианты. Купив телефон Samsung — а они есть у многих — вы увидите, что многие программы принадлежат Samsung. При желании вы можете установить программы от Google, но это не обязательно и даже не нужно. Amazon с Kindle Fire пошел еще дальше — собственная версия его программы даже есть в Play Store. Хотя Amazon не назовешь маяком для всего открытого, но компания показала, что Android без Google возможен. Возникает вопрос: можем ли мы сделать нечто подобное — пользоваться телефоном или планшетом Android без закрытых приложений Google или кого-то еще?

Прежде чем отвечать на этот вопрос, зададимся другим, более фундаментальным: а зачем нам это? Программы Google работают, и очень хорошо; стоит ли нам лишаться комфорта хорошо протестированной и популярной экосистемы Google? Многие пользователи, которые просто хотят, чтобы их телефон работал, вполне могут ответить «нет». Но вы не обычный пользователь, иначе вы бы не стали читать журнал о компьютерах, тем

более о Linux. Пользователи Linux хотят пользоваться тем, что лучше, а не легче всего; отчасти поэтому Linux и развился до своего нынешнего уровня.

Следующий вопрос: вы хотите вообще избавиться от всего проприетарного кода или только от Google и его сервисов? И то, и другое возможно, хотя первый вариант сложнее. Отнюдь не склонные избегать трудных вопро-

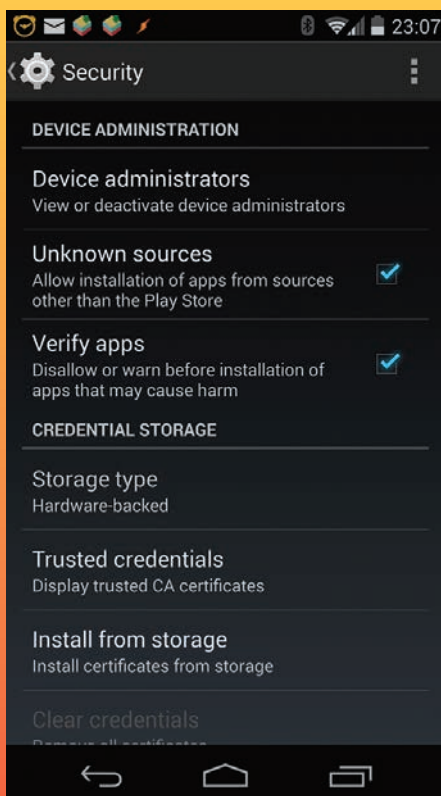
## «Это не полная блокировка Android от Google. Большая часть Android открыта.»

сов, мы обратимся к первому и скажем, что разберемся с ним позже. Для второго варианта нам нужна программа под названием *F-Droid* (<https://f-droid.org>).

## F-Droid

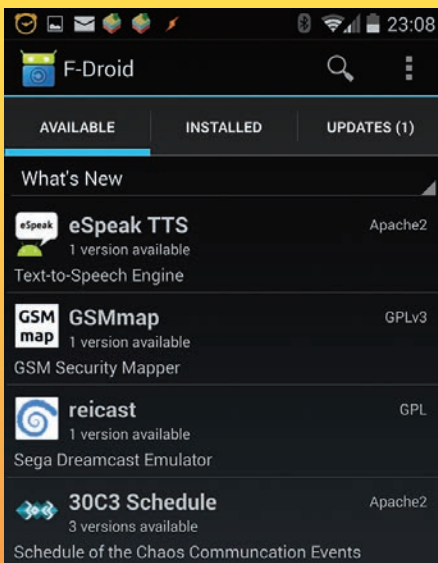
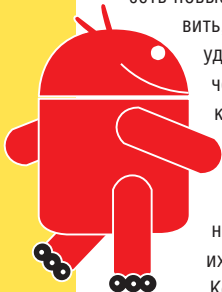
Название F-Droid носят две отдельные, но связанные между собой вещи. Во-первых, это репозиторий приложений для Android, подобный своему предшественнику Play Store Android Market, с одним важным отличием. Каждая программа в нем бесплатна, и исходный код ее открыт. Во-вторых, это клиент *F-Droid*, используемый для просмотра каталога F-Droid (каталог нельзя назвать рынком [market] или хранилищем [store], раз все здесь бесплатно). Такого клиента нет в Play Store, и он устанавливается прямо с сайта. Чтобы это сделать, нужно разрешить устройству загружать программы с других источников: зайдите в Settings > Security [Настройки > Безопасность] и поставьте галочку у Unknown sources [Неизвестные источники]. Учтите, что это считается риском для безопасности устройства, так как вы сможете устанавливать что угодно откуда угодно; но это ваш телефон, ваш выбор и ваше право устанавливать то, что вы хотите — даже если производители смартфонов считают иначе.

Теперь зайдите на <https://f-droid.org> и загрузите пакет *apk*. Если вы делаете это в браузере телефона, можно просто открыть файл пакета, и он будет установлен. Если вы загрузили файл на компьютер, загрузите файл на телефон через USB, файлообменник или карту памяти и откройте его. Другой вариант — открыть сайт в браузере на компьютере и отсканировать QR-код телефоном. Запустив *F-Droid*, вы увидите три вкладки: доступные приложения, установленные приложения и приложения, для которых есть обновления. Вас может удивить, что на вкладке



» Перед установкой *F-Droid* или любого другого пакета не из Play Store нужно разрешить установку программ из неизвестных источников.

«Установленные» уже есть программы — тут дело в том, что многие открытые программы F-Droid есть также в Play Store. Еще удивительнее то, что для некоторых из них есть обновления — хотя в Play Store они отображаются как обновленные, в F-Droid для некоторых из них есть новые версии. Попробовав их обновить, вы увидите, что сначала нужно удалить версию из Play Store, причем удалятся все ваши настройки. Некоторые программы резервного копирования, вроде *oandbackup* (естественно, из F-Droid) позволяют сохранить настройки и восстановить их после установки новой версии. Конечно, в каталоге F-Droid меньше приложений, чем в Play Store, но их довольно много, число их растет, и качество хорошее: здесь точно нет калькуляторов чаевых и имитации испускания газов. Приятная возможность, которой нет в Play Store — список новых программ, добавленных за последние две недели [What's New] — временной интервал, разумеется, настраиваемый.



➤ F-Droid показывает список не только всех доступных и установленных приложений, но и недавно добавленных программ. И все это открытое.

Мы коснулись приложений и репозитория; а как насчет основного компонента устройства — ПЗУ с ядром? Прошивка ПЗУ по большей части открыта — особенно ядро; это означает, что доступны и другие прошивки, например, Cyanogen Mod, о которой мы рассказывали в LXF177. Одно из преимуществ пользовательских прошивок в том, что во многих телефонах установлены измененные прошивки, отвечающие представлениям разработчика о внешнем виде и работе Android. В пользовательской прошивке этого нет: в ней (например, в семействе Nexus) используется версия ближе к версии по умолчанию, которая часто легче и быстрее. В некоторых прошивках добавлены собственные изменения, но раз уж вы решились запустить другую прошивку, то сумеете определиться со степенью ее приближения к стандартной. Преимущество пользовательских прошивок в том, что обычно они дают root-доступ (см. врезку на соседней странице) — хотя на моем Nexus 5 используется стандартная прошивка, но с открытым доступом root.

## Replicant

F-Droid, с которым вы пользуетесь только свободными программами — шаг в верном направлении; ну, а если вам нужна вполне свободная система? Это далеко не просто, из-за обилия проприетарных программ на наших устройствах. С одной стороны, это Google — все программы, которые работают с сервисами Play Services и пользуются сервисами Google. С этим можно бороться, хотя вы можете утратить часть возможностей, к которым привыкли. Сложнее с драйверами устройств — большинство из них в какой-то мере проприетарны. Эта проблема похожа на аналогичную с устройствами для настольных компьютеров и ноутбуков, но часто гораздо сложнее. В мобильной операционной системе без драйвера для телекоммуникационного оборудования столько же смысла, сколько

```
[nelz@hactor FreeAndroid/replicant 0%] for i in *.img
do
sudo fastboot flash $(basename $i .img) $i
sleep 5
done
sending 'boot' (2592 KB)...
OKAY [ 0.419s]
writing 'boot'...
OKAY [ 0.353s]
finished. total time: 0.772s
sending 'recovery' (4176 KB)...
OKAY [ 0.664s]
writing 'recovery'...
OKAY [ 0.543s]
finished. total time: 1.208s
sending 'system' (202532 KB)...
OKAY [ 31.807s]
writing 'system'...
OKAY [ 27.480s]
finished. total time: 59.287s
```

➤ Прошивка файлов образов Replicant в телефон.

в шоколадной каминной решетке. Те же проблемы касаются видеокарт, особенно аппаратного ускорения.

## «В каталоге F-Droid меньше приложений, чем в Play Store, но число их растет.»

Однако мир не стоит на месте, и появилась бесплатная замена для Android — это *Replicant* (<http://replicant.us>). Из-за проблем с драйверами устройств это не полноценная замена Android на всех устройствах — на момент написания статьи устройств на сайте 11, и не все функции работают на всех устройствах без установки платных прошивок. В основном в таких прошивках нуждаются Wi-Fi, Bluetooth и некоторые камеры.

Установка *Replicant* во многом напоминает прошивку пользовательского ПЗУ (чем по сути и является *Replicant*, только код ее полностью открыт). Как и всегда в таких случаях, сначала нужно сделать резервную копию, желательно с помощью Nandroid, если сможете ее восстановить, так как при перепрошивке ПЗУ сотрутся все ваши приложения и настройки. Также до перезаписи текущей системы прочтите подробную информацию о прошивке. Процесс установки различен в разных телефонах — в Samsung используется программа *heimdall*, а в Nexus — *fastboot*. Здесь мы рассмотрим установку на Nexus S, так как у нас есть один под рукой. Программа *fastboot* входит в Android SDK или, в некоторых дистрибутивах, в отдельный пакет *android-tools*. Ее также можно загрузить с сайта *Replicant*. Вам также понадобятся несколько файлов образов для записи на устройства, поэтому зайдите на <http://replicant.us/supported-phones> и пройдите по ссылке в вики для вашего устройства. После этого щелкните по ссылке для установки и затем по ссылке образа *Replicant*. Здесь нужно загрузить четыре файла образа — **boot.img**, **recovery.img**, **system.img**, **userdata.img** — и файл контрольной суммы MD5. После этого щелкните по ссылке Base URL и заберите *fastboot*

## Конфиденциальность

Одна из причин, по которым люди хотят уйти от Google (и других облачных сервисов) — конфиденциальность. Неудивительно, что люди хотят, чтобы за их перемещения не следили другие, особенно в свете последних новостей о PRISM и всяких там государственных агентствах. Вы можете решить, что вернули себе конфиденциальность, избегая программ Google, но полной конфиденциальностью на самом деле ни один из обладателей мобильных

телефонов не располагает. Как телефон способен довольно точно определить свое положение без GPS — с помощью сотовых вышек, так и ваш провайдер; и все ваши разговоры, данные и сообщения все равно проходят через него. АНБ и Центру правительственной связи не нужен Google; единственный способ сохранить полную конфиденциальность с мобильным телефоном — это не иметь его. Это не значит, что нужно сдаваться, но будьте реалистичны в своих ожиданиях.

из каталога **tools**. Сохраните все файлы в один каталог и перейдите в него в терминале. Если вы загрузили *fastboot*, сделайте ее исполняемой командой:

```
chmod +x fastboot
```

Если у вас уже установлен *fastboot*, замените `./fastboot` на `fastboot` в каждой из следующих команд. Переведите телефон в режим *fastboot*, выключив его и удерживая кнопки включения питания и увеличения громкости до появления экрана загрузчика с заголовком "FASTBOOT MODE [Режим FASTBOOT]". Теперь подключите телефон к компьютеру через USB. Если вы не меняли прошивку и не открывали `root`-доступ, понадобится разблокировать загрузчик. В случае сомнений посмотрите на строку "LOCK STATE [состояние блокировки]" на экране загрузчика. Если загрузчик заблокирован, выполните команду

```
sudo ./fastboot oem unlock
```

Для подтверждения выберите нужный вариант с помощью кнопок управления громкостью и нажмите кнопку включения питания, чтобы принять его. Теперь запишите образы в устройство следующими командами:

## «Приятно, что телефонная ОС с открытым кодом есть и всегда будет.»

```
sudo ./fastboot flash boot boot.img
sudo ./fastboot flash recovery recovery.img
sudo ./fastboot flash system system.img
sudo ./fastboot flash userdata userdata.img
```

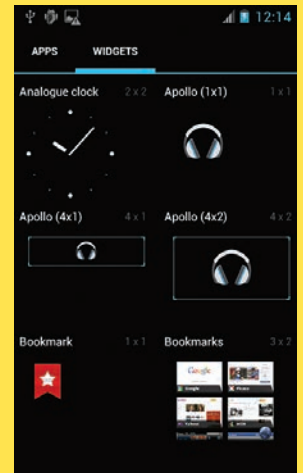
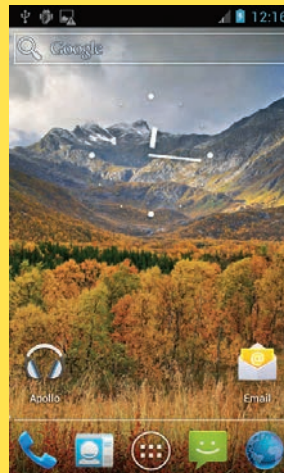
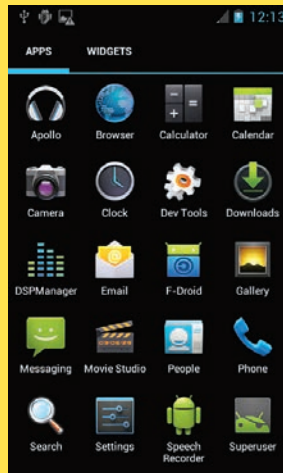
По завершении каждой команды на экране должна появиться надпись "Write Success! [Запись успешно завершена!]". На третьей команде понадобится больше времени: **system.img** — самый большой из файлов. Наконец, очистите кэш и перезагрузите телефон:

```
sudo ./fastboot erase cache
sudo ./fastboot reboot
```

При загрузке должен появиться логотип *Replicant*, после чего откроется стартовая страница. Вы поймете, что больше не прикованы к Google, по тому, что вас больше не попросят сначала войти в учетную запись Gmail.

## Сохраняем прошивку

Как мы упоминали, для некоторых функций нужны файлы прошивки. У *Replicant* их нет, и даже нет информации о том, как их получить, хотя необходимые файлы перечислены на странице вики для каждого устройства. Если в вашем телефоне уже открыт `root`-доступ, можно просто скопировать эти файлы в безопасное место и восстановить их после установки *Replicant*. Для этого после установки *Replicant* обычно нужно открыть `root`-доступ к телефону, установив программу восстановления. Я предпочитаю *TWRP* с сайта <http://>



» В *Replicant* солидная подборка приложений и виджетов по умолчанию; еще больше их в *F-Droid*.

[teamw.in/project/twrp2](http://teamw.in/project/twrp2). Загрузите файл образа для вашего устройства и запишите его в телефон с помощью *fastboot* командой

```
sudo ./fastboot imagefile
```

После этого можно перезагрузиться в загрузчик, выбрать `Recovery` [Восстановление], перейти в раздел `Mounts and Storage` [Точки монтирования и устройства хранения данных], смонтировать `/system` и отправить каждый файл с компьютера при помощи программы *adb*, которая находится там же, где и *fastboot*: это программы-компаньоны. Выполните команду

```
sudo ./adb push firmware.file /system/vendor/firmware/firmware.file
```

указав соответствующие пути для вашего устройства. Если поменять аргументы местами и указать **pull** вместо **push**, то файлы скопируются с телефона на компьютер; так удобно сделать с самого начала.

```
sudo ./adb pull /system/vendor/firmware/firmware.file firmware.file
```

С помощью такого восстановления можно сделать резервную копию Nandroid для восстановления системы в предыдущее состояние (при необходимости), поэтому стоит

установить его до установки *Replicant*. После загрузки *Replicant* начните исследовать образ. Он покажется вам знакомым — по сути, это Android. По умолчанию установлено несколько программ; гораздо больше доступно через *F-Droid*. Устанавливать *F-Droid* в *Replicant* не надо, он включен в *Replicant* как репозиторий по умолчанию и файловый менеджер. Если вы хотите вернуться к варианту по умолчанию, возможностей несколько. Если вы делали резервную копию в Nandroid, просто восстановитесь из него. Также можно установить одну из пользовательских прошивок обычным образом. Если у вас Nexus, можно вернуться к конфигурации по умолчанию, загрузив и прошив соответствующий образ с <https://developers.google.com/android/nexus/images>. Эти образы имеют формат tar-архивов, которые содержат файлы образа и скрипт для их установки. Достаточно распаковать архив и выполнить команду

```
sudo ./flash-all.sh
```

Если система кажется вам слишком ограниченной — ну, по крайней мере, у вас есть выбор. В любом случае, не отказывайтесь от *Replicant* — заходите на его сайт, ведь разработка продолжается. Приятно знать, что телефонная ОС с открытым исходным кодом есть и всегда будет. **LXF**

## Открываем root-доступ в телефоне

Об открытии `root`-доступа в телефонах есть много ложной информации. Если оставить в стороне контрактные обязательства по отношению к поставщику телефона, `root`-доступ не делает телефон более уязвимым, так как появление прав `root` у телефона не означает их появления у приложений. Это как команды **su** или **sudo** в обычных системах: они позволяют запускать команды как `root`, но не дают прав `root` кому-либо еще. Открытие `root`-доступа в телефоне

означает, что если вы хотите запустить программу от имени `root`, она должна спросить у вас разрешение, которое можно дать на один раз или навсегда. Если за несколько секунд вы не дадите разрешения, в нем будет отказано. Для пущей безопасности программа *SuperSU* по умолчанию дает доступ только запросившей его версии программы. По обновлению программы все придется повторить. Ни одна программа не запустится от имени `root` без вашего разрешения.

# Сравнение

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

## Клиенты VoIP

Маянк Шарма рассматривает ряд лучших приложений VoIP, которые помогут вам в общении, обмене файлами и сотрудничестве.



### Про наш тест...

Все программы Сравнения предусматривают самые популярные функции VoIP, но реализуют их по-разному. Мы расскажем об их интеграции с рабочим столом и протестируем их функции безопасности и способности к потоковому вещанию с настольного ПК. Мы также оценим их производительность в многопользовательских сессиях и поиграем с их функциями над звонками.

Самым популярным и широко используемым протоколом для общения через VoIP является SIP, но некоторые клиенты в нашем Сравнении поддерживают также ряд других, и это ставит их впереди тех, что не могут похвалиться такой поддержкой.

Чтобы честно испытать все клиенты VoIP, мы сначала использовали один и тот же клиент по обе стороны коммуникации. Еще мы тестировали связь клиента и сервиса (например, Empathy с Google Talk), но не в течение долгого периода. Все клиенты протестированы на протоколах Google Talk, XMPP и SIP.

### Наша подборка

- » Empathy
- » Ekiga
- » Jitsi
- » Skype
- » Yate

**Д**ля многих пользователей звонки и собрания — ежедневная рутина. Однако они могут стать трудным (и недешевым) делом, если ваши собеседники разбросаны по всему миру.

Тут-то и приходит на помощь VoIP. Он позволил абонентам снизить затраты, перенаправляя звонки через Интернет вместо традиционной телефонной сети PSTN. VoIP не бесплатен, однако сопутствующие

ему расходы (в первую очередь, на широкополосную связь) значительно ниже.

Есть и другие причины использовать VoIP, помимо экономии средств на переговоры с географически рассредоточенной аудиторией: он также весьма удобен в закрытой сети благодаря предлагаемыми им функциям, невозможным при традиционной форме общения.

Даже для начинающих настройка конференц-звонка с несколькими участниками не сложнее прогулки по парку. VoIP дорос до работы с несколькими видами медиа помимо голоса. Вместе с аудио вы можете передавать изображения,

видео и текст, а что еще удобнее — просто выдать другим доступ к своему ПК.

Это делает VoIP весьма эффективным для разных сценариев в плюс к обычному разговору. Он делает встречи более интерактивными. Можно применить его для обеспечения удаленного содействия, или для поддержки контакта с другими игроками во время многопользовательской игры, или для добавления нового измерения в хостинг подкастов.

На следующих страницах мы рассмотрим пять лучших инструментов, способных помочь вам полностью задействовать всю мощь VoIP.



# Функции вызова

## Способны ли они обеспечить видеозвонок?

**К**роме клиента *Yate*, который способен только на аудиозвонок, все остальные приложения в нашем Сравнении умеют проводить как аудио-, так и видеопереговоры.

Все клиенты поддерживают разнообразные кодеки VoIP. *Jitsi*, *Ekiga* и *Skype* еще и позволяют включать или отключать определенные кодеки и даже выставлять их приоритет. Все поддерживают популярный аудиокодек Speex, который минимизирует занятость канала, варьируя скорость передачи данных. Кроме того, они поддерживают кодек iLBC, очень устойчивый к потерям пакетов, а *Jitsi* и *Ekiga* — еще и аудиокодек SILK от *Skype*.

Для улучшения качества звука некоторые клиенты, такие, как *Empathy*, *Ekiga* и *Jitsi*, также умеют включать функции эхо- и шумоподавления.

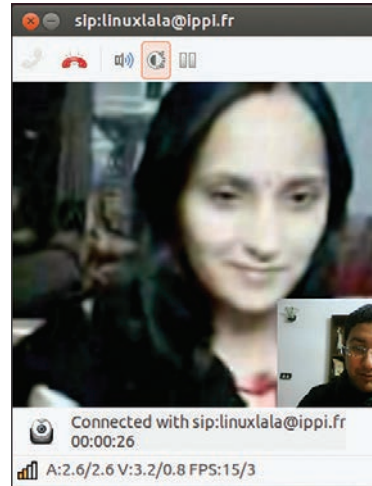
*Skype* использует H.265 для видео с высоким разрешением и кодек VP8 для стандартного видео, и все это налицо у *Jitsi* — как и кодеки H.264 и H.263, имеющиеся у *Ekiga*.

Все клиенты Сравнения справлялись с одиночными аудио- и видеозвонками

относительно легко, причем практически не наблюдалось задержек звука или искажений видео. Мы также запускали клиенты, когда сеть была занята скачиванием торрента. Даже в этой ситуации ни у одного клиента не возникло трудностей с аудиозвонками. Но с видео ситуация была иной. Только *Skype* и *Jitsi* смогли установить соединение с достойным качеством видео.

Далее, *Ekiga* и *Jitsi* также позволяют настроить разрешение видео. *Ekiga* поддерживает разрешение от 176×144 до 704×576, а *Jitsi* работает с разрешениями от 160×100 до 1280×720, и даже позволяет настроить индивидуальную скорость фреймов и битрейт для видео.

Некоторые клиенты ограничивают функциональность протоколов, которые они поддерживают. Например, *Jitsi* предлагает совершать аудио- и видеозвонок только через SIP и XMPP. Насчет *Jitsi* мы еще заметили, что хотя он справляется с любым сервисом XMPP и SIP, соединения, установленные с помощью его сервиса Jit.si XMPP, стабильнее других сервисов.



В *Jitsi* также встроена функция записи разговора: можно записывать аудио, созданное через SIP, XMPP или даже учетные записи Google Talk. В *Empathy* и *Ekiga* нет функции записи звонка, но для записи звонков можно обратиться ко внешним приложениям, например, *Audacity*. В *Skype* есть сторонние инструменты, такие как *Skype Call Recorder*.

» Благодаря протоколам обнаружения сервиса типа Zeroconf, некоторые клиенты VoIP автоматически определяют других пользователей внутри LAN.

### Вердикт

**Ekiga**  
★★★★★  
**Jitsi**  
★★★★★  
**Skype**  
★★★★★  
**Empathy**  
★★★★★  
**Yate**  
★★★☆☆

» Все клиенты, кроме *Yate*, обеспечивают достойное качество VoIP и видеозвонков.

# Конференц-звонки

## Можно ли звонить нескольким абонентам?

**К**онференц-звонки — инструмент не только бизнеса. Вы можете использовать их, чтобы спланировать встречу с друзьями или обсудить домашнее задание. Кроме *Empathy*, все клиенты так или иначе предлагают опцию конференц-связи.

В *Ekiga* нет опции хостинга конференц-звонков, но если вы используете VoIP-сервис *ekiga.net*, то можете получить доступ к его конференц-серверу или присоединиться к общественному конференц-залу, позвонив на 501@ekiga.net. Или можете создать частный конференц-зал, добавив

любой четырехзначный номер к 501, например, 5012233@ekiga.net, и ваши друзья смогут набрать этот номер и присоединиться к вам.

*Skype* и *Jitsi* позволяют стать хостом конференц-звонка. Если вы и ваши собеседники используете *Skype*, к звонку можно бесплатно добавить до 25 человек. Можете также сделать конференц-звонок видео или выдать доступ на просмотр своего рабочего стола, если у вас есть подписка Skype Premium. В любой момент любой участник беседы может добавить другого участника.

*Jitsi* тоже умеет выбирать участников конференц-звонка. Одна из приятных функций конференц-звонка *Jitsi* — добавление контактов из различных сервисов и протоколов и включение их в единый разговор. Помимо этого, разработчики *Jitsi* создали специальный компонент XMPP под названием Jitsi Videobridge, который обеспечивает многопользовательские видеозвонки.



» Вы можете увидеть Jitsi Videobridge в действии, настроив видеоконференцию с помощью сервиса [meet.jit.si](https://meet.jit.si) для многопользовательских видеозвонков.

### Вердикт

**Jitsi**  
★★★★★  
**Skype**  
★★★★★  
**Yate**  
★★★★★  
**Ekiga**  
★★★★★  
**Empathy**  
★★★★★

» Jitsi выделяется в этой области благодаря функциям типа Videobridge.

# Удобство использования

Являются ли они эффективной опцией?

**К**оличество функций еще не делает клиент VoIP чемпионом. Следует обратить внимание на то, как эти функции предлагаются пользователям.

Нашпиговать функциями интерфейс звонка — не самая лучшая идея, однако абсолютно голый интерфейс, который заставляет вас постоянно

возвращаться к основному приложению для запуска каждой функции, тоже далеко не ушел.

Мы также рассмотрим интеграцию клиента с рабочими столами. Основные дистрибутивы рабочих столов потратили немало времени и трудов на свою систему уведомлений, и клиент должен ее придерживаться.

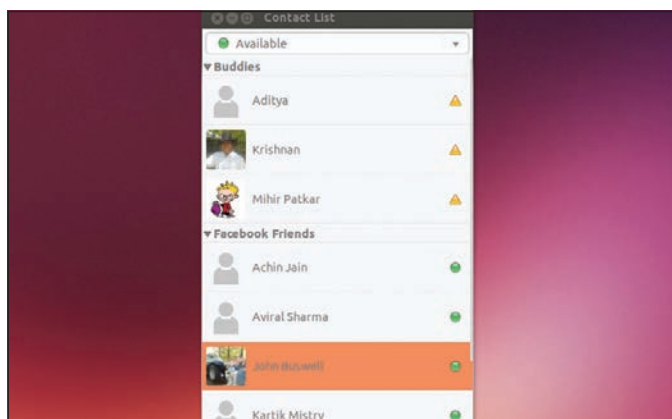
Следующий вопрос — кроссплатформенная доступность приложения. Конечно, наш журнал посвящается Linux, однако вы вряд ли захотите ограничить свой круг общения сплошными линуксоидами. Следовательно, мы бы не рекомендовали вам клиент, который недоступен на других платформах.

## Ekiga ★★★★★

В противоположность *Empathy*, *Ekiga* в первую очередь является клиентом VoIP, который работает как в Gnome, так и в KDE. Он поддерживает протоколы H.323 и SIP и ряд популярных кодеков. Клиент также размещает собственный SIP-сервис и сервис Call-Out для звонков на постоянные номера, и в нем имеется удобный набор функций входящих звонков — например, функция передачи звонка на другой SIP-номер.

*Ekiga* умеет определять других пользователей *Ekiga* в локальной LAN, что сильно упрощает ее использование в закрытых сетях, например, в офисах и школах. Однако на данный момент эта функция не поддерживается Windows-версией клиента.

Кроме того, хотя многие функции *Ekiga* с виду привлекательны, они не всегда работают как заявлено, и клиент не дотягивает по производительности до своих коллег.

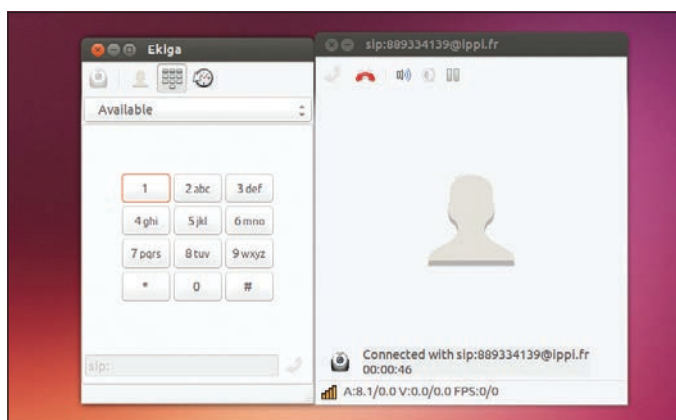


## Empathy ★★★★★

*Empathy* — это унифицированный клиент с поддержкой текстового, голосового и видеочата и передачи файлов через разные протоколы. Это клиент обмена быстрыми сообщениями по умолчанию в Ubuntu и в нескольких дистрибутивах на базе Gnome, включая Fedora.

Приложение хорошо интегрировано с этими дистрибутивами. При первом его запуске оно автоматически запустит приложение Online Account для добавления учетной записи. Большинство дистрибутивов не предоставляют *Empathy* с компонентами Telepathy для соединения с сетями VoIP, но это легко исправить, обратившись к соответствующим менеджерам пакетов.

Против приложения работает лишь его недоступность вне Linux. Кроме того, в отличие от своих собратьев, *Empathy* не поддерживает обмена сообщениями OTR, а также не делает звонков на телефоны PSTN. *Empathy* — хороший клиент IM, но не вписывается в ряд чистокровных клиентов VoIP.



# Протоколы и сервисы

В состоянии ли они заменить собой IM-клиентов?

**Б**ольшинство клиентов VoIP одно поддерживают множество популярных сервисов обмена быстрыми сообщениями. Два основных исключения — *Skype* и *Ekiga*. Если вы — пользователь сервиса *Skype*, то сможете соединиться только с другими пользователями *Skype*. Чтобы соединиться с работающим на SIP PBX, вам придется подписаться в *Skype* на Connect и открыть счет. Ну, а *Ekiga* — просто клиент VoIP. Он поддерживает исключительно протоколы VoIP — SIP и H.323.

*Yate*, помимо этого, поддерживает еще Jabber, который использует для соединения с учетными записями Google Talk, а также включает список предустановленных провайдеров, типа *ekiga.net*. Имеется опция добавления учетной записи Facebook через Jabber, но она не соединяется.

*Empathy* в первую очередь клиент IM; он поддерживает все протоколы, поддерживаемые клиентом *Pidgin*. Вы можете использовать *Ekiga* для соединения с Google Talk (Jabber/XMPP), MSN, AIM, Facebook, Yahoo! Messenger, ICQ, и прочими. Есть

также и поддержка IRC. Возможность зарегистрировать учетную запись на сервисе зависит от того, установлены ли в вашем дистрибутиве соответствующие менеджеры соединения Telepathy.

*Jitsi* тоже поддерживает ряд сервисов, в том числе Google Talk, Facebook, MSN, Yahoo! Messenger, AIM, ICQ и прочие, а также SIP и XMPP. Более того, он делает очень удобную регистрацию во многих сервисах. При первом запуске клиента он позволит вам настроить вход во многие учетные записи пользователя за один щелчок.

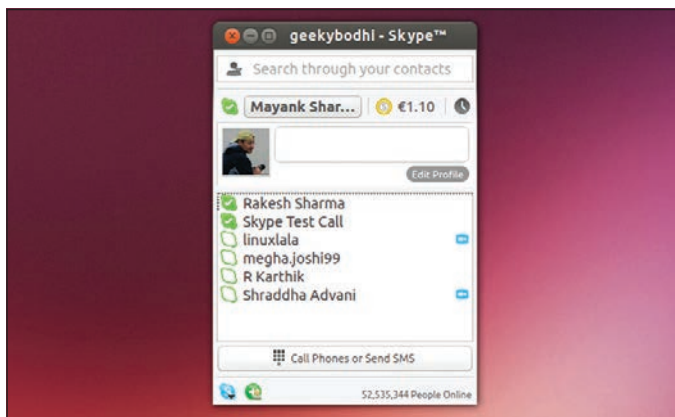
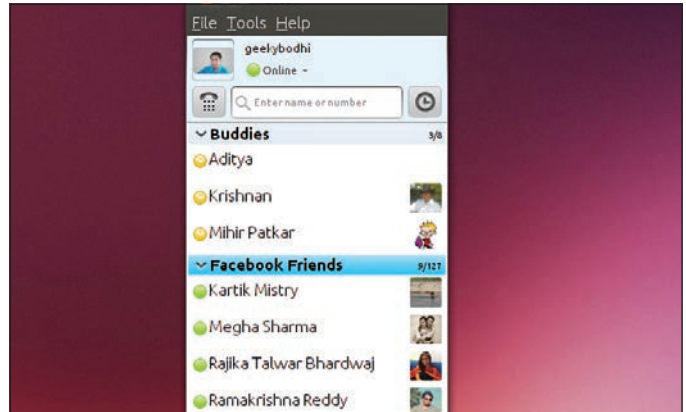
### Вердикт

<b>Empathy</b>	★★★★★
<b>Jitsi</b>	★★★★★
<b>Yate</b>	★★★★★
<b>Ekiga</b>	★★★★★
<b>Skype</b>	★★★★★

» Empathy может поддерживать протоколы VoIP, но вам придется установить Telepathy.

## Jitsi ★★★★★

*Jitsi* предлагает отличное взаимодействие с пользователем уже с момента установки. Приложение написано на Java и доступно на нескольких платформах. Это еще одна причина, по которой оно загружается не сразу; но уж когда запустится, то предложит вам все функции, которых только можно ожидать от клиента VoIP. Вы можете соединить клиент с любым сервисом VoIP, включая его собственный. В дополнение к VoIP, приложение также поддерживает ряд других популярных IM-сервисов, таких, как Facebook Chat и Google Talk. Клиент также может делать нерегистрируемые звонки SIP другим пользователям *Jitsi* по локальной сети. Приложение хорошо интегрировано с рабочим столом. Вы получаете всплывающие сообщения, которые отображают, когда вам пишут сообщение, или показывается само сообщение, если окно сообщения не открыто. Также в приложении имеются дружественные для предприятия функции, например, поддержка директорий LDAP.



## Skype ★★★★★

Если отвлечься от его проприетарной природы, *Skype* является одним из лучших приложений VoIP для установки и работы. Чтобы освоить *Skype*, не нужно никаких усилий. Вы устанавливаете приложение, регистрируете свою учетную запись *Skype* — и все. *Skype* определяет и настраивает ваше аудио- и видеооборудование. Во время наших тестов мы не столкнулись ни с одной неожиданностью в процессе звонков другим пользователям *Skype*. Клиенты работали хорошо и в одиночных, и в конференц-звонках. Подобно *Jitsi*, *Skype* тоже переполнен функциями, подобающими клиенту VoIP.

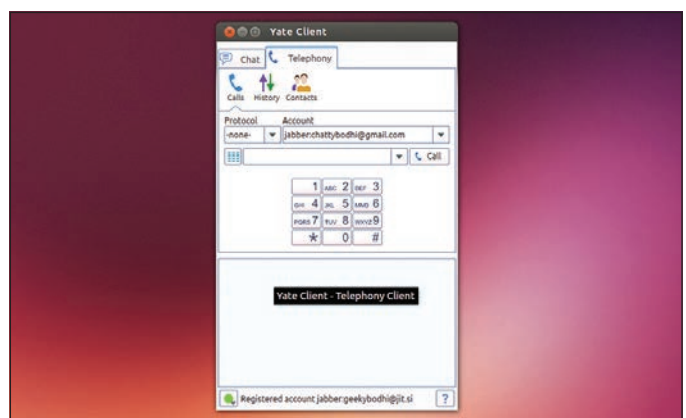
Более того, *Skype* — единственный клиент из протестированных нами, в котором есть официальные клиенты для мобильных платформ, в частности, Android. К сожалению, не весь рабочий стол работал безукоризненно для аудио- и видеозвонков Android. В некоторых был плохой звук, а некоторые вообще отказывались соединяться.

## Yate ★★★★★

Клиент *Yate* — это незатейливый старый клиент VoIP с поддержкой протоколов VoIP H.323 и SIP, а также Jabber и IAX. Основанный на Qt, клиент доступен на нескольких платформах. Он может делать аудиозвонки, но не видео. Более того, он даже не поддерживает никаких вспомогательных видеофункций, например, просмотр рабочего стола.

Но при этом клиент поддерживает относящиеся к VoIP расширенные функции, например, возможность добавлять контакты в текущий звонок, превращая его в аудиоконференцию, а также переводить активный звонок на другого пользователя.

Раздражает, что при звонке на номер SIP вам приходится набирать этот номер вручную. У вас не получится просто выбрать его из списка контактов, как это делается в других сервисах.



# Безопасность и шифрование

### Справятся ли они с любителями подслушивать?

**Е**сли вас заботят проблемы безопасности, вы не захотите использовать публичный сервис, не убедившись сначала, что ваши сообщения будут зашифрованы. В *Ekiga* и *Yate* вообще нет прямой поддержки шифрования, хотя в *Yate* вы можете заставить его использовать для шифрования сообщений TLS при добавлении учетной записи Jabber или Google Talk.

Однако золотым стандартом шифрования сообщений IM является протокол Off-The-Record (OTR), который использует комбинацию нескольких алгоритмов

безопасности для обеспечения сильного стандарта шифрования. Клиенту *Empathy* не хватает конфиденциальности и функций безопасности через OTR.

*Skype* заявляет, что использует меж абонентское шифрование, но за прошедшие годы несколько человек выражали сомнения по поводу шифрования *Skype*, в первую очередь из-за его спецификации протокола с закрытым кодом. И благодаря разоблачениям Эдварда Сноудена [Edward Snowden] мы теперь знаем, что АНБ и ФБР могут подслушивать звонки по *Skype*. Так

что, несмотря на все публичные заверения в безопасности *Skype*, доверие утрачено.

И опять же, требуемые усилия к обеспечению безопасности и конфиденциальности прилагает один *Jitsi*. Он использует протоколы SRTP и ZRTP для шифрования всех аудио- и видеозвонков SIP и XMPP. Также в нем есть механизм аутентификации контакта по отпечаткам пальцев. Клиент использует OTR для шифрования быстрых сообщений во всех протоколах и хранит подробности ваших учетных данных в зашифрованной папке с паролями.

## Вердикт

- Jitsi ★★★★★
- Skype ★★★★★
- Yate ★★★★★
- Ekiga ★★★★★
- Empathy ★★★★★

» *Skype* действительно шифрует звонки, но кто ж ему поверит после разоблачения PRISM?

# Потоковое вещание рабочего стола

Получится ли у них выступить в роли VNC-клиентов?

**М**ы проверяли, могут ли клиенты VoIP размещать конференц-звонки и передавать видео вместе с аудио. Мы также использовали их в традиционных сетях обмена быстрыми сообщениями, и некоторые из них могут выполнять функции вашего обычного клиента IM. А что еще они могут делать? Как насчет того, чтобы помочь поделиться рабочим столом и файлами?

*Empathy* поддерживает предоставленные доступы к рабочему столу через клиент VNC *Vinagre* и придерживается опций,

установленных на сервере распределенного доступа к рабочему столу *Vino*. Когда вы используете *Empathy*, чтобы выдать доступ к вашему экрану, он автоматически предоставит весь ваш экран.

Если вам нужна большая гибкость при распределении доступа к рабочему столу, выбирайте *Skype* или *Jitsi*. Оба позволяют делиться доступом к экрану, или к отдельной области, размер которой можно менять. С помощью *Jitsi* вы даже можете позволить вашему собеседнику удаленно управлять вашим рабочим столом. Помимо этого, в *Jitsi* обе стороны смогут поделиться друг с другом доступом к рабочим столам; в *Skype* это работает не так хорошо.

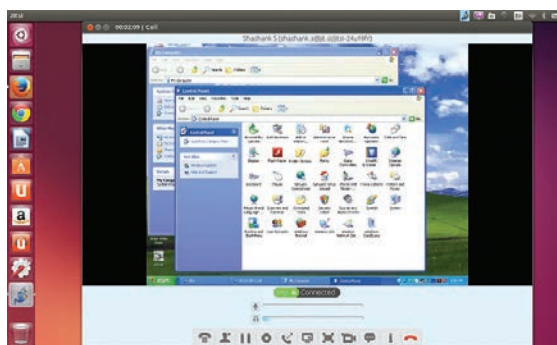
А вот что работает в *Skype* отлично, так это пересылка файлов. Фактически, это единственный клиент в нашем Сравнении, который позволяет

выделять и отправлять множество файлов одновременно. Все, кроме *Ekiga*, могут отправлять только по файлу за раз.

*Ekiga* уникально плох тем, что не позволяет ни передавать файлы, ни обеспечивать поток с удаленного рабочего стола. И ведет себя клиент *Ekiga* довольно беспорядочно. Например, он несколько раз отправил текстовое сообщение одному и тому же адресату.

*Yate* тоже не умеет делиться доступом к удаленному рабочему столу, но может передавать файлы бесперебойно, и то же касается *Empathy*. Кроме ограничения по отправке только одного файла за раз, нам не понравился тот факт, что опция передачи файлов *Yate* не слишком явная. Вы можете передавать файлы или через окно чата, или щелкнув правой кнопкой по адресату.

Все это даже сравнивать нельзя со *Skype*, который не только позволяет отправлять сразу несколько файлов, но и имеет весьма удобное диалоговое окно передачи файлов. У вас также есть опции по автоматической загрузке и даже по изменению директории скачиваний по умолчанию перед приемом файлов.



» В *Jitsi* вы можете безо всяких усилий делиться и обеспечивать удаленное управление рабочим столом.

**Вердикт**

Skype	★★★★★
Empathy	★★★★★
Jitsi	★★★★★
Yate	★★★★★
Ekiga	★★★★★

» Здесь побеждает Skype, благодаря своим превосходным функциям передачи файлов.

# Функции входящих звонков

Могут ли они заменить служебные телефоны?

**С**лужебные телефоны умеют делать намного больше, чем просто совершать или принимать звонки. Они способны перенаправлять звонки, удерживать звонящего на линии, выполнять автоматическую переадресацию звонков, и т.д.

В *Empathy* есть лишь минимум функций. Он позволяет отключить свою камеру или перевести поток локальной камеры в другой угол или в окно видеочата. Полезная функция — смена камер во время самого звонка одним щелчком, если web-камер у вас несколько. В других клиентах для переключения web-камер придется влезать в соответствующий раздел их настроек.

*Yate* имеет наименьшее число функций входящих звонков, но все они очень полезны. В клиенте имеются кнопки для переключения звонка или для приглашения к участию в разговоре других собеседников и превращения звонка в конференцию.

Когда вы выбираете любую из этих опций, *Yate* представляет вам выпадающий список людей, которым вы звонили раньше. Вы даже можете переводить звонок на другой номер SIP с помощью *Ekiga*.

Когда вы ответите на звонок с помощью *Jitsi*, вам покажут строку символов для подтверждения соединения с контактным лицом, и обеспечивается безопасность вашего видео и аудио. Очень интересна кнопка записи, позволяющая сохранить звонок в формате MP3. Также у вас есть кнопки для создания конференц-звонка или переадресации звонка, и обе они открывают диалоговое окно для выбора контактного лица.

В *Skype* тоже есть подобные опции; есть кнопки для переключения чата, web-камеры и микрофона, и возможность перевести звонок в режим ожидания. Две интересные опции — просмотр профиля контактного лица с показом местного времени и возможность делать отправку кон-



» *Empathy* предлагает минимальный набор функций во время звонка, в отличие от *Jitsi*.

тактным лицам, которая автоматически добавляет их в список получателей.

Одна из основных функций, которых нет в *Skype*, это переключение звонка, хотя можно добавлять людей к разговору. Также в клиенте Linux нет некоторых опций, имеющихся в клиенте Windows — например, возможности сделать фото, просмотреть информацию о качестве звонка и настроить некоторые параметры во время звонка.

**Вердикт**

Jitsi	★★★★★
Skype	★★★★★
Ekiga	★★★★★
Empathy	★★★★★
Yate	★★★★★

» Можете с успехом использовать *Jitsi* как PBX в MS.

## Клиенты VoIP

## Вердикт

Если вы использовали клиент VoIP раньше, то велика вероятность, что это был *Skype*. Этот клиент — сильный игрок на арене видеоконференций. Но несмотря на тот факт, что клиент работает на нескольких платформах и предлагает хороший набор функций бесплатно, ему не удалось впечатлить основных пользователей открытого кода из-за своих проприетарных кодеков и закрытого кода: это вызвало не слишком хороший резонанс в сообществе. Да и недавние разоблачения в СМИ по поводу правительственной слежки тоже не украсили его имидж.

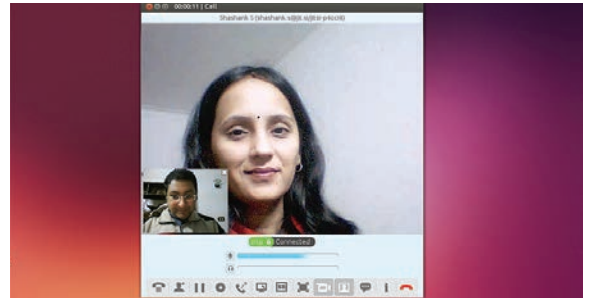
Задолго до приобретения *Skype* Microsoft и скандала с PRISM сообщество открытого кода активно искало альтернативы, и он был в списке высокоприоритетных проектов, нуждающихся в свободной альтернативе от Free Software Foundation.

Долгие годы эту мантию носил *Ekiga*, и хотя в *Ekiga* весьма впечатляющий список функций, все это сводится к нулю, потому что во время звонков клиент выдает

множество ошибок. Эта проблема усугубилась, когда мы говорили с теми контактными лицами, которые использовали *Ekiga* на разных платформах.

Клиенту *Empathy*, который заменил *Ekiga* на ниве обмена быстрыми сообщениями в большинстве самых популярных дистрибутивов рабочего стола, по-прежнему недостает функций, чтобы соперничать со *Skype*. В первую очередь это приложение обмена быстрыми сообщениями, которое также может выступать в роли клиента VoIP. Помимо отсутствующих функций, настройка *Empathy* как клиента VoIP требует некоторого времени на возню с менеджером пакетов вашего дистрибутива.

Если у вас есть время, вы, возможно, захотите взглянуть на *YateClient*, который лучше всего работает, если его использовать с сервером *Yate*. Это, однако, не отменяет того присоробного факта, что *Yate* умеет делать только аудиозвонки.



Однако у этой истории счастливый конец, и имя ему *Jitsi*. VoIP-клиент *Jitsi* является лучшей альтернативой *Skype* с открытым кодом, соответствуя ему в плане функций, а по некоторым даже превосходя его. Вы можете использовать его, чтобы поделиться аудио, видео или доступом к своему рабочему столу с одним или несколькими контактными лицами. Клиент также неплохо ладит с другими сервисами, так что вы можете перенести свои контакты из Google и *Skype* на одну страницу. В некоторых Сравнениях бывает трудно принимать решение. Но не в этом: получайте *Jitsi*.

» *Jitsi* использует стандартные протоколы для шифрования чата, голосовых и видео-звонков.

«VoIP-клиент *Jitsi* является лучшей альтернативой *Skype* с открытым кодом.»

## I Jitsi ★★★★★

Сайт: [www.jitsi.org](http://www.jitsi.org) Лицензия: LGPL Версия: 2.2  
» Ответ открытого кода *Skype*.

## IV Ekiga ★★☆☆☆

Сайт: [www.ekiga.org](http://www.ekiga.org) Лицензия: GNU GPL Версия: 4.0.1  
» Увы, славные дни этого клиента остались далеко в прошлом.

## II Skype ★★★★★☆

Сайт: [www.skype.com](http://www.skype.com) Лицензия: Freemium Версия: 4.2.0.11  
» По количеству функций это по-прежнему один из лучших клиентов VoIP.

## V Yate ★☆☆☆☆

Сайт: <http://yateclient.yate.ro> Лицензия: GNU GPL Версия: Yate-q14  
» Воистину Yet Another Telephony Engine [Еще Один Движок Телефонии] — эх, если бы он еще работал с видео...

## III Empathy ★★★★★☆

Сайт: <http://live.gnome.org/Empathy> Лицензия: GNU GPL Версия: 3.10  
» Если вы уживетесь с его ограничениями, то он уже в вашем дистрибутиве.

## Обратная связь

Вы согласны с нашим выбором? Может быть, вы выбрали бы другой клиент VoIP? Присылайте нам свои мнения по адресу: [ixf.letters@futurenet.co.uk](mailto:ixf.letters@futurenet.co.uk).

## Рассмотрите также...

Если вы ищете решение VoIP общего назначения, то нет ничего лучше победителя нашего Сравнения, *Jitsi*.

Мы хотели включить в наше Сравнение клиент *Blink*, но он сам разрушил все свои шансы, отказавшись запускаться на наших тестовых машинах. Мы также обдумывали вариант включить *Mumble*, но это приложение было создано с конкретной

целью — для геймеров и геймерских команд, чтобы они могли общаться во время игры; а большинству пользователей это не подходит. Однако это идеальная замена для геймеров, вынужденных за игрой переговариваться через проприетарные приложения вроде *TeamSpeak*.

Есть еще *LinPhone* — именно его мы используем на наших телефонах Android для соединения

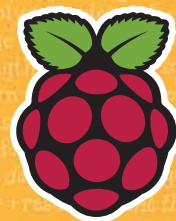
с клиентами рабочего стола VoIP. У *LinPhone* есть также очень способная версия рабочего стола.

Если вы не против проприетарных программ, то у вышеупомянутого приложения *TeamSpeak* есть и Linux-клиент (при желании можно настроить свой приватный TS-сервер). Можете также использовать его с основанным на браузере Google+ *Hangouts*, установив его плагин для Linux. **LXF**

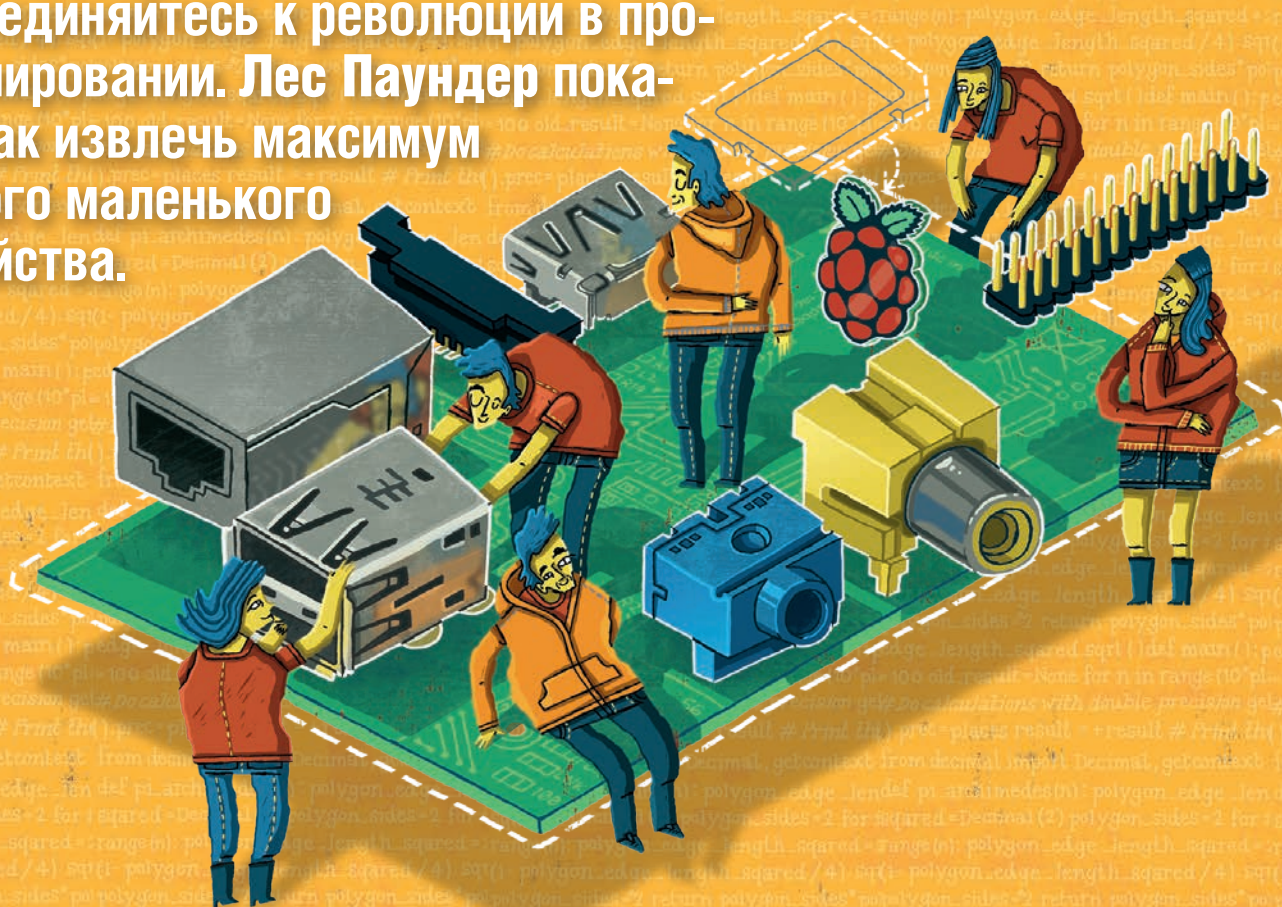
Первые шаги с Raspberry Pi

# Освойте чудесный

# Raspberry Pi



Присоединяйтесь к революции в программировании. Лес Паундер покажет, как извлечь максимум из этого маленького устройства.



**П**рошло года два с тех пор, как Raspberry Pi появилась на сцене, и за это время компьютерный ландшафт сильно изменился. Это миниатюрное устройство было разработано фондом Raspberry Pi Foundation, в тревоге об уменьшении числа студентов, выбирающих для изучения в колледжах и университетах компьютерные дисциплины.

Теперь, два года спустя, на этих дешевых компьютерах учатся программировать шестилетние дети, и становится очевидно, что на подходе новое поколение детей, которые хорошо смыслят в программировании и с раннего детства прекрасно ориентируются в цифровом мире.

До появления Raspberry Pi на школьных уроках по информатике обычно изучали программы

Microsoft Office; отдельные дети добивались до веб-разработки. Но сейчас наступил переходный период, когда учителя обращаются к открытым сообществам за помощью, чтобы научиться действовать и думать, как программисты. И мы видим это в появляющихся проек-

**«На этих дешевых компьютерах учатся программировать шестилетние дети.»**

тах, например, в околосмических миссиях Дейва Акермана [Dave Akerman] ([www.daveakerman.com](http://www.daveakerman.com)), роботах Саймона Уолтерса [Simon Walters] на основе Scratch (<http://cymplecy.wordpress.com>) и экспертных комиссиях, таких как OCR [Oxford,

Cambridge, and RSA Examinations, Британский экзаменационный совет], включающих программирование в свои учебные программы.

Удивительно думать, что сейчас мы на пороге обширных, радикальных изменений в образовательной системе Великобритании и появления поколения программистов, готовых овладеть Интернетом.

Raspberry Pi — отличная забава для людей всех возрастов, которая сделала вычисления и электронику доступной и веселой благодаря своей гибкости и низкой стоимости владения. В этой статье мы расскажем о настройке Raspberry Pi, а попутно вы получите новые знания в программировании и аппаратуре. Это создаст прочную основу для ваших будущих проектов — итак, начнем!

# Начинаем

Установим на Pi Raspbian, оптимизированную ОС на базе Debian.

**R**aspberry Pi работает под управлением процессора ARM частотой 700 МГц, и хотя не обладает большой вычислительной мощностью, но прекрасно справляется со своей задачей — обучением программированию и экспериментам с устройствами. У Pi также есть 512 МБ оперативной памяти — более чем достаточно для рекомендуемой операционной системы, Raspbian.

Raspbian — это облегченная версия Debian для процессоров ARM. Большинство пользователей и программистов берут эту ОС за основу для своих проектов. Это позволило улучшить систему и уровень поддержки внешних устройств настолько, что теперь Raspbian — полноценная и надежная система. Впрочем, есть и другие ОС для Raspberry Pi (см. «А если не Raspbian?», стр. 52).

## Установка Raspbian

Итак, у вас есть Raspberry Pi и все необходимые внешние устройства, но теперь вам нужно загрузить и установить на него операционную систему. Сейчас это стало довольно рутинной задачей, но когда Raspberry Pi только появился, приходилось копировать загруженный образ на SD-карту командой `dd`, а такое было уж точно не для новичка.

В ответ Raspberry Pi Foundation выпустила *NOOBS* (New Out of the Box Software — новое ПО из коробки), которое представляет собой загружаемый ZIP-архив с несколькими операционными системами (Pidora, Raspbian и два варианта XMBC), каждую

## «Большинство пользователей и программистов берут Raspbian за основу для проектов.»

из которых можно записать на SD-карту Raspberry Pi. Для установки на Pi Linux понадобится SD-карта объемом не менее 4 ГБ, отформатированная в FAT32. Для загрузки ZIP-архива *NOOBS* перейдите по ссылке [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads). Это файл объемом 1,3 ГБ, который загружается минут за 20 — успеете выпить чашечку чая.

Загрузив архив, вставьте в компьютер SD-карту и распакуйте на нее содержимое ZIP-архива. После этого извлеките карту



и вставьте ее в Raspberry Pi. Убедитесь, что все подключено правильно, и включите питание. При первой загрузке *NOOBS* автоматически загружает меню со списком операционных систем.

В этой статье мы сосредоточимся на Raspbian, поэтому выберите его и нажмите Install OS [Установить ОС] на панели меню. После установки Raspbian на SD-карту, вставленную в Pi, *NOOBS* попросит вас перезагрузиться — сделайте это и дайте Pi полностью загрузиться.

После перезагрузки вас попросят ввести логин и пароль. Имя пользователя по умолчанию — `pi`, пароль — `raspberry`. Войдя в систему, наберите в командной строке `raspi-config`, и откроется утилита для обновления и настройки Pi. Первое, что нужно сделать — обновить `raspi-config`. Для этого понадобится работающее интернет-соединение. Затем нужно изменить загрузку Raspberry Pi так, чтобы загружался рабочий стол.

По завершении обновления выйдите из `raspi-config` и наберите `reboot`, чтобы перезагрузить Pi и загрузить сделанные вами изменения. После успешной перезагрузки сразу откроется рабочий стол *LXDE*.

В интерфейс *NOOBS* можно вернуться в любое время после установки ОС. Для этого удерживайте клавишу Shift при загрузке. Так вы сможете попробовать другие операционные системы или перезаписать поврежденную карту свежее установленным Raspbian.

» RPi удобно использовать с макетами для сборки электрических схем без пайки.

»

## Установка программ на Raspberry Pi

Raspberry Pi — типичный компьютер с ОС на базе Debian. Это означает, что устанавливать пакеты можно напрямую через терминал (в меню Пуск) командой `apt`.

`Apt` — это сокращение от “Advanced Packaging Tool [продвинутая утилита управления пакетами]”. Эта утилита позволяет устанавливать, удалять и отслеживать все программы. Для доступа к ее возможностям нужны права `root`, которые легко получить на время, написав перед командой `sudo`.

Перед установкой новой программы стоит обновить список установленных программ, для чего набрать `sudo apt-get update`. Эта команда сравнит список ваших программ с тем, что доступно на серверах, и обновит список.

Обновив существующие программы, для установки новой наберите `apt-get install <имяпакета>`, заменив `<имяпакета>` названием устанавливаемой программы. Для удаления программы наберите `apt-get remove <имяпакета>`, указав название

удаляемой программы. `Apt` — мощное средство, поэтому пользуйтесь им осторожно; а если есть сомнения, спросите друга.

Другой, более простой метод установки программ — воспользоваться специализированным Pi Store, но для этого потребуются загрузить клиент — <http://store.raspberrypi.com/download>. Все программы открытые, написаны тысячами самоотверженных программистов, и обновляются автоматически.

## Развивимся с GPIO

Управляйте датчиками и сенсорами с помощью Python через GPIO.

**Л**учшее, что есть в Raspberry Pi — невинный с виду набор из 26 медных контактов. Эти контакты, называемые входами/выходами общего назначения (General Purpose Input Output, или сокращенно GPIO) позволяют воспользоваться новыми методами ввода и вывода данных, такими как датчики и светодиоды. Любой из компонентов, которыми вы пользуетесь на Arduino, можно подключить и к Raspberry Pi, позволив создать дешевые и функциональные проекты. Но чтобы оживить проекты, нужно написать программы, и для Raspberry Pi это делается на Python.

Python — невероятно гибкий язык, используемый везде: от небольших устройств вроде Raspberry Pi до дата-центров Google. Доступны две версии Python: 2.7 и 3.3. Версия 2.7 считается стабильной, но более старой, с прекрасной совместимостью, а 3.3 — новинка с более обширным набором функций, но на данный момент не вполне совместима с частью библиотек Python. В этой статье мы сосредоточимся на 2.7, но нашим кодом при желании легко воспользоваться в версии 3.3.

Пользоваться Python с Raspberry Pi очень просто, так как в Raspbian он уже установлен. Мы будем пользоваться интегрированной средой разработки Python под названием IDLE (см. рис. 1).

### Программирование на Python

По окончании загрузки IDLE откроется оболочка Python, где можно быстро проверить код в интерактивной среде. Понять, что вы в оболочке, можно не только по заголовку окна, но и по трем символам «больше» (>>>) в окне оболочки.

Но нам нужен редактор, в котором можно писать много кода. Чтобы его открыть, просто выберите в левом верхнем углу File > New Window [Файл > Создать].

Лучший способ изучения Python — перейти сразу к делу и написать какой-нибудь код. Наберите следующий код в окне редактора, сохраните его, затем выберите Run > Run Module [Запустить > Запустить модуль] и посмотрите на результаты. Это будут числа от 0 до 9, по одному на строку:

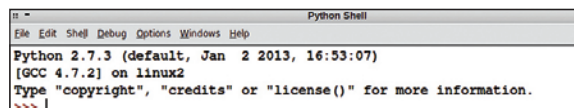


Рис. 1. IDLE — Interactive Development Environment (Интерактивная среда разработки) — оболочка Python.



Гибкость — одно из важнейших достоинств Raspberry Pi. Pi поддерживает I<sup>2</sup>C (для подключения низкоскоростных периферийных устройств) и SPI (что позволяет обеспечить обмен данными «ведущий/ведомый»), а также обычную функциональность GPIO — и все это под управлением Python.

```
a = 0
while a < 10:
    print a
    a = a + 1
```

В первой строке — переменная, которую мы назвали **a**. Переменная — это место, в котором можно временно хранить какие-то данные — нечто вроде пустой коробки с надписью, где можно хранить вещи. Вторая строка — начало цикла, который в данном случае является управляемым, с условной логикой. Это означает, что пока наша переменная **a** меньше 10, будет выполняться код, указанный ниже. Третья строка кода — это функция печати, которая выводит на экран значение переменной **a**. Третья строка автоматически форматируется отступом, чтобы показать, что это код, который будет выполняться, если условие во второй строке верно. Как правило, в качестве отступа в Python используются четыре пробела. В последней строке наша переменная увеличивается на 1 с каждым проходом цикла.

Так в этом маленьком фрагменте кода мы продемонстрировали несколько понятий из области программирования:

» **Последовательность Шаги**, необходимые для выполнения задачи.

## Безопасная работа с GPIO

Входы/выходы общего назначения — чувствительные компоненты, и перед их использованием нужно усвоить несколько базовых принципов техники безопасности.

Номера контактов относятся к их расположению на плате: нечетные — в левом столбце, четные — в правом (см. рис. 2).

Контакты 1 и 19 — выходы с напряжением 3,3 В, это означает, что при подключении светодиодов или других чувствительных компонентов лучше

последовательно включить резистор для защиты компонента.

Контакты 2 и 4 — выходы с напряжением 5 В, и на них распространяется то же правило с резистором; но также старайтесь не использовать все 5 В на этих контактах, иначе Raspberry Pi сразу начнет работать нестабильно и может выйти из строя.

Максимальный безопасный ток для входов, который мы бы посоветовали — 500 мА. Не превышайте это значение, не то поджарите процессор. Макси-

мальный выходной ток — от 250 мА (миллиампер) до 500 мА (0,5 А); если его превысит, Raspberry Pi может автоматически отключиться.

Подключите кабели разъемов (разъем «мама» к Pi и разъем «папа» к печатной плате) можно при отключенном питании, но убедитесь, что не ошиблись с номерами контактов и не устроили короткого замыкания, соединив контакты напрямую.

Помня об этих правилах, не забывайте получать удовольствие от забав с GPIO!



» **Переменные** Хранилище данных.

» **Условия** Встроенная в Python логика, поддерживающая логические и математические операции.

» **Циклы** Способы повторить действия несколько раз.

Благодаря своей простоте и удобочитаемости Python — превосходный язык для изучения. Мы уже видели немного кода на Python, теперь двинемся дальше и посмотрим, как с помощью Python можно управлять входами/выходами общего назначения.

## Управление GPIO

Для работы с GPIO в Python нужно установить несколько дополнительных программ, поэтому откройте терминал, выбрав пункт меню **Menu > Accessories > LXTerminal** [Меню > Стандартные > LXTerminal], и наберите

```
sudo apt-get update
sudo apt-get install python-dev
sudo apt-get install python-rpi.gpio
```

Установив эти дополнения, мы сможем с помощью Python управлять объектами материального мира. Посмотрим на код, с помощью которого мы зажигаем светодиод, подключенный к Raspberry Pi.

```
import RPi.GPIO as GPIO
from time import *
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
led_pin = 8
GPIO.setup(led_pin, GPIO.OUT)
while True:
    GPIO.output(led_pin, True)
    sleep(0.5)
    GPIO.output(led_pin, False)
    sleep(0.5)
```

В начале этого файла мы видим две операции импорта. «Что это?» — спрашиваете вы. За долгие годы было создано множество библиотек и модулей Python, и чтобы воспользоваться ими в нашем коде, их нужно импортировать. Базовые библиотеки, например, с функциями работы с числами и временем, уже установлены и доступны, но модуль `RPi.GPIO` к ним не относится. Импортировав этот модуль, мы получаем функции для управления стандартными входами/выходами общего назначения с Pi. После импорта нужно настроить GPIO для использования в нашем коде, и для этого мы делаем следующее:

```
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
```

`GPIO.setwarnings(False)` отключает все сообщения об ошибках, и программа выводит меньше сообщений на экран. `GPIO.setmode(GPIO.BOARD)` изменяет схему нумерации GPIO на более логичную, при которой контакт 1 — это контакт, ближайший к разъему SD-карты, а контакт 2 находится справа от первого. Все нечетные номера находятся в левом столбце, а четные — в правом. Потом мы видим переменную, в которой хранится номер контакта, используемого для управления светодиодом.

```
led_pin = 8
```

## Светодиоды под управлением Python

Записав номер контакта в переменную, а не жестко закодировав его, мы можем быстро менять свой код для управления другими контактами. Следующая строка кода управляет самим контактом и переводит его в режим входа или выхода. Как видите, мы объявляем переменную `led_pin` и затем говорим, что это выход.

```
GPIO.setup(led_pin, GPIO.OUT)
```

Следующей строкой кода начинается бесконечный цикл, или, в терминологии Python, цикл `while True`. Булевская логика говорит

3.3V	1	2	5V
I2C1 SDA	3	4	5V
I2C1 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
GROUND	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GROUND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
SP10 MOSI	19	20	GROUND
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
GROUND	25	26	SP10 CE1 N

» Рис. 2. Расположение контактов GPIO. Импортировав модуль RPi.GPIO, можно изменить схему нумерации контактов на более логичную для более удобного управления светодиодами.

нам, что программа должна выполняться всегда, так как условие установлено в `True` (истинно). Код, выполняемый в этом цикле, включает и выключает светодиод на полсекунды. Для управления

## «Установив дополнения, мы сможем с помощью Python управлять объектами.»

длительностью мы используем функцию `sleep` из библиотеки, импортированной в начале программы. Эта функция фактически означает «подождать», поэтому мы подаем питание на контакт вызовом `GPIO.output(led_pin, True)` и ждем полсекунды. Затем мы на полсекунды отключаем питание вызовом `GPIO.output(led_pin, False)`, создавая эффект мигания.



# Ваш аппаратный проект-дебют

## Превратите Pi в маленький игровой автомат.

**Ц**ель Raspberry Pi в том, чтобы научить детей программировать и познакомить их с концепциями программирования, преодолевая резкое снижение числа студентов, выбирающих для изучения в университетах компьютерные дисциплины. Заниматься с Raspberry Pi очень весело, и это помогает в процессе обучения. Для первого проекта мы воспользуемся кодом, рассмотренным в предыдущей части (см. стр. 39), так как это исключительно плодотворный проект для студентов с любым уровнем знаний.

Мы подробно разобрали код в предыдущей части, а теперь сосредоточимся на аппаратуре. Наша схема относительно проста, но невероятно плодотворна, как всегда, когда делаешь первый шаг в восхитительный мир электроники. Наша задача — зажигать светодиод каждые полсекунды, и для этого нужно подключить светодиод к Raspberry Pi (как показано на рис. 2, стр. 39).

На диске к журналу и на сайте LXF также можно найти копию этого кода, который мы советуем скопировать и вставить в Raspberry Pi, а затем запустить с правами root. Код для управления GPIO можно запускать только с правами root, полученными с помощью команды `sudo`.

Разберем аппаратную часть проекта. Подключим к макету контакт 8 и последовательно включим резистор сопротивлением 220 Ом, который снижает ток от Raspberry Pi к нашему светодиоду. Резистор подключается к длинному контакту светодиода, это «плюс» (анод). Половина нашей схемы готова, но теперь нужно соединить контакт 4 (Земля, GND) с коротким контактом светодиода — это «минус» (катод). Получится замкнутая цепь.

В терминале перейдите в каталог с кодом (для перехода в каталог пользуйтесь командой `cd`, а для просмотра содержимого текущего каталога — `ls`) и наберите:

```
sudo idle flash.py
```

Теперь появляются два варианта поведения светодиода. Мы видим, как светодиод загорается и затем гаснет, а значит, первая стадия нашего проекта закончена. Bravo! Вы вступили на путь освоения Raspberry Pi.

В следующем разделе мы познакомимся еще с одним устройством ввода для нашей игры — микропереключателем. С его помощью мы будем имитировать нажатие кнопки настоящего игрового автомата. После нажатия кнопки код выберет три разных фрукта и отобразит их названия на экране. Если все они совпадут, появится слово «Джекпот», и светодиод начнет мигать. Посмотрим на код, который разобьем на части, чтобы было проще на него ссылаться.

```
from random import *
from time import *
import RPi.GPIO as GPIO
button_pin = 8
led_pin = 12
GPIO.setmode(GPIO.BOARD)
GPIO.setup(button_pin, GPIO.IN)
GPIO.setup(led_pin, GPIO.OUT)
```

Прежде всего нужно импортировать модуль `random` из библиотеки Python (как мы уже сделали с функциями времени и GPIO). Мы воспользуемся модулем `random` для получения трех случайных предметов; модулем `time` управляет скорость игры; модуль GPIO — для работы с контактами GPIO.

## Игровой автомат

Затем объявляются две переменные: `button_pin` и `led_pin`. Эти переменные позволяют хранить номер контакта GPIO — так удобнее. Если нам вдруг захочется его изменить, проще изменить переменную в одном месте, чем конкретное значение в нескольких строках кода.

Эта часть кода заканчивается тремя строками, которые задают настройки GPIO. `GPIO.setmode(GPIO.BOARD)` сообщает Python, что мы используем базовую схему нумерации контактов, с нечетными в левом столбце и четными в правом. Функция `GPIO.setup()` позволяет управлять ролью, в какой может выступать контакт: для `button_pin` это «вход», а для `led_pin` — «выход».

```
reel1 = ["cherry", "lemon", "bell"]
```

Здесь у нас есть список; иногда его называют массивом. Список может хранить несколько значений, и в нашем коде их три — `reel1`, `reel2` и `reel3`. Они содержат выпавшие значения на трех барабанах игрового автомата.

```
while True:
```

Это начало цикла, который будет выполняться всегда или пока мы не нажмем Ctrl+C.

```
print("Нажмите кнопку для начала игры")
```

```
sleep(1)
```

Теперь у нас выводится первоначальное сообщение с инструкциями для игрока, затем после ожидания в течение одной секунды пользователь может нажать кнопку.

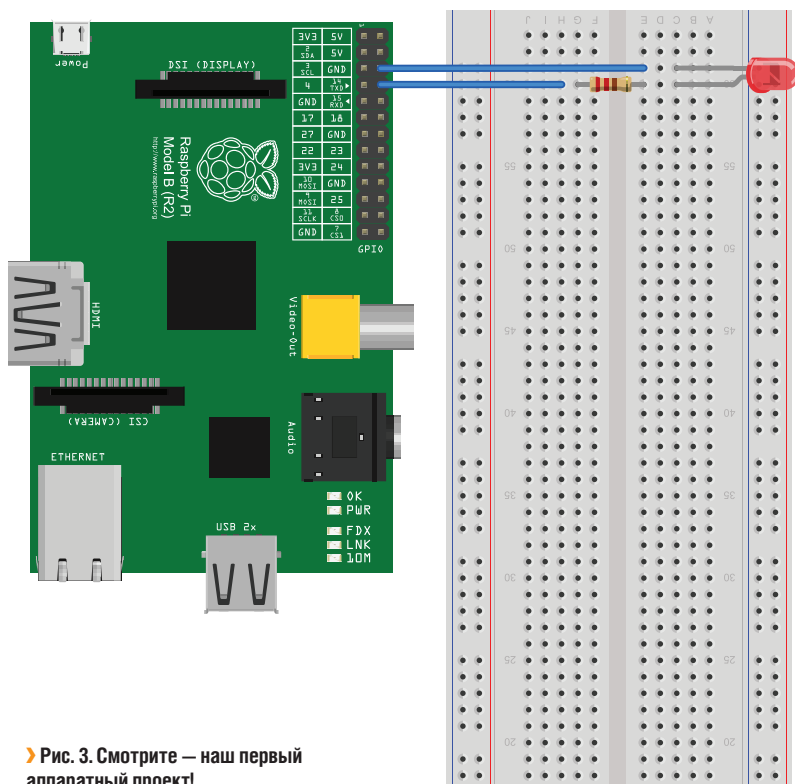


Рис. 3. Смотрите — наш первый аппаратный проект!

```

if GPIO.input(button_pin)==1:
    for i in range(0,3):
        r1 = choice(reel1)
        r2 = choice(reel2)
        r3 = choice(reel3)

```

Теперь перейдем к логике нашей игры. Прежде всего, мы начинаем цикл с условием, которым является нажатие кнопки. Если условие верно, мы с помощью цикла for выбираем три фрукта для наших барабанов и сохраняем их в три переменных: **r1**, **r2** и **r3**.

```

print ("++Играем++")
print (" ")
print (r1) + " + " + (r2) + " + " + (r3)
print (" ")
sleep(2)

```

Эта часть кода с помощью операторов print управляет тем, что мы видим на экране. Обратите особое внимание на третий оператор print, так как в нем используется объединение строк. Символ '+' можно представить себе в виде клея, который скрепляет три переменные вместе, добавляя между ними пробелы:

```

if (r1) == (r2) and (r1) == (r3):
    print ("=====")
    GPIO.output(led_pin, True)
    sleep(0.5)
    GPIO.output(led_pin, False)
    sleep(0.5)
    print ("Jackpot")
    GPIO.output(led_pin, True)
    sleep(0.5)
    GPIO.output(led_pin, False)
    sleep(0.5)

```

После этого идет еще один цикл с условием, в котором каждый фрукт сравнивается с другими на наличие совпадений. Если все три совпадают, на экран выводится надпись 'Jackpot' и светодиоды начинают мигать, чтобы результат для игрока выглядел эффектнее.

## Программирование логики

Здесь мы сравниваем первую переменную со второй и с третьей; позже мы повторяем эту логику с оператором **elif**, который по сути означает «если первое условие неверно, проверить условие elif». Таких операторов может быть сколько угодно, и в нашем коде их еще два: они сравнивают значение второй переменной с первой и с третьей, и, наконец, мы сравниваем третью переменную с первой и со второй. В коде, относящемся к условиям **if...elif**,

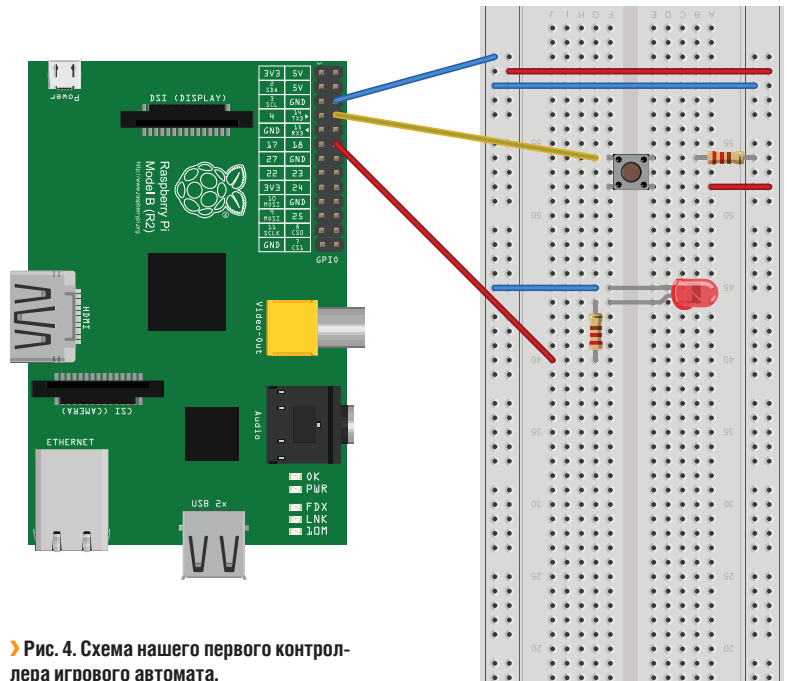


Рис. 4. Схема нашего первого контроллера игрового автомата.

нужно повторить тот же самый код, который выводит на экран 'Jackpot' и зажигает светодиод.

Итак, мы загрузили код в Raspberry Pi и собрали схему; теперь запустим код и поиграем в наш маленький игровой автомат.

## «Забавляться с Pi очень весело, и это помогает в обучении.»

Лучший способ запустить код — открыть IDLE с командой **sudo**, чтобы мы могли работать с GPIO, поэтому в терминале откройте каталог, где сохранили файл **fruitGPIO.py**, и наберите:

```
sudo idle fruitGPIO.py
```

На экране должно появиться сообщение, приглашающее вас нажать кнопку — нажмите и удерживайте кнопку в течение 1–2 секунд, и игра начнется.

Конечно, это только начало. Добро пожаловать в мир экспериментов, опытов и непрерывного обучения!

## Устройства ввода и вывода

Большинство из нас, наверное, привыкло думать об устройствах ввода и вывода в традиционном компьютерном смысле, когда используется какое-то устройство ввода, например, клавиатура или мышь, а вывод отправляется на монитор или принтер. Но в электронике диапазон устройств гораздо больше:

» **Устройства ввода** Это может быть простой переключатель, замыкающий цепь, или сложный внешний датчик, собирающий данные и отправляющий их для хранения и дальнейшей обработки. Существует множество разных типов датчиков: одни отслеживают температуру и влажность, другие умеют определять перемещение и измерять

расстояние с помощью инфракрасных или ультразвуковых импульсов.

» **Устройства вывода** В электронных компонентах выход — это то, что предоставляет пользователю какую-то обратную связь. В проектах для новичков традиционно используются светодиоды, но вместо них также можно использовать зуммеры, гудки и сервоприводы.

Устройствами вывода могут управлять устройства ввода: например, робот, решающий задачу по выходу из лабиринта, может определять положение стен с помощью ультразвукового датчика, который передает команды двигателям робота, чтобы он не наткнулся на стены и двигался по лабиринту.

Стоимость компонентов может варьироваться от нескольких пенсов до многих фунтов, но когда вы делаете первые шаги, разумно купить несколько дешевых компонентов для знакомства с миром электроники. Следующий набор обойдется не дороже £10:

- » макетная плата;
- » 10 светодиодов;
- » 20 резисторов — для простых проектов достаточно 220 Ом;
- » кабели «мама-папа» для соединения GPIO с платой;
- » кабели «папа-папа» для соединения компонентов на плате.

## Совершенствуем свой Pi

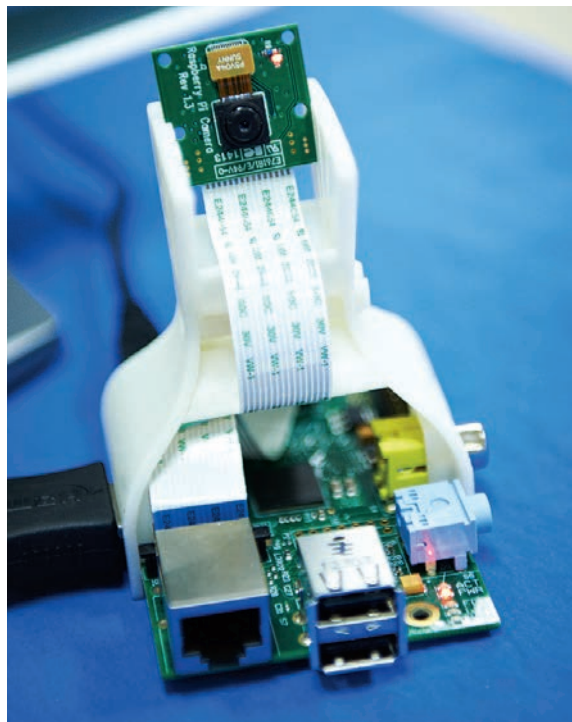
Небо — не предел, если у вас есть Pi, говорит Дейв Акерман.

**М**ы обсудили несколько классных идей и проектов от настройки Pi до создания собственного игрового автомата на Pi. Пора перейти к серьезным проектам и рассмотреть сложные элементы управления, утилиты и проекты, которые направили дух изобретательства и экспериментов прямо к границам космоса.

*BerryIO* — умный проект, который поможет добавить в Pi новую функциональность. Это набор элементов управления, которые можно контролировать удаленно с ноутбука или смартфона и которые позволяют работать со входами/выходами общего назначения и камерой Pi, сидя на диване или, если предпочитаете, через Интернет. *BerryIO* устанавливается в Raspbian, и команда разработчиков подготовила установочный скрипт, который можно загрузить с их страницы на Github: <http://bit.ly/1j5DN6Y>.

Это web-приложение, написанное преимущественно на PHP, и это косвенно означает, что *BerryIO* установит стек LAMP, по сути превратив ваш Pi в маломощный web-сервер.

*BerryIO* имеет прекрасный набор возможностей; самые интересные — возможность настройки камеры Pi и просмотра изображения в live-режиме (что уже достойно отдельного проекта). Набор возможностей регулярно обновляется — недавно были



➤ Сочетание камеры Pi с *BerryIO* позволит вам наблюдать за тем, что происходит в доме, когда вас нет.

### «Мишка по кличке Бэббидж улетел в космос — и прихватил с собой Raspberry Pi.»

добавлены возможности управления определенными моделями подключаемых плат расширения с ЖК-экраном и более гибкого управления GPIO (и это прекрасная возможность, которая просится в наш следующий проект — мы хотим сделать дирижабль с удаленным управлением, чтобы он передавал картинку с установленной камеры в прямом эфире).

Начать пользоваться *BerryIO* довольно просто:

```
wget -N https://raw.github.com/NeonHorizon/berryio/master/scripts/berryio_install.sh
chmod +x berryio_install.sh
sudo ./berryio_install.sh
berryio help
```

Теперь заходите в Pi через web-интерфейс, используя стандартные имя пользователя и пароль для терминала.

### Google Coder

Обучение программированию — прекрасный опыт, и обычным занятием сегодня считается web-программирование. Несколько сотрудников Google увидели эту связь и решили что-то сделать — так появился Google Coder (<http://bit.ly/1dNkgsh>). Это бесплатная «песочница» для web-разработки, предоставляющая каждому стабильную среду для написания отличных web-приложений на HTML 5, CSS 3 и JavaScript. Эти приложения компилируются и запускаются на Pi, а в качестве web-сервера служит Node.js.

В основе Node.js — среда выполнения JavaScript в Chrome. Сейчас Node.js считается будущим серверных масштабируемых приложений.

Среда Google Coder разбита на части для каждого языка, что позволяет легко ориентироваться в своем проекте и просматривать результаты выполнения кода. У этой среды при использовании как она есть имеется несколько ограничений: на ней нельзя развернуть публичный web-сервер, так как на Pi нет серьезных политик безопасности; кроме того, в каждый момент времени с Google Coder может работать только один пользователь, и чтобы реализовать этот проект в классе, всем ученикам придется раздать по Pi — но Google Coder бесплатен, и это прекрасный способ обучения web-программированию.

### WebIOPi

Цель жизни *WebIOPi* — предоставить вам простой интерфейс для управления GPIO Raspberry Pi. Он не так богат возможностями, как *BerryIO*, зато прост. Его web-интерфейс представляет собой схему контактов GPIO. На ней можно изменить состояние любого контакта, его роль (вход/выход), а также включить и выключить контакт.

Загрузить *WebIOPi* можно с сайта <http://bit.ly/1iOCCOq>.

Установочный скрипт автоматически загрузит и установит все необходимые дополнения с помощью *apt-get*. Если у вас не Raspbian, может потребоваться установить *GCC* и Python вручную. Для запуска *WebIOPi* воспользуйтесь командой

```
sudo webiopi -d -c /etc/webiopi/config
```

Если вы подключаетесь к Raspberry Pi напрямую, используйте адрес <http://localhost:8000>; в противном случае, если Pi в локальной сети, замените localhost IP-адресом Pi.

## Raspberry Jams

Raspberry Jams — мероприятия, на которых энтузиасты Pi встречаются друг с другом и делятся проектами в дружелюбной атмосфере. Их 18 месяцев назад придумал Алан О'Донохоу [Alan O'Donoghue]. Сначала было всего несколько встреч в Великобритании, но в конце 2013 года прошло множество Raspberry Jams по всему миру, включая такие отдаленные места, как Новая Зеландия.

В общих чертах, Raspberry Jams — это информационные собрания, организаторы которых представляют докладчикам оборудование, необходимое для демонстрации проектов. Некоторые встречи более формальны и имеют четкое расписание докладчиков, но обычно расписание составляется так, чтобы у вас было много времени на общение

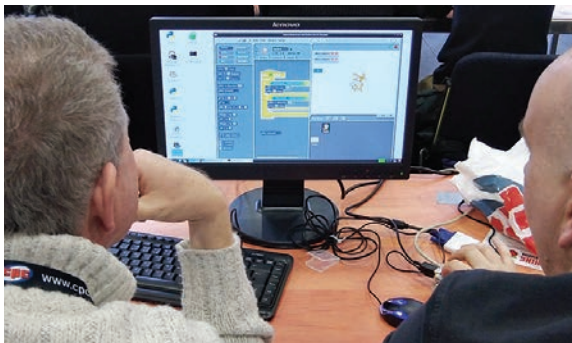
и обмен идеями. Но главное, эти встречи — отличный способ получения новых знаний, так как некоторые группы проводят ежемесячные занятия по программированию или электронике.

Если такая встреча проводится рядом с вами, примите в ней участие. Если нет, почему бы не организовать собственную, например, в рамках встречи местной группы пользователей Linux?

В Великобритании несколько тысяч поклонников Pi, и найти приятелей с похожими интересами проще, чем кажется. Райан Уолмсли [Ryan Walmsley] даже создал отличную карту поклонников Pi по всему миру: <http://rastrack.co.uk>. Можете найти кого-нибудь рядом с вами и, возможно, поговорить о своих замечательных проектах.



➤ Благодаря популярности Raspberry Pi в мире появилось множество групп и клубов по интересам.



➤ Scratch — мощный и доступный инструмент для обучения программированию, и он успешно используется для управления проектами с использованием GPIO, типа роботов и светофоров.

## Полетели со мной!

14 октября 2012 года Феликс Баумгартнер [Felix Baumgartner] установил рекорд для прыжков с низкой околоземной орбиты, совершив прыжок с высоты 24 мили в рамках проекта Red Bull Stratos с бюджетом около £30 млн. Но знаете ли вы, что игрушечный мишка по кличке Бэббидж (названный, разумеется, в честь Чарльза Бэббиджа) похожим способом улетел в космос и прихватил с собой Raspberry Pi?

Это правда. Бэббиджа в космос запустил Дейв Акерман в августе 2013 года, взяв шар, заполненный водородом, и мишку с Raspberry Pi, камерой и батареей. У Бэббиджа также были GPS-передатчик и 3G-модем, чтобы команда на земле могла определить местоположение отважного медведя. Проект Дейва был амбициозным и успешным, но главное — недорогим, около £300. Подробности об этом классном проекте можно прочитать на официальном сайте Акермана: [www.daveakerman.com](http://www.daveakerman.com).

## Передатчик на Pi

Каждый год PA Consulting проводит в начальной и средней школе конкурс применения технологий на пользу обществу. Победители прошлых конкурсов придумали датчик качества воздуха, автоматическую систему выдачи лекарств под контролем врача и охранную систему — и все это на Raspberry Pi.

За последние годы мы навиделись природных катастроф, которые происходили во многих изолированных сообществах по всему миру. В таких чрезвычайных ситуациях очень часто сразу выходят из строя средства связи — что может вызвать

смертельные последствия. В конкурсе 2014 года начальная школа Мерисайд в Блэкпуле работает над проектом, который исследует исчезновение связи и обращение к экстренным службам в подобных случаях. Проект включает водонепроницаемый пластмассовый буй, где находится Raspberry Pi с батареей и солнечной батареей. Внутри корпуса к 4-му контакту Pi подключен проводник, выступающий в роли антенны, и дети написали и установили на Raspberry Pi скрипт на Python, который запускается при срабатывании аварийного переключателя. Это действительно блестящий проект, и тот факт, что он родился в головах учеников начальной школы, поражает. Подробнее о проекте передатчика Pi можно узнать на сайте <http://bit.ly/1d0zRam>.

Raspberry Pi — прекрасный и увлекательный гаджет, и чем больше времени вы с ним проведете, тем больше получите в ответ. Лучший совет, который мы можем дать — попробуйте разное и посмотрите, что пробуждает ваш интерес.

И Scratch (см. наш проект на стр. 88), и Python отлично подходят для изучения программирования, и в обоих языках есть средства для работы с GPIO. Эксперименты с макетами и Pi — отличный способ больше узнать об электронике. Главное — получать удовольствие от того, что вы делаете, а с обилием проектов для Pi научиться можно еще многому. **LXF**



➤ Raspberry Pi очень популярен среди энтузиастов трехмерной печати и самодельных, и они сделали Pi сердцем многих интересных проектов.



# Человек Pi

Пит Ломас спроектировал первый Raspberry Pi. Нейл Мор беседует с ним об истории Pi и о том, почему Манчестер — столица компьютерной вселенной.



Случайное знакомство в Imperial College в Лондоне привело Пита Ломаса [Pete Lomas] — эксперта в области электроники из Манчестера — в Кембридж к Эбену Аптону [Eben Upton] и заключению неофициального соглашения создать альфа-версию платформы для обучения программированию. *Linux Format* удалось посидеть с Питом на заводе Sony, недалеко от Кардиффа, чтобы обсудить Pi, образование и чуть-чуть *Minecraft*.

**Linux Format: Как вы пришли в проект?**

**Пит Ломас:** Я разработал большой комплекс оборудования для Imperial College и отправился на день открытых дверей, чтобы посмотреть, как все работает. И уселся рядом с неким Аланом Майкромфтом [Alan Mycroft], оказавшимся профессором компьютерных наук из Кембриджа.

Потом мы бродили по Гайд-парку и обсуждали, как упало качество знаний тех людей, что сегодня приходят на собеседования, ведь они не имеют представления о таких элементарных вещах, как конденсатор или резистор, или с какой стороны паяльник горячий.

Он отвечал, что в программировании та же ситуация, ведь они не отладили ни одной программы. Похоже, мы много потеряли с исчезновением Sinclair Spectrum и BBC Micro. Теперь они просто сразу становятся геймерами и потребителями электронных устройств.

Он рассказал, что в Кембридже есть такой Эбен Аптон, который задумал создать небольшую макетную плату для обучения программированию. Суть в том, чтобы в июне она уже попала к студентам, и им можно было бы дать на лето задание сделать что-то полезное еще до того, как они придут в университет. Что-нибудь написать или отладить.

Я подумал, что это прекрасная идея, и отправился туда, чтобы с ним поболтать. Всего через полчаса сдуру я сказал, что готов к изготовлению: оборудование у меня есть, пришлите мне чипы, скажите, что плата должна уметь, а я ее спроектирую. Давайте просто сделаем и посмотрим, что из этого выйдет. Я рад, что слез у них с крючка: сейчас счет уже перевалил за два миллиона образцов, такое нам было бы трудно потянуть.

Работа над Pi была мне по душе, потому что позволила вернуться к тому, что я умею. Я понимал, что здесь все лежит на мне: все чертежи, вся разводка печатных плат, — я могу вложить все свои умения в успех этого проекта и заниматься этим в свое удовольствие. Слегка озлился из-за того, что надо было месяцев пять торчать по вечерам в офисе каждый день, но в результате вышел Raspberry Pi.

**LXF: Манчестер ведь как-то связан с ARM, так?**

**Пит Ломас:** Стив Фербер [Steve Ferber], некогда работавший над BBC Micro Computer, теперь профессор в Манчестере. Манчестер — это вообще центр вселенной, от этого никуда не денешься. Даже



Би-Би-Си туда переезжают, сдали их сюда, так что и лучшие повара теперь у нас.

**LXF: Пожалуй, в этом был главный недостаток Манчестера: отсутствие хороших ресторанов.**

**Пит Ломас:** Теперь там появились прекрасные местечки, чтоб поесть. Такой же резкий взлет, как и у Raspberry Pi. Мне было трудно делать сразу и Pi, и свою повседневную работу, и я очень рад, что кембриджский офис взял на себя большую часть рутины.

Важно помнить, почему мы оказались в таком болоте, и учить детей информатике, а для этого нужен штатный директор по образованию, которым у нас является Клайв Бил [Clive Beale], и штатный управляющий директор — Лэнс Говарт [Lance Howarth]. На них все крутится, чтобы попечители могли уделять больше времени стратегии развития и продвижению.

Нас с Эбеном обычно называют Билл и Бен. Эбен взял на себя большую часть работы с Би-Би-Си, так как он прикомандирован к Broadcom, которым мы очень благодарны: ведь если бы они не поддержали нас в самом начале, позволив Эбену включиться в проект, ничего бы не было... моя

**LXF: Удачная метафора и для вашего Фонда.**

**Пит Ломас:** Это просто уму непостижимо. Сначала мы планировали изготовить две или три тысячи [плат Pi] за три года, и меня угораздило ляпнуть: «У меня есть своя производственная площадка, присылайте чипы. Я спроектирую и сделаю платформу сам». С этого все и началось — я сделаю это для вас, ведь это такая хорошая идея. А раз она работает в Кембридже, почему бы не попытаться и где-нибудь еще.

Я инженер, и мне свойственно быть осторожным. Мы всегда сначала пробуем и смотрим, что получится. Мы взялись за дело, и, естественно, поработали над конструкцией Pi, и сделали много. Мы сделали альфа-прототип, и надо было впихнуть его в Pi и уложиться в стоимость. И мы от \$110 пришли к \$35.

Рори Селлан-Джонс [Rory Cellan-Jones] обратился к Дэвиду Брейбену [David Braben, один из создателей *Elite*], тоже попечителю, и сказал: «Мы тут кое-что придумали, не хотите взглянуть?» Это была ранняя версия Pi, но уже с полнофункциональным графическим процессором, и они сделали ролик для YouTube. Когда за несколько недель он набрал 600 000 просмотров, нам стало ясно, что мы чего-то да достигли.

**ОБ АЛЬФА-ПРОТОТИПЕ**

**«Надо было впихнуть его в Pi и уложиться в стоимость. И мы от \$110 пришли к \$35.»**

**LXF: То есть вы поняли, что на Raspberry Pi будет спрос?**

**Пит Ломас:** Многие люди читают про Лам-

борджини, но покупают ее где-то пятеро. Почему-то мы думали, что и с Pi так будет, с чего бы нам рассчитывать на такую же востребованность, как у Arduino? Им-то уже много лет, проделана колоссальная работа. Они росли постепенно, с чего бы у нас быть иначе? И вдруг — как прорвало!

компания со мной попрощалась, но я все равно ее владелец. Мы сейчас находимся на кривой роста, а любой рост — штука болезненная. Мой сын вчера сказал: «Мне ногу больно», и мы понимаем, что это значит: он растет, его кости растут, ботинки стали жать, и нужно купить новые.



**LXF:** На том этапе Arduino еще был скорее внешним модулем к уже существующим системам?

**ПЛ:** Arduino мог использоваться самостоятельно, но компилировать программы нужно было все равно на чем-нибудь другом. Фишка Pi была как раз в его самодостаточности. Когда-то записать SD-карту можно было только при помощи ПК, собственно, как и сейчас, если она у вас испортится. Главную роль, на мой взгляд, сыграло появление сообщества, людей, которые могли сказать: «я сделаю то, а я возьмусь за это».

Как раз тогда сообщество изобретателей стало вновь обретать популярность... Существовало оно много лет, но про него мало кто знал, и они не вовлекались в подобные проекты. А тут оно начало набирать обороты, в США даже стал издаваться журнал Make. Моему сыну нравится идея создания газонокосилки с удаленным управлением, чтобы гоняться за собакой. Такое вызывает творческое мышление, а это же главное, это основа всего. Научившись мыслить творчески, а затем системно, все задумки можно воплотить в жизнь.

**LXF:** Кажется, как раз когда появился Pi, все были ошарашены новостью, что в английских школах не учат программированию.

**ПЛ:** Меня в школе этому тоже не учили, но тогда и компьютеров не было. Свой первый компьютер я увидел уже когда учился в Техническом колледже — стоял он в комнате размером вот с эту [делает жест в сторону большого зала заседаний], и в штате числился специальный человек, который каждый день стирал с него пыль. Это был PDP-8, и я изучил машинный код и Basic, и тогда меня осенило, что раз я могу, рассуждая прагматически, написать набор инструкций, чтобы сделать то-то и то-то, значит, я смогу заставить машину сделать это за меня.

Первая программа, которую я написал, декомпилеровала результаты работы одной из моих лабораторий электроники, которая была не очень-то успешной: я решил заставить машину откатить все

назад и выполнить заново, пока результаты не покажутся мне пригодными.

**LXF:** То есть вы взломали собственный код?

**ПЛ:** Да, я из ранних хакеров. Я еще много чего с машиной делал, но это не для печати, поскольку кое-кто из преподавателей не обрадуется. Я мог заставить телетайп ASR 33 звонить, как телефон. Публика была не в восторге, ну, вы понимаете, куда дело шло.

Строго говоря, здесь важен сам процесс. Если у вас есть цель, суть не в том, как это запрограммировать, а в том, чего еще это поможет добиться. Таким образом все это программирование проникает в жажду познания.

**LXF:** То есть вместо того, чтобы засадить кого-то за изучение кода, надо представить ему конечный результат?

**ПЛ:** Это такой цикл for: все сидят и думают, зачем же нужен цикл for? Какой в этом толк? Другое дело, если дать им что-то вроде Sonic Pi, сделанного в Кембриджском университете, и сказать: вот так получается музыка. Чтобы повторить музыкальный фрагмент, нужно сделать так-то, и это называется циклом for, а затем вот здесь задать, сколько раз он будет повторяться. И сразу становится понятно, как работает вот этот кусок кода, зачем он нужен. Это один из приемов, освоив которые, вы научитесь лучше писать музыку, — и они его освоят.

Когда вы дойдете до программирования на Python, они уже будут кое с чем знакомы. А раз знакомы, то им это будет легче даваться.

**LXF:** Один профессор из Ньюкасла утверждает, что чем меньше детям рассказывать, тем больше они узнают. Там набрали несколько контрольных групп, и лучшие результаты в изучении некоего устройства показала та, где участникам вообще ничего о нем не говорили.

**ПЛ:** Вот именно, они экспериментировали. Над Scratch еще осталась куча работы, чтобы улучшить его быстродействие в Pi, но я все равно пошел на один из ранних Raspberry Jam. Была там пара ребят, настроенных в духе «Ну и что мне здесь делать?» И нужно было убрать этот исходный барьер. Тогда мы сказали: вот вам персонаж — кот. Можете перетащить его под кнопку Старт и задать, как далеко он сможет бегать и куда поворачивать. Нажимаете на кнопку Старт, и вперед.

Тут меня кто-то вызвал. Когда я вернулся, этот кот у них уже танцевал по всему экрану. То есть мы подсказали им, как это работает, они сами разобрались с командами и поняли, что они могут делать много чего...

**LXF:** Потом вы добавили Minecraft...

**ПЛ:** Должен сказать, это прекрасная программа. Впервые узнал о ней от своего сына, тогда восьмилетнего. Он сказал: «Я хочу Minecraft на свой компьютер», — пришлось раскошелиться. Потом вижу: он нашел тротил и взрывает все подряд.

Я решил, что с этой игрой все ясно. Но стоило мне посидеть с ним рядом пару часов, когда он показал мне, что там можно делать, как я понял, что это штука потрясающая. Сама возможность полностью отстроить этот цифровой Lego, включая водопроводы, электронику — и он сказал, что может изготовить даже пушку, стреляющую тротилом...

Важно, что [в Pi] можно выйти за пределы экрана, благодаря GPIO. Конечно, с Grand Theft Auto в отношении графики он не сравнится, но зато может заставить что-нибудь двигаться на вашем столе. Я бы сказал, что в такой встраиваемой среде незримо присутствует гораздо больше компьютеров, чем заметно миру.

**LXF:** Сегодня у нас есть все эти удивительные устройства: планшеты, телефоны, но как они устроены, мы не знаем. Pi успешен в том числе и потому, что вы видите все, что есть внутри.

**ПЛ:** Я это называю луковицей. Можно начать снаружи и работать с тем, что на поверхности, затем копнуть глубже настолько, насколько вас это интересует. Если вы хотите дойти до самой сути, до двоичных основ — пожалуйста.

Это логически вытекает из самого принципа расширения структуры. Вы начинаете с простого и надстраиваете все остальное. Ваша первая программа, «Hello World», это отправная точка, пройдя которую, вы чувствуете себя увереннее. Взяться за другой язык программирования проще, когда один у вас уже в кармане.

Так же и в инженерии: создав один процессор, вы уже знаете, что нужно делать. Может встретиться и что-то новое, но в целом вы уже к этому готовы.

Детям нужно давать инструменты для развития воображения. Это все равно что учить их писать, научив языку, но не разрешая сочинять свои истории. Просто зубрить грамматику. Разве это не подло?

Это труднейшая методическая задача. Введение новых предметов в образовательную программу требует усилий, ведь для этого нужно много что понять.

**LXF:** Должен ли Raspberry Pi как-то этому способствовать?

**ПЛ:** Было бы абсолютно неправомерно сказать учителям: вот вам новые предметы в программу, потрудитесь освоить и внедрить. Откуда сам собой возьмется постоянный профессиональный рост? Необходимо предоставить им такую возможность, ведь они изо всех сил стараются учить детей. На мой взгляд, существование Pi и знание компьютерных наук призваны вдохновить их на творчество.

Вспомним самое начало: тогда перед нами не было такой задачи. Она появилась по ходу развития Pi. Сейчас мы понимаем, что это в наших силах, потому-то у нас и есть директор по образованию и развитию.

Нам очень интересно, что получится, если они появятся в школах. Ведь нужно, чтобы они как-то



вписались в учебные планы. Цели их использования должны быть достижимы за конечный период времени и внутри конкретной школы. Такая программа должна быть хорошо структурирована, и нужно многое сделать, чтобы она себя оправдала.

Ведь можно принести Pi в школу, похвастаться, и на этом все. Чтобы действительно внедрить его в образовательную программу, потребуются гораздо больше труда, и сейчас мы над этим работаем.

**LXF:** И насколько он туда вписывается?

**ПЛ:** Без сомнения, Pi это под силу, равно как и какому-то другому проекту. Мы не хотим ничего навязывать — по-моему, это самая фаталь-

**ARM было 25 000 транзисторов — поразительно, что он появился, после Intel 286.**

**ПЛ:** Ну, миниатюрность и есть причина его популярности. Такова философия ARM: в том же 8086 каждая команда интерпретируется микропрограммой, и он непередаваемо сложен. Создатели ARM пошли другим путем, упростив все операции. Поэтому микропрограмм в нашей модели нет, мы предпочитаем экономную структуру — утяжеленной, которую использует Intel.

**LXF:** Сложный набор команд — это вроде одной машины внутри другой...

**ПЛ:** Стало быть, вы знакомы с проектированием электронных систем; я когда-то сам его преподавал в Манчестерском университете, под руководством великого Тома Килберна [Tom Kilburn], ныне покойного, изобретателя компьютера с хранимой программой совместно с Фредериком Уильямсом [Frederic Williams].

## ОБ ОБУЧЕНИИ

**«Чтобы понять, что не так с кодом, нужно в нем очень хорошо разбираться.»**

ная ошибка. Я бы даже скорее хотел, чтобы учителя посещали Raspberry Jam, чтобы оценить собственные навыки и наблюдать за навыками детей.

У нас в команде есть преподаватель-технарь Алан Донохоу [Alan Donohoe] из Престона, он очень много делает для того, чтобы убедить учителей попробовать Pi в школах. У него прекрасные результаты, и он с невероятным энтузиазмом говорит о Pi как инструменте обучения. Он еще и организует Raspberry Jamboot [усмехается, — LXF].

Ну что поделаешь? Естественно, с таким названием [jamboot — англ. «пирушка»] на вас сыплются всякие ссылки на рецепты, заливки для пирогов, основы для пирогов, недопеченные пироги. И прочие шутки в таком духе; но ведь здорово, что это дает людям возможность посмеяться. Ведь цель не только в учении, но и в развлечении. А если вы смотрите на все, что можно купить, как на потенциальное приращение к этой экосистеме — все в ваших руках.

**LXF:** Функция ввода-вывода не требует особо мощного процессора.

**ПЛ:** Да, но если вы подключаете, например, фотоаппарат, вся обработка данных ведется графическим процессором. Почти вся нагрузка с центрального процессора перекладывается на него. Происходит это за счет Broadcom SoC, он создает дампы кадров. Если вам нужно сжатие JPEG, он и это делает. Если говорить о чипах в плане нагрузки, то вот этот неказистый малыш в углу будет на первом месте, а уж потом — графический процессор и электропитание.

**LXF:** Мне нравится, как рванули эти чипы — а чип ARM там есть? Мы в него заглядывали не так давно, и обнаружили, что в исходной конструкции

Снова идет надстраивание, опирающееся на начальное знание. Я считаю это эффективным способом идти к своей цели. Делать это нужно постепенно, не гонясь за всем сразу. Я предпочитаю начинать с простых вещей, а затем учиться их совмещать. Если подумать, именно так мы программируем. Из простых блоков вы собираете большую программу, а потом неделю разбираетесь, почему она не работает.

То же и с электроникой. Вы разбираете схему на маленькие сегменты, определяете, что с чем нужно соединить и как эта цепь будет работать, затем помещаете их на плату, подключаете ее и пытаетесь понять, почему не работает. Отладка очень многому учит. Ведь чтобы понять, что не так с кодом, нужно в нем очень хорошо разбираться.

**LXF:** Каковы ваши впечатления от преодоления двухмиллионной отметки продаж?

**ПЛ:** Думаю, нам очень повезло, Sony — великолепный производственный партнер. У нас все хорошо с лицензиями, благодаря RS и Farnell. Продукт будто сам собой попал в руки именно тех людей, кому он нужен.

Точности ради, можно было бы найти способ, чтобы найти тот самый миллионный Pi или двухмиллионный Pi. А потом что, 250 000-й Pi? Pi с любым случайно выбранным номером можно считать важным.

**LXF:** Можно найти Пи-миллионный Pi, — 3 141 592-й Pi!

**ПЛ:** Главное, что их используют, с их помощью люди творят и учатся. **LXF**





В Elphel всего три сотрудника: Олег, Андрей и Ольга. Но открыта вакансия Linux-программиста, и после некоторого испытательного срока компания готова спонсировать рабочую визу.

# Принципы GNU/Linux в «Железе»

Игорь Штомпель побеседовал с Андреем и Ольгой Филипповыми из компании Elphel.



Компания Elphel — известный производитель видеокamer на базе открытого и свободного как аппаратного, так и программного обеспечения. Андрей Филиппов — ее президент, ведущий инженер и исследователь. Ольга Филиппова управляет делами компании.

**LXF:** Расскажите, чем вы занимаетесь.

**Андрей Филиппов:** Занимаюсь я тем, что начал в 2001 — попытался применить принципы из буквально потрясшего меня GNU/Linux (когда я о нем узнал — <https://web.archive.org/web/20050315233332/http://www.linuxdevices.com/articles/AT2171151224.html>) к железкам (кстати — <http://old.computerra.ru/hitech/tech/31862/>). Ранняя история компании описана в тех ссылках, а сейчас можно уже подвести некоторые (надеюсь, промежуточные) итоги того, что вышло, а что нет. Для меня главное, наверное, что вышло — возможность заниматься тем, чем интересно, так, как сам считаешь нужным. И чтобы такое «хобби» давало средства к существованию.

Ну, а что не вышло — напрасны были наивные надежды, что, как в свободном софте, удастся только незначительную часть делать самому, а по большей части использовать/адаптировать чужие наработки. Ну и получать новый код/схемы, сделанные другими. В нашей новой камере мы только планируем впервые использовать GPL-ный код для FPGA, написанный не нами — и это будет впервые за 12 лет существования фирмы. Конечно, это ни в коей мере не означает, что я разочаровался в свободном железе; просто для нас фокус в другом. Мы буквально продаем свободу. Свободу как продукт, свободу модифицировать, менять, копировать, если угодно, и наши клиенты это вполне ценят.

Когда стало понятно, что все равно практически все придется делать самим, не надеясь

на поступление кода/дополнений к железу от пользователей, это нас не отвратило от свободного железа. Понятно, что доля тех, кто хочет и может помочь, среди пользователей всегда небольшая. Но если свободные программы используются миллионами человек, то у нас на подписке чуть больше ста человек, а всего счет пользователей идет на сотни, а не тысячи. Конечно, есть свободное железо, распространяемое огромными тиражами — мы и сами используем Arduino, но такие простые и дешевые устройства — это все-таки отдельная ниша. А нам интереснее заниматься чем-то более сложным, новым, создавать то, чего еще нет, без скидок на «свободное» или «проприетарное». Мне когда-то было обидно слышать объяснение, что «GIMP — это такой Фотшоп для бедных». И, к сожалению, хотя я сам пользуюсь исключительно Гимпом, я все никак не дождусь там 16-битного режима, который мне нравился в Фотшоппе в прошлом веке. Да, я знаю, что такой режим можно «прикрутить», но хочу, чтобы он был «из коробки», сразу.

В высокотехнологичном железе есть и дополнительные сложности, которые не в пользу свободных разработок. Во-первых, рынок таких устройств всегда будет ограничен, ведь объективно свободное и сложное железо нужно не всем,

направлении. Т.е. нужно угадывать такие направления и стараться не обрубать потенциальные точки роста — как в самом железе, так и в его программном обеспечении. А это — дополнительные условия, которые требуют дополнительных усилий и/или аппаратных элементов.

Дальше. Разрабатывая свои камеры, мы не можем использовать, например, небесплатные средства разработки FPGA (ПЛИС). Для нормальных производителей такие программы за «смешные» деньги (менее \$10 тыс.), конечно, не сильно удорожают продукт. Но такие суммы значительно превышают стоимость большинства самих наших камер, т.е. за возможность пересобрать прошивку устройства нашему пользователю пришлось бы выложить сумму в несколько раз больше той, которую он уже заплатил. Конечно, в идеальном мире такие программы должны быть свободными, а не freeware (<http://blog.elphel.com/2013/10/fpgas-for-freedom/>), но этого пока нет. Ну и, конечно, мы не можем использовать такие компоненты (или протоколы), где требуется давать подписку о неразглашении. Единственный раз я подписывал NDA с Кодяком, но это было на заранее оговоренный срок, который истек до того момента, когда мы планировали продавать устройства, основанные на этом сенсоре.

И еще. Как верно написал в своей статье Марк Левой [Marc Levoy] (профессор Стэнфорда, автор Франкенкамеры), проприетарные производители камер постоянно нарушают патенты друг друга, и это им обычно сходит с рук, т.к. обнаружить это непросто. Со свободными разработками жизнь патентных троллей облегчается — нам, например, с ними приходилось иметь дело 4 раза (правда, не по части камер, а по подаче питания через кабель локальной сети). И хотя им самим от нас ничего не досталось, на юристов мы уже потратили не одну тысячу долларов.

По-моему, если стараться, чтобы продукт имел сразу три желаемых качества: «свободный», »

**«Мы продаем свободу как продукт, и наши клиенты это вполне ценят.»**

а в основном ученым и разработчикам, т.е. в тех областях, где готовых решений пока нет. Ну, а маленький тираж производства аппаратуры, естественно, сильно повышает ее цену. Во-вторых, чтобы эту свободу можно было продавать, т.е. чтобы были люди, готовые платить деньги не только за принцип, разработка с самого начала должна быть удобной для модификации, причем в неизвестном нам, как разработчикам,

## Короткая справка

» **Андрей Филиппов** — выпускник МФТИ 1978 г., кандидат физико-математических наук (1989 г.). Работал инженером и младшим научным сотрудником Института общей физики в Москве с 1978 по 1995 г. С 1995 по 2001 г. — в компании Cordin (Солт-Лейк-Сити) на позиции ведущего инженера.

С 2001 года стал президентом, ведущим инженером и исследователем в компании Elphel.

» **Ольга Филиппова** в 1996 году закончила МАрХИ, а в 2000 г. — Университет Юты по специальности «архитектор». С 2008 года является управляющей делами компании Elphel.

«дешевый» и «высокотехнологичный», то ничего не выйдет. Можно сделать свободный и дешевый, но простой — выйдет Arduino. Можно — дешевый и высокотехнологичный: выйдет мобильник. Наша ниша — свободных и высокотехнологичных (но недешевых) продуктов.

### LXF: Что входит в текущую линейку продуктов Elphel?

**Ольга Филиппова:** Несколько моделей камер, основанных на модели NC353L: базовая камера NC353L состоит из системной платы 10353 и сенсорной платы 10338 с 5-мегапиксельным сенсором (Artina) + интерфейсная плата 10369 = NC353-369 камера, которая обеспечивает дополнительные интерфейсы: SATA, USB, последовательный порт, облегчает отладку ПО, т.к. с камерой можно общаться через консоль, + мультисенсорная/мультиплексорная плата 10359 = NC353-369-359, позволяющая подключить до 3-х сенсоров к одной системной плате; также позволяет программировать FPGA (ПЛИС), такой же, как на системной плате; такая конфигурация электроники используется в стереокамере и других мультисенсорных камерах, например, Hammer, которая обеспечивает 360-градусный угол обзора, используя 2 сенсорных платы и 2 широкоугольных объектива Fisheye. К любой из перечисленных моделей можно добавить встроенный жесткий диск или CF-карту, подключить GPS и IMU. Варианты представлены в таблице на [http://www3.elphel.com/model\\_353\\_cameras](http://www3.elphel.com/model_353_cameras).

Еще одна линия — калиброванные панорамные камеры Eyesis4Pi. Полная модель Eyesis4Pi обеспечивает угол обзора 360°×180°, т.е. сферическую панораму. Точная калибровка опико-электронных модулей камеры позволяет использовать камеру как стереофотограмметрическую — вычислять расстояния до объектов в кадре, используя только полученные фотографии. Мы производим и фотограмметрические камеры NC353L-PHG и NC353L-PHG3, с одним и с тремя опико-электронными модулями. Ведется разработка новой модели NC393, с более мощным процессором и ПЛИСом — подробно о ней см. в блоге: <http://blog.elphel.com/category/model-393/>.

### LXF: Как организовано производство камер Elphel? Какие стандарты лежат в основе этого процесса?

» **ОФ:** В Elphel работают 3 постоянных сотрудника. Разработка камер (замысел, схемы, разводка

печатных плат, набор компонентов, дизайн корпуса и чертежи — все делает Андрей. Проверка и тестирование новых прототипов тоже его. Первую плату он всегда собирает (паяет) сам (кроме BGA), т.к. это помогает отловить ошибки и оптимизировать дизайн.

Производство организовано так: платы заказываются в Китае, сборка сложных плат — на заводе в 10 минутах от офиса Elphel, алюминиевые корпуса для камер — в Китае, сложные алюминиевые детали (всего их в камере Eyesis4Pi больше 100) вытачиваются в США (2 мастерские, тоже в Солт-Лейк-сити, специализирующиеся на разных размерах и точности). Анодирование, сварка, специальные прокладки, объективы и пр. — все заказывается либо в Америке, либо в Китае. Срок поставки — от 2-х дней до 2-х месяцев. Детали за-

## «В процессе изготовления камер — все элементы современного производства.»

казываем из расчета на 100–200 камер NC353L и на 3–5 камер Eyesis4Pi.

Общение с поставщиками, заказ деталей, инвентаризация, а также и бухгалтерия, получение заказов на камеры — этим в основном занята.

Сборка, настройка и особенно калибровка панорамных камер осуществляется у нас. Наш новый офис/производство спроектирован и построен под калибровку: главное помещение — это калибровочная комната, специально оборудованная, с калибровочной мишенью размером 7×3 м и калибровочной машиной, она может быть полностью изолирована от света. В то же время стены остальных помещений расположены под углом, чтобы иметь возможность калибровать камеры с длиннофокусным объективом. Также имеется чистая комната для сборки оптических элементов, склад деталей, компьютерная, помещение для сборки и тестирования. Сборкой, тестированием, отладкой, настройкой камер и программного обеспечения занимается Олег Джимиев, инженер.

Вообще, несмотря на крошечный размер нашей компании (высокоэффективной), в процессе изготовления камер присутствуют все элементы современного производства, включая аутсорсинг из Китая. Производство высокотехнологичных камер и так очень дорого, и камеры Elphel, конечно, гораздо дороже массовых, широко используемых камер, а без поставок из Китая цены выросли бы втрое или больше.

Конечно, в компании очень много других дел: техническая поддержка через список рассылки — Олег и Андрей; переписка и телефонные разговоры с клиентами — Ольга; в прошлом году мы все подавали на государственный грант — это примерно 180 человеко-часов; подготовка к выставкам и конференциям тоже занимает очень много времени всех сотрудников.

Наверное, в новой статье стоит больше рассказать про панорамную и фотограмметрическую камеру Eyesis4Pi и про наши разработки методов калибровки, т.к. это абсолютно новая линия продукции и новое направление, которого не было в 2009 году, когда была опубликована первая статья. Точная калибровка Eyesis4Pi позволяет автоматически склеивать полученные фотографии в панораму, используя метод pixel-mapping, а также позволяет определить геометрические свойства объектов по изображениям. Геометрические свойства калибровочной мишени (наклеенной на ровную стену) посчитаны с помощью Eyesis4Pi и описаны в статье про офис: <http://blog.elphel.com/2013/06/elphel-new-camera-calibration-facility/>.

Также в смысле производства — с панорамной камерой количество деталей и поставщиков увеличилось в несколько раз. Вся механическая и опико-механическая часть камеры тоже разработана на Elphel, помимо электроники.

Камеры Elphel использовались в Google Streetview, и, кстати, панорамы Москвы и Санкт-Петербурга сняты камерами Elphel.

Сейчас ведется разработка новой камеры NC393. Эта разработка была давно запланирована, но мы ждали подходящего FPGA/процессора; NC393 спроектирована как мультисенсорная (можно подключить 4 сенсора одновременно) для поддержки мозаичных систем и 3D, в том числе real-time 3D, для летающих БПЛА.

**АФ:** Сенсорные порты камеры подключены непосредственно к FPGA, и их легко «перепрофилировать» — например, вместо 4-х сенсоров оставить 2, а освободившиеся порты использовать для управления моторами квадрокоптера — быстрое действие программируемой логики здесь никак не помешает и позволит разгрузить центральный процессор. Кстати, камера получается удобной для космических применений — с одной стороны, в самом чипе предусмотрен режим исправления ошибок оперативной (и внутренней) памяти, а с другой — в камере реализованы два альтернативных способа загрузки: с внутренней NAND Flash и со стандартной карточки MicroSD. Причем и сама «кнопка» перезагрузки, и выбор источника определяется через USB-порт (по совместительству с последовательным портом системной консоли). Поэтому устройство-супервизор (например, Arduino) может перезагрузить камеру. Мы думаем сделать адаптер к сенсорному порту, чтобы можно было подключать shields от Arduino — мы применяем камеры для управления многими устройствами, а для Arduino продается множество специализированных интерфейсных плат.

В этой новой разработке мы надеемся осуществить давнюю мечту — использовать свободный код для FPGA, написанный другим. За все 12 лет существования фирмы мы этим пользовались только для ПО, а свободный код программируемой логики был исключительно наш.

Вы спрашивали, какие стандарты лежат в основе процесса производства камер. Не очень понимаю, что именно вы имеете в виду. А то некоторые стандарты (вроде IEEE802.3af) очень дорого обходятся — не одну тысячу долларов пришлось заплатить юристам для отражения многократных атак патентных троллей. Конечно, патент, например, на использование фантомного питания вряд ли действительная инновация (мне про подачу питания по сигнальным проводам отец в детстве рассказывал), но суд для признания патента недействительным требует средств, многократно превышающих те, которыми мы обладаем.

Фотографии нашего офиса можно посмотреть на <http://blog.elphel.com/2013/06/elphel-new-camera-calibration-facility/>. Там же есть ссылки на панорамные изображения офиса.

## LXF: Расскажите о своем инструментарии. Какие языки программирования и средства разработки вы используете?

**АФ:** В принципе, то, что написано в статье 2009 года, верно и сейчас, с небольшими добавлениями: Icarus Verilog с GTK Wave для симуляции FPGA (ПЛИС), C — drivers Linux, приложения Java, на основе *ImageJ* для обработки изображений, PHP, Python, JavaScript, средства разработки: *Xilinx ISE*, *Eclipse* (не очень хочется рекламировать проприетарное и вообще ничем не замечательное ПО от Xilinx, но приходится его использовать). Подробнее об этом я писал недавно в <http://blog.elphel.com/2013/10/fpga-is-for-freedom/>.

## LXF: Какое программное обеспечение использует компания Elphel в своих проектах?

**ОФ:** Программное обеспечение для работы с камерой: SDK, Elphel Toolkit, позволяющий запустить систему с Live USB или установить и обновлять ее на своем компьютере. Открытое программное обеспечение, распространяемое под лицензией GNU/GPL. Текущая и все предыдущие версии выложены на SourceForge, а также текущая версия поставляется с каждой камерой. Для панорамных камер мы разработали ПО для получения изображений, коррекции аберраций, дисторсии, цвета и пр. у полученных изображений и склеивания отдельных фотографий в панорамы. Изображения синхронизированы с данными, полученными с IMU и GPS, и привязаны к карте. Подробнее о калибровке панорамных и фотограмметрических камер написано в нашем блоге: <http://blog.elphel.com/category/panoramic/>.

**АФ:** Сейчас мы осваиваем OpenEmbedded и что изменилось в ядре Linux (в основном, в смысле устройства/написания драйверов) со времен 2.6.19 — версии, которая используется в наших прежних камерах. Именно на основе OE и строится ПО для новой ([http://blog.elphel.com/wp-content/uploads/2013/11/system\\_393.jpeg](http://blog.elphel.com/wp-content/uploads/2013/11/system_393.jpeg)) камеры:

<http://sourceforge.net/p/elphel/meta-elphel393/>. Сначала, правда, нам пришлось написать замечательную проприетарную загрузку от Xilinx (<http://sourceforge.net/p/elphel/ezyng/ci/master/tree/>, <http://sourceforge.net/p/elphel/meta-ezyng/>). Производители этой SoC Zynq, которую мы используем в камере (объединяющую двудядерный ARM и мощную FPGA), умудрились к свободным GNU/Linux и U-Boot привесить-таки проприетарный предзагрузчик (FSBL — first stage boot loader), для генерации которого требуется использование проприетарного ПО, а сам сгенерированный файл невозможно легально распространять. Т.е. наши пользователи не смогли бы сами пересобрать образ. В бесплатном варианте, кстати, это проприетарное ПО устанавливается вместе с обязательной программой, которая может посылать фирменному производителю любую информацию с вашего компьютера. Причем в лицензионном соглашении оговаривается, что Интернет есть Интернет, и они не могут гарантировать конфиденциальность этой информации во время пересылки. Подробно про это написано в <http://blog.elphel.com/2013/10/fpga-is-for-freedom/>.

Сейчас эта работа уже завершена, и в камере не используется никакого проприетарного ПО (за исключением, конечно, кода во встроенном ПЗУ процессора). Изменений, конечно, немало, и очень обрадовало, что для половины использованных устройств I<sup>2</sup>C нашлись готовые драйверы, достаточно было только прописать их в Device Tree. Ну, а для тех микросхем, для которых готовых драйверов не оказалось, на них я и решил подготовиться к написанию (частично — к портированию нашего существующего кода) драйверов для собственно «камерной» части.

Другая категория ПО, которое мы используем (и разрабатываем) — это программа *ImageJ* (<http://ru.wikipedia.org/wiki/ImageJ>). Именно в виде плагинов для нее ([sourceforge.net/p/elphel/ImageJ-Elphel/](http://sourceforge.net/p/elphel/ImageJ-Elphel/)) написаны наши программы по обработке изображений, в основном для мультисенсорных камер. И под эти программы (точнее, конечно, под сам процесс калибровки камер) мы даже преобразовали наш офис (или его лучше назвать лабораторией?) — <http://blog.elphel.com/2013/06/elphel-new-camera-calibration-facility/>.

Конечно, все ПО, которое мы разрабатываем сами, лицензируется под GNU GPLv3. «Железки» — как электронные платы, так и механические детали — под двойной лицензией, CERN OHL и GNU FDL, вся документация (включая чертежи и STEP (3d CAD) файлы) доступна на [wiki.elphel.com](http://wiki.elphel.com). Здесь показано, как посетители рассматривают сборочный чертеж свободной камеры на SIGGRAPH: <http://blog.elphel.com/wp-content/uploads/2012/08/elphel-at-siggraph02.jpeg>; ну, а каждую указанную там деталь можно найти хоть Гуглом: <http://www.google.com/search?q=0353-19-66>.

## LXF: В каких крупных проектах были использованы решения Elphel? Какие из ведущих компаний стали вашими клиентами?

**ОФ:** Самый крупный клиент — Google. NC353L использовались для Google Streetview (встроены

в гугловскую панорамную камеру), и еще раньше мы разработали камеру NC323 для сканирования книг в проекте Google Books. В основном наши заказчики — это университеты и научные лаборатории всего мира. Их очень много — можете посмотреть на карту: <http://map.elphel.com/>, но мы не можем назвать их крупными заказчиками, т.к. каждый покупает 1–5 камер.

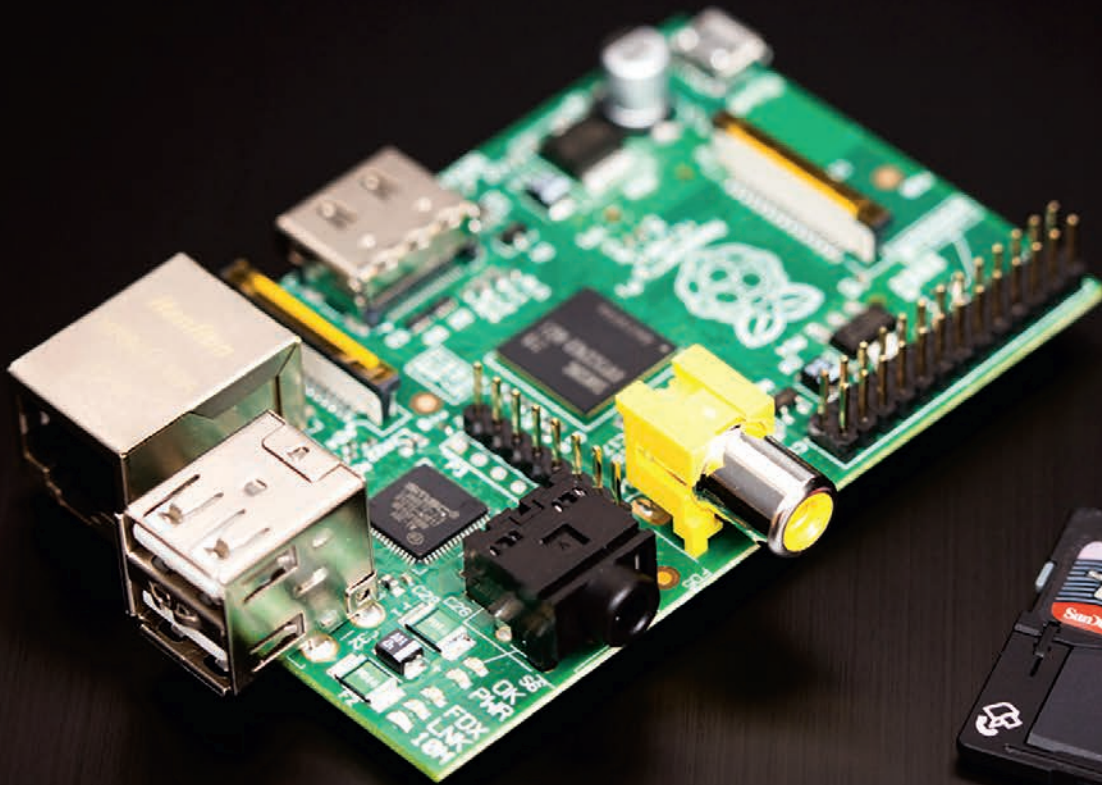
Применения камер: роботы/машинное видение, подводные роботы, беспилотные летающие аппараты — квадрокоптеры и пр., астрономические наблюдения, кинематография, фотофиниш, микроскопы, панорамные изображения, фотограмметрия, видеонаблюдение и т.д.

## LXF: В каких мероприятиях компания принимала участие, в каких будет принимать в текущем и следующем году? Является ли Elphel участником каких-либо из них на регулярной основе?

**ОФ:** Первое участие нашей компании в международной конференции — CineGrid, которая проходит ежегодно в Сан-Диего, Calit2. Насколько мы поняли, тема конференции — изображения и видео высокого разрешения: их получение, обработка, запись, хранение, применения для киноиндустрии и разные другие применения. Они сами в Calit2 построили панорамы-комнаты, в которых можно интерактивно общаться с реальностью на экранах. Система называется CAVE. Андрей представлял там наши многосенсорные камеры: панорамный Eyesis4Pi и другие разработки. В 2012 году Elphel принимал участие в выставке SIGGRAPH, демонстрируя Eyesis4Pi и методы калибровки (<http://blog.elphel.com/2012/07/elphel-at-siggraph-2012/>, <http://blog.elphel.com/2012/08/the-last-chance-to-see-us-at-siggraph12/>). В следующий раз мы планируем подать заявку на выступление на конференции SIGGRAPH (2014, 2015, как получится). Последние 3 года Elphel регулярно делает презентации в Университете Юты, на факультете программирования [computer engineering], а также в институте SCI (Scientific Computing and Imaging). Презентации выложены здесь: <http://www3.elphel.com/presentations>.

В октябре 2010 и 2009 Elphel участвовала в местной выставке и конференции: Utah Open Source Conference, демонстрируя камеры, и рассказывая про Elphel, про наш опыт разработки и производства открытого «железа» и ПО и ведения бизнеса с открытыми продуктами (<http://blog.elphel.com/2010/10/elphel-participation-in-russian-open-source-hardware-mini-conference/>, <http://blog.elphel.com/2009/10/elphel-at-utah-open-source-conference-2009/>).

В октябре 2010 Elphel дистанционно принимала участие в мини-конференции Russian Open Source Hardware в Санкт-Петербурге (<http://blog.elphel.com/2010/10/elphel-participation-in-russian-open-source-hardware-mini-conference/>), в семинаре в Бордо на RMLL-2010 (<http://blog.elphel.com/2010/06/elphel-workshop-in-bordeaux-during-rml-2010/>). В июле 2008 — выставка Linux World в Сан-Франциско; ноябрь 2007 — Linux World в Гуанчжоу, Китай. 2004, 2005, 2006, 2007 — LinuxTag, Германия: Висбаден, Карлсруэ, Берлин. **LXF**



# А если не Raspbian?

Отложив в сторонку официальную операционную систему для Raspberry Pi, **Дэвид Хейвард** проверяет три других заметных дистрибутива, чтобы вы тоже их попробовали.

**R**aspberry Pi — самая суть творчества, свободы и открытости — всего того, за что стоит Linux и мир открытого ПО. Неудивительно, что он так любим.

Нет конца оригинальным и творческим вариантам использования этого чудесного компьютера-крошки, или платы для разработки. Мы видели, как Raspberry Pi выходит за пределы атмосферы, ныряет в морские глубины, превращается в старый игровой автомат, и как несколько Pi, объединенных вместе, предоставляют возможности суперкомпьютера.

Их заключали в дерево, картон, металл, конструктор «Лего» и штукатурку. Ими пользовались машиностроительные предприятия, школы, колледжи и университеты. Все ограничивается, кажется, только воображением их владельцев. Однако большинство из нас использовало Pi в нескольких

домашних проектах и для того, чтобы лучше понять Linux, электронику, моторизованных роботов, сеть и программирование. В основном это стало возможным благодаря самоотверженным командам разработчиков, создавших операционные системы, на которых работает Raspberry Pi. Несомненно, 99% пользователей RPi берут офи-

**«99% пользователей Pi берут Raspbian, но есть и другие варианты для экспериментов.»**

циальную операционную систему Raspbian — Linux-дистрибутив на базе Debian; но есть и другие варианты для экспериментов. Существует масса операционных систем для Raspberry Pi, с помощью которых можно добиться большего, применяя Pi в специализированных областях: для

медиа-центра, альтернативных систем, ретро-игр или для вычислений. Мы взглянули на верхушку айсберга операционных систем для Raspberry Pi, чтобы увидеть, какие проекты проплывут через наше воображение.

Пожалуй, самый популярный вариант — медиа-центр на Pi. Благодаря HDMI-порту, достаточной вычислительной мощности, небольшому размеру и бесшумной работе Raspberry Pi свил уютное гнездышко под многими телевизорами.

Сначала пользователи были довольно сильно ограничены в своих возможностях, но благодаря сообществу пользователей RPi внедрено много ценных улучшений. Читая эту статью почти наверняка пробовали операционную систему XBMC, но многим ли из вас знаком *RasPlex*, клиент *Plex Media Server* для Raspberry Pi?

# RasPlex

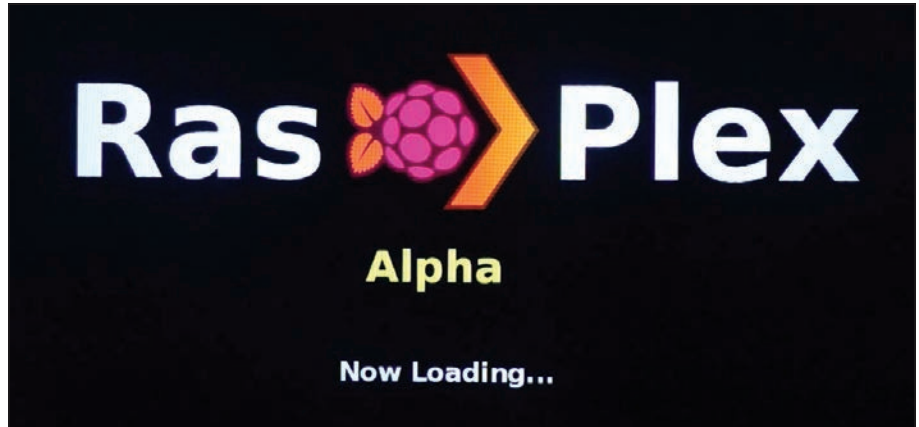
Превратим Pi в медиа-центр.

Для начала вам потребуется установленная версия *Plex Media Server*; это программа с закрытым исходным кодом, но бесплатная для использования. Платформы с *Plex Media Server* существуют для разных операционных систем: Windows, OS X, Linux, NAS, FreeBSD и даже нескольких мобильных ОС, поэтому шансы на то, что вы сможете найти платформу, которая подходит вам больше всего, высоки. При возникновении проблем с серверной частью установки воспользуйтесь отличными пошаговыми инструкциями на [goo.gl/rld6P5](http://goo.gl/rld6P5).

Настроив серверную часть и потратив несколько минут на организацию своей коллекции мультимедиа, а также проверив, что серверная часть *Plex* нашла все файлы и их местоположение указано верно, переходите к настройке RasPlex. Но сначала убедитесь, что у вас есть все необходимое для этого: (очевидно) Raspberry Pi, SD-карта



С помощью загрузчика RasPlex Downloader записать образ ОС на SD-карту проще.



Для первой загрузки RasPlex из-за «холодного кэша» понадобится вечность, поэтому кэш лучше разогреть. Кому бутылочку с горячей водой?

объемом 1 ГБ или более — желательно класса 10 или выше — и доступ к домашней сети. Разумеется, нужно убедиться, что Raspberry Pi подключен к монитору или телевизору, а к нему подключены клавиатура и мышь. После этого можно поместить Pi в какой-нибудь корпус. А пока можно просто оставить как есть.

Также стоит отметить, что хотя RasPlex будет довольно хорошо работать на первых версиях Raspberry Pi с объемом ОЗУ 256 МБ, по очевидным причинам лучше запускать его на более новых версиях с ОЗУ 512 МБ. Разработка RasPlex еще в самом начале, поэтому хотя разработчики пользуются дистрибутивом и тестируют его на версиях с 512 МБ, то же самое можно сделать в версиях с 256 МБ — просто не стоит ждать, что все будет работать так же быстро, как на меньшем брате.

Существует три типа установщиков RasPlex — для Windows, OS X и Linux. Их можно загрузить с [goo.gl/8LPXc9](http://goo.gl/8LPXc9), и в них есть простой графический интерфейс для переноса образа на SD-карту.

Все, что нужно сделать — вставить SD-карту, запустить программу установщика, выбрать версию RasPlex для загрузки (мы бы посоветовали текущую версию, если у вас нет каких-то особых мыслей), затем выбрать загруженный образ и нажать кнопку Write to SD Card для записи новой ОС на карту. После записи образа на SD-карту просто вставьте ее в Raspberry Pi и включите питание Pi для первой загрузки RasPlex.

Именно при загрузке RasPlex все начинается, и поэтому в первый раз графический интерфейс RasPlex будет работать очень медленно — в версиях Pi с ОЗУ 256 МБ почти зависать. Это нормально, и называется «холодным кэшем»: как написано в FAQ на сайте RasPlex, «у RasPlex есть клиентский кэш изображений, миниатюр и мета-данных. При первой загрузке кэш „холодный“, и ему требуется время на разогрев. Если при первой загрузке RasPlex работает медленно, можно принудительно разогреть кэш».

Чтобы в следующий раз все работало быстрее, можно принудительно «разогреть» кэш, войдя в раздел All [Всё] и загрузив All Videos [Всё видео], All TV Shows [Все телепрограммы] и т.д., пока не загрузятся все миниатюры и весь фан-арт — тогда все они добавятся в кэш. Обычно делать это не обязательно, но последующий рост производительности определенно стоит времени, потраченного на кэширование всего содержимого.

## Помогите RasPlex

Как мы уже говорили, RasPlex только начинает свою жизнь в Raspberry Pi, и почему бы не помочь сделать его лучше? Зайдите на [goo.gl/vlsUC](http://goo.gl/vlsUC), и вы узнаете несколько прекрасных способов поддержать существование и постоянный рост проекта. А в свободное время вы сможете наслаждаться фильмами в HD-качестве на Pi.

## XBian

RasPlex основан на XBMC, но многие предпочитают всем ответвлениям оригинал. Если вы из их числа, то в XBian вы найдете полноценное и передовое решение в виде минимального образа Raspbian с XBMC.

Дистрибутив впечатляет, и его поддерживает обширное сообщество, на которое можно положиться при возникновении проблем на любом этапе. Все, что нужно сделать — зайти на [www.xbian.org](http://www.xbian.org), последовать по ссылкам для загрузки и попробовать XBMC на Raspberry Pi.

XBian имеет несколько преимуществ перед RasPlex. Для начинающих, во-первых, не нужна

серверная часть *Plex* — подойдет диск NAS, медиа-сервер или просто USB-диск с вашими файлами. Во-вторых, XBian нетребователен к системным ресурсам и сразу поддерживает массу беспроводных сетевых адаптеров. В-третьих, графический интерфейс очень плавный — благодаря своей легкости, XBian помещается на SD-карту объемом 2 ГБ, и файлы мультимедиа обычно воспроизводятся очень хорошо.

Наконец, хотя личные предпочтения обычно одерживают верх, всегда лучше протестировать обе системы и решить, какая подходит вам больше.

# RISC OS

## Приобщитесь к Элите.

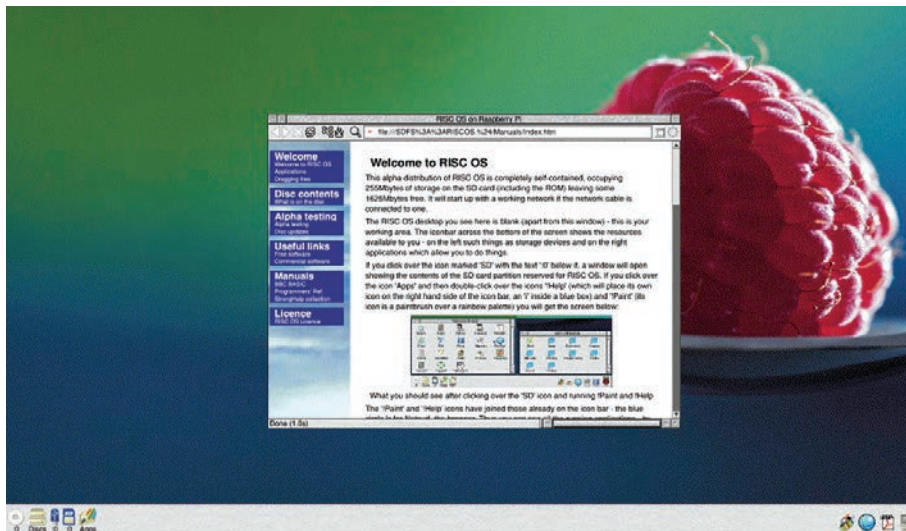
Ретро сейчас явно в моде. Нельзя включить телевизор, открыть газету или зайти на сайт без того, чтобы перед глазами не пронеслась ваша вычислительная юность. Право, в этом есть что-то волшебное; а что может быть более ретро, чем превосходная *Elite*?

*Elite* была прекрасной среди игр, и запуск ее на исходных системах, будь то Spectrum, Commodore, BBC, способен выжать ностальгическую слезу даже из стеклянного глаза. И, сглотив комок в горле, мы подумали: было бы забавно запустить *Elite* на Raspberry Pi с установленной RISC OS — нечто вроде «шведского стола» на ретронебесах.

## Что такое RISC OS?

RISC OS была выпущена в 1987 году под названием Arthur 1.20 и разрабатывалась Acorn Computers для их компьютеров Archimedes на базе ARM. Эта конкретная версия для Raspberry Pi, разумеется, бесплатна, но за £35 можно приобрести набор программ *NutPi*. В нем есть масса дополнений, достаточных для того, чтобы сделать RISC OS RPi полноценной системой на каждый день.

На момент написания статьи версия RISC OS для RPi — 5.19 RC6, а в наборе программ по умолчанию вы найдете браузер, текстовый редактор и даже BBC BASIC. Так что, если надоест *Elite*, вы всегда сможете написать свою игру. Первый пункт повестки дня — загрузить последнюю версию RISC OS для Raspberry Pi. Это не слишком сложно: зайдите на сайт Raspberry Pi, перейдите в раздел Downloads [Загрузки] и прокрутите



► RISC OS — функциональная, красивая и быстрая система, и одновременно ретро.

список вниз до раздела RISC OS. Там вы сможете загрузить образ через торрент или напрямую, и так как образ весит всего 97,9 МБ, много времени это в обоих случаях не займет. После загрузки образа распакуйте его и воспользуйтесь *Win32DiskImager* в Windows, утилитой *RPi-sd card builder* в Mac OS X или командой **dd** в Linux, в зависимости от своей системы и предпочтений.

После загрузки вы заметите, что RISC OS выглядит очень приятно, особенно если к RPi подключен приличный монитор разрешения 1080p, так как вся красота RISC OS видна только на полном

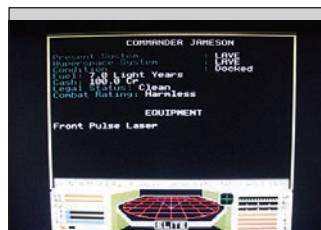
экране. К сожалению, вы не сможете воспользоваться сетью, так как по умолчанию она отключена; но это легко исправить. Закройте сообщение об ошибке NetSurf и дважды щелкните по иконке Configure [Настроить] на рабочем столе. Один раз щелкните по иконке Network, затем по иконке Internet. Поставьте галочку у Enable TCP/IP protocol suite [Включить набор протоколов TCP/IP], и три иконки внутри должны загореться. Теперь нажмите Close [Закреть], затем Save [Сохранить] из предыдущего окна и перезагрузитесь кнопкой Reset Now [Перезагрузиться сейчас].

## Elite и эмуляторы

Так как *Arc Elite* была написана для компьютеров на базе ARM2 и ARM3, она не запустится в среде ARM6 на Raspberry Pi. Поэтому надо эмулировать более старую версию RISC и ARM с помощью *ArcEm*.

Эмулятор и соответствующие прошивки можно загрузить и по отдельности, но проще взять один архив, со всем необходимым для эмуляции *Elite*. Однако при попытке открыть файл *ArcEm* вместо *Elite* откроется *StrongArm*. К счастью, запустить *Elite* можно так: загрузите архив *ArcEm.zip* (<http://www.phlamethrower.co.uk/misc2/arcem.zip>), затем распакуйте его на флешку. В RISC OS устанавливать программы не обязательно: они загрузятся с внешнего устройства.

Вставьте флешку в Raspberry Pi и подождите, пока RISC OS обнаружит новое устройство. Щелкните левой кнопкой мыши, чтобы открыть файловый менеджер, затем в NetSurf перейдите на [goo.gl/6TIOE](http://goo.gl/6TIOE) и загрузите последнюю версию файла *ArcEm*.



► Прямо-таки чувствуешь себя в 1984 году.

Это архив, но нам нужно открыть его с помощью приложения RISC OS под названием *sparkFS*. Щелкните по иконке SD-карты в левом верхнем углу окна, чтобы открыть содержимое файловой системы, затем дважды щелкните по папке *Utilities*. Найдите программу *sparkFS* и дважды щелкните по ней — теперь она должна появиться рядом с набором иконок слева внизу. Перетащите загруженный файл *arcem-1.50-alpha2-riscos* на иконку *sparkFS*, которая теперь находится слева внизу. Откроется окно с содержимым, которое нужно перетащить и бросить на исходное содержимое архива *ArcEm*. Тогда файл *ArcEm* и каталог *hostFS*, а также текстовый файл заменятся

на те файлы, которые будут работать в данной системе.

Теперь зайдите на [goo.gl/wEGK2](http://goo.gl/wEGK2) и загрузите файл **b5052410.arc**. Как и ранее, перетащите его на иконку *sparkFS*, затем перетащите распакованное содержимое в каталог *hostFS*. Чтобы запустить *Elite*, дважды щелкните левой кнопкой мыши на новом файле *ArcEm*, который загрузит старую платформу Archimedes, затем дважды щелкните на диске *HostFS*. После этого должна запуститься *Elite*, хотя в нескольких случаях нам приходилось дважды щелкать по иконке *Elite*. На вопрос, имеет ли ваш экран высокое разрешение, можете ответить «Y [да]». Секунд через десять *Elite* запустится.



# RetroPie

Сделаем из Pi классическую игровую консоль.

**К**оль скоро мы говорим об играх, и в частности, о старых играх, что может быть лучше, чем превратить Raspberry Pi в старинную игровую консоль? Хотя старые игры и не для всех, есть несомненное очарование в том, чтобы поиграть в игры вроде *Doom* на компьютере, который умещается в ладони — и к тому же это отличный способ провести время с детьми долгими дождливыми вечерами.

RetroPie — это операционная система «все в одном», построенная из урезанной версии Raspbian с несколькими дополнениями для баланса. Ее придумал и создал petRockBlock, поклонник RPi и ретро — он массу времени промучался с поддержкой и обновлением проекта RetroPie нашей пользы ради. Перейдите на [goo.gl/34tFal](http://goo.gl/34tFal) и загрузите последнюю версию с архивом образа RetroPie. Затем распакуйте архив и скопируйте образ на SD-карту на 4 ГБ или больше, одним из стандартных способов, описанных на сайте Raspberry Pi или в одном из наших предыдущих руководств. После этого подключите Raspberry Pi к подходящему монитору или телевизору, подключите клавиатуру, мышь и USB-джойстик, чтобы играть было еще интереснее, и включите Pi.

При первой загрузке запустится утилита настройки контроллера, используемого для навигации по пользовательскому интерфейсу



» Экран настройки используется для навигации по пользовательскому интерфейсу EmulationStation.



» Наслаждайтесь старыми играми, среди которых, в конце концов, есть и лучшие игры всех времен.

EmulationStation, который по сути представляет собой меню, где можно выбирать различные игры и эмуляторы. Однако учтите, что для большинства операционных систем первая загрузка на Raspberry Pi может быть очень медленной. Но последующие загрузки будут быстрее, и игры будут запускаться как обычно.

По окончании настройки вы вернетесь в Terminal или в командную строку, поэтому просто наберите 'emulationstation' для загрузки пользовательского интерфейса RetroPie.

## Прошивки и все такое

Игры, с которых вы начнете — довольно ограниченные платные версии *Duke Nukem 3D*, *Doom* и какая-то доселе неведомая игра *Apple 2*, поэтому вам наверняка понадобятся еще несколько. Но здесь встает вопрос законности загрузки образов игр и игры в них. Одними образами можно пользоваться вполне легально, другими — нет. Мы не будем влезать в нелегальную «серую» область, где

не сможем защититься, и по части прошивок, наверное, лучше будет предоставить вас самим себе. Мы почти уверены, что вы хорошо знаете, как обойти Google, и сами поймете, что делать дальше.

Также не забудьте заглянуть на страницы с описанием адаптера GPIO на сайте petRockBlock ([goo.gl/QFMWCw](http://goo.gl/QFMWCw)), где есть пошаговое руководство по созданию устройства, позволяющего подключать к RPi игровые контроллеры.

Мультимедиа, старые игры и полноценная альтернативная ОС — лишь верхушка айсберга, когда дело касается Raspberry Pi. Существует еще множество более полных и специализированных операционных систем, с которыми можно поэкспериментировать.

Список исчерпывающий, но теперь вы игрались с RPi уже достаточно, чтобы задуматься о создании собственной ОС. Кстати, а почему бы не сообщить об этом нам? Напишите нам и расскажите, какой вы придумали собственную ОС для Raspberry Pi и почему. **LXF**

## Поддерживаемые системы и эмуляторы

Судя по сайту petRockBlock, список поддерживаемых эмуляторов и систем для версии RetroPie 1.9 довольно впечатляющий:

- » Amiga (UAE4All)
- » Apple II (Basilisk II)
- » Arcade (PiFBA, Mame4All-Rpi)
- » Atari 800
- » Atari 2600 (RetroArch)
- » Atari ST/STE/TT/Falcon
- » C64 (VICE)
- » CaveStory (NXEngine)
- » Doom (RetroArch)
- » Duke Nukem 3D
- » Final Burn Alpha (RetroArch)
- » Game Boy Advance (gpSP)
- » Game Boy Color (RetroArch)
- » Game Gear (Osmose)
- » Intellivision (RetroArch)
- » MAME (RetroArch)
- » MAME (AdvMAME)
- » NeoGeo (GnGeo)
- » NeoGeo (Genesis-GX, RetroArch)
- » Sega Master System (Osmose)
- » Sega Megadrive/Genesis (DGEN, Picodrive)
- » Nintendo Entertainment System NES (RetroArch)
- » N64 (Mupen64Plus-Rpi)
- » PC Engine/Turbo Grafx 16 (RetroArch)
- » PlayStation 1 (RetroArch)
- » ScummVM
- » SNES Super Nintendo Entertainment System (RetroArch, PiSNES, SNES-Rpi)
- » Sinclair ZX Spectrum (Fuse)
- » PC/x86 (rpx86)
- » Эмулятор Z Machine (Frotz)

Если здесь нет того, что вам нужно, почему бы не поработать над поддержкой этого самим и не рассказать об этом сообществу?



# GCC

## Универсальный компилятор

Быстрая, доступная, расширяемая GNU Compiler Collection подняла свободное ПО на невиданные доселе высоты. Ричард Хиллсли рассказывает историю GCC, великого универсального компилятора.

**П**ервый релиз GCC вышел 22 марта 1987 года. Изначально название было акронимом от GNU C Compiler; GCC был разработан Ричардом Столлменом (Richard Stallman) (с помощью Лена Тауэра (Leonard Tower) и других) как строительный блок для операционной системы GNU. Первые версии GCC поддерживали только язык C на ограниченном количестве архитектур, но со временем GCC и его ответвления развились в общий компилятор для самых разных языков и платформ.

Написание компилятора — непростая задача, и с самого начала Столлмен надеялся найти готовый компилятор, способный облегчить работу по созданию полностью свободной ОС. Вот его слова в «Открытый Код: Голоса революции открытого кода [Open Sources: Voices from the Open Source Revolution]»: «Незадолго до начала проекта GNU я услышал о Free University Compiler Kit, известном также как VUCK (голландское слово, означающее „свободный“, начинается с буквы V)». И Столлмен решил написать его автору, Энди Таненбауму (Andy Tanenbaum), который, по словам Столлмена, «ответил иронически, сказав, что университет-то свободный, а вот компилятор — нет. Тогда я решил, что моя первая программа для проекта GNU будет многоязычным многоплатформенным компилятором». С этой целью Столлмен поставил перед собой

тяжкую задачу написания интерфейса C для свободного компилятора Pastel, многоплатформенного компилятора, разработанного в Ливерморской лаборатории [Lawrence Livermore Lab]. Pastel был производным от языка Pascal.

Столлмен начал портировать Pastel на компьютер Motorola 68000. Все было неплохо, пока он не наткнулся на проблему: «Компилятору нужен был стек на много мегабайт, а имеющаяся система 68000 Unix могла предоставить только 64К», вспоминает Столлмен.

Он рассказывает: «Компилятор Pastel работал посредством разбора всего файла ввода в дерево синтаксиса, превращая все дерево синтаксиса в цепь „инструкций“ и затем генерируя выходной

Pastel; но тогда мы не сможем компилировать Pastel, поэтому все нужно полностью переписать на C», сказал Столлмен.

Столлмен адаптировал и заново использовал интерфейс C, который он написал для компилятора Pastel, и реализовал идею использования RTL (Register Transfer Language) в качестве средства хранения промежуточных представлений транслированного кода, внедрив «некоторые идеи по оптимизации» от Portable Optimizer, написанного в Университете Аризоны Джеком Дэвидсоном [Jack Davidson] и Кристофером Фрейзером [Christopher Fraser]. Пол Рубин [Paul Rubin] написал большую часть предпроцессора и, по словам Столлмена, Леонард Тауэр написал части парсера, генератора RTL, определений RTL и описания машины VAX (Internals of GNU CC, <http://bit.ly/1eSCE56>).

Майкл Тиманн [Michael Tiemann], которому в грядущее десятилетие предстояло стать самым значимым участником проекта GCC, сказал, что день выхода первого релиза GCC был «самым волнующим и жутким днем моей жизни (до того момента)... мне нужно было принять решение. Я мог присоединиться [к Столлмену], мог начать конкурировать с ним или начать мечтать заново. Я скачал GCC версии 1.0 и начал сотрудничество, которое продлилось десять лет» (Open Source Initiative, <http://bit.ly/1gKatC2>).

**«Свободный портируемый компилятор привел свободное ПО туда, где прежде его не бывало.»**

файл без освобождения памяти», что привело его к выводу о необходимости писать новый компилятор с нуля.

В феврале 1986 года в выпуске GNU's Bulletin он написал: «Проблема в том, что большая часть компилятора написана на Pastel» и в том, что он назвал «супер-лохматым расширенным Pascal» — единственном компиляторе для этого языка. «Чтобы сделать его меньше, нам надо устранить лохмы, необходимые для компиляции

Через год-два GCC начал изменять сцену для компиляторов, и свободный портируемый компилятор привел свободное ПО туда, где прежде его никогда не было. В 1980-е компиляторы были дорогим удовольствием. GCC был быстр, доступен и расширяем, и стал прорывом для свободного ПО — равно полезный для производителей чипов и устройств и для программистов, которые писали программы, предназначенные для портирования в проприетарные реализации UNIX или иных платформ. Универсальный компилятор упрощал решение проблем и обеспечивал совместимость и портируемость для широкого ряда платформ.

Создатели проприетарных компиляторов были склонны добавлять «функции языка», которые создавали неявные шероховатости и разноречивость на разных машинах. GCC предлагал простой способ отладки программ, поскольку работал в широком диапазоне ОС, процессоров и устройств.

GCC не всегда отвечал стандартам ANSI, хотя в последнее время это изменилось. Как и в большинстве коммерческих компиляторов, часть расширений была вписана в более ранние версии компилятора GCC. Разница была в том, что GCC портировался на многие архитектуры, и для коммерческих кодеров это упрощало портирование их программ: они могли рассчитывать получить тот же результат на разных платформах. В последние годы разработчики GCC были более тверды в поддержке стандартов языков программирования, даже если это приводило к трениям среди разработчиков — прежде всего, с Линусом Торвалдсом [Linus Torvalds].

Торвалдс заявил: «По какой-то причине разработчики компиляторов, похоже, слишком оторвались от „реальной жизни“, развив тенденцию говорить: „так требуют спецификации“ вместо того, чтобы сказать: „вот какая есть проблема“» (источник: <http://bit.ly/1b3ayk0>).

Достоинство спецификаций в том, что неожиданные расширения языка не дают неожиданных результатов, и «свободный» компилятор не поощряет проприетарных расширений.



Создатель GCC Ричард Столлмен сидит на дереве со своим нетбуком Lemote.

Побочным эффектом раннего успеха GCC как универсального компилятора было то, что он стал важным средством распространения идеи свободного ПО. Программистов и разработчиков соблазняло его удобство и доступность, и к концу 1980-х инструменты GCC и GNU постоянно прописались в арсенале программистов UNIX. GCC дал возможность Линусу Торвалдсу скомпилировать ранние версии Linux на его 486, а Linux'у — стать переносимым на разные архитектуры.

К 1989 году GCC стал двигателем первых коммерческих предложений свободного ПО,

для свободного ПО. В *Open Sources: Voices from the Open Source Revolution* Тиманн писал: «Основная идея была проста: открытый код объединит усилия программистов по всему миру, и компании, предоставляющие коммерческие услуги (индивидуальную настройку, расширения и поддержку) на базе этого ПО, составят капитал на масштабной экономике этого нового вида ПО».

«Сначала я пытался приводить свои доводы так же, как Столлмен приводил свои: говоря о достоинствах. Я объяснял, как свобода делиться может привести к потрясающим инновациям при более низких расходах, более масштабный эффект через более открытые стандарты, а люди повсеместно отвечали: „Отличная идея, но этому не бывать, потому что никто не станет платить за свободное ПО“». Два

года путешествовав по всему миру вестником этой идеи, Тиманн заключил: «Если всем кажется, что идея отличная — вероятно, так оно и есть, а если никто не верит, что она работает — что ж, у меня не будет конкурентов!»

По словам Тиманна, «Манифест GNU — весьма рациональный документ. Он анализирует природу ПО, природу программирования, великую традицию академического образования, и подытоживает, что независимо

## «GCC стал двигателем первых коммерческих предложений свободного ПО.»

посредством Cygnus Solutions, основанного в 1989 году Джоном Гилмором [John Gilmore] и Майклом Тиманном. На момент появления первого стабильного релиза GCC, Тиманн создал GNU C++ Compiler и портировал GCC на другие архитектуры.

### Успех Cygnus

Cygnus был первой компанией, доказавшей возможность жизнеспособного маркетинга

## Простая яичница

Пусть GCC не всегда был гладким. К 1997 году Free Software Foundation назначил куратора GCC. Но, как отметил разработчик Cygnus Ди-Ви Хенкель-Уоллес [D V Henkel-Wallace], начались жалобы от «Cygnus, от ребят из Linux, от ребят из PGCC, Fortran и многих других», что их труды над GCC «не вошли в официальное древо GCC, несмотря на годы, потраченные на его реализацию».

Проблема достигла критической точки, когда Хенкель-Уоллес объявил по электронной почте

(источник: <http://bit.ly/1aFYn0Y>), что «некоторые из нас (включая программистов Fortran, Linux, Intel и RTEMS) решили начать более экспериментальный проект разработки...Только на сей раз сообщество, работающее с нами, куда больше! Мы назвали этот проект EGCS (произносится как eggs — англ. «яйца») или Experimental/Enhanced GNU Compiler System.

Ответвление EGCS оказалось успешным и собрало все разрозненные элементы, на которые

не обратили внимания специалисты GCC. EGCS обгонял официальную ветку, и к началу 1999 года примирение двух ответвлений стало неизбежным: «FSF согласился перейти на код EGCS для GCC, а проект EGCS согласился раствориться, перейдя в исходный проект GCC».

Воссоединение завершилась с выходом GCC 2.95 в июле 1999 г., когда GNU C Compiler был переименован в GNU Compiler Collection, и проект принял подход EGCS, пересмотренный оргкомитетом.

от финансовых последствий, существует моральный и этический императив бесплатно и свободно делиться информацией, которой, в свою очередь, бесплатно и свободно поделились с вами. Я пришел к другому выводу — мы часто спорили о нем со Столлменом: он заключался в том, что свобода использовать, распространять и модифицировать ПО будет превалировать над любой моделью, стремящейся ограничить эту свободу. Но не по этическим соображениям, а по соображениям конкуренции и законов рынка».

«Осознав, как непривлекательна позиция конкуренции с ПО с открытым кодом, — заключает Тиманн, — я понял, что время этой идеи настало».

Успех Cygnus основан на GCC и отладчике GNU, и среди его клиентов — Intel, Sony, AMD, 3Com и Adobe. GCC упростил задачу компиляции кода для новых языков и архитектур, и компании стали обращаться к Cygnus,



► Майкл Тиманн: «Осознав, как непривлекательна позиция конкуренции с ПО с открытым кодом, я понял, что время этой идеи настало».

заказывая ему работу. Все дополнения к коду были отданы на усмотрение Free Software Foundation, но большая часть основных расширений GCC в течение 1990-х были плодом труда разработчиков Cygnus.

Постепенно Cygnus стал значимым провайдером поддержки инструментов разработчика свободного ПО во многих областях индустрии, и обеспечивал компании открытого кода прототипами, которым они и следовали. Cygnus стал первым серьезным приобретением Red Hat после их IPO в 2000. Джон Гилмор позднее сказал: «Наконец мы стали получать миллионные контракты — например, контракт от Sony на создание компиляторов и эмуляторов PlayStation. Это позволило разработчикам игр начать работу за год до появления оборудования PlayStation».

Вот иллюстрация широты и повсеместного распространения GCC к 1999 году: набор инструментов Cygnus GNUPro включал поддержку GCC для более 175 сочетаний хостов и адресатов (источник: <http://bit.ly/L50pxK>).

## Наша цель — непрозрачность

Ради простоты можно сказать, что компилятор GCC состоит из трех частей: «входной [front-end]», которая содержит код для компиляции определенного языка, «средней [middle-end]», или фазы оптимизации кода, и «выходной [back-end]», которая передает скомпилированный код на конкретную архитектуру. Между этими «стадиями» передаваемый код хранится в виде «промежуточного представления [intermediate representation, IR]», которое на разных стадиях именуется Generic, Gimple и RTL. GCC включает разные «входы» (для каждого языка) и «выходы» (для каждой архитектуры), и состоит из более чем двух миллионов строк кода.

Иногда утверждают, что код GCC принципиально избегает ясности (и промежуточная семантика непрозрачна), и это сработало против великих целей разработки GCC. В идеале код

должен быть чистым и модульным, но имеет тенденцию быть непрозрачным. Процесс отделения «входов» и «выходов» от остальной части кода не так прост, как мог бы быть, и это означает, что путь наименьшего сопротивления для тех, кто хочет обогатить GCC, добавив другой язык или расширение для конкретной машины, заключается в выпуске своих изменений под GPL, что, в свою очередь, сделает языки и архитектуры более доступной для других свободных программ.

Переносимость GCC улучшила масштабируемость и переносимость ядра Linux, а относительная непрозрачность GCC сделала его неуязвимым и крайне затруднила задачу превращения его в проприетарный. За долгие годы тенденция создателей компилятора и производителей чипов следовать по «пути наименьшего сопротивления» сильно способствовала универсальной совместимости GCC на благо пользователей и разработчиков.

Джо Бак [Joe Buck], участник проекта GCC, написал (источник: <http://bit.ly/1aojDVo>): Столлмен «полагает, что люди, которые расширяют GCC и надеются сделать свои расширения проприетарными, обнаруживают, что это им не удастся, и тогда решают внести свой вклад в код... иначе они не смогут распространять его, и им придется вечно поддерживать его самим или пойти на риск нарушения закона. У RMS [Столлмена] есть свидетельства того, что это иногда случается (C++, Objective-C, многие предложенные back-end'ы). Так что намерение заключается не в том, чтобы помешать людям его использовать, а в том, чтобы эти люди вносили сделанные ими изменения как всеобщее достояние, даже если им это и не нравится».

GCC — флагман движения свободного ПО, а цель данного движения — сделать программы свободными. Конечно, это устраивает не всех. Код компилятора может раскрыть секреты производителей и открыть архитектурные инновации гипотетическим конкурентам, хотя нет никакого обязательного

## Cygnus: Коммерциализация фактора GNU

Cygnus был основан Майклом Тиманном, Дэвидом Уоллесом и Джоном Гилмором. Когда вышел GCC, Тиманну было 22, и он мечтал написать «Великий американский компилятор»: «Мое восхищение Столлменом и проектом GNU росло по мере того, как я начал понимать глубину его технических задумок, красоту его подходов к реализации и то, как это все было защищено для меня, а не от меня с помощью GPL» (источник: Open Source Initiative, <http://bit.ly/1gKAtC2>).

Скачав GCC, Тиманн решил портировать его в National Semiconductor 32032, и он отметил:

«Когда я принялся за эту работу, обычная компания выставила бы счет в миллионы долларов за услугу поставки компилятора в течение 12–18 месяцев, а иногда и больше». В течение нескольких недель Тиманн создавал в GCC код, работающий на 40% быстрее, чем созданный в проприетарном компиляторе этой фирмы.

«В то лето я писал множество других портов, — написал он, — демонстрируя с завидным постоянством, что, программируя на GNU, я могу постоянно создавать лучший продукт, и быстрее, чем целые команды создателей компиляторов, работающие в крупных компаниях. К концу того

года я достиг новой цели, которая заключалась в написании компилятора GNU C++. Для меня GNU C++ был Великим Американским компилятором C++: новой и улучшенной версией моей изначальной цели».

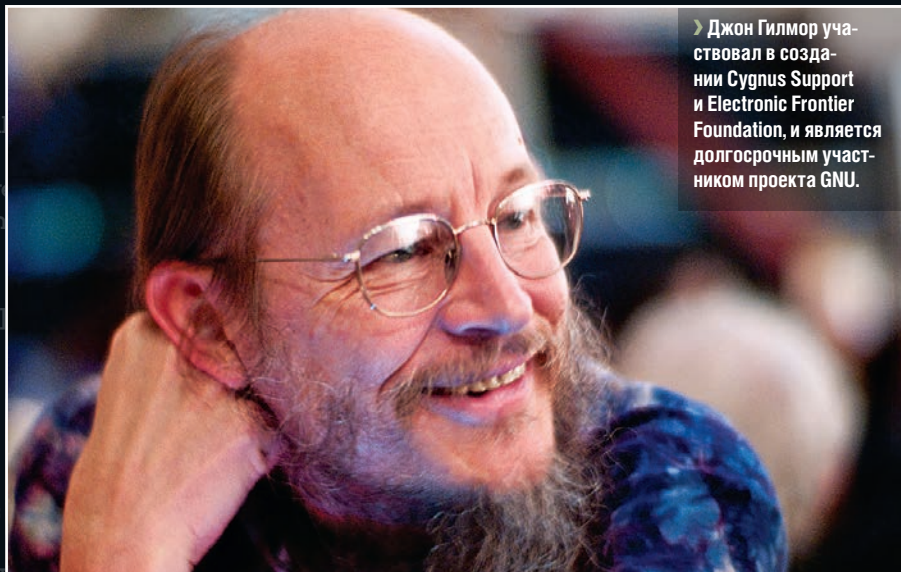
Тиманн продолжил писать и организовывать «первое всестороннее обучение по внутренним компонентам компилятора GNU, передавая свои 100+ слайдов LaTeX Столлмену», и был организован Cygnus Support. «Два года спустя я официально коммерциализировал коммерческие преимущества программ GNU, а остальное, как говорится, история».

требования выпускать код под GPL до выхода чипа, и закрытые back-end'ы ограничивают возможности экспериментировать и исследовать для программистов. И наоборот, для некоторых производителей выпуск кода под GPL сработал в их интересах, что иллюстрирует использование GCC в Sony для поощрения создателей игр к их написанию для ранних версий PlayStation.

Трудности использования кода, не соответствующего GPL, могут быть случайным побочным продуктом ранней разработки GCC, а не сознательным конструктивным решением, но, как говорит Бак: «RMS скажет вам, что у нас есть компилятор GNU C++ только потому, что работодатель Майка Тиманна не смог сделать его проприетарным, и у нас есть компилятор GNU Objective-C, потому что Стив Джобс [Steve Jobs] не смог сделать его проприетарным. Будь в то время эквивалент форматов dump, у нас был бы только C. (Конечно, это обратная проблема: проприетарные входы, соединенные с выходами GNU). Превращая всего лишь наработки для GCC и выпуск кода под GPL в путь наименьшего сопротивления, RMS надеется мотивировать еще больше людей создавать свободное ПО». Однако Бак признает некую оговорку: «все это хорошо, но когда дойдет до защиты непрозрачной, нерасширяемой архитектуры, у нас будут проблемы» (источник: <http://bit.ly/1d6BqQV>).

Возможно, усложнение процесса использования проприетарных дополнений и было полезно в те времена, когда свободное ПО считалось противоположностью бизнес-практике, но времена изменились, и производители программ и оборудования меньше сопротивляются открытости своего кода.

Насчет этого разработчики GCC в последнее время сделали радикальный шаг, разрешив



Джон Гилмор участвовал в создании Cygnus Support и Electronic Frontier Foundation, и является долгосрочным участником проекта GNU.

плагины к коду. Плагины могут выпускаться под другими лицензиями и предлагать сторонним разработчикам подобие модульности.

Здесь на сцену выходит LLVM. LLVM считает себя альтернативой GCC с открытым кодом, и о его выходе было объявлено в виде документа под названием «Архитектура для GCC следующего поколения [Architecture for a Next-

Аргумент и за, и против LLVM заключается в том, что он легче разрешает третьим сторонам сохранить контроль за своим кодом, и его структура не стеснена ограничениями лицензии.

Безусловно, LLVM доступнее и проще для встраивания языкового компилятора, чем GCC, и, вероятно, по этой причине LLVM стал так популярен среди разработчиков более модных языков программирования, таких, как Ruby, Python, Haskell, Pure и Lua. Модульность и упрощение открыли дорогу другим достижениям, таким, как опция JIT (just in time — строго вовремя) или инлайн-компиляция, и облегчили для компаний вроде Apple, которые используют LLVM для создания самых последних версий OS X, включение проприетарных языковых расширений или встраивание LLVM в проприетарную среду программирования, если они того пожелают.

Появление LLVM дало Apple возможность рассмотреть альтернативы, благодаря чему им удалось добиться большей гибкости и контроля. Для LLVM это и хорошая, и плохая новость. Здоровая конкуренция между GCC и LLVM улучшила возможности для всех, и она держит разработчиков GCC в форме, но в конечном итоге у GCC есть преимущество высочайшей переносимости и достойная история, а также присмотр оргкомитета, собранного из намного более широкого спектра промышленности.

Появление LLVM дало Apple возможность рассмотреть альтернативы, благодаря чему им удалось добиться большей гибкости и контроля. Для LLVM это и хорошая, и плохая новость. Здоровая конкуренция между GCC и LLVM улучшила возможности для всех, и она держит разработчиков GCC в форме, но в конечном итоге у GCC есть преимущество высочайшей переносимости и достойная история, а также присмотр оргкомитета, собранного из намного более широкого спектра промышленности.

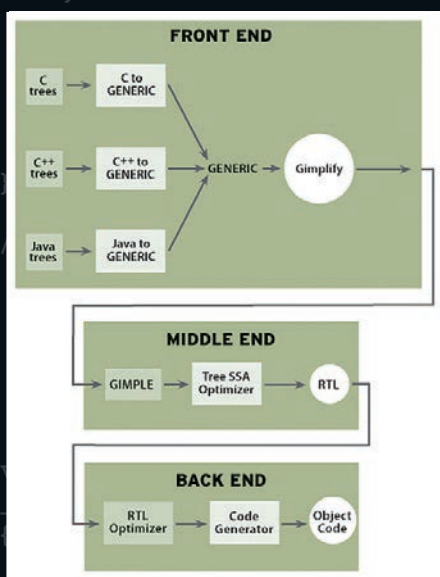
GCC независим от любых коммерческих единиц, и это обеспечивает его приверженность стандартам, ведя за собой свободу от споров и волнений — мнимых или реальных — по поводу патентов и проприетарного контроля за языками программирования, которые являются самым важным инструментом свободы самовыражения программиста. **LXF**

## «LLVM считался модульной, более прозрачной версией GCC.»

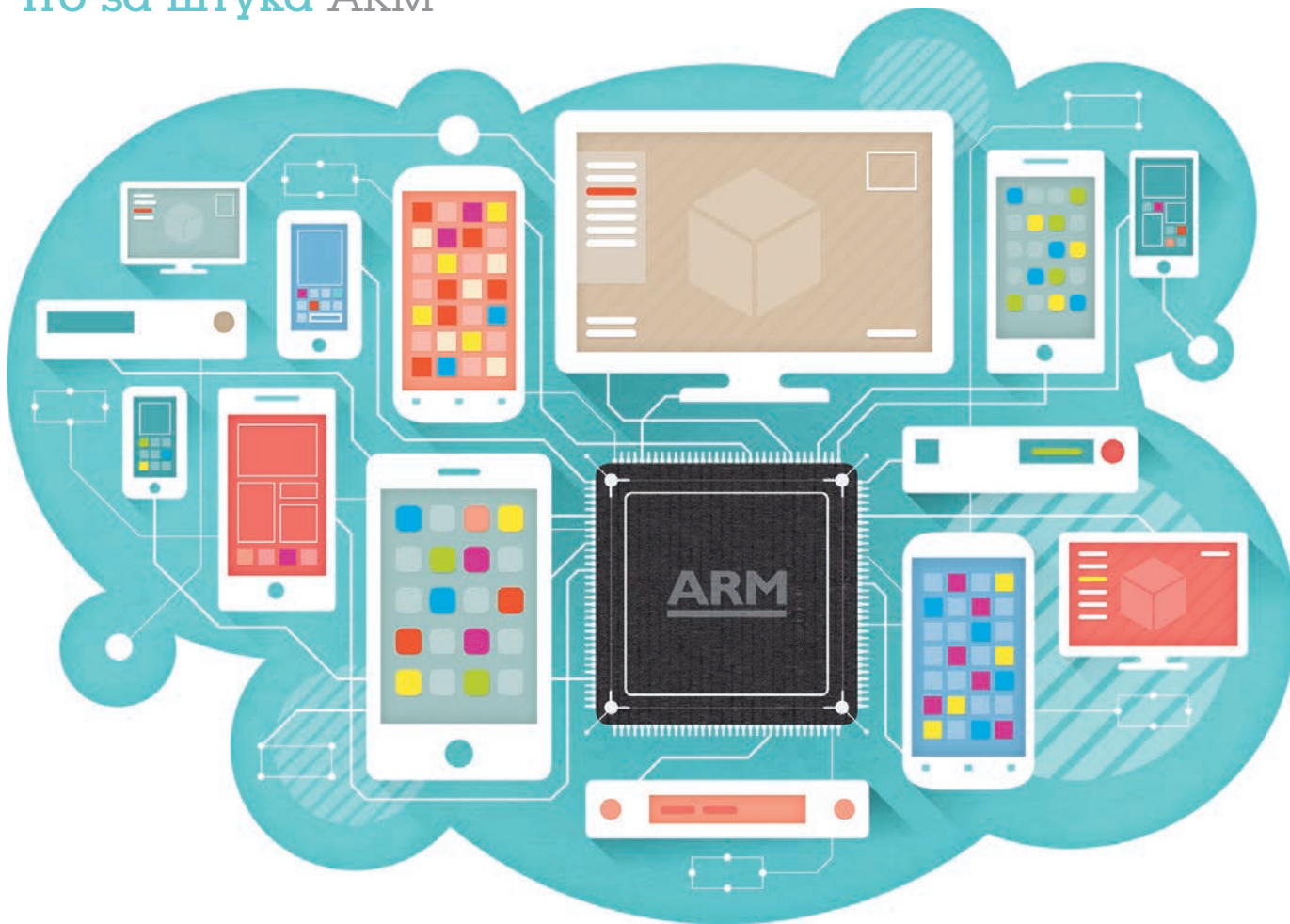
Generation GCC]», который был представлен на Ежегодном саммите разработчиков GCC [Annual GCC Developers Summit] в 2003 году Крисом Латтнером [Chris Lattner] и Викрамом Адве [Vikram Adve]. LLVM считался модульной, более прозрачной версией GCC, и в результате достиг успеха.

### Яблоки и апельсины

LLVM выходит под лицензией BSD, и он заменил GCC в качестве предпочитаемого компилятора в операционных системах Apple и BSD. Сторонники BSD утверждают, что лицензия BSD более чистая и свободная, поскольку она не налагает на конечного пользователя никаких ограничений (кроме требования атрибуции). С точки зрения перспективы свободного ПО, основное различие между лицензиями в том, что лицензия BSD разрешает использовать LLVM в проприетарных продуктах и не создает обязательства обнародовать изменения. Код GPL, а также любые изменения или расширения к этому коду должны быть выпущены как свободное ПО, и в результате еще больший объем программ остается свободным для последующих поколений разработчиков свободного ПО.



Базовая структура компилятора GCC: front-end, middle-end и back-end.



# Что за штука... ARM?

Ежедневно продают более 16 миллионов этих крошечных микропроцессоров, теперь уже и в 64-битном формате. **Мэтт Хэнсон** толкует о будущем ARM...

**В** Ну теперь-то я не полный ноль в вопросе. Я слышал о процессорах ARM, они не новинка, так ведь?

**О** Можно сказать, что и так. ARM стал чем-то вроде национального достояния Британии, с тех пор как в 1980-х впервые появился в качестве сопроцессора в BBC Micro, нежно памятном до сих пор. ARM — один из главных технологических прорывов Соединенного Королевства.

**В** То есть процессоры ARM довольно популярны?

**О** Более чем, и в недавнем времени они получили вторую жизнь благодаря популярности небольших устройств, требующих большой производительности. От смартфонов и электронных читалок до роутеров и Smart TV. На данный момент, в мире продано более 30 миллиардов процессоров на основе архитектуры RISC.

**В** Э, придержите коней. Что еще за архитектура RISC такая?

**О** RISC — это сокращение от Reduced Instruction Set Computer, и эта архитектура создана специально для микропроцессоров с более узким набором (или же более простых по структуре) команд. Суть в том, что поскольку их меньше, процессор становится более эффективным. Что, в свою очередь, позволяет оптимизировать его работу и сами команды для еще большего улучшения производительности.

Это означает, что процессор на базе RISC может выполнять в миллионы раз больше команд в секунду, чем машина на базе CISC (Complex Instruction Set Computer). Поскольку для каждого типа команд требуются дополнительные транзисторы и схемы, а процессоры RISC — меньше, проще, и, как следствие, дешевле для производства. Что, опять же, идеально подходит для небольших специализированных устройств.

**В** Звучит более чем разумно. Кому же пришла в голову такая светлая мысль — создать RISC?

**О** Термин RISC принадлежит Дэвиду Паттерсону [David Patterson], преподавателю Калифорнийского университета, хотя само представление существовало до этого. Его создателем часто называют Джона Кока [John Cocke], исследователя от IBM из Нью-Йорка. Именно он, в 1974 году, открыл, что 80% работы компьютера выполняется с помощью примерно 20% команд. В 1980, создатели IBM PC/XT учли эту идею Кока, а еще позднее, в IBM RISC System/6000 довели ее до совершенства. А сейчас архитектура RISC используется в старых добрых процессорах ARM.

**В** А почему именно сейчас за ARM так ухватились?

**О** Помимо того, что на них основана куча устройств у вас дома, во многих из которых используется ОС Linux, теперь процессоры ARM появились и в 64-битном формате.

**В** Ну, давно пора бы. Но что же конкретно это нам дает?

**О** Ну, раньше процессоры ARM были только 32-битными — в прошлом нам этого хватало, но сейчас мы становимся более требовательными к тому, что умеют наши телефоны. Переход от 32-битной к 64-битной архитектуре дает целый ряд преимуществ — главным образом, возможность иметь более 4 ГБ ОЗУ.

**В** Разве 4 ГБ ОЗУ для мобильного устройства — это не за глаза? Да это больше, чем у моего — признаться, старенького — настольного ПК!

**О** Пару лет назад я бы с вами согласился, но времена меняются. Чем более многофункциональными становятся наши смартфоны, чем больше мы хотим от наших роутеров и телеприставок, тем больше ОЗУ им требуется. Сегодня 4 ГБ ОЗУ в телефоне кажутся приветом из отдаленного будущего, где памяти произвольного доступа будет хоть отбавляй, но Sony и Samsung намерены выпустить устройства с таким объемом уже в этом году. Разрыв между устройствами в наших карманах и настольными ПК все сокращается. Уже существует процессор Apple A7, также на основе 64-битной архитектуры ARM, для iPhone 5S и последнего поколения iPad'ов.

**В** А у меня все еще монохромная Nokia 5110, которая по-прежнему умеет играть в эту странную Змейку и старорежимно звонить по телефону. И что, меня порадует переход ARM на 64 бита?

**О** Почти наверняка. Как уже было упомянуто, у вас дома есть куча устройств с процессорами ARM. Большим достижением является появление ARM в среде серверов. Маломощные, экономичные серверы становятся предметом явного интереса серверных центров, которым переход на ARM позволил бы сократить энергозатраты, уже без ограничений на память, присущим 32-битной версии.

**В** То есть нас ждет появление 64-битных серверных процессоров ARM?

**О** Бьюсь об заклад. Applied Micro уже предлагают средства для разработки под их новый 64-битный чип ARM X-Genie. И хотя стоит такой комплект \$5000, они уже пользуются популярностью. Calxeda, AMD и прочие готовятся выпустить собственные 64-битные серверные процессоры ARM либо в этом году, либо в начале 2015.

**В** А кроме Apple A7, есть другие изготовители 64-битных чипов? Этот-то огород я предпочитаю обходить стороной.

**О** Да, многие производители уже принялись за дело. Недавно в Qualcomm объявили о выходе их первого 64-битного ARMv8 SoC (System on a Chip) — Snapdragon 410. Предназначен он для популярных мобильных устройств, и в нем будет 4 ядра Cortex A53 ARM, с частотой около 1,2 ГГц. В пресс-релизе Qualcomm говорится, что «Чипсет Snapdragon 410 станет первым из целого серии 64-битных процессоров».

**В** Такое впечатление, что Qualcomm взялись за 64-битные ARM всерьез.

**О** Верно, но так было не всегда. Когда в Apple объявили о выходе процессора A7, один из исполнительных директоров Qualcomm сказал, что 64-битная архитектура — всего лишь рекламный трюк. Позднее, однако, последовало официальное опровержение его слов.

**В** Ого... значит, говорите, в Snapdragon 410 будет процессор Cortex-A53?

**О** Вот именно. Это процессор ARMv8, главные свойства которого — эффективность и способность поддерживать команды как 32-битные, так и 64-битные; и вследствие его малых габаритов и низкого энергопотребления мы предполагаем увидеть его в бюджетных смартфонах. Что вовсе не говорит о его низкой производительности. Перейдя на 64-битную архитектуру, он значительно превзошел своего предшественника, Cor-

## «64-битные ARM-процессоры могут открыть дивный новый мир компактных и дешевых ПК.»

tex-A7. Он требует меньшей мощности, меньше по размеру, но его производительность выше, чем и у Cortex-A9. Так что это идеальное решение не только для простейших смартфонов и прочих мобильных устройств, но и, благодаря высокой масштабируемости, для корпоративных систем, в конфигурации 'big.LITTLE'.

**В** Минуточку, разве 'big.LITTLE' — не тот довольно дебильный комедийный дуэт из конца 1980-х?

**О** Нет, те назывались Little and Large. Big.LITTLE — нечто другое. Это технология оптимизации энергопотребления, разработанная ARM и сочетающая энергоемкость одних чипов ARM, например, Cortex-A53, и высокую производительность других, таких как Cortex-A57.

**В** И в чем преимущества процессоров ARM с поддержкой big.LITTLE?

**О** Во многом, это объединение лучших качеств того и другого. Вы получаете высокую как усредненную, так и максимальную производительность и повышение эффективности параллельных процессов при существенном снижении энергопотребления. Технология big.LITTLE рационально распределяет нагрузку между процессорами в зависимости от выполняемой задачи. Возьмем, к примеру, смартфон с поддержкой big.LITTLE в процессоре ARM. Изначально при загрузке сайта или запуске игры нагрузка переходит на более мощный процессор, заставляя его тянуть более сложные задачи. А для простого пролистывания web-страниц, отправки SMS или выполнения других несложных задач будет использоваться другой, менее мощный, но более энергоемкий

процессор. Смена происходит автоматически и за микросекунды — пользователь и не заметит. По данным ARM, новая конфигурация big.LITTLE позволит сократить энергопотребление процессора на 75% и увеличить производительность на 40% при большой нагрузке. Короче, вы получаете мощный смартфон, у которого батарея не садится в одночасье.

**В** Выходит, это хороший пример того, когда правая рука знает, что делает левая?

**О** Полагаю, что можно и так сказать, коли вам угодно.

**В** Ну, вы просто мистер хорошие новости. И я могу прямо сейчас пойти и купить 64-битное устройство на Android?

**О** Пока нет. Последняя версия Android, KitKat, еще 32-битная, а значит, и приложения для Android пока не получают всех преимуществ

64 бит — пока. Поскольку в основе Android лежит Linux, который поддерживает 64-битную архитектуру вот уже 12 лет, появление 64-битного Android уже не за горами. Особенно если труды Samsung и Sony OS увенчаются успехом в этом году.

**В** Значит, процессоры ARM предназначены только для мобильных устройств и серверов?

**О** Совсем нет. Один из главных плюсов 64-битных ARM в том, что еще больше устройств смогут перейти на полноценный Linux, с поддержкой самых современных приложений. Новые 64-битные ARM процессоры могут открыть нам дорогу в дивный новый мир компактных и дешевых ПК на этих крошечных чипах, которые съедят все те программы, которые вы используете ежедневно. Речь заходит даже о создании процессоров Intel на базе ARM, хотя с этим придется-таки повозиться.

**В** Простите, я не ослышался? Intel будет разрабатывать чипы ARM?

**О** Правильно. Это уже дело решенное. В конце прошлого года один из крупнейших партнеров Intel, Altera, объявил о том, что в новом Stratix 10 SoC будет использован 4-ядерный процессор ARM Cortex-A53 — и что Intel приступит к производству Stratix 10 в этом году.

**В** Ух ты, значит, даже бывшие соперники вроде Intel переходят на ARM! Похоже, за этими процессорами будущее.

**О** Так и есть, и приятно видеть такие успехи британской компании в области технологии, гордиться сознанием их лидерства. **LXF**



# По рецептам доктора Брауна

## Д-р Крис Браун

Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

## Эзотерическое системное администрирование из причудливых заворотов кишок серверной

### Перебор выбора?

**П**ойдем в магазин. Возьмем немного хлеба и сыра. Вот хлеб — багеты, батоны, булочки, рогалики, хлеб из непросеянной муки, с пшеничными зародышами, сдобные булочки, булочки с корочкой, что-то странное с сыром и нарезанные батоны... А вот прилавок с сыром — здесь чеддер, эдамский сыр, гауда, бри, стилтон, пор-салью, маскарпоне, четыре вида бебибел, эммонталь и уэнслидейл... разве не чудесно? Что?.. Вам пакетик чипсов?

Рядом с супермаркетом наш местный торговый центр Linux, Tux 'R' Us. Вот витрина с дистрибутивами. Есть Mint, Debian, Fedora, CentOS, Scientific, шесть вариантов Ubuntu, SUSE, Sabayon, Arch, Puppy, Crunchbang, Slackware, Knoppix... а вот дистрибутивы рабочих столов — Unity, KDE, несколько версий Gnome, Xfce, LXDE, Cinamon, Mate...

А что у нас с почтовыми клиентами? *Evolution*, *Thunderbird*, *Kmail*, *Claws mail*, *Mailx*, *Elm*, *Pine* и *Mutt*. Браузер? Как насчет *Firefox*, *Chrome*, *Sea Monkey*, *Epiphany*, *Konqueror* или *Opera*? Хотите текстовый редактор? Пожалуйста: *Nano*, *Gedit*, *Kedit*, *Kate*, *Emacs*, *Vi*... прекрасный ассортимент, не так ли?

Что? Вы уже затосковали по iPad?

Может ли выбор быть слишком большим? Боюсь, что чрезмерное обилие открытых программ мешает людям попробовать их. Они боятся, как боялся я, когда, придя в «Старбакс», в первый раз заказал чашку кофе вместо средней порции трехслойного диетического латте маккиато со льдом и карамелью.

[chris.linuxformat@gmail.com](mailto:chris.linuxformat@gmail.com)



## У вас есть время?

Сетевой протокол времени NTP — удобный и эффективный способ синхронизации часов компьютера.

**К**стати о времени: когда вы дома, вы сами можете выбрать, который сейчас час, в зависимости от того, подошли вы к микроволновке, плите или холодильнику. Синхронизировать эти часы можно только одним способом: нажав у каждого прибора очередное неопишное сочетание кнопок, выбранное разработчиком для установки времени, изменить время вручную.

Синхронизировать время на компьютерах гораздо проще благодаря NTP (Network Time Protocol) — сетевому протоколу времени. Этому протоколу около 30 лет, и он позволяет синхронизировать компьютеры по всему миру с точностью до 0,1 с или даже точнее, и с точностью до 1 мс

в локальной сети. В NTP используется многоуровневая иерархия серверов. В компьютерах верхнего уровня (слой 0) время определяется с помощью хитрых атомных часов; потом эти измерения просачиваются вниз через серверы на уровнях 1, 2 и 3.

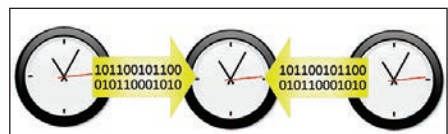
Конечно, простые смертные вроде нас с вами взаимодействуют только с нижними уровнями этой структуры. На самом деле, настроить свой компьютер в качестве клиента NTP нетрудно. Обычно для этого достаточно лишь установить пакет *ntp* и убедиться, что демон *ntpd* запустился. У *ntpd* есть файл настройки (*/etc/ntp.conf*), где нам важнее всего строки "server", которые говорят демону, где находятся серверы времени; впрочем, в вашем дистрибутиве в этих строках уже заданы подходящие значения по умолчанию. Будет здорово, если перед запуском демона вы более-менее точно выставите время (командой *date*). При расхождении во времени более 1000 секунд демон теряет ощущение реальности и сбдается.

Демон *ntpd* обычно работает постоянно, аккурратно синхронизируя компьютерное время. Но можно синхронизировать время и принудительно:

```
# ntpd -q
ntpd: time set +42.230535s
```

Некоторые предпочитают периодически запускать эту команду как задание *cron*, вместо того, чтобы демон работал постоянно.

Впрочем, доступность дешевых GPS-приемников, предоставляющих время с точностью до нескольких десятков наносекунд, позволяя каждому достичь точности сервера уровня 0, лишает подобные ухищрения смысла. »



► NTP позволяет синхронизировать время в Интернете и в локальной сети.

### Почему важно синхронизировать время?

Расхождение во времени между компьютерами может вызывать самые разнообразные проблемы. Программы вроде *make*, которые сравнивают временные отметки файлов, могут работать неверно, особенно если увидят время модификации файла, которое, по их мнению, находится в будущем.

В хронологии записей в лог-файле сложно разобраться, если эти записи оставляют компьютеры с рассинхронизированным временем. Некоторые протоколы аутентификации принимают логин и пароль только с отметкой в узком временном диапазоне для защиты от повторных атак.



# ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 13.10

## А ТАКЖЕ

версии для юриди-  
ческих лиц —  
с лицензионным  
договором  
присоединения

USB Flash 8 Gb

**495** рублей

[www.linuxcenter.ru/shop/distros/usb/](http://www.linuxcenter.ru/shop/distros/usb/)

Москва  
+7 (499)

**271-49-54**

Санкт-Петербург  
+7 (812)

**309-06-86**

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)

Linux  center

# Безопасная оболочка

Идеально для агорафобных лежебок — присоединяйтесь к Доктору в исследовании удивительного мира, скрытого за портом 22.

**В**озможно, вы уже читали об SSH в этой рубрике. Я подробно писал о ней в **LXF121**, и многие другие писали о ней на страницах нашего журнала. Но это важная утилита, и я не буду извиняться за то, что вернусь к ней в этом месяце.

SSH — сокращение от 'secure shell [безопасная оболочка]', но это не совсем точное название. SSH — на самом деле не оболочка (как, например, *Bash*). Скорее это средство безопасного подключения к удаленному компьютеру. Чаще всего SSH используется для подключения к настоящей оболочке на этом компьютере, причем выглядит все так, как будто вы работаете в оболочке за этим компьютером. Но, как мы убедимся, с ее помощью можно делать и другие вещи. Самый простой способ использования SSH выглядит так:

```
[chris@ubuntu1204 ~]$ ssh centos64
chris@centos64's password:
Last login: Mon Dec 2 06:35:29 2013 from ubuntu1204.home
[chris@centos64 ~]$
```

Здесь первая оболочка — оболочка на моем компьютере (ubuntu1204), а вторая — оболочка на удаленном компьютере (centos64). Предполагается, что имя 'centos64' является разрешимым (например, указано в файле */etc/hosts*). Если нет, прекрасно подойдет IP-адрес. По умолчанию предполагается, что вы хотите подключиться к компьютеру под тем же именем пользователя, что и на локальном компьютере; в примере выше это 'chris'. Если нет, можно сделать так:

```
[robert@ubuntu1204 ~]$ ssh bob@centos64
```

С помощью SSH можно выполнить на сервере одну произвольную команду. Например, создать каталог:

```
$ ssh centos64 mkdir /home/chris/demo
```

Для пользователя все это выглядит довольно просто, но внутри происходит сложная криптографическая работа. Во-первых, сервер безопасно аутентифицирует себя клиенту, чтобы мы были уверены, что подключаемся именно к тому компьютеру, к которому хотим. Во-вторых, генерируется случайный «ключ сессии», и компьютеры безопасно обмениваются им друг с другом. Весь трафик между клиентом и сервером шифруется этим ключом. Для этой первоначальной установки связи требуется, чтобы на сервере была сгенерирована пара из открытого и закрытого ключа (обычно это происходит при установке демона *ssh*) и чтобы у клиента была копия открытого ключа.

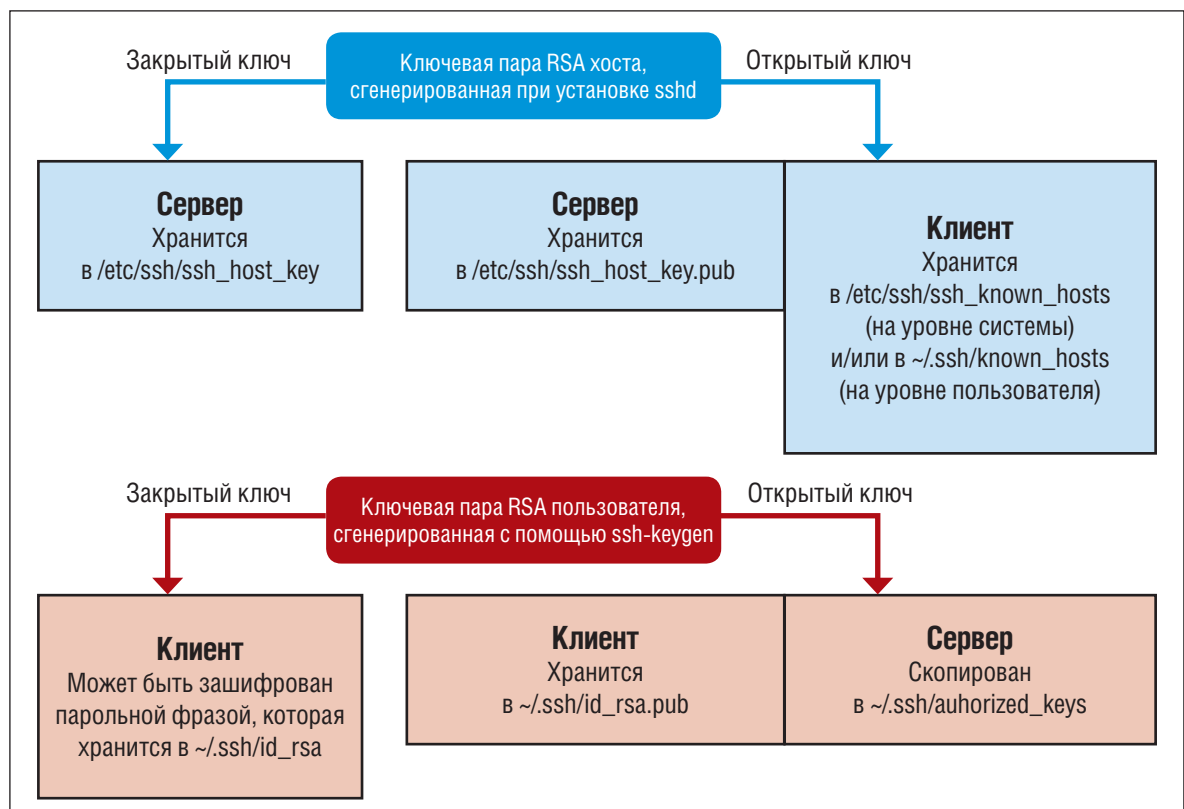
При первом подключении к компьютеру по SSH у вашего клиента может не быть открытого ключа сервера. Сервер предоставляет ключ клиенту, но клиент ему не доверяет, так что в этот момент вы видите довольно жуткое сообщение:

```
$ ssh centos64
Аутентичность хоста 'centos64 (192.168.1.79)' не установлена.
RSA key fingerprint is f2:97:9f:44:e3:d5:41:0f:6f:1e:49:76:df:1a:15:f6.
Продолжать соединение (yes/no)?
```

Будь вы полным параноиком, вы потребовали бы личной встречи с администратором сервера и попросили бы его подтвердить цифровую подпись ключа сервера. Администратор может просмотреть цифровую подпись, зайдя на сервер и выполнив команду

```
$ ssh-keygen -lf /etc/ssh/ssh_host_rsa_key.pub
```

Затем администратор прочтет вам цифровую подпись, и вы сравните ее с той, которую сообщил клиент SSH. Часто ли



➤ В SSH для аутентификации сервера клиенту и клиента серверу используются пары закрытых и открытых ключей. Все просто!

этот идеализированный сценарий воплощается в жизнь? Наверное, никогда. Обычно вы просто набираете 'yes' в ответ на страшное сообщение, и клиент молча добавляет открытый ключ сервера в локальный файл (обычно `~/.ssh/known_hosts`), чтобы это сообщение больше не появлялось. Но помните, что в этом случае вы фактически говорите: «Я беру на себя ответственность по проверке аутентичности сервера», и в этот момент вы уязвимы для атаки типа «посредник [man-in-the-middle]».

Открытый ключ сервера в домашнем каталоге будет работать, разумеется, только для вашей учетной записи. Если в системе есть другие пользователи, у них должна быть своя копия ключа. В этой ситуации лучше всего переместить файл `known_hosts` в `/etc/ssh/ssh_known_hosts`, доступный всем пользователям.

Но SSH еще не выполнила аутентификацию. Ее следующая задача — аутентифицировать вас, пользователя, серверу. Это можно сделать разными способами, но мы рассмотрим всего два. Во-первых, можно использовать старую добрую схему с именем пользователя и паролем — с тем преимуществом, что они передаются по уже установленному защищенному каналу. Для свежее установленной конфигурации SSH это, наверное, естественный метод, который будет работать.

Но в SSH также можно аутентифицироваться с помощью открытых и закрытых ключей. Для настройки такой аутентификации придется немного поработать, и это тест на право быть системным администратором. Во-первых, сгенерируйте пару ключей на клиенте:

```
$ ssh-keygen -t rsa
```

Согласитесь с каталогом размещения ключа по умолчанию, если у вас нет причин менять его. Вас попросят ввести парольную фразу для защиты закрытого ключа, но пока мы оставим ее пустой. Это не самая безопасная опция, и вскоре я к ней вернусь. Теперь у нас есть закрытый ключ в файле `~/.ssh/id_rsa` и соответствующий ему открытый ключ в файле `~/.ssh/id_rsa.pub` (обратите внимание, что эти ключи привязаны к пользователю, а не к компьютеру, как мы уже сказали).

Затем открытый ключ копируется на сервер. При желании это можно сделать «вручную» — просто возьмите файл `~/.ssh/id_rsa`.

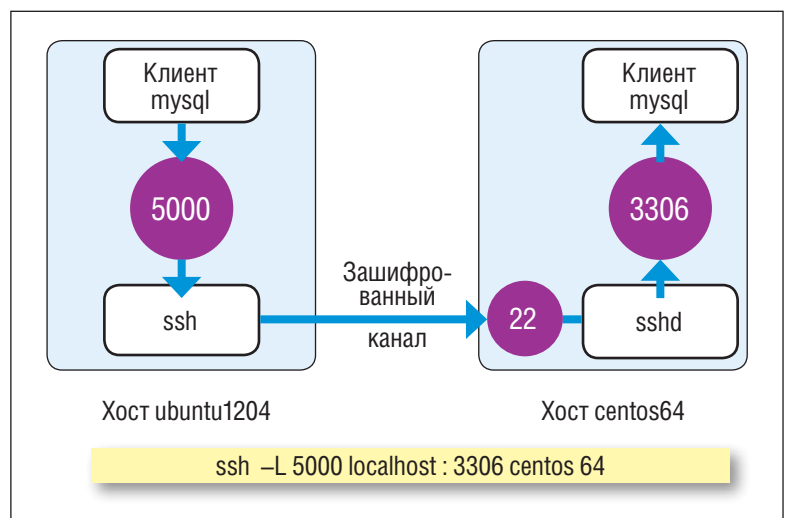
`pub` на клиенте и добавьте его в `~/.ssh/authorized_keys` на сервере. Но проще сделать это так:

```
$ ssh-copy-id centos64
```

Этот скрипт выполняет подключение по SSH, поэтому вам придется ввести свой пароль на centos64. Если все хорошо, при следующей попытке подключения к серверу вы подключитесь сразу без ввода пароля (если это не работает, обратитесь к врезке «Экспресс-диагностика» внизу).

Кстати, если вы когда-либо разворачивали систему Linux на Amazon Web Services, то, наверное, проходили через процесс создания и загрузки пары ключей. Аутентификация пользователя на основе RSA обычно используется для получения root-доступа к этим системам.

Подключаться к SSH без ввода пароля удобно. Однако стоит подумать о безопасности такой схемы. Если закрытый ключ не защищен парольной фразой, то любой человек, у которого есть доступ к вашей учетной записи на локальном компьютере, »



» **Локальное перенаправление портов. Вот результат запуска команды `ssh -L 5000 localhost : 3306 centos64`.**

## Экспресс-диагностика недугов SSH от Доктора

Сначала прочтите сообщения об ошибках. Если не поняли их с первого раза, попробуйте прочесть еще раз, помедленнее. Загляните в лог-файлы, особенно на сервере (скорее всего, `/var/log/secure`,

но это зависит от настроек системы). Попробуйте запустить SSH с параметром `-v (verbose)` — подробный вывод) — и она может сказать вам, в чем дело.

Также засекийте, скоро ли появляется ошибка. Долгая пауза перед появлением сообщения об ошибке означает, что какой-то период ожидания медленно истекает.

**Симптом:** ssh: подключиться к хосту centos64, порт 22: Невозможно подключиться к хосту [No route to host], ошибка появляется после длинного таймаута.

**Диагноз:** Это, вероятно, означает, что сервер не работает. Эта проблема не связана с ssh. Можете пропинговать сервер?

**Симптом:** ssh: подключиться к хосту centos64, порт 22: Невозможно подключиться к хосту [No route to host], ошибка появляется сразу же.

**Диагноз:** Подключение может блокировать брандмауэр сервера. Убедитесь, что порт 22 открыт.

**Симптом:** ssh: подключиться к хосту centos64, порт 22: Соединение закрыто [Connection refused], ошибка появляется сразу же.

**Диагноз:** Сервер работает, но сервис sshd не слушает соединения. Он установлен? Запущен?

**Симптом:** Пара ключей RSA пользователя сгенерирована, но ssh все равно запрашивает пароль при входе.

**Диагноз:** Загляните в лог-файл на сервере. Убедитесь, что открытый ключ скопирован на сервер, и что права доступа к файлам `~/.ssh` и `~/.ssh/authorized_keys` заданы правильно (700 и 600 соответственно).

**Симптом:** Во время входа в систему по ssh перед вводом пароля появляется длинная пауза (порядка 15 секунд).

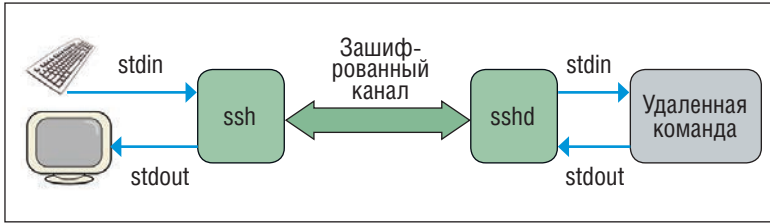
**Диагноз:** Это распространенная проблема. Обычно она вызвана таймаутом, необходимым для обратного DNS-запроса на сервере. Это можно исправить, либо исправив сам DNS-сервер, либо (более практичный вариант) отключив DNS-запросы в конфигурационном файле сервера `/etc/ssh/sshd_config`. Просто добавьте в него строку

```
useDNS no
```

Раньше мне также помогало отключение аутентификации GSSAPI с помощью строки:

```
GSSAPIAuthentication no
```

Не забудьте перезапустить сервер после изменения этого файла.



SSH направляет стандартные потоки ввода и вывода между локальным компьютером и удаленной командой туда и обратно.

получит доступ и к учетной записи на сервере. Задать парольную фразу для защиты закрытого ключа нетрудно, и обычно это делается прямо при создании ключа, но можно сделать это и впоследствии, командой `ssh-keygen -p`. Если вы задали парольную фразу, ее придется вводить при каждом подключении по SSH, что с точки зрения пользователя так же неудобно, как вводить исходный пароль. Однако во многих дистрибутивах есть программы для управления закрытыми ключами, такие как *Gnome Keyring Manager*, *Seahorse* или *SSH-agent*, которые будут управлять ключами и разблокировать их при необходимости. Но я решил не писать о них в этой статье, иначе вряд ли смогу загнать всех червей обратно в банку.

## Поиграем с канализацией

По зашифрованному каналу, созданному SSH, можно передавать трафик для других сервисов. Предположим, вы администрируете стек LAMP на удаленном сервере. На нем есть *Apache* и *MySQL*. Конечно, порт 80 на брандмауэре сервера открыт — для доступа к сайту, но как насчет порта *MySQL* (3306)? Этот порт не должен быть открыт для всех, так как к базе данных обращаются только *Apache* и PHP, запущенные на том же компьютере. Но как тогда ее администрировать? Здесь на помощь приходит локальное перенаправление портов. Если выполнить команду

```
$ ssh -L 5000:localhost:3306 centos64
```

то клиент SSH будет слушать порт 5000 (на локальном компьютере). Если к этому порту подключается клиент *MySQL* или *MySQL Workbench*, SSH принимает соединение и перенаправляет трафик на порт 3306 удаленного компьютера. Название 'localhost' несколько сбивает с толку — на самом деле оно интерпретируется на удаленном компьютере (centos64). Преимуществ у такого подхода два. Во-первых, не нужно открывать порт 3306 в брандмауэре на сервере. Во-вторых, обмен данными с сервером *MySQL* проходит по зашифрованному каналу.

SSH также может выполнять перенаправление удаленных портов, хотя в моем случае это менее полезно. Вот пример:

```
$ ssh -R 4023:ubuntu1204:23 centos64
```

В данном случае, сервер SSHD (запущенный на centos64) будет слушать порт 4023. При подключении клиента он принимает подключение и перенаправляет трафик на порт 23 вашего компьютера (ubuntu1204).

Вот еще один пример того, что можно отнести к «играм с канализацией». Предположим, что у меня на удаленном компьютере (снова centos64!) есть tar-архив, который я хочу восстановить на локальном компьютере. Конечно, можно было бы скопировать архив на локальный компьютер командой `scp` и восстановить

их там, но тогда мне придется хранить две копии архива. Вот более компактное решение:

```
$ ssh -n centos64 cat /tmp/demo.tar | tar xvf -
```

С точки зрения локального компьютера, мы запустили две команды: **SSH** и **tar**. Канал находится на локальном компьютере и связывает стандартный вывод **SSH** со стандартным вводом **tar**. Команда `cat` выполняется удаленно; ее вывод копируется через SSHD (по зашифрованному каналу) в стандартный ввод SSH. Это не особый случай — обычно именно так и происходит. Даже когда я просто вхожу в систему по SSH —

```
$ ssh centos64
```

вывод удаленной оболочки *Bash* передается через SSHD в стандартный вывод SSH, который появляется у меня на экране (см. рисунок).

Наш последний (но важный) пример «игр с канализацией» — это перенаправление X11. Эта технология позволяет запустить графическое приложение на удаленном компьютере и туннелировать трафик обратно на X-сервер на локальном компьютере. Кажется, что это сложно, но на самом деле нет. Вся работа выполняется параметром `-X` SSH. Так, например, команда

```
$ ssh centos64 -X mysql-workbench
```

запустит на centos64 графическую программу *mysql-workbench*, а пользовательский интерфейс будет находиться на локальном компьютере. Это не только впечатляюще просто, но и полностью безопасно, так как весь трафик X11 шифруется.

## Друзья и родственники

Близкий родственник SSH — *scp* — программа защищенного копирования ('secure copy'). Она очень похожа на старую добрую команду `cp`, с тем исключением, что с ней можно копировать файлы на удаленные компьютеры или с них. Вот простейший рабочий пример, который у меня получился:

```
$ scp gnomes.txt centos64:
```

Двоеточие в конце имеет значение. Без него 'centos64' интерпретировалось бы как имя файла, и *scp* повела бы себя как обычная команда `cp`. Но *scp* считает часть до двоеточия как имя компьютера и войдет на сервер по SSH, а затем скопирует файл **gnomes.txt** в ваш домашний каталог на этом компьютере.

Быть может, длинный пример понять будет проще:

```
$ scp gnomes.txt centos64:/home/chris/pointyhats
```

Здесь мы указали файл назначения по абсолютному пути (на сервере). Если для SSH используется аутентификация по умолчанию (имя пользователя и пароль), *scp* каждый раз будет запрашивать пароль, что как минимум утомляет. Настроив аутентификацию с помощью ключей, как я описывал, можно пользоваться *scp* «неинтерактивно» (даже в unattended скрипте), что гораздо удобнее.

В моих примерах использовался SSH-клиент Linux. А если у вас Windows? Ну, здесь есть несколько вариантов, включая программу от bitvise ([www.bitvise.com/tunnelier](http://www.bitvise.com/tunnelier)) и SSH-клиент в *Cygnwin* (о нем мы рассказывали в **LXF180**). Но самый популярный вариант, пожалуй, *putty* ([www.putty.org](http://www.putty.org)). А если вы хотите подключиться к своим серверам, загорая на пляже в Брайтоне, воспользуйтесь клиентами SSH для Android (*ConnectBot* и *JuiceSSH*). **LXF**

## Безопасность

Если вы запускаете сервер SSH на компьютере, открытом для доступа в Интернет, настоятельно рекомендую отключить вход в систему с пользователем `root` по SSH, поскольку `root`-доступ — более чем очевидное направление атаки. Сделать это просто:

позаботьтесь, чтобы в файле настройки `sshd /etc/ssh/sshd_config` присутствовала строка

```
PermitRootLogin no
```

После этого вам придется входить в систему как обычный пользователь,

а затем переключаться на `root` командой `su` (или `sudo`). Конечно, просто отключить вход в систему от имени `root` на своих серверах — тоже хорошая идея; в Ubuntu и ее производных так сделано по умолчанию.

Наши эксперты помогут вам с любым приложением Linux!



**ЕВГЕНИЙ БАЛДИН**  
Подтвердивший  
свою квалификацию  
физик.

## Хромбуки тоже сгодятся

Никогда не доверяй компью-  
теру, который не можешь  
выбросить из окна.

Стив Возняк

**М**не нравятся недорогие и легкие ноутбуки с большой батареей. Для них еще маркетологи специальное имя придумали: нетбуки. Их заряда хватает как раз на полный рабочий день конференции без необходимости судорожно искать место недалеко от розетки. А мощности — на то, чтобы подправить презентацию, почитать почту, удаленно запустить задание в терминале. Вне конференции такой ноутбук может стоять на столе, служа окном в виртуальный мир, пока за завтраком распивается чай и поглощаются бутерброды.

Нетбук EeePC от Asus давно служит мне мобильной печатной машинкой. Из-за внезапного желания ублажить свой взор я даже картинку на задней крышке заказал художнику, с пингвинами на закате. Но время идет, а технический прогресс за последние пять лет все-таки демонстрирует некие успехи, и пора искать замену долгоживущей машинке.

И тут меня ждало разочарование: привычная линейка EeePC де-факто вымерла. К счастью, возникла новая: Chromebook. Спасибо «корпорации Бобра». Да, там тоже не всё шоколадно. Да, для комфортной работы придется надставлять SSD. И не у всякой модели достаточно емкий аккумулятор. И, как обычно, GNU/Linux придется ставить самим. Но недорогое железо так доступно истинным энтузиастам. И в новой его линейке ядро Linux бьется изначально...  
e.m.baldin@inp.nsk.su

## В этом месяце вы научитесь...



**Вещать аудио ..... 68**  
Нейл Ботвик показывает, как использовать *Subsonic*, инструмент создания музыкального сервера, и получать доступ к своей фонотеке через Интернет.



**Рудить SteamOS ..... 70**  
Команда из Нейла Мора и Мэтта Хэнсона все объяснит про игроцентрический дистрибутив, в том числе как его установить и, что важно, зачем он вам вообще.



**Автоматизировать новости ..... 74**  
Вы уже не пропустите важные новости или реформы своего любимого сайта, благодаря автоматически загружаемым лентам RSS от Нейла Ботвика.



**Отдавать журналы ..... 76**  
Системные журналы полезны, когда надо найти неполадки в вашей системе. Ричард Смедли разъяснит, как организовать к ним удаленный доступ.



**Строить сервер Git ..... 80**  
Станьте хозяином своего проекта, обойдясь без Github. Джон Лэйн любезно поможет вам создать собственный Git-сервер.

## Код в учебниках

Строки исходного кода помещаются в цветные плашки. Если строка кода не умещается в колонке, ее остаток переходит на следующую строку, в той же плашке:

```
procedure TfrmTextEditor.  
mniWordWrapClick  
A плашки разделены зазорами:  
begin  
mniWordWrap.Checked := false
```

## АКАДЕМИЯ КОДИНГА



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

## В этом номере

### Ada

Джульетта Кемп вникает в язык милитаристского происхождения, но с именем прекрасной дамы **с. 84**

### Scratch

Лес Паундер восхищается легкостью освоения языка для малолеток и на радостях создает игру **с. 88**

# Subsonic: Поток

Нейл Ботвик разбросал все свои музыкальные файлы по разным устройствам, и *Subsonic* хоть как-то упорядочивает этот хаос в его фонотеке.



Наш эксперт

У Неила Ботвика по компьютеру в каждой комнате, но по соображениям безопасности он нипочем не расскажет, где находится центральный сервер.

**S**ubsonic — это web-приложение потокового медиа-вещания, и оно запускается как сервис на компьютере, собирая вашу музыку, а потом вы можете воспроизводить ее на других устройствах, соединенных с сервером. Хотя оно воспроизводит видеофайлы, все же упор в нем сделан на воспроизведении потокового аудио, что мы здесь и рассмотрим. Через свой веб-интерфейс *Subsonic* предоставляет обычные функции, которые мы ожидаем от музыкальных плееров: плей-листы, оформление, тексты и т.д. Сервер может работать с несколькими соединениями, вы можете слушать разные треки в разных комнатах, или даже находясь вне дома.

*Subsonic* написан на Java. Если ваш менеджер пакетов включает самую свежую версию, установите ее обычным способом. В ином случае используйте отдельный архив. Мы используем **subsonic-4.9\_beta2-standalone.tar.gz**, хотя к тому моменту, когда вы будете это читать, может уже с успехом быть доступен полный релиз 4.9 (вот почему мы не используем 4.8). Перед установкой из архива позаботьтесь об установке движка Java Runtime Engine (JRE) — или Java от Sun, или варианта с открытым кодом IcedTea. Архив не додумался аккуратно поместить все файлы в одной директории, и вам придется ее создать.

```
sudo mkdir /opt/subsonic
sudo tar -xvf subsonic-4.9_beta2-standalone.tar.gz -C /opt/subsonic
```

Жесткого и быстрого правила по поводу того, куда надо устанавливать *Subsonic*, не существует, но стандартным местоположением всяких программ вроде этой является **/opt**. Затем вы можете даже установить его в свою директорию **home**, если только решите запускать его от имени одного пользователя. Чтобы запускать *Subsonic*, нужно создать пользователя и домашнюю директорию для него:

```
sudo useradd --home /var/subsonic --create-home --usergroup subsonic
```

Также нужно также добавить транскодеры. *Subsonic* использует их, чтобы мгновенно транскодировать в аудиоформаты, и без них вы далеко не уйдете. Встроенный плеер браузера даже не воспроизводит файлы Ogg Vorbis, поэтому перед воспроизведением их нужно транскодировать в MP3. *Subsonic* использует

для этого стандартные инструменты, такие, как *ffmpeg*, но не будет осуществлять их поиск по системе — нужно сделать ссылку на них в директории *Subsonic*, **transcode**:

```
sudo mkdir -p /var/subsonic/transcode
cd /var/subsonic/transcode
sudo ln -s $(which ffmpeg)
sudo ln -s $(which flac)
sudo ln -s $(which lame)
```

Не стоит и говорить, что эти инструменты должны быть обязательно установлены, и в большинстве дистрибутивов они есть.

## Настройка Subsonic

Если вы делали установку из вашего менеджера пакетов, он должен был добавить скрипт *init*, чтобы запускать *Subsonic* во время загрузки. Если вы используете отдельный архив, то скрипт есть в **/opt/subsonic/subsonic.sh** (предполагая, что вы сделали установку в **/opt/subsonic** как описано выше). Отредактируйте скрипт, настроив путь к вашим папкам по умолчанию, затем запустите его для запуска сервера *Subsonic*. Потом откройте свой любимый браузер и направьте его на <http://localhost:4040>, где вам предложат задать пароль пользователя-администратора вместо **admin** по умолчанию. Пока вы этого не сделаете, любой, у кого есть доступ к вашей сети, сможет войти в *Subsonic* с правами администратора, так что сделайте это сразу же после первого запуска *Subsonic*. Учтите, если вы запускаете *Subsonic* на медленном оборудовании, при первом запуске вам придется дождаться, пока web-сервер не станет доступен — на Raspberry Pi мы прождали больше минуты.

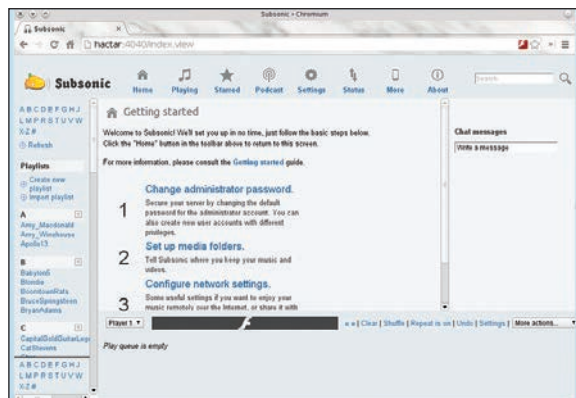
Войдя в систему, вы увидите всю музыку из установленной вами папки по умолчанию. Чтобы добавить папок, перейдите в Media Folders на странице Settings. *Subsonic* не пользуется тэгами ID3 для сортировки музыки, хотя применяет их для отображения информации; список ваших музыкальных файлов основан на расположении файлов в файловой системе. Лучше всего это работает, если ваша музыка организована по иерархии исполнитель/альбом/трек.

Транскодеры, которые мы только что настроили, используются не только для конвертирования из одного формата в другой: они также пригодны для изменения скорости передачи битов в потоке. Это важно при работе с удаленными клиентами, особенно если у вашего сервера ограничена пропускная способность на выход, или когда удаленные клиенты используют соединение 3G. Таким образом вы можете сохранять свою музыку в высококачественных форматах для наилучшего локального воспроизведения, но в то же время слушать их, находясь в другом месте.

Режим работы по умолчанию — получать доступ к *Subsonic* через web-браузер, в который включен Flash-плеер. Из него можно просматривать имеющиеся треки, создавать и воспроизводить плей-листы, скачивать отдельные треки или целые плей-листы и делать все, что полагается в музыкальном плеере.

Можно также использовать и внешний плеер, если вам нужны дополнительные функции или вы не охотник до Flash. Это делается посредством создания плей-листа в стандартном формате

» Web-интерфейс *Subsonic*. Первым делом измените пароль администратора, затем можете добавлять музыку.



# АУДИО

## Дополнительные функции за наличку

*Subsonic* предлагает сервис Premium, который включает определенные функции: например, удаленный доступ через ваш собственный адрес [user.subsonic.org](http://user.subsonic.org); возможность использовать мобильные клиенты для прослушивания музыки

вне пределов сети; скачивание подкастов и другие функции. Это стоит \$1 в месяц, поскольку он использует чужие серверы, но первый месяц бесплатен.

Если вы ощущаете в себе творческую жилку, можете воплотить многие из этих

функций в свободной версии, воспользовавшись такими функциями, как динамический DNS-сервис и VPN. С другой стороны, доллар в месяц уберет вас от лишних проблем и поспособствует дальнейшему развитию *Subsonic*.

M3U и отправки его плееру. Затем внешний плеер получает файлы для воспроизведения с сервера *Subsonic*. Следовательно, *Subsonic* должен сработаться с любым музыкальным плеером с поддержкой плей-листов M3U. Однако он не работает с браузером *Chromium*: тот пытается сохранить плей-лист вместо того, чтобы попытаться его открыть. *Firefox* по умолчанию передает плей-лист в выбранный вами медиа-плеер.

Поскольку вы можете настроить несколько профилей плеера на странице *Settings*, вы можете отвести один из них на встроенный плеер, а другой на внешнюю программу. Также вы можете использовать эту страницу для настройки плееров с ограничением скорости передачи данных на случай более медленных сетевых соединений.

## Удаленный доступ

Хотя вы можете использовать *Subsonic* с любого компьютера в своей сети, можно также получить доступ к вашей музыкальной коллекции вообще отовсюду, через Интернет. Для этого надо настроить переадресацию с вашего роутера на компьютер, где работает *Subsonic*. Если ваш роутер поддерживает UPnP, это делается через web-интерфейс; в ином случае вы просто переадресуете соответствующий порт — по умолчанию это 4040. Если у вас сервис Premium (см. вверху *Дополнительные функции за наличку*), у вас будет собственный адрес [subsonic.org](http://subsonic.org), что снимет необходимость в статическом IP-адресе или динамическом DNS-сервисе.

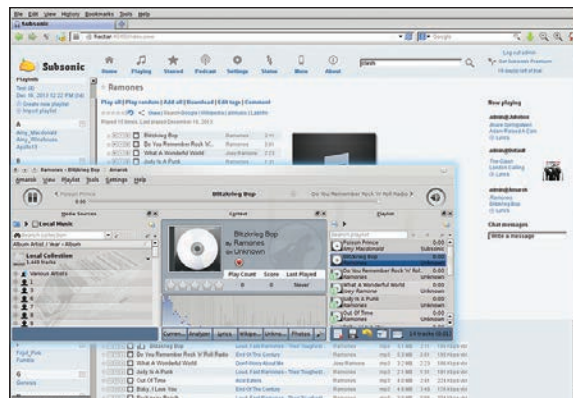
Есть приложения для большинства мобильных ОС, но можно обойтись и браузером. Сервис Premium также необходим, если для воспроизведения вашей коллекции вы хотите использовать приложение Android, даже при соединении через Wi-Fi с вашей локальной сетью. Однако вы можете воспроизводить и из браузера или чего-то вроде *Bubble UPnP Player* с бесплатным сервисом, при условии, что ваш сервер доступен через Интернет и у вас есть публичный адрес, с которым можно установить связь.

В *Subsonic* имеется также загрузчик подкастов: дайте ему URL ваших любимых подкастов, и он найдет для вас новые серии, как только они станут доступными. Однако это функция Premium. В порядке альтернативы можете использовать какой-нибудь *FlexGet* (см. Учебники, стр. 74), чтобы скачать подкасты и загрузить их в папку подкастов *Subsonic*.

## Уровни пользователей

Первый пользователь, который настраивает *Subsonic*, может создавать пользователей с очень тонко настроенными разрешениями. Нажмите на *Settings* в верхнем меню и перейдите в *users*, где вы сможете настроить всплывающее меню на *New user*. Затем вы можете решить, что позволить этому пользователю. Настройки для нового пользователя по умолчанию позволяют ему только воспроизводить файлы. Вы можете добавить разрешения скачивать или загружать файлы, комментировать и влиять на рейтинг песен, делиться песнями с другими и т. д. Вы также можете настроить максимальную скорость передачи для каждого пользователя, что важно, если у вас есть несколько пользователей, в частности, те, кто, вероятно, будут пытаться одновременно получить доступ к сервису удаленно, поскольку они могут съесть всю пропускную способность вашего сервера на выход.

*Subsonic* обычно работает по классической модели клиент/сервер, где сервер отправляет музыку, а клиент ее воспроизводит. В нем также есть режим Jukebox, доступный только для пользователя с соответствующими привилегиями, в нем музыка воспроизводится прямо на сервере. Если сервер у вас работает прямо на ПК в вашей гостиной, можете использовать мобильный клиент или любой другой ПК для управления им. Режим Jukebox управляется настройками плеера на web-клиенте или посредством нажатия кнопки RC на клиенте Android. **LXF**



» Не нравится web-интерфейс? Не любите Flash-плееры? *Subsonic* отправит ваши песни на любой плеер по вашему выбору.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

# SteamOS: Valve-дистрибутив

Компания Valve — ее считают «спасителем игр для ПК» — создала свой дистрибутив на базе Debian Wheezy. Мэтт Хэнсон оценил первый бета-релиз.



Наш эксперт

Мэтт Хэнсон во-зится с операционными системами со времен Amiga Workbench 2.0. Поработав с DOS и Windows, он перебрался на Linux.



Еще в 2003 году, когда Valve впервые выпустила свой Steam, друзей у них было немного. Изначально Steam задумывался как платформа для автоматического обновления игр при добавлении DRM и анти-читинговых мероприятий, однако развился в невероятно популярный сервис с 65 миллионами учетных записей, созданных к концу 2013 года; и, согласно подсчетам, 75% всех цифровых игр приобретаются для ПК через Steam. Хотя мы не очень комфортно чувствуем себя с проприетарным приложением с DRM, нельзя отрицать, что решительный настрой Valve (она присоединилась к Linux Foundation в прошлом году) помог перевести на Linux больше игр, чем когда бы то ни было раньше. Когда Valve объявила о создании собственного дистрибутива Linux, возможности открылись захватывающие.

Через несколько месяцев шумихи вышла первая бета-версия SteamOS. Однако радость быстро угасла: новый дистрибутив сопровождался сообщением от Valve, предупреждающим, что этот релиз очень ранний, с функциями лишь самыми основными,

и рекомендуется только тем, кто уже знаком с Linux. Видеокарты от AMD и Intel не поддерживались, хотя это и было исправлено. Но были и другие ограничения по оборудованию. Вместе с 64-битным процессором Intel или AMD нужен диск объемом 500 ГБ или больше и материнская плата с поддержкой UEFI. На данный момент его можно установить только с USB, и он не позволяет двойной загрузки — он будет настаивать на очистке вашего первоначального жесткого диска и установки на него. Все эти проблемы (надемся) будут решены в будущих обновлениях, но все же если мы решим попробовать SteamOS, препятствий будет немало. К счастью, есть иная возможность.

Valve сделали SteamOS свободной и с открытым кодом, а Ye Olde SteamOSe — это модифицированная программа установки SteamOS, которая решает ряд проблем стандартной версии — и ориентирована на старые машины. Ее можно также использовать на виртуальных машинах — в этом изначальная SteamOS не сильна. Ее можно установить с DVD, и кроме UEFI она поддерживает BIOS. Ей нужно не 500 ГБ на диске, а более реалистичные 40,5. А главное, она поддерживает двойную загрузку, и вы сможете опробовать ее, не жертвуя своим основным дистрибутивом.

Есть несколько причин попробовать дистрибутив Valve, хотя все они — по крайней мере, сейчас — касаются игр. Будем надеяться, SteamOS расширят и добавят функций. Один из ее главных плюсов — повышение производительности, так что при переходе с Windows многие игры улучшатся. В сравнительном тесте Valve стрелялка зомби *Left 4 Dead 2* показала в Linux ускорение на 16,4%. Тесты Valve проводились на 32-битной Ubuntu 12.04, а значит, любой дистрибутив получит те же преимущества, и клиент Steam может запускаться на ряде дистрибутивов — с включенным пользовательским интерфейсом Big Screen, дружелюбном к ТВ. SteamOS обещает много, но Valve придется-таки попотеть, чтобы завоевать популярность. К счастью, у них репутация трудяг.

## Паровые машины

Объявляя о SteamOS, Valve также пообещала ряд Steam Machines — компьютеров, созданных третьими сторонами с предустановленной SteamOS. На недавней выставке CES 2014 в Лас-Вегасе мы наконец-то взглянули на первое поколение Steam Machines и поняли, чем они могут похвастаться в плане компонентов и в плане цены. Было 13 Steam-машин от разных производителей компьютеров, и цены варьировались от £500 до £3000. И хотя даже

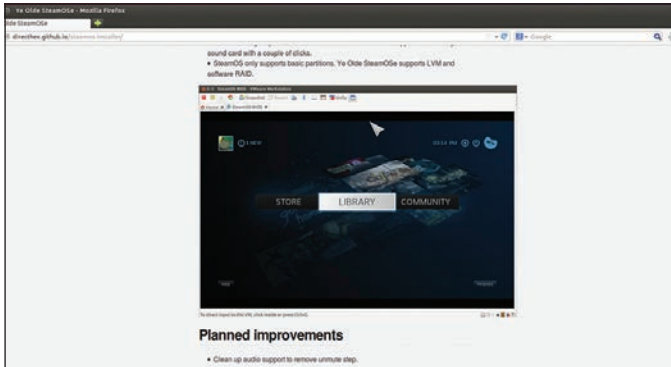
самая низкая цена не была достаточно низкой, чтобы обеспечить массовую популярность, все же был заметен здоровый интерес со стороны производителей ПК, да и большой выбор для потребителей — во многом именно так, как и обещали Valve.

К особо примечательным Steam Machines относится Tiki от Falcon Northwest, у которого есть множество вариантов конфигурации, в том числе с высококлассной видеокартой GTX Titan, 16 ГБ ОЗУ

и 6 ТБ на жестком диске, при цене, подбавляющей высокотехнологичным компонентам — между £1500 и £5500. Имеется также куда более скромный по характеристикам (и стоимости) CyberPowerPC, который идет с 3,9-ГГц AMD A6 CPU, 2-ГБ Radeon R9 270 GPU и 8 ГБ ОЗУ. AMD CPU и GPU можно заменить на процессор Intel 3,5 ГГц Core i3 и видеокарту Nvidia GeForce GTX 760, и он будет продаваться примерно за £499.

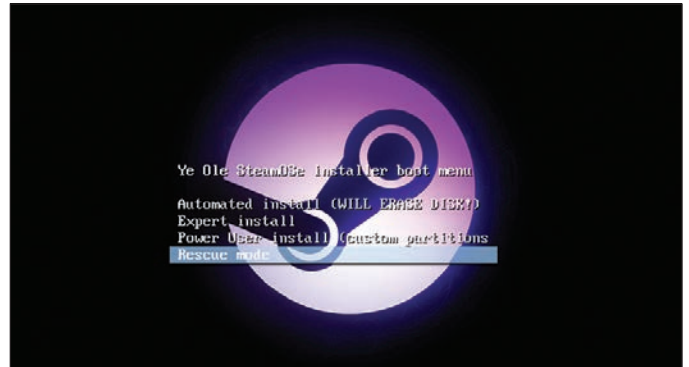


## Устанавливаем SteamOS с помощью Ye Olde SteamOSe



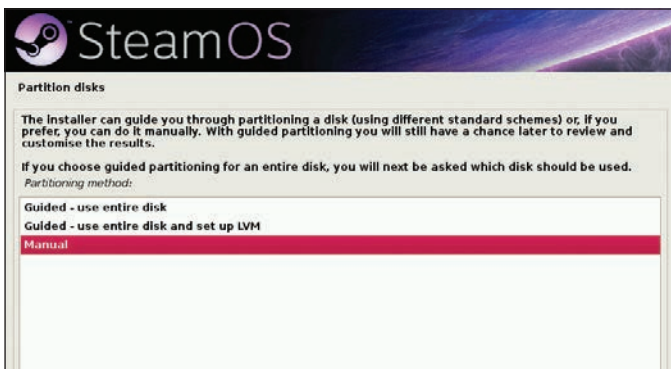
### 1 Скачайте и прожгите Ye Olde SteamOSe

Перейдите на <http://directhex.github.io/steam-os-installer>, чтобы скачать самую свежую версию Ye Olde SteamOSe через торрент или magnet-ссылку. После загрузки нужно записать файл .iso на пустой DVD или создать загрузаемую флэшку USB. Сделав это, перезагрузите свой ПК и загрузитесь с созданного вами USB или DVD.



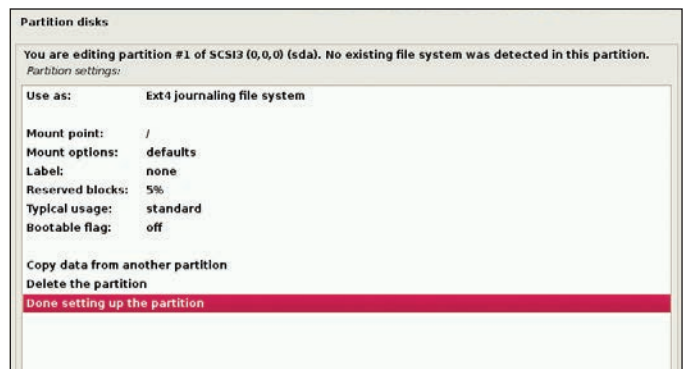
### 2 Выберите свой метод установки

Когда загрузится программа установки, вам предложат сделать выбор из разных возможностей установки. Простейшая установка — автоматическая, но она полностью удалит содержимое диска. Если вы устанавливаете SteamOS вместе с другой ОС, для двойной загрузки, то выбирайте или установку Expert, или Power User.



### 3 Установка Power User

Для двойной загрузки рекомендуем установку Power User. На следующем экране выберите Manual, чтобы назвать раздел диска, на который запишется установка. На следующем экране вы увидите разделы и диски вашей системы (и сможете изменить размер любого раздела, если надо). При установке на свободный диск дважды щелкните по диску и выберите Yes.



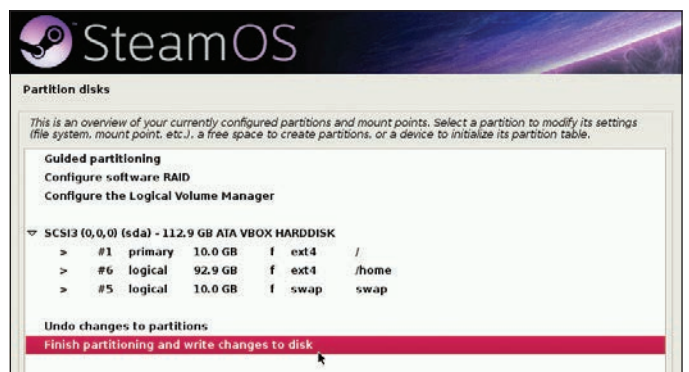
### 4 Чтение с основного раздела

Укажите свободную область, где вы хотите установить SteamOS — не менее 10 ГБ. Выберите Create a new partition [Создать новый раздел], если вам это нужно, и задайте его размер. Убедитесь, что он Primary [первичный], затем нажмите на Continue [Продолжить], проверив, что в поле Use as [Использовать как] указано Ext4 journaling file system [ФС ext4 с журналированием]; затем выберите Done setting up the partition [Настройка раздела завершена].



### 5 Создание других разделов

Щелкните по свободному месту, чтобы создать раздел. Укажите ему размер около 10 ГБ — это будет ваш раздел подкачки [swap]. Сделайте его логическим и создайте его в конце оставшегося места, когда появится эта опция. Попад на итоговый экран, дважды щелкните по Use as и выберите Swap area. Дважды щелкните по оставшемуся пространству, оставьте размер раздела как есть и убедитесь, что при надписи Mount point [Точка монтирования] у вас стоит /home.



### 6 Завершение установки

Выберите Finish partitioning and write changes to disk [Завершить создание разделов и записать изменения на диск], затем Yes. SteamOS начнет настройку и установку. По окончании нажмите на Continue, чтобы перезапустить вашу машину, и загрузите SteamOS. После загрузки SteamOS нужно будет ввести пароль desktop. И, наконец, нажмите на Activities в верхней части экрана, выберите Applications и откройте Терминал. Введите steam, чтобы обновиться до последней версии. **LXF**

# SteamOS: Даете Движок Steam

Всем некогда создавать настоящий движок Steam, поэтому давайте создадим виртуальный с помощью **Нейла Мора**.



## Наш эксперт

**Нейл Мор** играет в игры со времен *3D Monster Maze* на ZX81, но так до сих пор и не сумел пройти *Jet Set Willy*. Ну, вам-то это по плечу, да?

**М**ы увидели, как SteamOS работает на настоящем оборудовании, но даже с самым свежим обновлением *AI-chemist*, добавившим поддержку драйвера отображения AMD и Intel, создание настоящей машины на Steam — штука проблематичная. Поскольку в наши дни модно работать на виртуальной машине (VM) в целях обеспечения безопасности, давайте рассмотрим, как установить и запустить прихрамывающую SteamOS на базе Debian на *VirtualBox*.

Мы будем иметь дело с куда менее дружелюбным стандартным дистрибутивом, который настаивает на наличии оборудования *UEFI* и всего остального. Возьмите самую последнюю сборку **SteamOSInstall.zip** на <http://repro.steampowered.com/download>. Это более 1 Гб. Распакуйте содержимое архива и скомпилируйте файл ISO из его содержимого — проще всего сделать это в терминале, командой

```
mkisofs -R -o steam.iso SteamOSInstaller/
```

Мы запустим его из *VirtualBox*, поэтому установите его и создайте новую машину. В идеале она должна иметь 4 Гб ОЗУ, выбранную загрузку *EFI* и не менее 50 Гб на диске и включать 3D-ускорение со 128 Мб памяти. Укажите виртуальному CD-приводу на созданный вами ISO и запустите его.

Система настроена на использование *UEFI*. При этом первом запуске она, вероятно, выдаст вам ошибку *Grub* — не обращайте внимания, просто немного подождите. Постепенно ISO загрузится, секунд через 10 промедления. Если он перейдет в *EFI*, вам нужно будет вызвать код загрузчика *EFI* из *UEFI*. В командной оболочке введите:

```
FS0:\EFI\BOOT\BOOTX64O
```

Появится меню установки SteamOS — выбирайте установку по умолчанию. В этой точке вы можете передать все в руки вашей виртуальной машины. Минут через пять SteamOS установит все, что, по ее мнению, ей нужно. Проверьте установки VM и убедитесь,



➤ Контроллер Valve Steam стар, как мир, и его может производить любая компания.

что ISO размонтирован, затем разрешите перезагрузку. Появится меню *Grub* с возможностью загрузиться в SteamOS или *Recovery*. Выберите *Recovery*, и вы окажетесь в терминале. Вам нужно убрать драйверы по умолчанию и установить драйверы *VirtualBox*, чтобы мы могли видеть на экране происходящее. Удобно. Сначала удалите драйвер по умолчанию и настройте *X Windows* следующим образом:

```
apt-get purge ".*nvidia.*"
```

```
dpkg-reconfigure xserver-xorg
```

Драйвер *VirtualBox* является частью ISO *VBoxAdditions*. Если вы уже использовали раньше *VirtualBox*, то, вероятно, у вас уже есть этот ISO. Его довольно просто скачать; выделив окно виртуальной машины, выберите *Install Guest Additions* [Установить гостевые дополнения]. Если у вас его пока нет, оно загрузит для вас ISO и подмонтирует его. Вернитесь в терминал VM — нужно будет также подмонтировать диск и запустить программу установки:

## Берем управление на себя

Valve участвовала в CES 2014, демонстрируя созданные ими прототипы *Steam Machine*. Мы даже сумели лично познакомиться с прототипами, благодаря любезности AMD. Одной из областей особого интереса были джойстики, которые производит Valve для игр на ПК. У них сохранился знакомый дизайн геймпада, но, в отличие от остальных знакомых вам геймпадов, они задействуют для управления две больших сенсорных панели. Первым нашим впечатлением было то, что привыкать к ним придется долго. И даже в этом случае

нам было трудно понять, как использовать их для определенных видов игр — в первую очередь, для жанра шутеров в реальном времени, где нужна большая точность в выборе боевых единиц. Возможно, мы просто представители старой школы, привычной к клавиатуре и мыши: даже стандартные джойстики кажутся нам довольно сложными в применении.

Однако в *Borderlands 2* панель работала отлично. Вы используете левый контроллер для движений типа WASD, а правый — для обычного вида мыши.

Панель дает также осязательные ощущения через оба контроллера с помощью «двойных линейных резонансных приводов». Прототипы предлагают только четыре плохо сделанных кнопки; ожидается, что в окончательной версии контроллеров будет восемь.

Перейдите на *Civ 5*, и все становится не столь безоблачно — как мы и подозревали, переход оказался более сложным. Вы пытаетесь добиться точности работы с мышью на 600+ DPI, впахнув ее в двухдюймовую сенсорную панель.

```
mount /dev/cdrom /media/cdrom
sh /media/cddrom/VBoxLinuxAdditions.run
reboot
```

Это приведет базовую SteamOS в нормальную форму, чтобы *VirtualBox* мог с ней справиться. После перезагрузки надо будет разобраться с загрузчиком UEFI. В окне приглашения EFI введите следующие две строки — важно поставить обратный слэш:

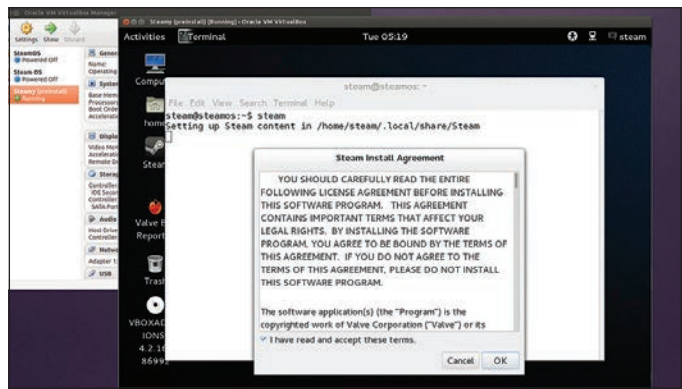
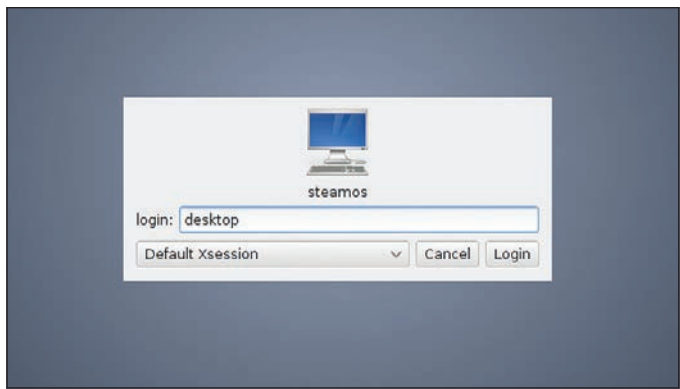
```
FS0:
efi\steamos\grubx64.efi
```

Этого достаточно, чтобы установить и запустить SteamOS в *VirtualBox*, но это еще не все. Следуйте прохождению, чтобы разобраться со всеми элементами до конца. Конечно, мы по-прежнему рекомендуем сборку Ye Olde Steam OS. Она устраняет множество проблем ОС от Valve. Увы, *VirtualBox* работает не настолько быстро, чтобы пробовать какие-то игры, но зато вы сможете поэкспериментировать с рабочим столом и режимом Big Picture, который предлагает SteamOS. Плюс к тому вам, разумеется, понадобится учетная запись Steam — впрочем, у кого же ее нет?

Исходный релиз SteamOS был подмочен отсутствием широкой поддержки оборудования. Это был ускоренный вариант Steam Machine от Valve. Мы считаем ошибкой, что не включили драйверы для AMD и Intel. Даже при поддержке более новых GPU это, вероятно, отпугнуло многих пользователей не из Linux. Брань, доносящаяся с той стороны офиса, где сидит *PC Format*, показывает, что у пользователей Windows явно все не так уж хорошо ладится со SteamOS.

Мы остаемся открытыми для нового дистрибутива. Он даст производителям оборудования возможность предложить игры Linux новой аудитории. Вопрос в том, купит ли его кто-нибудь? Valve хорошо известна тем, что не спешит выпускать продукты, пока они не готовы к релизу. Это будет дорогая консоль с потенциально более широкими встроенными возможностями в Linux и мультимедиа, привязанными к игровой платформе Steam для продажи игр. Ее будет трудно продать, даже если SteamOS окажется идеальной, особенно когда мы сможем просто запускать Steam на любой существующей машине Linux в режиме Big Picture. Ну давай, Valve, удиви нас! **LXF**

## Подчистим Steam

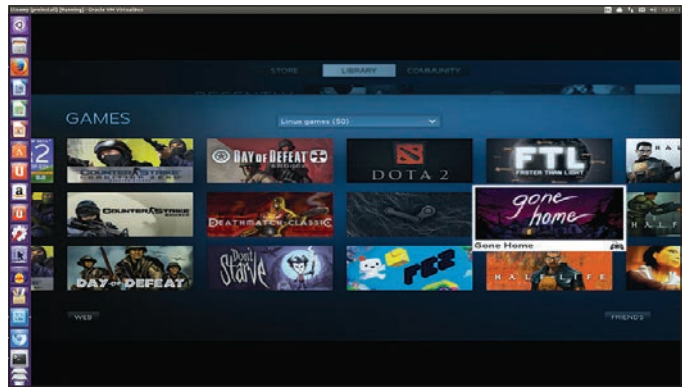
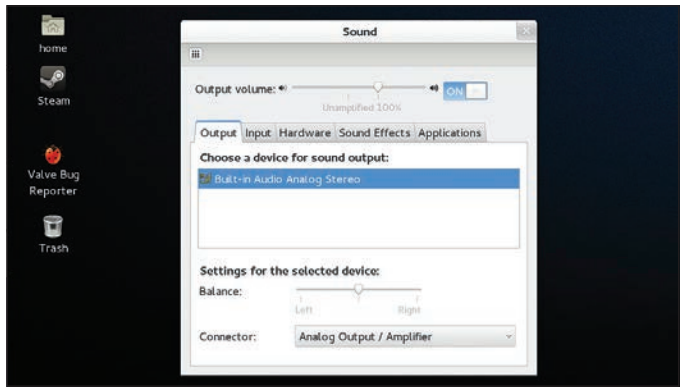


### 1 Пожалуйте на рабочий стол

По умолчанию SteamOS идет с рабочим столом Gnome 3; в окне приглашения вы должны ввести в качестве имени пользователя Desktop и desktop. И вы попадете на рабочий стол по умолчанию. Оказавшись в интерфейсе SteamOS Big Picture, можно выбрать Return to Desktop [Возврат в рабочий стол] в качестве резервной опции вместо выключения или перезагрузки.

### 2 Установка Steam

Вы еще не установили Steam? Вы установили ОС, но не проприетарный клиент Steam, потому что тот не может поставляться вместе с дистрибутивом — и по другим причинам, связанным с лицензией. Выберите activities вверху слева, введите 'Terminal' и выберите его. Введите 'Steam' и нажмите на Return. В этот момент вам понадобится подключение к Интернету, чтобы скачать клиент.



### 3 Лучший звук

Возможно, вас не удивит наличие одной странной проблемы — среди прочих странных проблем — с аудио. Хотя на сей раз все исключительно из-за пристрастия Valve к поддержке ASRock-интегрированного HDMI-аудио для своего компьютера. Выберите значок с микрофоном, откройте панель управления звуком [Sound Control] и включите аудио для рабочего стола здесь.

### 4 К Steam на всех парах

Теперь вы можете выбрать меню сверху справа и выбрать Log Off. В меню входа выберите SteamOS и войдите, используя 'steam' как имя пользователя и пароль. Запустится режим Big Picture, и вы сможете войти обычным способом. Под значком настроек находится опция перехода на рабочий стол Linux. **LXF**

# FlexGet: ЛЕНТЫ

Нейл Ботвик показывает, как настроить и использовать *FlexGet*, чтобы постоянно скачивать самые свежие обновления контента.



## Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он нипочем не скажет, где находится центральный сервер.

Ленты RSS — отличный способ следить за множеством сайтов. Чаще всего используемые для отображения заголовков новостей или блогов, они могут также анонсировать новый контент, от подкастов до дистрибутивов. Здорово, когда можно узнать о наличии новой версии, но еще лучше, если она скачается автоматически. Одна из программ, способных сделать это для вас — *FlexGet* (<http://flexget.com>). *FlexGet* читает ленты RSS и другие источники и находит новые файлы для скачивания, а затем поступает с ними согласно вашему файлу настройки.

*FlexGet* написан на Python. Если в репозиториях вашего дистрибутива свежая версия, установите ее. Если нет, установите пакет *python-pip* и запустите его, он все скачает и установит:

```
sudo pip install flexget
```

Для работы *FlexGet* нужен файл настройки — с информацией, какие ленты RSS ему проверять, что искать и что делать с найденным. Место по умолчанию для этого файла — `~/flexget/config.yml`; *FlexGet* должен запускаться от имени обычного пользователя. Файл настройки использует язык разметки YAML с достаточно суровым синтаксисом. Подобно Python, YAML использует отступы для структурирования иерархии настроек; в отличие от Python, ему важно то, как делаются отступы. Отступы кратны двум пробелам — если использовать табуляцию или нечетное число пробелов, у вас будут проблемы, так что будьте осторожны. Зато синтаксис файла настройки вполне прямодушный. Вот пример, как найти и скачать подкаст серии BBC *Discovery*:

```
tasks:
  bbc_discovery:
    rss: http://downloads.bbc.co.uk/podcasts/worldservice/
    discovery/rss.xml
    accept_all: yes
    download: ~/podcasts/bbc_discovery
```

Первая строка запускает список задач, которые все расположены на одном уровне отступов. Следующая строка начинает определение задачи и дает ей имя. Определение задачи состоит из ряда плагинов. Их можно разбить на группы, и в основной задаче будут плагины ввода и вывода и обычно как минимум один фильтр между ними. Плагины ввода определяют, где искать информацию — в этом случае используется плагин *rss*, который берет единственный аргумент URL ленты. Фильтром здесь будет *accept\_all*. Если использовать его в новом правиле, то при первом запуске скачать может очень много. Плагин вывода — это скачивание, которое дает указание *FlexGet* скачать файл и берет аргумент, где его сохранить. Хотя мы показывали плагины в логической последовательности — ввод, фильтр, вывод, порядок здесь значения не имеет, как и порядок задач. Опять же, *FlexGet* весьма привередлив по поводу синтаксиса своего файла настройки, но, на наше счастье, есть опции проверки файла перед его использованием. Первая проверяет файл на наличие ошибок в синтаксисе.

```
flexget --check
```

Эта команда дает список всех задач в файле, вместе с сообщениями о правильности или ошибке. Предполагается, что для файла настройки вы используете стандартное местоположение; в противном случае, сообщите об этом *FlexGet* посредством

опции `-c`. Убедившись, что синтаксис правилен, можете проверить, делают ли созданные вами задачи именно то, что вам надо, с помощью

```
flexget --test -c /some/odd/location/config.yml
```

У вас запустится *FlexGet* в режиме подробной информации, но ни одна из задач по выводу при этом не выполняется; вам сообщат, что будет делаться. Если ряд задач у вас определен и вы добавляете другие, вы вряд ли захотите заново тестировать те, которые заведомо работают, поэтому можно определить к прогону отдельные задачи:

```
flexget --test --task bbc_discovery
```

или

```
flexget --test task bbc_*
```

Вторая команда иллюстрирует использование в именах задач спецсимволов. Возможно, придется заключить имя задачи в кавычки, чтобы оболочка не раскрывала спецсимвол как имя файла.

## Автоматизируем

Противоположность опции `--test` это `--cron`. Вы, видимо, уже догадались, что она предназначена для запуска *FlexGet* из *cron*; при этом сокращается объем вывода, но все действия выполняются. В *stdout* не поступает сообщений об успехе работы, что весьма полезно: *cron* может сообщать по электронной почте обо всех результатах выполняемых им задач, и вы получите уведомления обо всех ошибках. *FlexGet* записывает информацию, которую в ином случае он должен отправить в *stdout* в лог-файле, расположенном в той же директории, что и файл настройки.

*Cron* — стандартный способ запуска *FlexGet*, вместе со многими другими программами. В *cron* лучше использовать абсолютный путь, поскольку он не запускается во всей полноте среды, имеющейся у вас в терминале; *FlexGet* предоставит вам полный путь до программы, затем добавьте его в свой *crontab*, запустив

```
crontab -e
```

который загрузит ваш пользовательский *crontab* в редактор по умолчанию. Добавьте такую строку:

```
10 * * * * /usr/bin/flexget --cron
```

чтобы запускать *FlexGet* на десять минут каждый час. Запускать *FlexGet* ежечасно вполне разумно: вам вряд ли понадобится запускать его чаще, поскольку некоторые провайдеры лент блокируют IP, перегружающие их серверы. Можно запускать его реже:

```
30 8 * * * /usr/bin/flexget --cron
```

```
0 *4 * * * /usr/bin/flexget --cron
```

Первый пример запускает его раз в день, в 8:30 утра, а второй — каждый четыре часа, на час.

Некоторые настройки могут быть одинаковы для целого ряда задач, поэтому *FlexGet* разрешает использование пресетов, определенных в отдельной от задач секции. Если у вас есть задачи для ряда подкастов, которые вы хотите скачать в одну директорию, можете сделать пресет и использовать его в задачах.

```
preset:
```

```
  podcast_dload:
```

```
    download: ~/owncloud/podcasts
```

```
tasks:
```

# RSS — АВТОМАТОМ

```
bbc_discovery:
  rss: http://downloads.bbc.co.uk/podcasts/worldservice/
  discovery/rss.xml
  accept_all: yes
  preset: podcast_dload
```

Перед запуском пресеты сливаются в каждую задачу, так что tasks:

```
bbc_discovery:
  rss: http://downloads.bbc.co.uk/podcasts/worldservice/
  discovery/rss.xml
  accept_all: yes
  download: ~/owncloud/podcasts
```

по сути то же самое. Зато пресет можно использовать во многих задачах, и если вы решите изменить настройку, надо будет только отредактировать одну строку в файле настройки. Чтобы использовать в задаче несколько пресетов, синтаксис будет таким:

```
preset:
  - preset1
  - preset2
```

## Работа с torrent-файлами

Некоторые ленты RSS объявляют о наличии новых торрентов, а не о файлах как таковых; скачать торрент недостаточно, вам нужно загрузить его в свой BitTorrent-клиент. Некоторые клиенты поддерживают директорию watch — сохраненные там торренты запускаются автоматически, но для этого нужно, чтобы на момент сохранения работала программа BitTorrent.

*FlexGet* также может напрямую интегрироваться с популярным ПО BitTorrent, а значит, вы будете получать уведомление, если добавление торрента не удалось. Следующий пример проверяет Linux Tracker на предмет наличия определенного дистрибутива и отправляет все обнаруженные торренты в *Transmission*.

```
presets:
  torrent:
  transmission:
    username: nelz
    password: 1337
```

```
tasks:
  geexbox:
    rss: 'http://linuxtracker.org/rss_torrents.php?feed=dl&cat[]=203&cat[]=591&pid=00000000000000000000000000000000'
    accept_all: yes
    preset: torrent
```

Пресет использует плагин передачи [transmission] с деталями логина; это делает задачу более удобной для чтения и упрощает переключение на разные программы с поддержкой BitTorrent, такие, как *Deluge*. Обратите внимание на одиночные кавычки вокруг RSS URL, потому что они содержат специальные символы.

Когда выходит новый торрент, *FlexGet* находит его и начинает скачивать его и раздавать. Единственный фильтр, который мы использовали, это **accept\_all**. Есть и множество других, например, **series**. Это удобно, когда у лент используется стандартная нотация сезон/эпизод [season/episode] — возможно, вам не нужны все доисторические файлы, которые вы загрузили иными средствами. Например, вот запись (сейчас неактивная) в **config.yml**:

```
tuxradar:
  rss: http://www.tuxradar.com/files/podcast/podcast_ogg.rss
```



> [LinuxTracker.org](http://LinuxTracker.org) — весьма полезный ресурс.

```
series:
  - tuxradar:
    begin: S04E11
  preset: dload
```

Опция begin показалась, когда я начал использовать *FlexGet* для скачивания подкастов. Документация по всем фильтрам и прочим плагинам находится на сайте *FlexGet*. Альтернативный подход — использовать фильтр **exists**:

```
exists: ~/podcasts/
```

Он не станет скачивать файл, если в данной директории уже есть файл с таким именем. От скачивания файлов и складирования их на жестком диске мало проку, если вы об этом не знаете. Поэтому *FlexGet* поддерживает несколько методов уведомления вас об этом. Самый очевидный — **email**, который определен на высшем уровне в **config.yml**.

```
email:
  from: flexget
  to: me@example.com
  smtp_host: mail.example.com
```

Возможно, вам понадобится настроить **smtp\_username** и **smtp\_password**. Вы также можете использовать **email** в отдельных задачах, добавляя

```
email:
  active: False
```

к задаче, чтобы она не отправляла сообщений по электронной почте. Альтернатива — добавить это к основной настройке и затем приписать **active: True** только к тем задачам, от которых вы хотите получать сообщения.

Другие плагины уведомлений умеют вызывать уведомления рабочего стола, отправлять сообщения на iPhone или отправлять сообщения [Pushover] на устройства Android и iOS. Вот пример отправки уведомлений на все устройства, зарегистрированные со специальным ключом Pushover, что наилучшим образом определяется, как пресет.

```
push_over:
  pushover:
    userkey: uVhHwfRQf6igoekc4fRv1sf1Br8jhE
    apikey: atPVXxm4iiXThEzNpQsaAeCbaAzQtC
    title: FlexGet
    message: Downloaded {{task}}
```

Переменная в скобках при отправке уведомления заменяется на имя задачи. Другие переменные, которые вы можете использовать, перечислены в <http://flexget.com/wiki/Jinja>. **LXF**



# Журнала

чтобы отбить у вас охоту обращаться к *Syslog-ng*, если вдруг окажется, что в *Syslog-ng* есть необходимая вам функция, которой в *Rsyslogd* не хватает.

Мы покажем вам основы настройки удаленного сервера журнала, настройки клиента для отправки на него системных записей, и рассмотрим *Apache* как пример сервиса с собственными системными записями, который можно убедить использовать (*r*) *syslogd*.

Мы будем придерживаться журналов на файловой системе — это намного проще для анализа с помощью имеющихся инструментов; но если вы хотите установить *rsyslog-mysql* через свой менеджер пакетов, позаботьтесь, чтобы он автоматически настроил вашу базу данных *MySQL*, создав базу данных под названием *Syslog* и пользователя *rsyslog* с автоматически сгенерированным паролем и правильными параметрами в `/etc/rsyslog.d/mysql.conf`. Имеется также плагин вывода *PostgreSQL*.

## Настройка сервера

Если вы создаете сервер с нуля, то все, что вам надо — это минимальная установка дистрибутива на жестком диске или его разделе, чтобы большая часть диска была отведена под лог-файлы. Выберите версию программы установки вашего дистрибутива с ручной настройкой (возможно, в меню при загрузке со средства установки она помечена как расширенная установка), не то окажется, что большая часть раздела на диске отведена под `/` или `/home`.

Если вы используете готовый сервер, который все пишет на дополнительный диск, переместите существующее содержимое `/var/log` на новый диск после форматирования, затем добавьте новый диск в `/etc/fstab` — файл, который система читает при загрузке, чтобы увидеть, какой диск монтировать — вместе с чем-то вроде

```
/dev/sdb1 /var/log ext4 0 1
```

Теперь смонтируем диск с помощью

```
mount -a
```

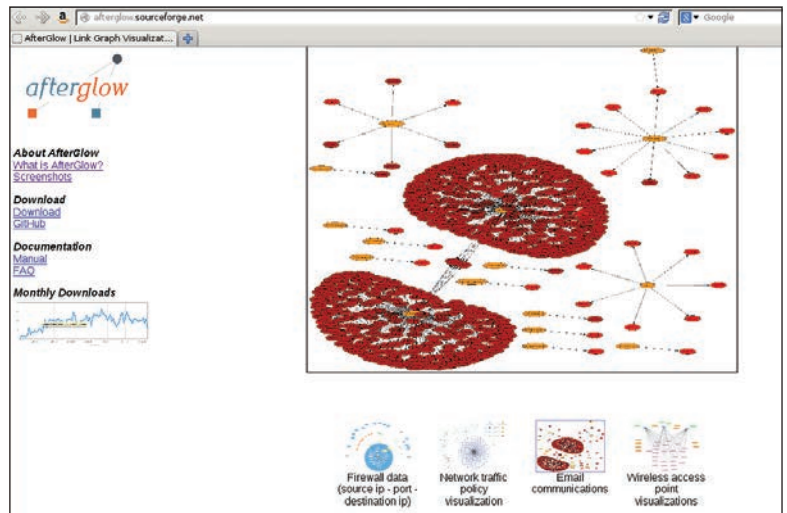
и пусть ваши лог-файлы наслаждаются своим новым жизненным пространством.

Если вы пока не определились с любимым инструментом мониторинга сервера, таким, как *Nagios* или *Zabbix* [Учебники **LXF178/179**, стр. 84], чтобы получать предупреждения о потреблении диска и других проблемах на сервере, то сейчас самое время сделать это самым первым пунктом списка неотложных дел!

В Debian вы найдете `rsyslog.conf` прямо в `/etc`, с пустым `/etc/rsyslog.d/`. Для обратной совместимости *Rsyslogd* читает любой стандартный файл `syslog.conf` из более старого *Syslogd* в Unix, и файл по умолчанию `/etc/rsyslog.conf` пребывает именно в этом формате, с несколькими специальными опциями. `rsyslog.conf` начинается с разделов для Modules [Модули] и Global Directives [Глобальные директивы], но мы сразу перейдем к Rules [Правила], где и находится самое важное. Типичным вариантом по умолчанию, чтобы дать (*r*) *syslogd* команду, куда и какие делать записи, будет

```
auth,authpriv.* /var/log/auth.log
```

```
*.*:auth,authpriv.none -/var/log/syslog
```



► Пакет визуализации, подобный *AfterGlow*, обеспечивает красочную интерпретацию данных из журналов почты.

```
kern.* -/var/log/kern.log
```

```
lpr.* -/var/log/lpr.log
```

```
mail.* -/var/log/mail.log
```

```
user.* -/var/log/user.log
```

Как вы, вероятно, уже догадались, файл настройки просто указывает виды сообщений, которые нужно записать в первой части. Второе поле — это местоположение, которому не обязательно быть файлом:

```
*.info /var/log/messages
```

```
kern.emerg /dev/console
```

Первое поле разделено на функцию и уровень. Функция — это системное приложение, которое генерирует сообщение; выбор по умолчанию невелик, но позволяет отделять системные сообщения от пользователя, ядра и демона системы. Так, `auth` используется для событий безопасности, а `authpriv` — для событий, относящихся к управлению доступом. Уровень более полезен, поскольку дает выбор между `emerg` (panic), `alert`, `crit`, `err` (error), `warning` (warn), `notice`, `info`, `debug` — которые описаны в RFC 5424 и запоминаются (в обратном порядке) с помощью английской мнемонической фразы “Do I Notice When Evenings Come Around Early [Замечаю ли я, как рано стали наступать вечера]”. Специальный уровень `none` велит *Rsyslogd* не отправлять никаких сообщений `authpriv` в *Syslog*, что в противном случае вошло бы в `*.*` в нашем примере.

Обычно уровень включает в себя также все, что находится выше этого уровня важности. Все, кроме этого уровня, можно отметить знаком `!`, а просто указанный уровень можно отметить с помощью `=` перед уровнем, вот так:

```
*.=debug;auth,authpriv.none;news.none;mail.none /var/log/Debug
```

Некоторые программы генерируют массу отладочных сообщений, так что не оставляйте его работать — не будет полезно изучать взаимодействия, порождаемые вашим собственным кодом. ►►

## Скорая помощь

Документация *Rsyslog* не устанавливается по умолчанию. 100 страниц (или около того) обычно выделяются в дистрибутивах в пакет *rsyslog-doc*. Найдите его, чтобы извлечь оттуда нужную информацию.

►► Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Если вас интересуют дефисы (которые вы, вероятно, видели в файле настройки своего дистрибутива), то они дают команду оригинальному *Syslogd* не синхронизироваться после каждой записи в файл, чего *Rsyslogd* не будет делать в любом случае (если только вы не добавите специальное указание в раздел *Global Directives*). Они находятся там в целях обратной совместимости, и их можно благополучно игнорировать.

В нашем втором примере настройки уровня *info* (и выше) предупреждения об опасности [*alert*] записываются в **/var/log/messages**, и на консоли появляется паническая запись от ядра, что может быть полезно только для машин, с которыми вы соединены. Как и со многими файлами настройки, совпадение может быть более чем по одной строке, но первая строка в файле настройки, которая совпадет с событием, определяет действие, если оно изменяет то, что должно быть записано.

`msg, contains, "user nagios" ~`  
игнорирует сообщения *Nagios*. Иногда вам будет нужно не принимать что-то во внимание во время установки, а иногда вы запускаете программу, которая заполняет журналы информацией, которая вам никогда не понадобится. Обратите внимание, что строки исполняются слева направо, и сначала надо ставить общие правила, с особыми исключениями и дополнениями после запятой, где это необходимо.

Попробуйте кое-что изменить в файле **rsyslog.conf**, просто чтобы познакомиться с форматом, прежде чем приступить к удаленному прослушиванию. Чтобы *Rsyslogd* прочитал ваши изменения в настройке, запустите

`kill -HUP (cat /var/run/rsyslogd.pid)`

Вы можете протестировать свои изменения в **rsyslog.conf** из командной строки с помощью программы записи [*logger*], например:

`logger -p local6.warn "Это тест с помощью логгера."`

➤ Даже стандартный, поставляемый провайдером ADSL модем/роутер должен экспортировать свои лог-файлы, если его как следует попросить. Однако он может и не дать вам возможности использовать нестандартный порт.

## Слушаем трафик

Пора вернуться к тем ранним указаниям **rsyslog.conf**, которые мы было проигнорировали. В первую очередь займемся *Modules*. У вас должны быть по умолчанию

`$ModLoad imuxsock`

`$ModLoad imklog`

предоставляющие поддержку записей журналов локальной системы и ядра соответственно. Чтобы ваш сервер слушал удаленный трафик через UDP или TCP, нужно также загрузить соответствующие им модули:

`# provides UDP syslog reception`

`$ModLoad imudp`

`$UDPServerRun 514`

`# provides TCP syslog reception`

`$ModLoad imtcp`

`$InputTCPServerRun 514`

Конечно, номер порта у каждого свой, но множество машин предварительно настроено на стандартный 514. UDP не следит за состоянием — клиентская машина будет упорно слать лог-файлы, даже если ваш сервер журнала слишком перегружен, чтобы все их принять. TCP обеспечивает прохождение всей информации, но за счет повышения нагрузки.

Далее, раздел *Global Directives* работает с правами собственности на файл и разрешениями, но также позволяет указать поддиректорию для файлов настройки определенных приложений:

`$IncludeConfig /etc/rsyslog.d/*.conf`

Ubuntu делает это по умолчанию, а ваш сервер может включать такие файлы, как, например, **/etc/rsyslog.d/postfix.conf**, со строкой для создания дополнительного сокета в *chroot postfix*, чтобы не ломать журнала почты при перезапуске *rsyslog*:

`$AddUnixListenSocket /var/spool/postfix/dev/log`

Теперь, перед всеми другими правилами, вы можете начать отправлять свои лог-файлы на удаленный сервер регистрации. Сервер в нашем примере находится в локальной сети, по адресу **192.168.0.2**, и слушает стандартный порт, поэтому мы можем отправить туда наши лог-файлы командой

`*.* @192.168.0.2:514`

Вторая @, например, @@192.168.0.2:514 отправит лог-файлы через TCP вместо UDP. Однако вряд ли вам понадобится \*.\* от каждого клиента.

Мы уже видели **rsyslog.conf**, настроенный на локальные файлы и удаленные серверы и направленный в именованные потоки (**/dev/console**). Возможно, вам также понадобится отправить лог-файл определенному пользователю (например, *root*) или даже всем пользователям:

`kern.* root`

`*.emerg *`

но полезна и отправка лог-файлов в определенную консоль:

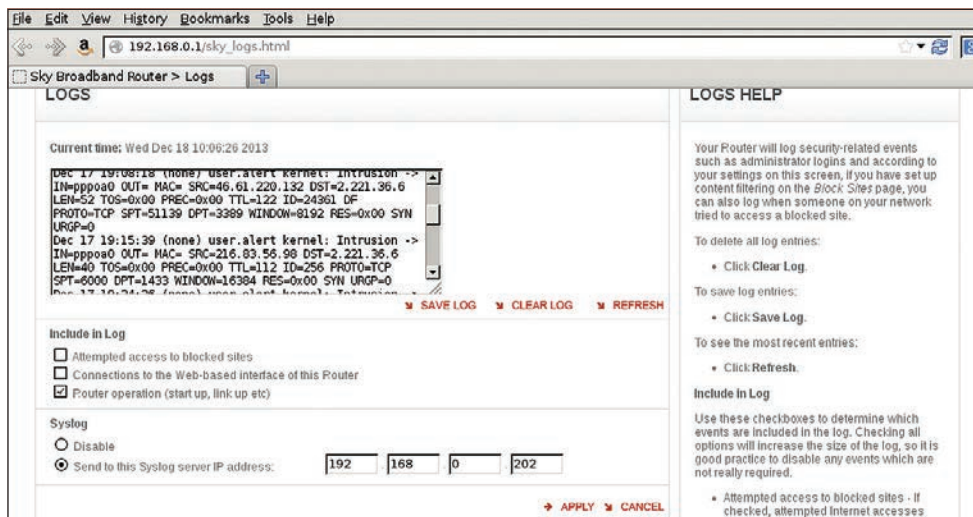
`auth,kern.* /dev/tty2`

Этот пример позволит вам увидеть любые проблемы с ядром и с тем, кто отправлял вам записи в журнал, и вы можете оставить монитор подключенным к серверу и видеть эту информацию, проходя мимо или с помощью *KVM*.

Немного работы с **/var/log** на типовом сервере откроет вам множество сервисов, делающих записи в собственные файлы или директории, без всякой помощи от (*r*)*syslog*. В большинстве случаев другие сервисы можно перевести на использование *syslog*, сильно упрощая перенос лог-файлов на удаленный сервер. Например, в *Apache* отредактируйте файл настройки в **/etc/apache2/conf.d** (в дистрибутивах на базе *Red Hat* — **/etc/httpd**), откомментировав все директивы по записям в журнал по умолчанию и добавив ту, что вызывает программу записи утилиты BSD, интерфейс оболочки в *syslogd*:

`ErrorLog "/usr/bin/logger -p local6.error" combined`

`CustomLog "/usr/bin/logger -t apache -i -p local6.info" combined`



➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



## Который из журналов?

Все ОС семейства \*nix хранят множество своих файлов системных журналов в **/var/log**, и вам стоит поближе познакомиться с самыми важными из них. Помните, что в системах на основе Ubuntu и Debian изрядно размыта грань между **/var/log/syslog** и **/var/log/messages**, но, говоря в широком смысле, последний каталог может хранить все важные, но не критически важные сообщения, тогда как в первый записывается практически все подряд, кроме сообщений **auth**. (Их вы можете обнаружить в **auth.log**.)

### » /var/log/dmesg

Кольцевой буфер ядра, доступный также через команду **dmesg**.

### » /var/log/auth.log

Этот файл, как уже упоминалось, регистрирует все входы и выходы из системы как обычных пользователей, так и системных процессов.

### » /var/log/Xorg.0.log

Лог-файл запуска X. Источник информации о проблемах с вашей настройкой X.

### » /var/log/btmp

Двоичный файл, регистрирующий неудачные попытки входа в систему, доступ к которому можно получить через команду **lastb**.

### » /var/log/wtmp

Двоичный файл, регистрирующий всех пользователей, входивших или выходивших из системы, доступ через команду **last**.

### » /var/log/daemon.log

Предоставляет подробную информацию о работающей системе и демонах приложений.

```
-rw-rw---- 1 root utmp 768 Dec 13 09:51 /var/log/btmp.1
root@rivendell:~# more /var/log/btmp.1

root@rivendell:~# last -f /var/log/btmp.1
richard tty7      :0      Fri Dec 13 09:51   gone - no logout
richard tty7      :0      Mon Dec  2 16:05 - 09:51 (10+17:46)

btmp.1 begins Mon Dec  2 16:05:23 2013
root@rivendell:~#
```

» Большинство записей журнала хранятся в виде простых текстовых файлов, но попытки входа в систему (**btmp** и **wtmp**) — по соображениям безопасности — являются основным исключением, и доступ к ним осуществляется через программу **last**.

Combined — это формат. Индивидуальные форматы можно определять через посредство директивы *Apache* LogFormat. **local6** — одна из восьми локальных функций в *syslog* для локального использования. Здесь не предусмотрено определенных стандартов, но обычная практика для *PostgreSQL* — использовать **local0**, а для *SpamAssassin* — **local3**.

Теперь нам необходимо переправить эти лог-файлы **local6** на удаленный сервер журнала. Возможно, вам понадобится хранить удобную локальную копию записей об ошибках в **/var/log/apache/** или даже с локальными файлами *Apache*:

```
local6.* @192.168.0.2:514
```

```
local6.error /etc/apache2/logs/error_log
```

На сервер журнала вы теперь можете направить входящие лог-файлы **local6** в их собственный файл:

```
local6.* /var/log/apache_logs
```

Не забудьте перезапустить *Apache* и перезагрузить файл **syslog.conf** и на удаленной, и на локальной машине, чтобы ваши изменения вошли в силу.

## Записи журнала мониторинга

Как мы видели, лог-файлы дают полезную информацию о том, что имело место в прошлом, однако лог-файлы мониторинга расскажут нам о том, что происходит в настоящем: например, не пытается ли кто-то атаковать ваш сервер, переполнив буфер. Простая утилита Perl, подобная *Swatch*, скажем, может быть настроена на отправку вам сообщения по электронной почте при каждом вызове чересчур длинного URL. Это делается посредством настройки его файла **conf** на отслеживание лог-файлов или на выполнение любых других действий по вашему желанию, даже на отправку вам сообщения на Twitter, если у вас установлено и настроено приложение командной строки **Ttytter**. Для этого в файле **~/swatchrc** необходимо указать

```
watchfor /File name too long/ ttytter -status="Buffer-overflow
attack in progress on my Web server."
```

*Logwatch* — еще один Perl-скрипт старой школы для присмотра за вашими журналами. Он разработан для запуска с удаленной машине или носителя без возможности записи, но его также можно без проблем запустить локально. Его безопасность достаточно высока, но сначала вам следует научиться писать для *Logwatch* собственные скрипты-фильтры. А если вы используете *multitail*, который показывает несколько лог-файлов, отфильтрованных и с выделением цветом в нескольких окнах, понадобится также знание регулярных выражений.

Если данный вид настройки — явно не ваша область, альтернативы есть: от корпоративных систем аналитики данных на Java, таких как *Graylog2* (способной управлять миллиардами событий и обрабатывать сотни тысяч новых событий ежесекундно), до систем обнаружения вторжений [Intrusion Detection Systems, IDS] вроде *OSSEC*, которые сочетают анализ лог-файлов с проверкой целостности файлов, мониторингом политики, опознанием *gootkit*, предупреждениями в реальном времени и активным откликом. Даже краткое сравнение лежит вне сферы нашего урока, но мы бы предложили вам, если только вы не управляете центром обработки данных или крупной корпоративной сетью, выбрать самое простое решение, которое отвечает вашим потребностям.

## Управление

Ну вот, вы накопили все эти журналы; надо как-то ими управлять. *Logrotate* — известная утилита Linux для сжатия и стирания лог-файлов. Внимательное ознакомление с **/etc/logrotate.conf** и отдельными скриптами в **/etc/logrotate.d** откроет вам несколько интересных и разумных настроек по умолчанию для отдельной машины. Небольшая настройка — и вы получите рабочий сервер журнала, но, вероятнее всего, вам придется некоторое время проводить мониторинг использования диска и роста числа записей, и продолжать делать это по мере добавления клиентов к вашей сети, добиваясь удовлетворительного баланса между местом на диске и наличием достаточного числа регистрационных записей, чтобы обеспечить вам душевное спокойствие.

Обязательность выполнения каких-либо нормативов в крупных компаниях и простые соглашения о том, что вы храните и архивируете от лица клиента, также выставляют свои требования к хранению. И тут мы подходим к рассмотрению режимов архивирования — использовать ли ленты, выделенные NAS/SAN, или даже просто записывать на DVD. Несмотря на наличие множества устрашающе сложных решений по архивированию, вызова *tar* в *Crontab* может вам вполне хватить для архивирования лог-файлов до того, как их сотрет *Logrotate*. Мы в **LXF** горой за простую жизнь — в конце концов, потому-то мы и используем Linux.

Кстати о сложностях: эти четыре страницы обрисовали вам лишь самые основы. Одна только опция фильтрации *Rsyslog* достойна целого учебника, но у каждого читателя свои требования, поэтому мы надеемся, что вы сумеете использовать наше введение для запуска удаленного сервера журнала, а потом приметесь копать глубже, чтобы ознакомиться со всеми функциями, необходимыми для вашей работы. **LXF**

# Git: Построим

Вам нравится идея Github, но хочется держаться подальше от облаков? Джон Лэйн объясняет, как создать собственный сервер.



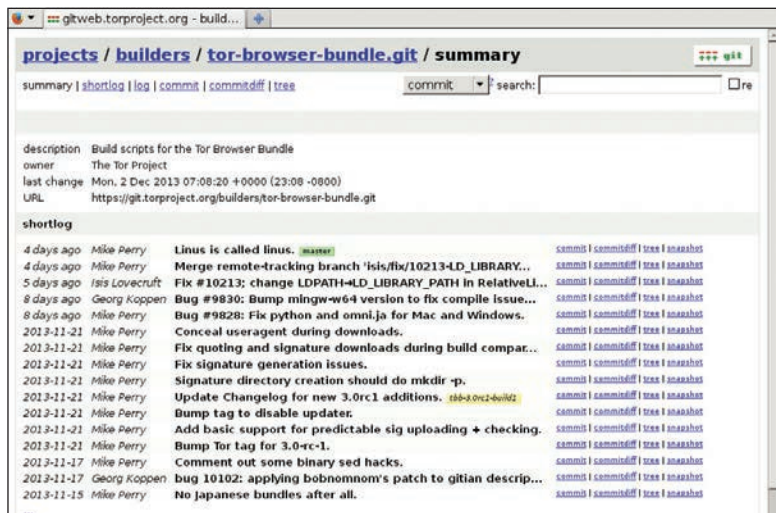
## Наш эксперт

Джон Лэйн ступил на путь Linux в 1994 году, да так с него и не сошел. Он содержит популярной Git-сервер где-то в укромном месте.



**G**it — одна из самых популярных систем управления исходным кодом [Source Code Management, SCM] в экосистеме Linux. Если вы работаете с исходным кодом, велика вероятность, что вы с ней знакомы. Один из принципов структуры *Git*, который и выделяет ее из многочисленного ряда систем SCM, предшествовавших ей (таких, как CVS и Subversion) — то, что она распространяется. Это означает, что каждый пользователь проекта, хранящегося в репозитории *Git*, будет

Так GitWeb работает в естественной среде обитания.



иметь собственную копию этого репозитория. Что весьма важно, у всех этих копий будет одинаковый статус: здесь нет одного мастера.

Вы можете спросить: а если мастер мне нужен? Если мне нужен сервер, куда каждый работающий над проектом может вносить свои изменения? Ну, по крайней мере, концептуально, вы можете счесть некий конкретный репозиторий мастером, и тогда все будет браться с него и храниться в нем. Так делают многие проекты, используя основанный на облаке хост типа GitHub или BitBucket как основной.

Потенциальные неприятности с использованием стороннего сервиса вроде GitHub включают тот факт, что если вы не заплатите за частные репозитории, все вами загружаемое будет в общем доступе. Также весьма спорный вопрос, стоит ли проекту полагаться на облачное хранение. Однако есть альтернатива, которую мы и рассмотрим на нашем уроке: создание собственного сервера *Git*.

В конце этого урока у вас будет дом, где вы разместите ваши репозитории *Git*, доступ к которым можно получить через SSH. У вас также будет web-сервер, в котором вы сможете просматривать свои репозитории и их контент. Для этого мы используем GitWeb, web-сервер, являющийся частью *Git*, и Gitolite, который предоставит интерфейс управления для ваших репозиторий.

На уроке мы будем использовать две машины: одну — в качестве сервера, а другую — в качестве настольной, на которой мы обычно работаем. На каждой нужна учетная запись обычного пользователя, а также доступ **sudo** (или **root**) к серверу, для установки и настройки. Если вы хотите на время урока создать виртуальный сервер, читайте врезку на стр. 82.

## Насущные потребности

В репозиториях *Git*, находящихся на сервере, нет ничего особенного; разве что обычно они создаются как минимальные. Минимальный репозиторий — это тот, который не содержит проверенной рабочей копии своего контента.

Смысл в этом есть, поскольку в репозитории на сервере никто не будет работать напрямую, что помогает снизить потребление ценного места хранения на сервере. Минимальный репозиторий содержит только то, что обычно присутствует в дереве директории **.git** внутри клонированной рабочей копии.

*Git* дополняется web-интерфейсом под названием GitWeb, и мы применим его так, чтобы можно было использовать браузер для просмотра репозиторий. Нам нужен web-сервер, который будет его размещать, и для нашего урока мы выбрали с этой целью *Apache*, но при желании вы можете использовать другой сервер по своему усмотрению. Мы настроим для нашего сервера *Git* виртуальный хост, и если вы уже используете *Apache* для других сервисов, добавление нового на них не повлияет.

Мы хотим защитить безопасность нашего сервера и управлять доступом к нашим репозиториям на запись, делая этот доступ возможным без предоставления прав входа на сервер. Этого можно добиться с помощью Gitolite, уровня управления доступом *Git*, который работает через изменение настройки в специальном

# СВОЙ сервер

## Instaweb: Экземпляр web-сервера Git

Вы, возможно, не в курсе, но *Git* идет со встроенными в стандартный пакет функциями web-сервера. Эта функция именуется *Instaweb*, и вы можете запустить ее в любом репозитории — все, что вам нужно сделать, это изменить корневую директорию репозитория (ту, где находится директория *.git*) и дать команду запуска web-сервера на <http://localhost:1234>:

```
git instaweb start
Предполагается, что в вашей системе присутствует lighttpd. Но если его нет, можно использовать другие серверы. Например, если у вас установлен Ruby, можете сделать
git instaweb start --httpd=webrick
Останавливают сервер командой
git instaweb stop
```

В порядке бонуса, если вы пропустили слово “start” при запуске сервера, тот все равно запустится, но при этом еще и откроет ваш браузер и направит вас куда надо. И вот вам мгновенный просмотр *Git*!

Вы найдете более подробную информацию в документации *Git* на <http://git-scm.com/docs/git-instaweb>

репозитории. Ему нужен только один ID пользователя на сервере, чтобы предоставить доступ на собственный сервер потенциально большому количеству пользователей, у каждого из которых есть индивидуальные права доступа к каждому репозиторию. Пользователям репозитория не нужны учетные записи для входа в систему на сервере. Можно осуществлять администрирование с любой машины в вашей сети (оно не осуществляется напрямую на сервере *Git*) от имени любого пользователя.

Пользователи *Gitolite* получают доступ на сервер с помощью инструментов *Git* для клонирования, вызова, хранения и извлечения репозитория, и это делается через протокол SSH. Они используют тот самый ID пользователя на сервере, но каждый пользователь аутентифицируется с помощью собственной индивидуальной пары публичных/приватных ключей. Аутентификация по паролю не разрешена. Администратор должен загрузить публичные ключи пользователей на сервер через *Gitolite*.

*Gitolite* аутентифицирует администратора точно так же. Для этого вы можете использовать свой публичный ключ, или создать новый именно для этой цели. Если хотите создать новый, войдите на машину, где будете производить администрирование, и создайте пару ключей:

```
ssh-keygen -t rsa -f ~/.ssh/gitadmin
```

Создастся пара ключей RSA. Мы использовали аргумент `-f` для указания имени файла ключа, иначе вы получите имя по умолчанию `id_rsa`. При выборе стоит принять во внимание, что *Gitolite* использует имя файла в качестве имени пользователя. Мы указали `~/.ssh/gitadmin`, предоставляя приватный ключ с таким именем и публичный ключ под названием `~/.ssh/gitadmin.pub`. Подобным же образом вы можете создавать ключи для всех пользователей. Скопируйте свой публичный ключ на сервер *Git*:

```
scp ~/.ssh/gitadmin.pub server_user@myserver:/tmp
```

Войдите на сервер и установите нужные пакеты:

```
sudo apt-get install apache2 apache2-suexec git gitweb gitolite highlight
```

Когда `apt-get` установит *Apache*, тот запустится автоматически, и вы сможете проверить его, направив на IP-адрес или имя хоста вашего сервера. Пакет *highlight* является опциональным дополнением, которое позволяет *GitWeb* выделять синтаксис исходного кода, а причина установить *apache2-suexec* выяснится немного позднее.

## Настройка Gitolite проста

Мы сказали, что *Gitolite* требует наличия специального ID пользователя на сервере, и что он предоставляет весь доступ к репозиториям через него. Создайте этого пользователя — вы можете назвать его как угодно, но *git* будет разумным выбором.

```
sudo adduser --system --group --shell /bin/bash --home /srv/git --disabled-password git
```

Мы настроим наш собственный сервер на хранение своих репозиториях в `/srv/git`. Иерархия файловых систем Linux определяет `/srv` как место для данных сайта, обслуживаемого этой системой, так что это тоже разумный выбор. Мы настроили домашнюю директорию пользователя *Git* на `/srv/git`.

Далее, настроим *Gitolite*:

```
gl-setup -q /tmp/gitadmin.pub
```

Параметр `-q` не даст окну редактора открыться, что позволило бы нам просмотреть и поменять настройки по умолчанию (это нормально — их можно изменить позднее).

Вот и все, чем настраивается *Gitolite*. Заглянув в репозитории `~git/repositories`, вы увидите, что созданы два репозитория: пустой тестовый репозиторий, который вы можете свободно использовать для тестирования, и репозиторий для особых задач администрирования, под названием `gitolite-admin`. Однако прежде чем приступить к администрированию, мы настроим *GitWeb* так, чтобы их можно было просматривать.

## Как Gitolite использует SSH

Мы уже объясняли, как весь доступ к серверу *Git* идет через одного и того же пользователя *Git*. Теперь разберемся, что творится за сценой.

Для начала вспомните, что администратор добавил публичный ключ SSH Боба к настройке *Gitolite*, чтобы настроить его как пользователя. Все, кого определяют как пользователя, должны для этого предоставлять свой публичный ключ.

Когда администратор внесет публичный ключ в репозиторий `gitolite-admin` и перекинет его на сервер, *Gitolite* сохранит этот публичный ключ в файле пользователя *Git* `authorized_keys` (на нашем сервере данный файл размещается в `/srv/git/ssh/authorized_keys`). Когда это будет сделано, ему будет предшествовать команда

```
command="/usr/bin/gl-auth-command bob"
```



Скорая помощь

Если вы когда-нибудь забудете свои команды *Git* — <http://bit.ly/1bNHfb9>.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

## Создайте виртуальный сервер

На нашем уроке мы решили использовать Ubuntu Server, потому что его репозитории по умолчанию содержат все необходимое.

Мы использовали Linux Containers для настройки сервера 13.04 Raring Ringtail с помощью образа

облака с <http://cloud-images.ubuntu.com>. Если в вашей системе есть LXC и вы намерены следовать уроку, не нанося ущерба своей системе, можете сделать нечто подобное. Но, конечно, если вам так больше нравится, ничто не мешает выполнить

все это и на реальной машине. (Имя пользователя по умолчанию — ubuntu; оно же заодно является и паролем.)

```
lxc-create -n gitserver -t ubuntu-cloud -- -r raring
lxc-start -n gitserver
```

Когда ключу SSH таким образом предшествует команда, сервер всегда будет исполнять указанную команду, независимо от того, что просил пользователь. Это ограничивает доступ с данным ключом до особой команды. Вы можете прочитать дальнейшее объяснение этой технологии на <http://oreil.ly/18F27kf>. Таким образом все ключи пользователей приписаны к пользователю стороны сервера git, но каждый исполняет команду Gitolite для определенного пользователя. Попытавшись войти через SSH на сервер, Боб увидел бы только отчет, показывающий его права доступа в Gitolite:

```
618 $ ssh git@git
hello bob, this is gitolite 2.3-1 (Debian) running on git 1.8.1.2
the gitolite config gives you the following access:
@R @W_ testing
Connection to git closed.
```

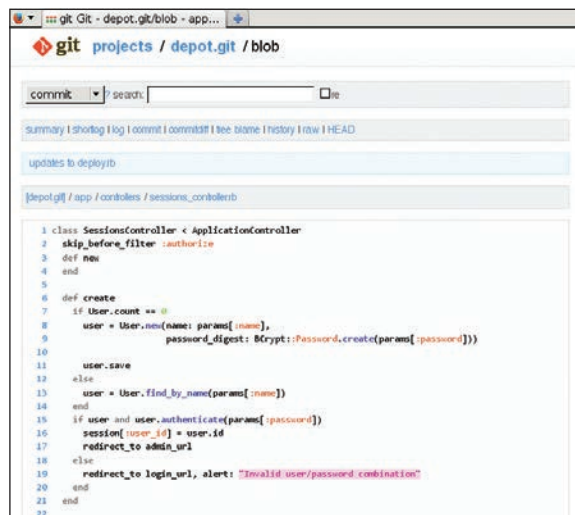
### Переключение пользователей

Для этого нам нужно, чтобы Apache запускал GitWeb от имени нашего пользователя git; но сервер Apache работает не от него. Здесь-то и появляется *apache2-suexec*: это модуль, позволяющий Apache запускать команды от имени другого пользователя. Он устанавливается в Ubuntu как `/usr/lib/apache2-suexec`; в других дистрибутивах это может быть `/usr/bin/suexec`. Подтвердите, где он сможет находить информацию:

```
sudo /usr/lib/apache2-suexec -V | grep AP_DOC_ROOT
```

В Ubuntu это дает `/var/www`. Создайте там поддиректорию, скопируйте в нее файлы GitWeb CGI и убедитесь, что они в собственности нашего пользователя git, потому что именно от его имени Apache будет их запускать:

```
sudo mkdir /var/www/git-suexec
sudo cp -a /usr/share/gitweb /var/www
sudo chown -R git.git /var/www/gitweb
sudo chown -R git.git /var/www/git-suexec
```



Затем напишите скрипт оболочки, который Apache будет использовать для вызова GitWeb, таким образом:

```
#!/bin/bash
# file /var/www/git-suexec/gitweb.cgi.suexec-wrapper
export GITWEB_CONFIG=/etc/conf.d/gitweb.conf
exec /var/www/gitweb/gitweb.cgi
```

Отредактируйте запись `$Projectroot` в `/etc/gitweb.conf`, указав на корень репозитория, созданного Gitolite:

```
$Projectroot = "/srv/git/repositories";
```

Если вам нужно включить выделение синтаксиса (что требуется установки пакета *highlight*), добавьте еще строку в `/etc/gitweb.conf`:

```
$feature['highlight']{default} = [1];
```

Последний шаг — настройка виртуального хоста для обслуживания GitWeb.

```
# file /etc/apache2/sites-available/gitweb
<VirtualHost *:80>
  ServerName git
  DocumentRoot /var/www/gitweb
  SuexecUserGroup git git
  ScriptAliasMatch "^/$" /var/www/git-suexec/gitweb.cgi.
  suexec-wrapper
</VirtualHost>
```

и затем активируйте модуль *suexec* и виртуальный сервер. Чтобы эти изменения вступили в силу, необходима перезагрузка сервера Apache, таким образом:

```
a2enmod suexec
a2ensite gitweb
service apache2 restart
```

Возможно, вам понадобится изменить расположение некоторых файлов, если у вас не Ubuntu, потому что структура настройки Apache зависит от дистрибутива. Вы можете разместить блок `VirtualHost` прямо в его основном файле настройки, если вам так нравится.

### Просматривайте свои репозитории

Выйдите с сервера, чтобы вернуться на рабочий стол и попробовать GitWeb. Конфигурация нашего виртуального хоста настроила его имя `ServerName` как `git`, а значит, GitWeb будет отвечать, когда браузер направлен на `http://git`. Чтобы протестировать это, добавьте запись к своему файлу хостов или DNS, который направляет git на IP-адрес вашего сервера и затем направляет web-браузер на `http://git`, чтобы увидеть два новых репозитория.

### Вопрос стиля?

Если вы полагаете, что основной стиль страниц GitWeb можно улучшить (скажем, вам жаль, что они не похожи на GitHub), попробуйте вот что:

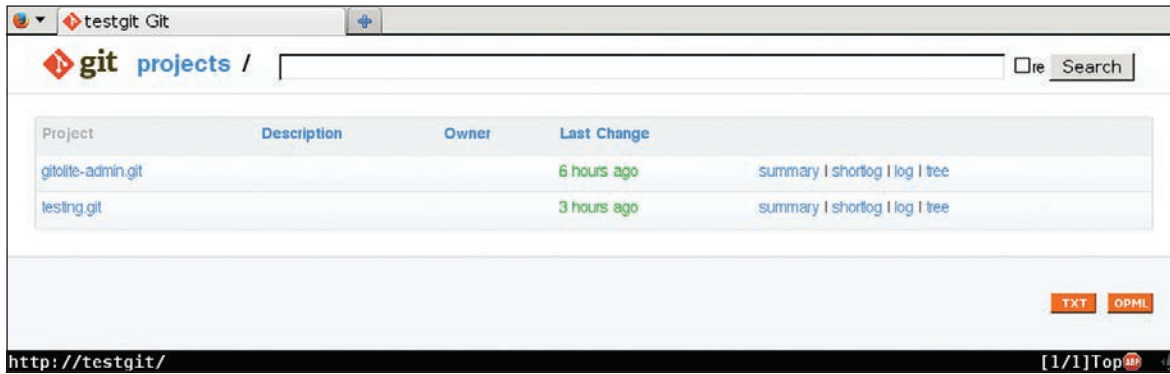
```
git clone https://github.com/kogakure/gitweb-theme
cd gitweb-theme
cp git* /var/www/gitweb/static
chown -R git:git /var/www/gitweb/static
```

### Скорая помощь

Если вам предложено ввести пароль при клонировании репозитория, к которому у вас есть доступ, вероятно, SSH использует неправильный ключ.

➤ Установив пакет *highlight*, наслаждайтесь разноцветным кодом.

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



› Смена стиля GitWeb заметно его улучшает.

Четыре файла GitWeb — `git-favicon.png`, `git-logo.png`, `gitweb.css` и `gitweb.js` — будут заменены, и внешний вид значительно улучшится.

Все администрирование с сервера убирается. Клонировать `gitolite-admin` на свой рабочий стол:

```
git clone git@git:gitolite-admin
```

Так будет установлено соединение с сервером через SSH от имени пользователя `git` и аутентификация по вашему собственному ключу. SSH должен быть в состоянии выбрать и использовать правильный ключ, и если это не ваш ключ по умолчанию, вам, возможно, нужно будет помочь ему, добавив небольшую настройку в свой файл `~/.ssh/config`. Это делается примерно так:

```
Host git
```

```
IdentityFile ~/.ssh/gitadmin
```

Это репозиторий администратора, и он содержит две директории, с именами `conf` и `keydir`.

В директории `keydir` вы определяете своих пользователей. Чтобы добавить Боба в качестве пользователя, просто поместите сюда его публичный ключ под названием `bob.pub` (помните, что Gitolite использует имя файла ключа как имя пользователя).

Настройка репозитория осуществляется в одном файле: `gitoliteadmin/conf/gitolite.conf`. Файл содержит определения для каждого репозитория на сервере, и вы создаете новый репозиторий, добавляя новое определение к этому файлу.

```
repo my-new-repo = "My new repository"
```

```
RW+ = me
```

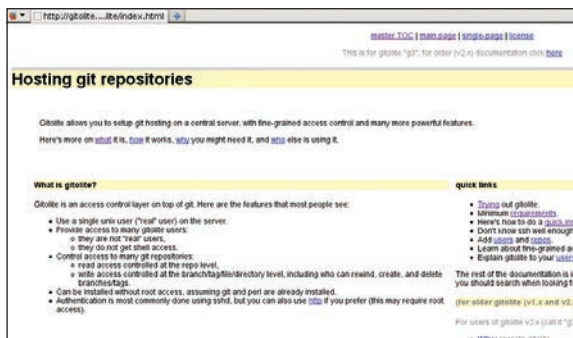
```
R = andrea
```

```
RW = barry
```

```
- = chantal
```

Каждое определение начинается со строки заголовка, предоставляя имя репозитория и опциональное описание — знак равенства и строку описания можно опустить, но если описания присутствуют, GitWeb будет их показывать.

Следующие за заголовком строки контролируют доступ пользователя к репозиторию, который может быть разрешен или запрещен. Основных уровней четыре:



› Gitolite поддерживается хорошо документированным сайтом — <http://gitolite.com>.

› Символ переноса (-) отказывает в доступе. В приведенном выше примере Шанталь [chantal] не в состоянии получить доступ к `my-new-repo`.

› **R** предоставляет доступ только на чтение [read-only].

› **RW** предоставляет доступ на чтение и запись [read-write], но записи ограничены быстрой перемоткой `push` и теми `push`, которые создают новые ветви или тэги.

› **RW+** предоставляет доступ для чтения и записи для любого типа `push`.

Между разрешением и знаком равенства вы также можете вставить то, что автор Gitolite называет `Refex` и определяет как значение опционального регулярного выражения, совпадающего с проталкиваемым `ref`. Так можно контролировать доступ к части репозитория. Например, добавление следующего в наше определение `my-new-repo` не позволит Дориану [Dorian] протолкнуться в главную ветвь.

```
- master = dorian
```

```
RW = dorian
```

Это очень мощная функция, и чтобы полностью ее понять, стоит заглянуть в документацию Gitolite на <http://gitolite.com> и <https://github.com/sitaramc/gitolite>.

Изменив конфигурацию и/или добавив публичные ключи новых пользователей, вы применяете их, перенося изменения в репозиторий `gitolite-admin` и проталкивая на сервер:

```
git add .
```

```
git commit -m "gitolite admin tasks"
```

```
git push
```

Gitolite автоматически обновит сервер, как это требуется для создания новых репозиториях и обновления публичных ключей, авторизованных для пользователя `git`.

Каждый имеющий соответствующее разрешение, полученное через репозиторий `gitolite-admin`, может клонировать репозитории с сервера от имени пользователя `git` через командную строку вида

```
git clone git@git:repo_name
```

Предполагается, что имя хоста сервера тоже `git`, как мы использовали в наших примерах в данном руководстве, и что `repo_name` — это репозиторий, расположенный на сервере.

Однако вы должны помнить, что разрешения Gitolite не применяются к доступу к GitWeb. Для защиты от несанкционированных просмотров репозиториях, обслуживаемых GitWeb, можете использовать стандартный механизм `Apache` — `.htaccess`. **LXF**

## Рассмотрите также...

Кроме рассмотренных на нашем уроке проектов, есть и другие решения для централизованных, размещенных в сети репозиториях Git. Вот некоторые из них:

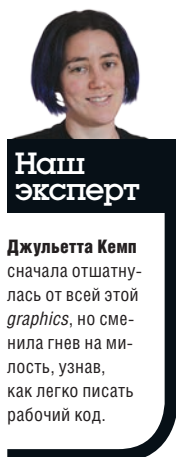
- › Gitorious <https://gitorious.org>
- › CGit <http://git.zx2c4.com/cgit>
- › GitBlit <http://gitblit.com>
- › GitLab <http://gitlab.org>

Скорая помощь

Модули `a2enmod` и `a2ensite` — предназначенные для Debian и Ubuntu расширения пакета `Apache`.

# Ada: Изучаем НОВЫЙ ЯЗЫК

Джульетта Кемп знакомит вас с языком, на котором во всем мире пишут программы с особыми требованиями к безопасности.



Наш  
эксперт

Джульетта Кемп сначала отшатнулась от всей этой *graphics*, но смилила гнев на милость, узнав, как легко писать рабочий код.

Язык Ada был разработан в конце семидесятых для Министерства обороны США и предназначался для систем реального времени и встраиваемых систем в очень больших проектах. Из четырех прототипов был выбран один, и после обширного исследования в 1983 году этот язык стал стандартом и получил название Ada в честь первого в мире программиста Ады Лавлейс [Ada Lovelace (1815–1852), кстати и дочь поэта Байрона].

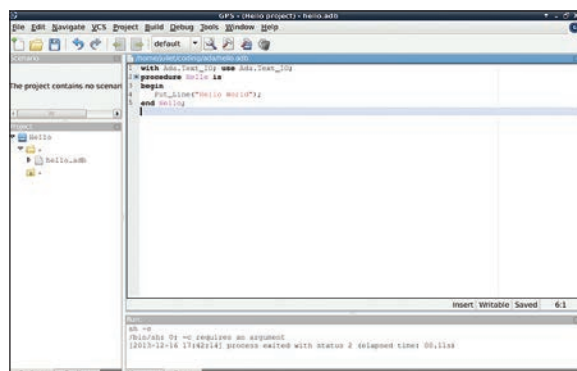
Ada — структурированный, статически типизированный и объектно-ориентированный язык со встроенной поддержкой параллелизма, многозадачности и синхронных сообщений. Благодаря особым требованиям к безопасности (а также наличию компилятора с обширными проверками и контролю ошибок во время выполнения) язык часто применяется в проектах, когда нужно минимизировать риски. Он все еще широко распространен в транспортной и космической отраслях, а также в медицине.

С момента выхода исходной версии было еще три обновления: Ada 95, Ada 2005 и Ada 2012. Исходная версия Ada уже была в некоторой степени объектно-ориентированной, но в Ada 95 появились возможности, сделавшие язык полностью объектно-ориентированным. В Ada 2005 улучшилась поддержка объектно-ориентированности и систем реального времени, а в Ada 2012 и то, и другое улучшили в очередной раз. Компилятор GNAT поддерживает все версии языка. В этом руководстве мы формально воспользуемся Ada 2012, но не будем касаться никаких новых возможностей. Мы познакомимся с языком, а затем напишем короткую программу для записи и отслеживания своих новых привычек или планов на текущий год.

## Начинаем

В Linux установка не представляет сложностей: *gnat* скорее всего есть в вашем менеджере пакетов. После установки напишем свою первую программу. Создайте файл **hello.adb**:

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Hello is
begin
  Put_Line ("Hello World");
```



➤ Если вам нужна графическая среда разработки на Ada, можете воспользоваться GPS (*gnatgps*).

```
end Hello;
```

Скомпилируйте файл командой **gnatmake hello.adb** и запустите его командой **./hello**. Ваша первая программа на Ada готова! Код на Ada может поддерживаться очень долго, причем не первоначальным разработчиком, а другими людьми, поэтому важно, чтобы он был хорошо читаемым.

*Ada.Text\_IO* — один из стандартных пакетов, который поддерживает, как нетрудно догадаться, ввод/вывод текста. Оператор **with** импортирует пакет, а **use** делает его видимым, позволяя вам пользоваться функциями этого пакета, такими как **Put\_Line()**. В Ada поощряется модульное программирование, поэтому почти во всех ваших программах будут использоваться пакеты.

После этого мы объявляем главную процедуру программы. Ее можно назвать как угодно (не обязательно **main**), но лучше, чтобы ее название совпадало с именем файла. Оператор между **begin** и **end** завершается точкой с запятой; после **end Hello**; тоже есть точка с запятой, завершающая определение процедуры.

## Чтение и запись

Пора перейти к полноценной программе. Наступивший год еще только в первой четверти, и мы напишем программу для записи и отслеживания своих новых (и надеюсь, хороших) привычек. Сначала нужно ввести строку (нашу новую привычку) с клавиатуры и вывести ее на экран. Создайте файл **habits.adb**:

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Habits is
  habit : String(1..10);
begin
  Get(habit);
  Put(habit);
  New_Line;
end Habits;
```

В первой строке процедуры объявляется переменная **Habit** [англ. Привычка] типа **String** [строка] и длиной 10. В коде процедуры, **Get()** записывает ввод из стандартного потока в указанную переменную, **Put()** выводит ее в стандартный вывод, а **New\_Line** переводит курсор на новую строку.

Скомпилируйте и запустите программу, введите строку, например, "floss teeth [чистить зубы зубной нитью]", и строка будет выведена на экран. Правда, на экране появится только "floss teet". В чем же дело? А видите объявление переменной типа **String** в начале процедуры? В нем создается строка **habit** длиной ровно 10 символов, и из того, что вы ввели, программа сохраняет только первые 10 символов. Однако если указать, например, 50, вы обнаружите, что после ввода строки и нажатия **Enter** программа ничего не делает. Она ждет, пока вы не введете 50 символов: строки в Ada должны точно соответствовать своей длине.

На эту проблему в Ada есть две стандартные библиотеки, *Ada.Strings.Bounded* и *Ada.Strings.Unbounded*. Для строк ограниченной длины [bounded] указывается максимальный размер, и они

работают быстрее; у строк неограниченной длины [unbounded] максимального размера нет, и они гораздо более гибки, но работают чуть медленнее, так как выделяются динамически.

```
with Ada.Text_IO, Ada.Strings.Unbounded, Ada.Strings.Unbounded.Text_IO;
use Ada.Text_IO, Ada.Strings.Unbounded, Ada.Strings.Unbounded.Text_IO;
procedure Habits is
  habit : Unbounded_String;
begin
  habit := Get_Line;
  Put_Line(habit);
  New_Line;
end Habits;
```

**Get\_Line** и **Put\_Line()** — аналоги **Get()** и **Put()** для строк неограниченной длины. В функции **Get\_Line** аргумент не передается в качестве переменной: вместо этого используется оператор присваивания :=. Двоеточие используется при объявлении переменных, а := — для присваивания. Запустите программу, и вы сможете ввести текст любой длины; и он будет выведен.

Хорошую привычку хотелось бы сохранить, поэтому попробуем вывести ее в файл, а не на экран.

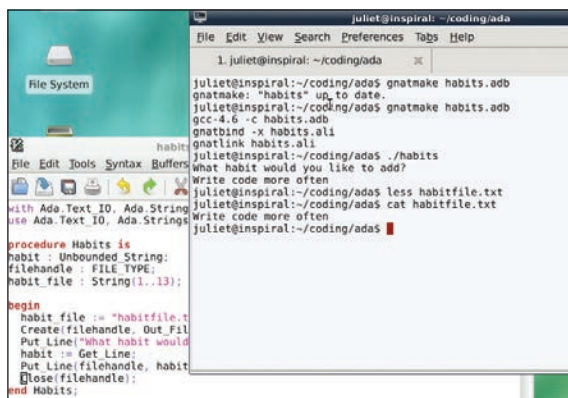
```
procedure Habits is
  filehandle : File_Type;
  habit_file : String(1..13);
begin
  habit_file := "habitfile.txt";
  Create(filehandle, Out_File, habit_file);
  habit := Get_Line;
  Put_Line(filehandle, habit);
  Close(filehandle);
end Habits;
```

Функция **Create()** создает файл, указанный в переменной **Out\_File**, и записывает его дескриптор в **filehandle**. Затем получаем строку ввода, выводим ее в файл, на сей раз указав второй аргумент в **Put\_Line()**, и закрываем файл.

Но если запустить программу снова, она перезапишет исходный файл, а мы-то хотим добавить данные к содержимому файла:

```
procedure Open_Datafile(File : in out File_Type; Name : in String) is
begin
  Open(File, Append_File, Name);
exception
  when Name_Error =>
    Create(File => File, Mode => Append_File, Name => Name);
end Open_Datafile;
begin
  Open_Datafile(filehandle, habit_file);
  -- remove Create line, rest is as before.
end Habits;
```

В Ada можно создавать подпрограммы внутри основной программы. Более подробно о процедурах и функциях можно узнать



» Также стоит вывести несколько подсказок пользователю, как показано на рисунке.

во врезке (на нашем уроке будут только процедуры). Наряду с конструкцией **begin ... end;** можно пользоваться и **begin ... exception ... end;**. В ней отлавливаются все возникшие исключения и выполняется указанный код. В данном случае мы отлавливаем исключение **Name\_Error**, которое возникает в функции **Open()**, если указанного файла не существует, и создаем новый файл для записи.

Вызов **Create()** чуть отличается от предыдущего. Такие 2 строки кода при компиляции и запуске дадут одинаковый результат:

```
Create(File, Append_File, Name);
Create(File => File, Mode => Append_File, Name => Name);
```

Однако вторая строка лучше читается, так как для каждого параметра в первой части указано имя параметра, а во второй — значение. Скомпилируйте и запустите программу, и теперь данные будут добавляться в файл.

## Циклы

Теперь можно добавить привычку; но ни одной из предыдущих привычек мы не видим. Поэтому теперь нужно написать подпрограмму, которая считывает данные из файла.

```
procedure Read_In_Datafile(File : in out File_Type; Name : in String) is
  Buffer : String(1..100);
  Length : Natural;
begin
  Open(File => File, Mode => In_File, Name => Name);
  While not End_Of_File(File => File) loop
    Get_Line(File => File, Item => Buffer, Last => Length);
    Put_Line(Buffer(1..Length));
  end loop;
  Close(File => File);
end Read_In_Datafile;
```

Тип **Natural** — это натуральное целое число (один из встроенных типов). Цикл **While** пояснять не нужно: мы перебираем файл построчно до тех пор, пока не дойдем до конца файла.

У **Get\_Line()** на сей раз больше аргументов: дескриптор файла, строка, в которую нужно записать данные, и номер позиции последнего символа в строке (т. е. длина строки, которую мы только

## Скорая помощь

Сделать объявление пакетов видимыми можно и другими способами, включая переименование и тип использования, но на нашем уроке мы будем придерживаться **use**.

## Подпрограммы: процедуры и функции

В Ada есть два типа подпрограмм: процедуры, которые значения не возвращают, и функции, которые возвращают значение. Базовая структура подпрограммы, процедура это или функция, не отличается от структуры главной процедуры: в начале объявляются переменные, за ними идет блок **begin/end**.

В процедурах (и функциях) есть три режима входных переменных:

» **in** означает, что параметр изменить нельзя; внутри процедуры он считается константой.

» **in out** означает, что параметр можно изменить. В процедуре такой параметр считается переменной, и любые изменения с ним сохраняются по выходу из процедуры.

» **out** означает, что значение будет присвоено параметру в процедуре, а его предыдущее значение

никакой роли не играет. Так можно «возвращать» значение из процедуры.

В функции **Open\_Datafile()** в основной статье есть функция **Open**. Этой функции требуется, чтобы ее параметр **File** можно было изменять, поэтому мы указываем для него режим **in out**.

А параметр **Name** можно рассматривать как константу.

## Массивы в Ada

Массивы в Ada обладают довольно большими возможностями. Массив можно создать из элементов любого типа, но все элементы должны быть однотипны. Аналогично, за индекс можно взять любой дискретный тип, но массив имеет фиксированный размер, изменить который нельзя. Это означает, что индексами могут быть буквы или даже слова. Так, количество животных у вас дома можно отслеживать с помощью следующего кода:

```
type Animals is (Dog, Cat, Rat);
My_Animals : array (Animals) of Natural;
My_Animals(Dog) := 1;
My_Animals(Cat) := 2;
My_Animals(Rat) := 50;
```

Для доступа к элементу массива надо указать его индекс в круглых скобках. Если указанный индекс выходит за пределы массива, формируется ошибка `Constraint_Error`. Массивы можно разделять на поддиапазоны или объединять (оператором `&`), и они могут быть многомерными. Размер массива остается постоянным, что бы вы с ним ни делали, а при создании нового массива из старого нужно правильно указать размер нового массива.

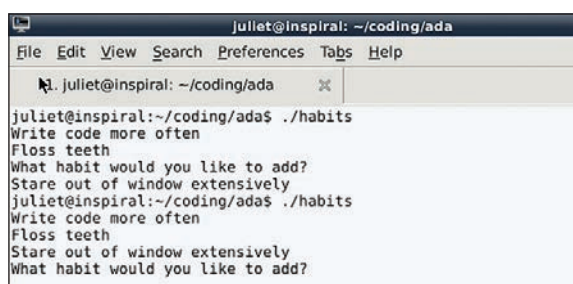
У массивов есть четыре особенно важных атрибута — **'First**, **'Last**, **'Length** и **'Range**, их назначение легко понять по названиям [Первый, Последний, Длина, Диапазон].

что прочитали). Это означает, что мы берем только ту часть переменной **Buffer**, которая содержит полезные данные. Если вывести **Buffer** полностью, вы получите тот мусор, который есть в конце введенной строки. Добавьте соответствующий вызов в главную процедуру и попробуйте. Пока мы записывали данные в строку; но теперь хотим все немного усложнить. Наряду с ведением списка привычек мы хотим следить за тем, выполняем ли мы эти действия каждый день. Поэтому, кроме самой привычки, мы должны записать день, в который мы обновляем данные, и количество дней в строке, которые обновили. Для этого воспользуемся составным типом `Ada.Record`. Сначала определим нашу запись:

```
with Ada.Calendar;
use Ada.Calendar;
procedure Habits is
type Habit_Record is
record
Habit_Name : Unbounded_String;
Date : Time;
Streak : Integer;
end record;
```

Итак, у нас есть название привычки, дата последнего обновления и продолжительность периода (количество последовательных дней, в течение которых мы придерживались этой привычки) с последнего обновления. Давайте создадим и выведем новую запись.

```
with Ada.Integer_Text_IO; use Ada.Integer_Text_IO;
procedure Habits is
-- as before
s : String(1..1);
procedure Output_Habit(Habit : in Habit_Record) is
Year, Month, Day : Integer;
Seconds : Day_Duration;
begin
Put_Line(habit.Habit_Name);
Split(habit.Date, Year, Month, Day, Seconds);
```



```
Put(Day, 3);
Put(Month, 3);
Put(Year, 5);
Put(habit.Streak, 3);
end Output_Habit;
begin
Put_Line("Input habit");
habit.Habit_Name := Get_Line;
habit.Date := Clock;
Put_Line("A сегодня вы это сделали? (y/n)");
s := Get_Line;
if (s = "y") then habit.Streak := 1;
else habit.Streak := 0;
end if;
Output_Habit(habit);
end Habits;
```

В процедуре **Output\_Habit()** дата разбивается на компоненты для вывода с помощью библиотеки `Ada.Integer_Text_IO`. К полям записи мы обращаемся с помощью конструкций **habit.Habit\_Name** и т.п. **Clock** возвращает текущую дату и время. Вы также видите блок **if ... then ... else ... end if**. Скомпилируйте и запустите программу, чтобы считать новую запись и вывести ее на экран.

После этого изменения мы уже не можем вывести данные в файл. Чтобы это исправить, можно разобрать запись на поля, записать их в текстовый файл и затем собрать запись при считывании. Но лучше работать с записью как с двоичными данными и записывать ее в файл целиком как объект.

```
with Ada.Directories; use Ada.Directories;
with Ada.Streams.Stream_IO; use Ada.Streams.Stream_IO;
procedure Habits is
filehandle : Ada.Streams.Stream_IO.File_Type;
fileaccess : Ada.Streams.Stream_IO.Stream_Access;
procedure Output_Habit_Records(File : in out Ada.Streams.Stream_IO.File_Type;
Name : in String) is
begin
Ada.Streams.Stream_IO.Create(File, Out_File, Name);
fileaccess := Ada.Streams.Stream_IO.Stream(File);
Habit_Record'Write(fileaccess, habit);
Close(File);
end Output_Habit_Records;
```

Записывать двоичные данные можно по-разному (см. врезку), но мы воспользуемся `Stream_IO`: это самый гибкий метод, и он работает с неограниченными строками. Нам нужно определить как переменную файлового дескриптора (но на сей раз с помощью библиотеки `Stream_IO`), так и поток, который будет использоваться после открытия файла. Как и прежде, мы создаем новый файл, но на этот раз с помощью библиотеки `Stream_IO`. Параметры надо указать полностью, чтобы гарантировать, что мы используем правильную версию **Create()**. Затем мы создаем поток для этого дескриптора файла. При записи в поток нужно указать тип объекта, который записывается — в данном случае, **Habit\_Record**. Атрибут **Write** преобразует объект в формат, пригодный для записи в поток. Укажите поток и сами данные в качестве аргументов — и все готово, данные записаны, и файл можно закрыть.

Добавьте строку с вызовом этой процедуры в основной блок программы, скомпилируйте и запустите ее, и вы увидите, что теперь вы можете записать таинственный двоичный файл со своими данными. Однако у нас остались две проблемы: данные нельзя прочитать снова, и каждый раз при добавлении новых данных мы будем перезаписывать старые. Добавим функцию для чтения существующего файла и для записи всех данных в новый файл.

```
with Ada.Directories; use Ada.Directories;
procedure Habits is
type Record_Array is array(1 .. 100) of Habit_Record;
```

### Скорая помощь

С помощью конструкции `Object'Attribute` можно получить или задать информацию о типе. С помощью конструкции `Object'Write` можно получить информацию в форме, годной для записи в поток.

➤ Однако теперь ваши привычки будут не просто выводиться — вас каждый раз будут просить добавить новую привычку.



```

records : Record_Array := Record_Array'(others =>
  (Streak => -1,
   Habit_Name => To_Unbounded_String(""),
   Date => Clock));
counter : Natural;
procedure Read_In_Habit_Records(File : in out Ada.Streams.
Stream_IO.File_Type; Name : in String) is
begin
  Open(File, In_File, Name);
  fileaccess := Ada.Streams.Stream_IO.Stream(File);
  counter := 1;
  While not End_Of_File (File) loop
    Habit_Record'Read(fileaccess, habit);
    Output_Habit(habit);
    records(counter) := habit;
    counter := counter + 1;
  end loop;
  Close(File);
end Read_In_Habit_Records;
procedure Get_New_Habit is
begin
  -- break out 'new habit' code into its own procedure, and add
  a line:
  records(counter) := habit;
end Get_New_Habit;
procedure Output_Habit_Records( *args as before* ) is
begin
  -- create file and stream access as before
  for H in records'Range loop
    if (records(H).Streak /= -1) then
      Habit_Record'Write(fileaccess, records(H));
    end if;
  end loop;
  Close(File);
end Output_Habit_Records;
begin
  if(Exists(habit_file)) then
    Read_In_Habit_Records(filehandle, habit_file);
  end if;
  Output_Habit_Records(filehandle, habit_file);
end Habits;

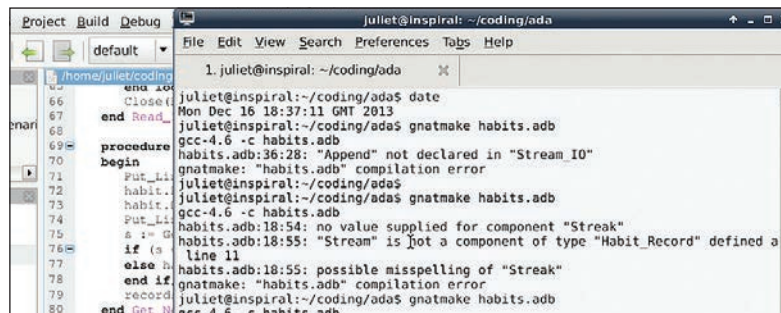
```

Сначала мы создаем массив **Habit\_Records** для считываемых данных, для чего надо создать новый тип. Затем создаем массив этого типа, инициализируем все его элементы пустыми строками, задаем продолжительность **-1** и текущую дату в качестве даты.

При инициализации массива в Ada можно инициализировать элементы массива с помощью списка. В Ada95 появился атрибут **others**, который позволяет присвоить одно указанное значение всем остальным элементам массива. Его нужно использовать последним после всех списков. Если отдельных значений нет, **others** применяется ко всему массиву. Без инициализации значения элементов массива могут быть произвольными.

Функция **Read\_In\_Habit\_Records()** делает что написано на упаковке. Мы открываем файл, устанавливаем счетчик в 1, затем считываем каждую запись, выводим ее и добавляем ее в массив записей, после чего увеличиваем счетчик. Обратите внимание на **While** и **End\_Of\_File** — с потоком они работают точно так же. Мы также добавили в **Get\_New\_Habit()** строку для добавления нового элемента в конец массива.

Использование счетчика **counter** означает, что перед вводом новых записей нужно считать старые, в противном случае данные в массиве будут перезаписаны. Более гибкий вариант — создать отдельный массив и объединить этот массив с существующим, оператором **&** при повторной записи. Для вывода записей нужно пройти весь массив в цикле. **arrayname'Range** — атрибут,



```

juliet@inspiral: ~/coding/ada
1 juliet@inspiral: ~/coding/ada
gcc-4.6 -c habits.adb
habits.adb:36:28: "Append" not declared in "Stream IO"
gnatmake: "habits.adb" compilation error
juliet@inspiral:~/coding/ada$ gnatmake habits.adb
gcc-4.6 -c habits.adb
habits.adb:18:54: no value supplied for component "Streak"
habits.adb:18:55: "Stream" is not a component of type "Habit_Record" defined a
line 11
juliet@inspiral:~/coding/ada$ gnatmake habits.adb
gcc-4.6 -c habits.adb

```

представляющий свободные индексы в массиве (если **Range** применить к типу, он представляет значения, доступные для данного типа). Это позволяет по сути превратить его в цикл **foreach** над массивом. В цикле с помощью инициализации мы проверяем, содержит ли эта запись «настоящее» значение. Это позволяет справиться с недостатком простых динамических массивов — без этой проверки мы бы вывели все привычки, которые уже есть, и набор случайных данных для остальных элементов массива до согого.

Перед вызовом **Read\_In\_Habit\_Records()** проверяем, что файл существует. Метод **Exists()** находится в библиотеке *Ada.Directories*.

## Объединяем все вместе

Чтобы сделать программу удобнее, добавим меню с вариантами действий при запуске и возможность добавить один день к продолжительности повторения привычки.

```

procedure Habits is
  result : String(1..1);
procedure Menu(Result : out String) is
begin
  Put_Line("Choose an option:");
  Put_Line("1 to show current habits and edit");
  Put_Line("2 to show current habits and add a new habit");
  Result := Get_Line;
end Menu;
procedure Ask_For_Update is
  choice : Integer;
begin
  Put_Line("Enter a number to add one to your streak, or 0 to
quit");
  choice := Integer'Value(Get_Line);
  if (choice = 0) then
    Put_Line("Quitting");
  else
    if (records(choice).Streak /= -1) then
      records(choice).Streak := records(choice).Streak + 1;
      records(choice).Date := Clock;
    end if;
  end if;
end Ask_For_Update;
-- add call to Menu(result) in main procedure; см. код на диске.

```

Процедура **Menu()** довольно проста: единственная интересная часть — возвращение значения с помощью **Result**.

**Ask\_For\_Update()** спрашивает, надо ли добавить один день к продолжительности повторения существующей привычки. Но **Get\_Line** возвращает строку, а для индекса массива нужно целое число. В отличие от некоторых современных языков, компилятор Ada не выполнит этого преобразования автоматически. Получить целое число из строки придется вызовом **Integer'Value()**. Затем мы проверяем, что данная привычка существует, обновляем продолжительность повторения и время последнего обновления.

Теперь программа делает все, что нам требовалось, хотя многое еще можно улучшить. **LXF**

► Обратите внимание на ошибки, выведенные компилятором в окно терминала: мою опечатку "Streak" вместо "Stream" и неудачный эксперимент с Append.

# Scratch: НЫРНЕМ

**Лес Паундер** проходите по основам Scratch для тех, кому не терпится приступить к кодированию или кто ищет способ учить кодированию других.



**Наш эксперт**

**Лес Паундер** обожает эксперименты со всеми новыми языками и гаджетами и считает jQuery огромным удовольствием.

Scratch использует пользовательский интерфейс с тремя столбцами, слева направо: Block Palette, Script Area и The Stage.

**К**ак вы учились программировать? Обычно мы вспоминаем мерцающий экран и человека перед ним, который одним пальцем медленно печатает магические слова, изначально для него бессмысленные. На заре программирования именно так обычно и выглядел этот процесс для энтузиастов, которые задалбливали язык, вводя длинные простыни кода, напечатанного в журналах.

В наше время, когда детей принимают учиться программировать в начальной школе, мы наблюдаем появление новых инструментов, чья цель — познакомить с программированием юное поколение, и самый популярный инструмент как раз и является темой нашего урока.

Scratch, созданный MIT — это визуальная среда программирования, которая предлагает использовать цветные блоки вместо фрагментов кода. Каждый набор блоков предлагает разные функции, и обучение концепции программирования проходит в весьма интересной и забавной манере. Дети могут использовать Scratch уже в возрасте шести лет, и ныне он активно используется в Великобритании как часть учебного плана из 3-х этапов (для возрастов 6 до 14 лет) и как часть деятельности Клубов по коду.

На нашем уроке мы используем текущую стабильную версию Scratch; на момент написания это 1.4. Версия 2 Scratch имеется в виде бета, и если верить отчетам, она весьма стабильна



Рис. 1.0. Логический блок для нашего спрайта — LXF-лого.

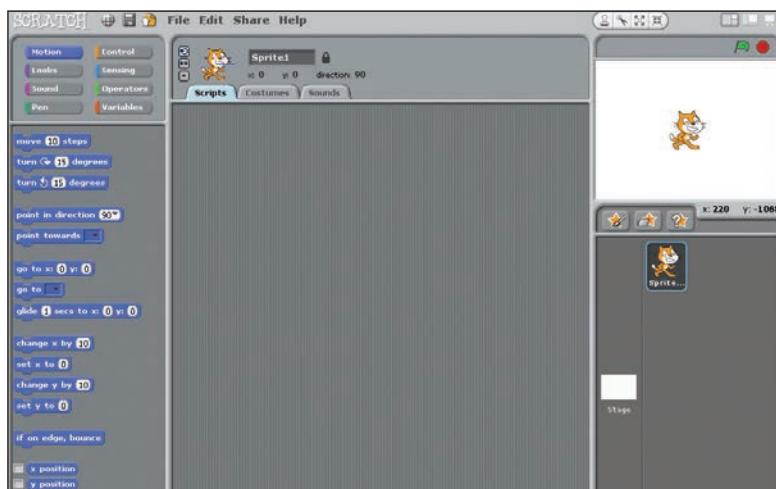


Рис. 1.2. Это код, который сбрасывает эффекты.

в использовании, однако по сравнению с версией 1.4 есть ряд различий в расположении блоков.

Как уже упоминалось, Scratch использует систему блоков, чтобы научить пользователей, как в программе работают разные функции. Ее можно разделить на следующие группы и цвета:

- » **Движение [Motion] (Синий)** Позволяет перемещать и контролировать графические объекты-спрайты [sprite] в игре.
- » **Управление [Control] (Оранжевый)** Эти блоки содержат логические узлы для управления вашей программой (циклы и инструкции) и триггерные события, нужные для запуска действий, такие как нажатие на клавишу. В Scratch 2.0 события, хранящиеся в Control, имеют собственную группу под названием Events [События].
- » **Внешний вид [Looks] (Темно-лиловый)** Эти блоки могут изменять цвет, размер или стиль спрайта, и вводят интерактивные элементы, такие как выноски с текстом.
- » **Уточнение состояния [Sensing] (Голубой)** Для уточнения состояния обрабатывается общая информация, необходимая вашей программе, например, нажатие клавиши, определение столкновения спрайтов и положение спрайта на экране.
- » **Звук [Sound] (Светло-лиловый)** Добавляет музыку и звук в вашу программу.
- » **Операторы [Operators] (Зеленый)** Позволяет использовать в своей программе математическую логику, например, Булеву, условные конструкции и случайные числа.
- » **Ручка [Pen] (Темно-зеленый)** Это для рисования на экране, во многом так же, как это делается с помощью Logo или Черепашки [Turtle].



## Scratch онлайн

Scratch доступен на многих платформах. Он предзагружен на каждый Raspberry Pi с Raspbian, и доступен для скачивания с <http://scratch.mit.edu>. Но знаете ли вы, что есть и онлайн-версия, со всеми теми же функциями, что и версия рабочего стола, но не требующая ни скачивания, ни установки?

Перейдите на сайт MIT (выше), и вы увидите тот же самый интерфейс и те же блоки кода, хотя

и с небольшими отличиями. Это самая последняя версия Scratch, и вы можете сохранить в облаке всю свою работу, готовую к использованию на любом ПК, который вам встретится.

Если вам захочется использовать код, созданный онлайн, на ПК без соединения с Интернетом, скачайте свой проект и преспокойно запускайте его у себя на настольном компьютере.

Можете также скачать самую свежую версию Scratch с сайта. Она пока что в стадии бета, но, по сообщениям, весьма стабильна и готова к использованию. Scratch 2.0 использует Adobe Air и доступен для всех платформ, включая Linux. Adobe прекратил поддержку Air для Linux несколько лет назад, но вы все равно можете скачать пакеты, необходимые для работы Scratch 2.0.



► Рис. 1.1. Код, используемый для того, чтобы Мэтт получал оповещение.

► **Block Palette [Палитра блоков]** Здесь хранятся и сортируются по функциям наши блоки кода.

► **Script Area [Область скрипта]** Во втором столбце находится область, куда вы можете перетащить наши блоки кода из Block Palette, чтобы добавить код в программу.

► **The Stage [Сцена]** Третий и последний столбец показывает результаты вашего программирования и может использоваться для взаимодействия с миром игры. В низу этого столбца вы найдете весьма удобную панель Sprites Pane. Она показывает спрайты и основное имущество, которые принадлежат именно вашей программе. Нажатие на спрайт изменит фокус на нем, позволяя вам написать код только для этого спрайта.

## Создаем свою игру

Чтобы написать код в Scratch, нам нужно переместить блоки «кода» из Block Palette в Script Area для каждого спрайта или фона, который мы хотим использовать в нашей программе.

В данном руководстве мы создадим игру-викторину, а Мэтт [Matt] и Нейл [Neil] будут нашими ведущими. Цель игры — набрать более 3 очков, и вы должны правильно ответить на вопрос, чтобы перейти в следующий раунд. Если у вас три неверных ответа в игре, то игра заканчивается. Каждый из наших спрайтов имеет собственный скрипт, срабатывающий, когда его запускает событие. Событие — это обычно щелчок по зеленому флажку, чтобы начать игру, но бывают события и посложнее, вовлекающие необходимость слушать, что выкинул триггер от другого спрайта.

Мы дали нашим четырем игровым триггерам имена: Нейл — это Sprite6, Мэтт — Sprite7, логотип *Linux Format* — LXF Logo — Sprite5, и изображение Game Over — Sprite8. С каждым из этих спрайтов связан собственный скрипт, так что давайте посмотрим, что делает каждый из них, начиная с LXF Logo. Я хотел, чтобы наш логотип появлялся прямо в начале игры, но чтобы его было видно не сразу, поэтому мой логический блок был таким:

```
When Green Flag clicked.
Hide.
Wait for 2 seconds.
Show.
Double in size.
Loop 10 times and each time reduce size by 15 %
```

То, как это визуально представляет Scratch, можно увидеть на рис. 1.0.

Теперь давайте посмотрим на логический блок спрайта Мэтта. У Мэтта есть пять разделов кода, которые реагируют на то, что именуется broadcast [оповещение], то есть способ соединения

► **Переменные [Variables] (Темно-оранжевый)** Создает контейнеры, которые могут хранить данные в вашей программе, и работает с ними.

Превращая язык в цветные блоки кода, Scratch позволяет быстро определить блок, который вам нужен. Дети обычно работают сначала через систему цветного кода, а затем через естественный игровой процесс понимают связи между блоками и их взаимодействие.

## Среда

Scratch использует понятную и структурированную раскладку, разделенную на три столбца.

Первый столбец —

триггерных событий между спрайтами в программе (вы найдете broadcast в Управлении). И так, нам нужно будет вызвать ряд событий, как только мы получим эти оповещения от Нейла и the Stage.

► **player\_name** The Stage отправляет оповещение, как только игрок водит свое имя.

► **support** Нейл отправляет оповещение Мэтту, как только игрок правильно отвечает на вопрос.

► **insult** Нейл шлет оповещение Мэтту, чтобы подразнить игрока.

► **Score** Нейл отправляет оповещение Мэтту, чтобы Мэтт сообщил игроку счет.

► **game\_over** Нейл отправляет оповещение Мэтту, чтобы завершить игру.

Как мы можем видеть (рис. 1.1, слева), для каждого из этих оповещений Мэтт вынужден запустить определенную последовательность кода. У Мэтта также есть скрипт, который запускается, если событием является нажатие на зеленый флаг. Рис. 1.2 (см. слева, стр. 88) показывает код, который удаляет все спецэффекты, использованные в его спрайте.

Перейдем к Нейлу. В этом спрайте намного больше кода, чем у Мэтта, потому что он — основная часть нашей игры.

Во-первых, есть событие Green Flag [Зеленого флажка]. И оно заставляет спрайт Нейла перезапускать любые используемые спецэффекты, а затем сохранять 0 в переменной под названием guesses [попытки] (см. рис. 1.3, сверху справа).

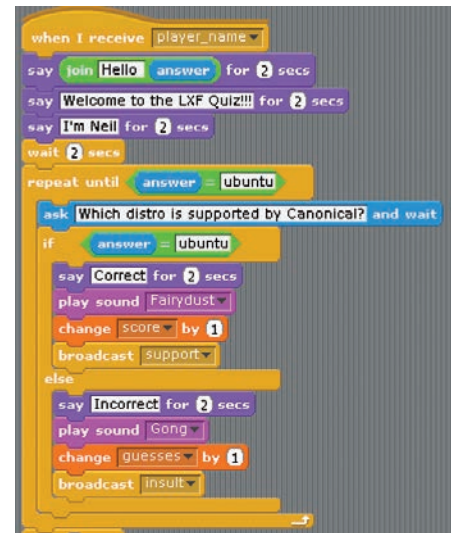
Вторая часть — это замкнутый цикл, который управляет нашей игрой. Этот большой цикл запускается, когда мы получаем имя игрока, которое обрабатывается через код, связанный с The Stage, к чему мы обратимся позднее.

Я разбил цикл Мэтта на три части. В начале игры есть раздел кода, связанный с The Stage, он спрашивает имя игрока. Затем имя запоминается как переменная под названием **Answer**, которая хранится в категории блоков Sensing. Когда пользователь вводит свое имя, скрипт, связанный с the Stage, отправляет оповещение под названием **player\_name**. Код Нейла ждет этого оповещения в качестве триггерного события. Когда код будет запущен, Нейл поздоровается с игроком, назвав его по имени на основании переменной **Answer**.


Разобравшись со всеми формальностями, перейдем к основному циклу, который управляет первым вопросом. Мы связываем



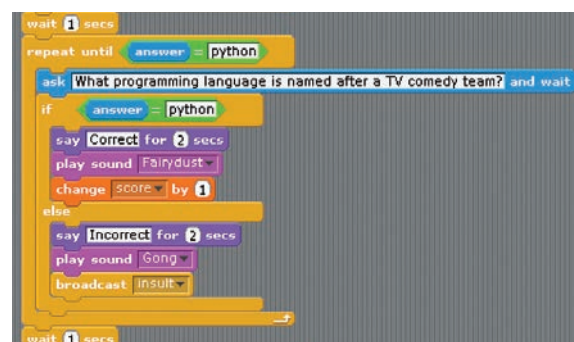
► Рис. 1.3. Это перезапускает спрайт Нейла и переменную guesses.



► Рис. 1.4. Часть основного кода игры.

**Скорая помощь** 

Щелчок правой кнопкой по блоку кода — быстрый способ продублировать или удалить ваш код.



► Рис. 1.5. Второй цикл кода Нейла внутри основного цикла.

## Скорая помощь

Блоки, которые можно соединять, имеют вокруг белый «ореол», который показывает, что их можно объединить.

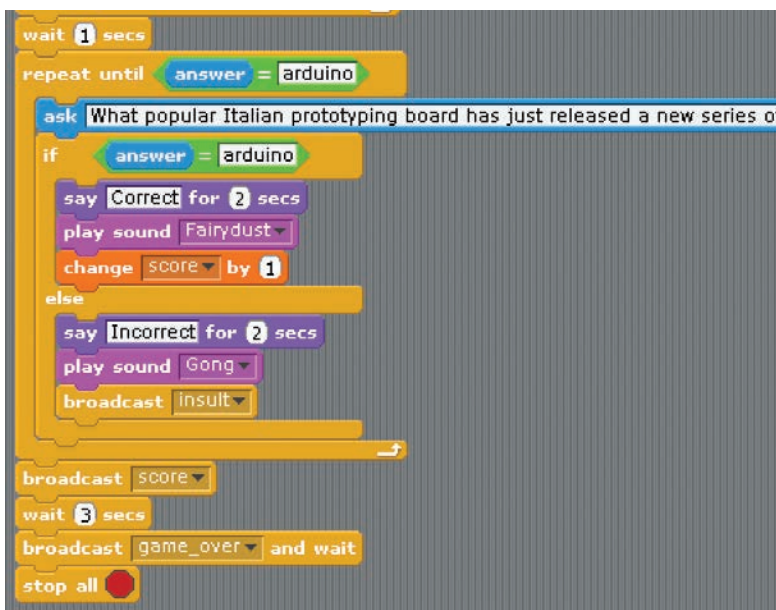
вопрос с циклом, который будет повторяться, пока не будет дан правильный ответ, т.е. **repeat until answer = ubuntu**.

Затем мы задаем вопрос, и запускаем второй цикл внутри основного цикла (см. рис. 1.5 на стр. 89 внизу). Этот второй цикл — утверждение **if**, которое выполняет некую последовательность кода, если условие истинно, а если оно ложно, то запускается код, связанный с **else**. Если ответ правильный, Нейл скажет, что ответ правильный, зазвучат фанфары в честь игрока, к переменной **score** добавится 1 очко и, наконец, отправится оповещение Мэтту, который скажет что-нибудь приятное. Если игрок дает неверный ответ, то Нейл скажет: «Неверно», будет воспроизведен звуковой эффект гонга, переменная **guesses** увеличится на 1, а затем будет отправлено оповещение Мэтту, который отпустит колкость в адрес игрока.

Вторая часть кода в точности такая же, как и основной цикл первой части, и причина в том, что мы дублировали код и изменили вопрос и ожидаемый ответ. Чтобы продублировать код в Scratch, можно просто щелкнуть правой кнопкой по блоку кода и выбрать **duplicate**, и вот так быстро вы продублировали свой код всего одним щелчком.

Первый раздел третьей части в точности такой же, как основной цикл из первых двух частей. Итак, давайте перейдем к последним четырем блокам кода (см. рис. 1.6 ниже). Наш первый

► Рис. 1.6. Последняя часть основного кода, связанная с Нейлом.



► Рис. 1.7. The Stage содержит спрайты, но в ней также имеются собственные скрипты.

блок отправляет Мэтту оповещение 'score', чтобы Мэтт сообщил нам счет, потом мы ждем три секунды, пока Мэтт закончит говорить. Затем мы отправляем Мэтту другое оповещение, которое старательно запустит окончание кода игры, связанного с этим оповещением. И, наконец, мы используем 'stop all', чтобы остановить все скрипты в игре.

Событие Green Flag — это код, управляющий количеством ответов, которое имеется у игрока. Мы прибегли к условной логике, чтобы сказать, что когда количество ответов равно 3, то мы передаем Мэтту оповещение **game\_over**, которое, в свою очередь, запускает завершение скрипта игры.

Со спрайтом Game Over связаны два скрипта. Первый просто прячет спрайт при нажатии на зеленый флажок. Второй скрипт запускается при отправке оповещения **game\_over**. Он заставляет спрайт проявиться и устанавливает его размер на 100%. Затем мы используем цикл из 10 повторов, чтобы изменить размер и поворот спрайта, что дает нам эффект увеличения и поворота — в точности как в классических 8-битных играх 1980-х.

## Весь мир — The Stage

The Stage — не только дом для наших спрайтов; здесь могут также содержаться собственные скрипты (см. рис. 1.7, вверху). Для нашей игры в The Stage у нас есть два раздела кода. Оба запускаются событием нажатия на Green Flag. Первая часть перезапускает две переменных под названием **guesses** и **score**. Затем мы просим игрока указать свое имя, которое затем передает Мэтту и Нейлу и запускает основной цикл кода, привязанный к Нейлу. Второй раздел кода является бесконечным циклом, который будет постоянно воспроизводить цикл DrumMachine и установит громкость на 50%.

## Концепции программирования

Использование Scratch — большое удовольствие, но отдаете ли вы себе отчет в том, что вы на самом деле учитесь программировать? Неважно, какой язык вы используете — у базовой концепции программирования весьма прочный фундамент. И, усвоив его однажды, его можно применять ко всем программным проектам. Основные концепции таковы:

- **Последовательности [Sequences]** Серии задач для выполнения в определенном порядке. Например, действия, необходимые при прохождении лабиринта.
- **Циклы [Loops]** Способ повторения последовательности. Они могут работать постоянно (**while true**) или управляться с помощью утверждения (**for x in range(0,3)**). В нашей игре мы использовали

множество циклов для управления результатами игроков.

- **Параллелизм [Parallelism]** Принцип запуска более одной последовательности кода в одно и то же время. Мы многократно использовали его в нашей игре Scratch, поскольку в каждом спрайте имеется собственный код, работающий параллельно кодам других спрайтов.
- **События [Events]** Триггер, запускающий последовательность кода; самое заметное событие в нашей игре на Scratch — щелчок по зеленому флажку для запуска игры.
- **Данные [Data]** Мы используем переменную для хранения значения нашего счета, и позднее можем загрузить и обработать счет, чтобы показать прогресс участников игры.

► **Операторы [Operators]** Основные математические действия, которые мы изучали в школе. Операторы применимы к тексту и числам, что позволяет выполнять вычисления в нашем коде и при необходимости итерировать данные.

► **Условные конструкции [Conditionals]** Образуют основу наших логических блоков и дают способ сравнения данных с информацией, предоставленной игроком.

Мы использовали в нашей игре условные конструкции для сравнения данного ответа с ожидаемым ответом. Если они совпадают, что в Булевой логике именуется истиной [True], то игроку начисляется очко. Если они не совпадают, что будет определено как Ложь [False], то игроку придется попробовать снова.

Как я уже говорил, переменные — это отличный способ хранения данных. Но прежде чем использовать их, надо их создать. Для создания переменной служит кнопка Variables в Block Palette. Там вы найдете кнопку Make a variable [Создать переменную] — щелкните по ней. И вы увидите рис. 1.9 (см. справа).

В нашей игре мы использовали две переменных, **score** и **guesses**, и они обе должны быть доступны для всех спрайтов, чтобы и Мэтт, и Нейл могли их использовать. Создав их, мы можем легко вставить эти переменные в наш код, позволяя использовать их значения много раз в игре, см. рис. 1.8 внизу — это пример, который мы сделали для нашей игры.

## Псевдокод

Пытаясь понять логику нашей игры, мы хотим написать псевдокод. «Псевдо — что?» — спросите вы. Это когда вы записываете логику работы вашей программы. Давайте рассмотрим простой пример:

```

a равно 0
До тех пор, пока a меньше 10:
  Печатать на экране значение a
  Увеличить a на 1
Итак, вот вам и псевдокод, но как выразить его на языке программирования? Сначала давайте сделаем это с помощью Scratch, а затем Python. В Scratch наш код будет выглядеть так:
[When Green Flag is clicked]
Set variable a to 0
forever if a < 10
say a for 2 secs
  
```



► Рис. 1.8. Как видите, в нашей игре две переменных: **guesses** и **score**.



► Рис. 1.9. Нажмите на кнопку Make a Variable, чтобы создать переменную.

```

change a by 1
Переменной a присваивается значение 0. Затем мы создаем условный цикл, который будет повторяться, только пока a меньше 10. Внутри цикла мы предложим Scratch вывести [print] значение a для 2 секунд, затем увеличить a на 1. Этот цикл будет продолжаться, пока мы не достигнем a = 9, и тут он остановится, потому что следующее значение, 10, уже не меньше 10.
  
```

А вот как выглядит наш код в Python:

```

a = 0
while a < 10:
  print a
  a = a + 1
  
```

Зачем же мы включили код Python в руководство по Scratch? Он просто иллюстрирует, что в логике обоих языков разница невелика и что при помощи Scratch можно понять логику работы многих приложений. И в Scratch, и в Python мы создали переменную **a** и установили ее значение в 0, откуда мы создали цикл, который будет повторяться, пока не достигнет показателя 9 (который меньше 10). При каждом проходе цикла мы увеличиваем **a** на 1, что позволяет нам сосчитать, сколько раз цикл повторился.

## Тестируем нашу игру

Мы сделали это! Мы сделали игру. А теперь давайте в нее сыграем. Процесс игры будет таким:

- Нажмите на зеленый флажок.
- У вас спросят ваше имя.
- Мэтт поздоровается с вами, а Нейл поздоровается с вами и назовет вас по имени.
- Оба предложат вам начать викторину.
- Мэтт предложит Нейлу задать первый вопрос.
- Нейл задаст вопрос.
- Появится окно, где вы должны дать ответ.
- Если ответ верный, Нейл вам об этом скажет, а Мэтт вас похвалит.
- Ваш счет увеличится на 1 очко.
- Если ваш ответ неправильный, то над вами посмеется Мэтт и количество попыток увеличится на 1, оставив вам только 2 попытки. Затем вам дадут второй шанс ответить на вопрос.
- Если вы угадали правильно, то перейдете к следующему вопросу, и это повторится дважды, поскольку у вас 3 вопроса.
- Если вы правильно ответили на все вопросы, Мэтт скажет вам ваш финальный счет, и затем сообщит, что игра окончена.
- На экране появится спрайт Game Over, и все спрайты игры будут отключены.
- Если в какой-то момент количество попыток достигло 3, игра автоматически переходит на экран Game Over.

И это только начало наших приключений в области Scratch. Возможно даже применять Scratch для управления электроникой через всеми любимый программируемый компьютер Raspberry Pi.

Надеемся, вам понравилось общение со Scratch; это отличный инструмент для того, чтобы понять принципы программирования, а кроме того, изучать его очень забавно. **LXF**

**Скорая помощь** 

Вы можете поделиться своим проектом онлайн на разных платформах через <http://scratch.mit.edu>.



# Scratch: Создаем игру-аркаду

Кто спасет наш прекрасный город от логотипов-убийц?!  
Кто же, как не Лес «Капитан» Паундер!

**Н**у вот, вы освоили игру в Scratch. Надеюсь, вы прошли викторину, и сделали ее больше и лучше. Учиться программировать — большое удовольствие, но придумать новые идеи не всегда приятно: иной раз лучше взять уже существующий фрагмент кода и адаптировать его под свои потребности. Это лучший аспект открытого кода: возможность изучать, изменять и заново использовать код от других программистов позволяет вам учиться у них.

В Scratch тоже есть эта функция. Если вы перейдете на <http://scratch.mit.edu> и зарегистрируете бесплатную учетную запись, вы тоже сможете поделиться своими проектами с динамичным и творческим сообществом программистов, которые делятся своими проектами с вами.

На этом уроке мы создадим душераздирающую стрелялку. Напомним: мы пользуемся текущей стабильной версией Scratch 1.4, и освежить свои знания о системе блоков Scratch можно на с. 88.

Итак, мы в городе, уже поздно, и вы патрулируете город, защищая его от захватчиков, и вдруг откуда ни возьмись в небе материализуются злобные логотипы *Linux Format* — спрайты LXF Logo — и нападают на вас. Ваша задача — уничтожить логотипы, получая очки за каждый разрушенный вражеский спрайт, но на спасение города у вас всего 30 секунд!

В данном проекте мы будем использовать:

- » Три спрайта LXF Logo.
- » Спрайт Game Over.
- » Фоновое изображение для сцен в городе.

Итак, давайте начнем создавать наш проект, и идеальное место для его начала — наш первый спрайт LXF Logo. Все файлы, требуемые для этого проекта, находятся на **LXFDVD** или в архиве.

Мы используем спрайт LXF Logo в качестве нашего захватчика, и нам надо уничтожить его за 30 секунд, или мы потеряем наш город! Но, пожалуй, хватит озвучки из малобюджетного фильма. Спрайт LXF Logo — это цель в нашей игре, и если нам удастся по нему щелкнуть, то он исчезнет, а наш счет увеличится.

Итак, сначала нам нужно открыть новую сессию Scratch, что мы и делаем, просто открывая приложение Scratch.

Перед вами должно открыться новое окно Scratch, с котом в разделе The Stage. Для этой игры кот нам не нужен, так что щелкните по нему правой кнопкой и выберите Delete. Теперь перед вами пустая сцена, готовая для нашего спрайта LXF Logo. Импортируем логотип в Scratch. Вы увидите в средней правой части экрана три значка. Первый — кисть со звездой позади — это для рисования нового спрайта. Второй — это папка со звездой позади

## Среда Scratch

### Палитра блоков [Block Palette]

Здесь хранятся и сортируются по функциям наши блоки кода.

### Область скриптов [Script Area]

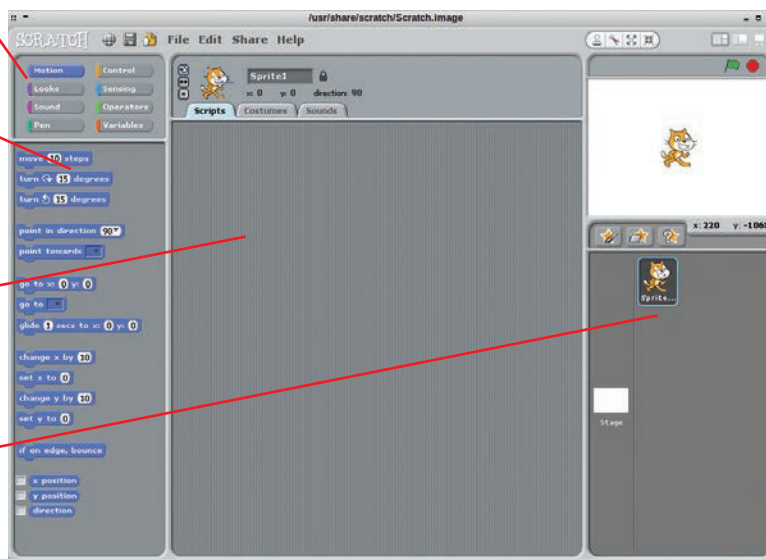
Во втором столбце находится область, куда вы можете перетащить наши блоки кода из палитры, чтобы добавить код в программу.

### Сцена [The Stage]

Она покажет результаты вашего программирования, и может использоваться для взаимодействия с миром игры.

### Панель спрайтов [Sprites Panel]

Эта панель показывает спрайты и основное имущество, которые принадлежат вашей программе. Нажатие на спрайт изменит фокус на нем, позволяя вам написать код только для этого спрайта.



ди, и она открывает диалоговое окно импорта, что нам и нужно. Третий импортирует случайно выбранный спрайт в вашу игру — на этом уроке мы не будем его использовать.

## Интерактивные спрайты

Значки, используемые для создания и импорта спрайтов в Scratch, снабжены всплывающими подсказками, чтобы быстро показать, что они делают — просто наведите курсор мыши на один из них, если сомневаетесь. Щелкните по второму значку, чтобы импортировать наш спрайт; вы должны увидеть диалоговое окно, всплывающее на экране (как показано на рис. 2.1). Расположение диалогового окна Import Sprite по умолчанию относится к ресурсам Scratch.

В диалоговом окне перейдите в местоположение спрайта LXF Logo, выделите его и нажмите на ОК. Теперь спрайт импортирован в Scratch, и появится на сцене. Далее нам надо предоставить нашему спрайту код, чтобы сделать его интерактивным. Для этого спрайта есть три фрагмента кода, и если выразить их псевдокодом, они будут работать так:

При щелчке по зеленому флагу.

Установить значение переменной "speed" как 5.

В бесконечном цикле.

Перемещать спрайт, используя переменную speed для контроля за числом шагов.

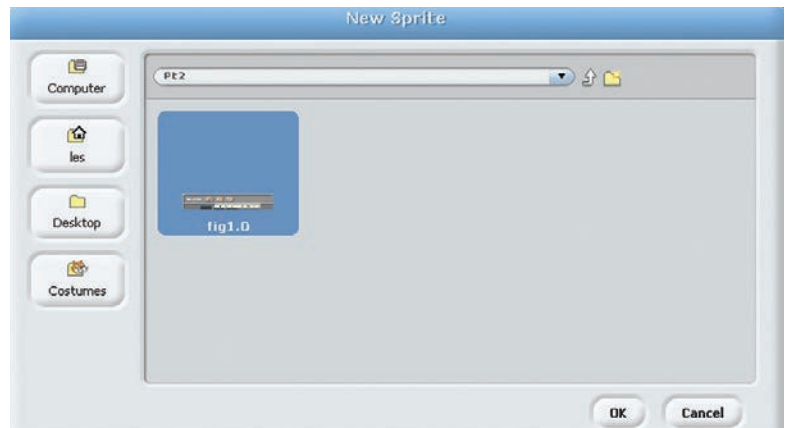
Если спрайт ударяется о край экрана, он отскакивает.

Код для перемещения спрайта LXF Logo довольно простой, и это первый шаг на пути превращения нашей игры в нечто захватывающее. Чтобы добавить этот код, вам сначала надо создать переменную с именем Speed, щелкнув по Variables Palette и по Make a Variable. В появившемся диалоговом окне назовите переменную Speed и настройте ее — For this sprite only [Только для этого спрайта]. (В диалоговом окне Make a Variable две опции, и вторая — For all sprites [Для всех спрайтов]; обе используются в нашей игре.)

## Псевдокод спрайта

Теперь у нас есть место для хранения скорости нашего спрайта, а значит, вместо того, чтобы задать скорость в коде, мы можем задать ее один раз и сделать ссылку на переменную в других местах, сэкономив немало времени и ресурсов.

Далее, вам понадобится продублировать код. Помните, что каждый блок имеет свой цвет, чтобы вы могли выбрать нужную палитру. Создав фрагмент кода, сохраните проект, и затем щелкните по Green Flag, чтобы запустить код. Теперь мы должны увидеть, как спрайт LXF Logo перемещается горизонтально по экрану и отпрыгивает от краев, но постойте-ка — что это? Когда спрайт отскакивает, он переворачивается! Надо исправить его поведение, и, к счастью, это не слишком сложно. Прямо над разделом, где мы писали код для спрайта, мы видим имя спрайта, и слева от него мы видим три серых значка. (Вы можете видеть значки на рис. 2.3, стр. 94). Сверху вниз, это:



► Рис. 2.1. Используйте диалоговое окно Import Sprite для добавления собственного спрайта в The Stage.

► **Верх [Top]** поворот на 360 градусов.

► **Середина [Middle]** Перевернуть вертикально.

► **Низ [Bottom]** Не переворачивать.

Для этого спрайта мы бы хотели, чтобы логотип не переворачивался, поэтому щелкните по нижнему значку. Теперь запустите код снова, щелкнув по Green Flag... все исправлено!

Эти мерзкие захватчики — спрайты LXF Logo — умны, и применяют технологию маскировки, чтобы скрыться от наших бесстрашных защитников. Однако они никогда не знают, насколько долго еще останутся скрытыми, ну и мы должны их отстреливать, как только заметим.

Шутки из малобюджетных фильмов опять отложим в сторону: мы должны настроить этот код, чтобы он работал, когда щелкают по зеленому флажку, так что давайте применим псевдокод, чтобы изучить логику.

При щелчке по зеленому флагу.

В бесконечном цикле.

Спрятать логотип

Ждать случайное число секунд между 1 и 3 при спрятанном логотипе

Показать логотип

Ждать случайное число секунд между 1 и 5, показывая логотип

Показ спрайта игроку — ключевой момент нашей игры, а использование случайных чисел делает это еще сложнее. Переделав код в Scratch, сохраните результаты и затем запустите код. Ваш спрайт пропал? Если да, то поздравляем! Ну, а если нет — убедитесь, что вы использовали Ждать, чтобы логотип был невидимым/видимым в течение случайного промежутка времени.

## Нам нужны лазеры

У последнего фрагмента кода для этого спрайта есть две цели: воспроизвести крутой звук лазера (пиу-пиу) и увеличить

**Скорая помощь**



Щелчок правой кнопкой по спрайту вызовет целый ряд действий, таких, как дублирование или удаление спрайта. Очень удобно и очень быстро.

## Использование Scratch с Raspberry Pi

Scratch установлен на каждом Raspberry Pi, на котором работает ОС Raspbian. Но знаете ли вы, что в Pi-версии Scratch легко работать с GPIO, ощущать окружающий мир и управлять им? Да, вы прочли все верно — с помощью Scratch можно управлять LED, двигателями и разными устройствами ввода через этот простейший в использовании интерфейс.

Если хотите такое попробовать, то лучший ресурс для этого — блог Саймона Уолтерса [Simon Walters]

Cymplecy, <http://bit.ly/cymplecy>. Саймон написал отличные статьи о создании проектов на Scratch, и его блог предлагает пошаговый подход к коду; так что вы сможете создать чудесные творения без особых предварительных знаний. Одно из исключительных достижений Саймона — ряд роботов для решения задачи лабиринта, запрограммированных в виртуальном лабиринте Scratch. Расстояния до объектов и стен роботы меряют сверхзвуковыми датчиками.

После того, как код прототипируется и проверяется в виртуальном лабиринте, он применяется к роботу, и лабиринт воссоздается в реальном мире, причем в роли препятствий выступают самые разные объекты.

Все содержимое блога Саймона свободно для использования, так что беспрепятственно налетайте — и попробуйте внедрить его творения в свой проект.



значение, хранящееся в новой переменной под названием Score, на 1. Итак, давайте опять обратимся к псевдокоду.

- При щелчке по Sprite1
- Спрятать спрайт
- Проиграть звук лазера
- Добавить 1 очко.

Нам нужно создать еще одну переменную: эта должна называться Score и быть настроена For all sprites, что означает следующее: другие спрайты могут изменить величину, хранящуюся в переменной score.

Поискем звук для нашей игры. В столбце, где мы редактируем наш код, мы можем видеть три вкладки: Scripts [Скрипты], Costumes [Внешний вид] и Sounds [Звуки]. Щелкните по Sounds, а затем по Import, и перед вами появится диалоговое окно, подобное тому, которое на рис. 2.2.

Через это диалоговое окно перейдите в папку Electronic и воспользуйтесь в качестве звука Laser1. Вы увидите, что звук импортировался во вкладку Sound.

Теперь, когда у нас уже есть основа, воссоздайте код выше. Завершив работу, сохраните ее и щелкните по Green Flag, чтобы запустить код. Поздравляем, вы закончили код для нашего спрайта LXF Logo. Как насчет чашечки чая, чтобы это отметить?

Вы, возможно, помните из первого урока, что The Stage также можно присваивать код, который будет работать параллельно коду для каждого из наших спрайтов. В данном руководстве у The Stage две очень важных роли.

Во-первых, он обнуляет score для каждой новой игры, и во-вторых, и это самая важная роль, он управляет таймером, чтобы тот давал игроку 30 секунд, чтобы набрать как можно больше очков.

Итак, давайте рассмотрим, как этот код работает, при посредстве псевдокода:

- При щелчке по зеленому флажку.
- Установить переменную score в 0
- Установить переменную timer to 30
- Создать условный цикл, который заканчивается, когда timer достигает 0
- Ждать 1 секунду
- Вычитать 1 из переменной timer при каждом проходе цикла.
- КОНЕЦ ЦИКЛА
- Переслать оповещение "gameover", которое получит слой нашего спрайта Game Over.

Нам нужно создать еще одну новую переменную, доступную для всех спрайтов, и она должна называться timer. Затем нам нужно установить переменные score и timer в 0 и 30 соответственно. Этот фрагмент кода перезапустит игру для нового игрока, дав ему 30 секунд времени и 0 очков.

Наш цикл условный (см. рис. 2.4): он будет работать, только пока переменная timer будет больше нуля. Чтобы это происходило, мы должны проинструктировать цикл ждать 1 секунду, затем вычесть 1 из переменной timer. Если бы мы не включили

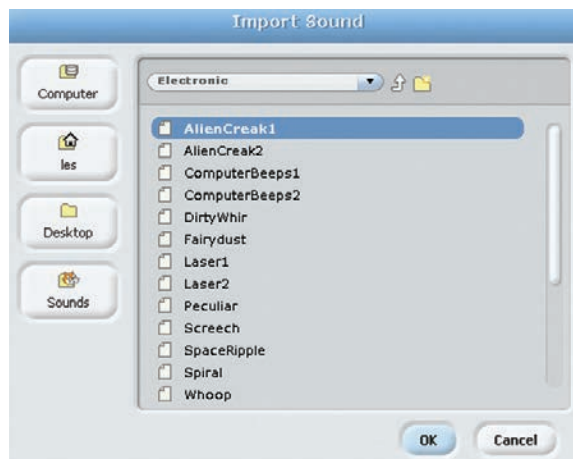


Рис. 2.2. Диалоговое окно Sound похоже на диалоговое окно Import Sprite, и по умолчанию относится к ресурсам Scratch.

ждать 1 секунду, то таймер начал бы обратный отсчет очень быстро. И, наконец, мы используем команду broadcast для общения между сценой и спрайтом Game Over, и это станет понятно, когда мы расскажем именно об этом спрайте в нескольких абзацах.

Воссоздайте код выше, сохраните результаты и затем запустите код, щелкнув по зеленому флажку. Теперь у вас есть таймер и метод перезапуска кода для нового игрока.

## Game over, чувак

У нашего последнего спрайта было две основных задачи, и они заключались в том, чтобы сообщать игроку его счет на момент окончания игры и останавливать все работающие скрипты. У этого спрайта также есть крошечный отдельный скрипт, который просто прячет спрайт при нажатии на зеленый флажок, и он появляется снова только в конце игры. Давайте заново рассмотрим логику кода основных задач с помощью нашего старого друга, псевдо-кода.

- Когда я получаю сообщение "gameover"
- Показать спрайт Game Over



Рис. 2.4. Вы также можете использовать команду Scratch broadcast в своем Raspberry Pi, чтобы управлять электронными компонентами.

### Скорая помощь

У вас пропал код? Вы уверены, что нажали на нужный спрайт? Это распространенная ошибка — убедиться, что нажали на нужный спрайт для редактирования его кода.



Рис. 2.3. Значки поворота спрайта в Scratch 1.4 размещаются здесь. Однако для 2.0 вам нужно щелкнуть по значку с информацией по спрайту, чтобы изменить его поворот.



Сказать “Вы набрали очков:” и приписать переменную “score” для завершения фразы

Остановить все скрипты в игре

Мы начнем с создания нового события (см. рис. 2.5, чтобы увидеть, как это выглядит в Scratch), которое будет запускать код для этого спрайта. Событие заключается в том, что этот спрайт получает сообщение `gameover`, а блок для него находится в Control Palette. Далее у нас есть блок, который показывает спрайт на экране, и этот блок вы найдете в Looks Palette.

Наш следующий блок — это смесь блоков Looks, Variable и Operator, смонтированных вместе.

Мы используем блок Looks `say Hello for 2 seconds` для вывода текста на экран. Там, где этот блок говорит Hello, мы помещаем блок оператора `Join Hello World block`, и в первом белом блоке мы пишем «Вы набрали очков:». Во втором белом блоке мы ставим переменную `score`. Оператор `join` использует конкатенацию для соединения двух строк текста: одной, которую мы ввели в первое окно блока `join`, и другой, которая содержится в переменной `score`.

После того, как вы создали этот основной фрагмент кода, продолжайте. И создайте меньший раздел кода, который будет скрывать спрайт при нажатии на Green Flag. По завершении сохраните результаты и щелкните по Green Flag, чтобы протестировать свой код. Отлично — вы закончили создание логического блока, управляющего вашей игрой.

Осталось украсить сцену фоном нашего города. Чтобы изменить фон сцены, нам надо щелкнуть по значку The Stage рядом с подборкой спрайтов. Теперь нам нужно перейти в область, где мы пишем наш код, и вы увидите вкладку Backgrounds (см.

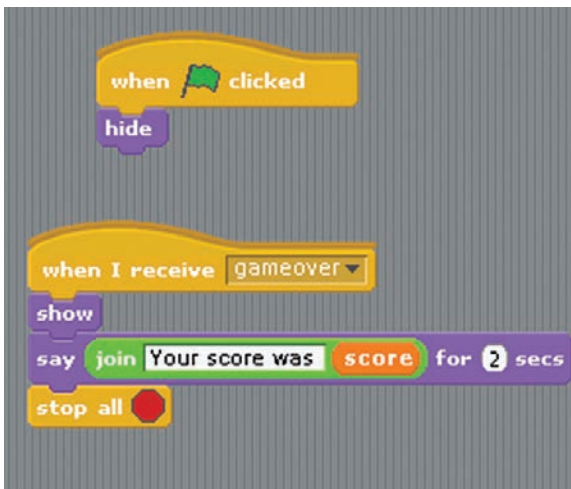


Рис. 2.5. Первая часть кода спрайта Game Over скрывает спрайт при запуске, а вторая — конец последовательности игры.

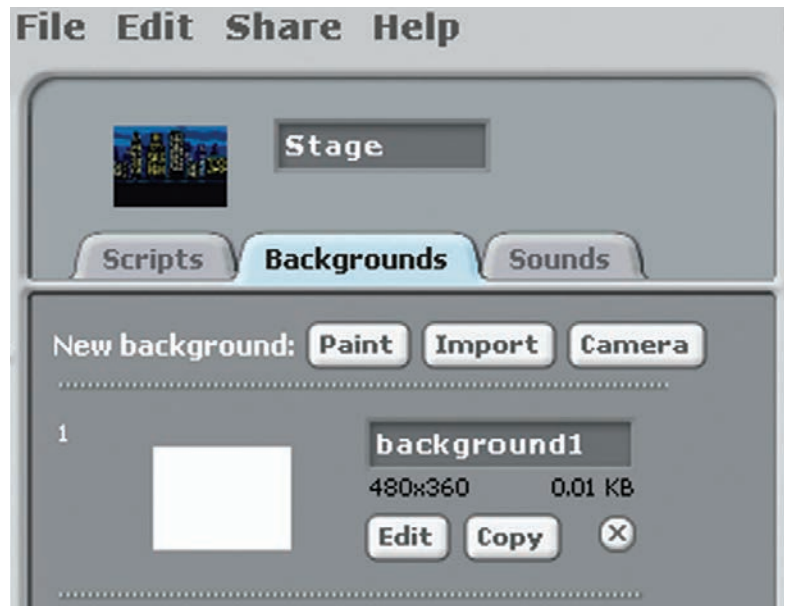


рис. 2.6). В ней может храниться множество разных фонов, и с помощью блоков Looks их легко изменять.

Находясь в этой вкладке, нам нужно щелкнуть по `Import`, и появится диалоговое окно; в этом окне нам нужно перейти в папку `Outdoors` и выбрать в качестве фона ночной город. Теперь, когда у нас есть фон, наша игра завершена, так что сохраните проект, а затем запустите код.

## Идем дальше

Мы можем продолжить, добавляя новые спрайты и программируя их с помощью того же логического блока, что и раньше, но это займет много времени и будет совсем не так интересно. Если вы щелкнете правой кнопкой по спрайту, который хотите скопировать, и выберете `Duplicate`, то и спрайт, и все связанные с ним скрипты будут дублированы за несколько секунд.

Теперь, когда у вас поприбавилось спрайтов, можете попробовать на каждом какую-нибудь новую идею. Почему бы не сделать один спрайт меньше и быстрее, и, если удастся его поразить, то увеличить счет на большее количество очков?

Наилучший способ продолжить развивать игру — программировать и изучать блоки. Сама природа блоков означает, что мы можем быстро пробовать новые идеи, не боясь потерять результаты нашей работы, когда мы разъединяем и заново соединяем блоки кода так, как нам хочется.

Наслаждайтесь Scratch! Это отличный инструмент для изучения концепции программирования и отличное приложение, радующее детей. LXF

Рис. 2.6. Вкладка Backgrounds может хранить множество фонов.

## Скорая помощь

Не бойтесь разбирать свой код на мелкие части, особенно при отладке своей работы. Если что-то работает не так, как вы рассчитывали, разбейте код на более мелкие фрагменты и выясните, в чем проблема.

## Онлайн-ресурсы Scratch

Scratch — только один из проектов, предоставленных MIT, и у этого учреждения есть и другие отличные инструменты, готовые для использования, например, *App Inventor*, который находится здесь: <http://appinventor.mit.edu/explore>.

Этот инструмент использует структуру блоков, похожую на Scratch, позволяя программистам создавать отличные по своему внешнему виду и функциональные приложения Android без необходимости изучать Java. *App Inventor* используется в школах по всему миру для создания школьных приложений.

Еще один инструмент, который стоит попробовать — *Blockly* (<http://bit.ly/1lACCDS>). Это не продукт MIT, но мне бы хотелось его продемонстрировать, поскольку мне он кажется весьма своеобразным.

*Blockly* использует похожую блочную структуру для написания кода, но он также пишет код для других устройств, и сохраняет код, соответственно. То есть, например, мы можем использовать похожий на Scratch интерфейс для создания кода для JavaScript и Python.

Загляните сюда, чтобы посмотреть на сетевую презентацию, которую вы сможете испытать сами для создания кода для JavaScript, Python и XML одновременно: <http://bit.ly/JHkXvX>.

Есть даже версия для экспорта кода, совместимого с Arduino. Чтобы скачать его, перейдите на страницу *BlocklyDuino GitHub*: <http://bit.ly/1bRCEz1>.

Все эти ресурсы стоят того, чтобы вы на них обратили внимание, и, что самое приятное, все они абсолютно бесплатные.

# ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу [answers@linuxformat.ru](mailto:answers@linuxformat.ru), и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Установку HotPicks
- 2 Проблемы с драйверами сканера Epson
- 3 BT Sport в Linux
- 4 Запуск сервера MythTV
- 5 Использование ZFS на главном диске
- 6 Повреждение жесткого диска ноутбука
- 7 Разбиение диска

## 1 Ставим HotPicks

На DVD с LXF178 есть две программы, которые прекрасно подходят для моей волонтерской работы: *fgallery* и *croppui*. Но ни для одной из них я не нашел руководства по установке. Не могли бы вы рассказать, как их установить? **Сбуфер, с форумов**

Не все пакеты содержат подробные инструкции по установке и использованию, и иногда после распаковки архива нужна небольшая детективная работа. Что касается *fgallery*, то в архиве есть файл *fgallery*, у которого установлен бит выполнения. Команда `file` показывает, что это и в самом деле файл программы, так что его можно запустить напрямую, зайдя в каталог *fgallery-1.0* и набрав в терминале

```
./fgallery --help
```

В этом каталоге также имеется пара файлов README: один — в формате HTML, другой — текстовый. Исполняемый файл *fgallery* можно переместить в какой-нибудь каталог из переменной PATH, например, в `~/bin` или `/usr/local/bin` (для первого нужны права root).

Для *croppui* все иначе; это скрипт на Python, но в архиве есть и установочный скрипт. Способ установки зависит от того, нужно ли установить программу в свой домашний каталог (для этого хватит прав обычного пользователя) или глобально (для чего потребуются привилегии root). Для установки в домашний каталог сделайте следующее:

```
tar xf croppui_0.1.1.tar.gz
```

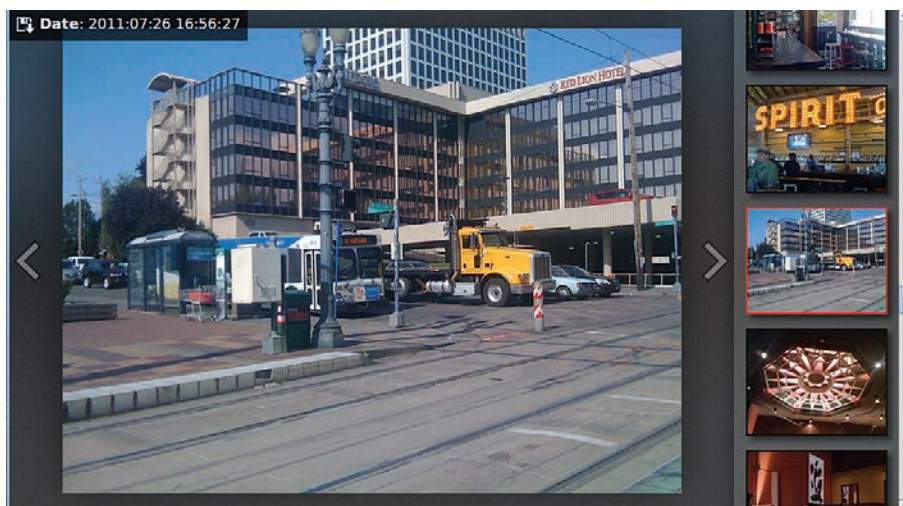
```
cd croppui
```

```
./install.sh -u
```

Для глобальной установки просто замените последнюю команду на такую:

```
sudo ./install -p /usr/local
```

Вы видите, что в пакете *croppui* есть файл *.desktop*. По этому файлу окружение рабочего стола определяет, в какое меню добавить программу, какой командой ее запустить и файлы каких типов стоит предлагать открывать этой программой. Это



Для установки программы с DVD обычно стоит поискать файл README или установочный скрипт.

означает, что данную программу можно запускать из меню на любом рабочем столе.

## 2 Сканер без драйверов

У меня Mint 15 с LXF DVD174, и проблем с ним до недавних пор не было. А теперь мне пришлось очистить систему и переустановить Mint 15, и хотя система работает нормально, я никак не могу загрузить свой сканер Epson V100. Раньше я загружал драйверы для этого сканера с AVASYS, и с ними было все хорошо. Но, похоже, они перестали делать драйверы, и приходится идти на сайт Epson. Пытался загрузить нужные драйверы, но ничего не вышло, так как сайт закикливается и снова выбрасывает меня на первую страницу.

Не подскажите, как решить эту проблему, или где еще можно загрузить драйверы?

**Джон Джексон (John Jackson)**

Драйвера Epson есть на их сайте. Драйвера для V100 доступны по ссылке [www.epson.com/cgi-bin/Store/support/supDetail.jsp?oid=72343&BV\\_UseBVCookie=yes&infoType=Downloads&platform=OSF\\_0\\_LINUX](http://www.epson.com/cgi-bin/Store/support/supDetail.jsp?oid=72343&BV_UseBVCookie=yes&infoType=Downloads&platform=OSF_0_LINUX).

Когда Вы примете лицензию, у Вас появится ошеломляющий набор вариантов загрузок. Для Linux Mint (как и для любого другого дистрибутива с пакетами *.deb*) Вам нужны файлы

```
iscan-data_1.24.0-2_all.deb
```

```
iscan_2.29.2-1-usb0.1.ltdl7_amd64.deb
```

Если у Вас 32-битная версия ОС, вместо второго файла воспользуйтесь файлом `iscan_2.29.2-1-usb0.1.ltdl7_i386.deb`. Первый файл одинаковый для всех архитектур. Файлы следует

устанавливать именно в этом порядке, для чего можно применить команду

```
sudo dpkg --install iscan-data_1.24.0-2_all.deb  
iscan_2.29.2-1-usb0.1.ltdl7_amd64.deb
```

Те, кто пользуется дистрибутивами на базе RPM, типа Fedora или openSUSE, могут загрузить RPM-эквиваленты пакетов и установить их командой

```
rpm -i iscan-data-1.24.0-2.noarch.rpm  
iscan-2.29.2-1-usb0.1.ltdl7.x86_64.rpm
```

Для сканеров Epson есть драйвер SANE, но на данный момент он не поддерживает V100, поэтому загружайте драйвера с сайта Epson.

## 3 BT Sport в Linux

В Я только что обновил свой канал на оптоволочный с бесплатным BT Sport. Но не могу заставить его работать. У меня Mint 14 (Mate), и сайт BT говорит, что нужно установить *Silverlight*. Я следую указаниям, а потом он просит меня установить *Mono*, что я и делаю, но результата все равно никакого. Зайти на BT Sport я могу, но при попытке что-то посмотреть он снова просит установить *Silverlight*.

Мой компьютер — AMD 64 с 2 ГБ памяти и Linux Mint 14 (32-битный). Причина во мне, в ОС или в BT? Возможно, не работает *Moonlight*.

**Коллин Мэгуайр (Colin Maguire)**

*Silverlight* — технология Microsoft, используемая для воспроизведения контента в браузере. *Moonlight* — свободная, открытая реализация *Silverlight*, разработанная проектом Mono и подходящая для использования

в Linux. К сожалению, *Moonlight* на несколько версий отстает от *Silverlight* и не включает возможности, необходимые для воспроизведения контента, защищенного DRM, к которому относится контент BT Sport. Это та же проблема, с которой сталкиваются пользователи Linux, пытающиеся воспользоваться Netflix, и решать ее нужно так же.

На самом деле, решений два — и оба связаны с *Wine*, слоем эмуляции Windows. Самый простой вариант привязан к Ubuntu, но должен работать и в Mint. Пакет **netflix-desktop** содержит специально зататанную для работы с *Silverlight* версию *Wine*, а также версии *Firefox* и *Silverlight* для Windows. Установите ее, добавив персональный архив пакетов (PPA), где находится этот пакет — для чего в терминале запустите команды

```
sudo apt-add-repository ppa:ehooover/compholio
sudo apt-get update
sudo apt-get install netflix-desktop
```

После установки запустите программу *netflix-desktop* — Вы обнаружите ее в меню (или в Dash в Ubuntu). Она загрузит и установит другие необходимые ей программы, а затем запустит *Firefox* в полноэкранном режиме, и обычно именно это и нужно для просмотра ТВ — на экране нет ничего лишнего. Закрывается браузер клавишами Alt+F4, а чтобы открыть меню и выйти из полноэкранного режима, нажмите Alt+F3.

Второй вариант установить сложнее — и, возможно, мы посвятим ему статью в ближайшем номере. Тут используется программа под названием *Pipelight* (<https://launchpad.net/pipelight>). Это плагин для браузеров Linux, который с помощью *Wine* запускает плагины Windows. Поэтому нам потребуется только плагин *Silverlight*, работающий через *Wine* — а браузер останется тем, к которому Вы привыкли в Linux. Преимуществом такого варианта является и то, что он полностью независим от выбранного дистрибутива. Подробная информация о нем есть на сайте *Pipelight*.

#### 4 Правила DVB

**В** у меня есть сервер *MythTV* и — как я понял, отправив вопрос в #mythtv-users — клиент *MythTV* с окном безопасности. Я знаю, что

без крайней необходимости запускать программы от имени root не следует, и при установке *MythTV* через ATrpm5 выбирал все опции по умолчанию. Однако ps говорит, что клиент работает от имени root, а если заглянуть в /etc/init.d/mythbackend, я вижу вот это:

```
# Does not work on Red Hat, due to missing audio/video groups. [He работает в Red Hat из-за отсутствия групп audio/video.]
# daemon --user mythtv $binary $OPTIONS
```

Мне посоветовали добавить группы и изменить правила udev для создания файлов устройств с правильными правами доступа, как описано на [www.reactivated.net/writing\\_udev\\_rules.html#ownership](http://www.reactivated.net/writing_udev_rules.html#ownership), но пока мои попытки не увенчались успехом. Я пробовал такой вариант:

```
KERNEL=="dvb/adaptor[0-9]*",
NAME="adaptor/%n", SYMLINK+="=%k",
GROUP="video"
```

Не могли бы вы направить меня на верный путь?

Mengoshmink, с форумов

Во-первых, устройства лучше не переименовывать: это может привести к конфликтам и к тому, что устройства начнут получать неожиданные имена. Вместе этого оставьте имя неизменным и просто добавьте свое, через символическую ссылку.

Во-вторых, /dev/adaptor0 и др. — это каталоги, и следуя концепции Unix о том, что все на свете — это файл, следует устанавливать права доступа для узлов внутри этих каталогов. Это можно сделать с помощью шаблонов по KERNEL, но гораздо проще произвести поиск по SUBSYSTEM, так как все эти записи создаются подсистемой dvb. Я и сам пользуюсь *MythTV* с картами DVB, и у меня в правилах udev есть следующее:

```
SUBSYSTEM=="dvb", GROUP="video"
```

Он задает video в качестве группы владельца всех устройств DVB. Права доступа должны быть такими: **rw-rw---**. Если нет, можно изменить правило, чтобы добавить и право на запись и чтение для группы, как показано ниже.

```
SUBSYSTEM=="dvb", GROUP="video", MODE=660
```

## 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**.

Конечно, также нужно обеспечить, чтобы пользователь, от имени которого запускается *MythTV*, входил в группу video, иначе доступ не будет ему предоставлен.

#### 5 ZFS всерьез

**В** Я последовал вашему учебнику по ZFS [LXF175, стр. 76] и, немного поэкспериментировав с запасным диском, решил поставить ZFS на основных дисках — паре дисков по 2 ТБ каждый, из которых я хочу сделать RAID 1. Трудно ли после создания zpool перенести в него



## Коротко про...

### at

**C**ron отлично подходит для запуска команд в заданное время или через регулярные интервалы, но что если команду нужно запустить всего однажды, но в заданное время? Можно воспользоваться **sleep** и подождать, пока это время наступит:

```
sleep 1h; команда
```

Но этак мы подвесим терминал, а после перезагрузки все исчезнет. Более удачной альтернативой будет демон *at*. Он работает как *cron*,

просыпаясь каждую минуту и смотря, нет ли для него заданий, но каждое задание выполняется только один раз. Команда **at** получает задание, считывая команду для выполнения из стандартного ввода. Так, команда

```
echo "команда --with-opts" | at 15:00
```

запустит указанную команду в три часа дня. Если время уже прошло, команда запустится на следующий день. Также можно задавать интервалы в часах, минутах и днях, и даже добавлять их:

```
echo "другаякоманда --with-opts" | at 5am + 2 days
```

С помощью параметра "now [сейчас]" можно создать задержку и без указания конкретного

времени — скажем, пример с командой **sleep** можно переписать так:

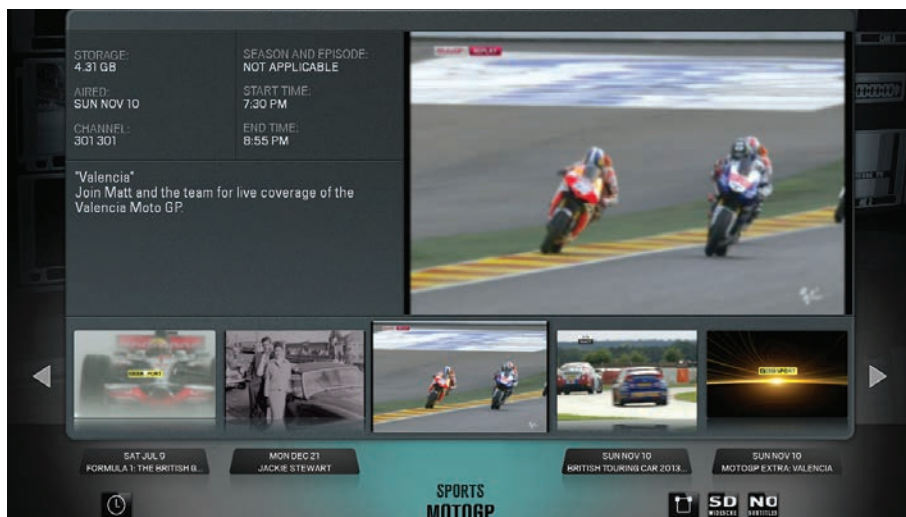
```
echo "команда" | at now + 1 hour
```

Запустив эту команду, можно закрыть терминал и даже перезагрузить систему, а команда все-таки будет работать. Для просмотра запланированных команд запустите **atq**, а если передумаете, **atrm** удалит задания из очереди.

За более подробным описанием временных параметров стоит заглянуть на тап-страницу **at**.

Если нужно запустить несколько команд, поместите их в файл по шутке на строку и скормите этот файл **at**:

```
at midnight <commands.txt
```



➤ **MythTV** не обязательно запускать с правами `root`, но для записи телепрограмм может потребоваться изменение правил `udev`.

существующие тома? Или лучше создать новые тома и скопировать на них данные через `rsync`?

**Эндрю Сондерс (Andrew Saunders)**

Если у Вас диски на 2 ТБ или больше, размер нового блока на них будет составлять 4К. Чтобы избежать проблемы выравнивания данных, которые могут серьезно уронить производительность, нужно сообщить ZFS о нужном выравнивании при создании пула с помощью `ashift`. С дисками 4К этот параметр нужно установить в 12 — по умолчанию ZFS пытается определить его сама, но некоторые диски притворяются, что пользуются блоками меньшего размера, чтобы не сбивать с толку более ранние версии Windows. Зеркало RAID1 на двух таких дисках создается так:

```
zpool create poolname mirror /dev/firstdevice /dev/seconddevice -o ashift=12
```

После создания пула для переноса отдельных томов воспользуйтесь командами `zfs send` и `zfs receive`. Команда `send` отправляет весь том (не только файлы, но и свойства файловой системы) в виде потока данных. Команда `receive` преобразует этот поток обратно в файловую систему в другом пуле (или на другом компьютере). `Send` работает со снимками, и подход, к которому Вы прибегнете, зависит от того, копируете ли Вы рабочую файловую систему и хотите ли сделать минимальным время простоя. Если файловая система размонтирована, выполните команды `zfs snapshot oldpool/volume@copy1`

```
zfs send -R oldpool/volume@copy1 | zfs receive newpool/volume
```

Это потребует времени, но зато больше ничего делать не нужно. Если Вы делаете все это в работающей системе, тоже выполните эти команды, но понадобится сделать еще кое-что. Если Вы не можете размонтировать том, либо переведите его в режим «только для чтения», либо запустите эти команды с Live CD, чтобы гарантировать, что с момента создания снимка содержание тома не изменится.

```
zfs unmount oldpool/volume
```

```
zfs snapshot oldpool/volume@copy2
zfs send -i -R oldpool/volume@copy2 | zfs receive newpool/volume
```

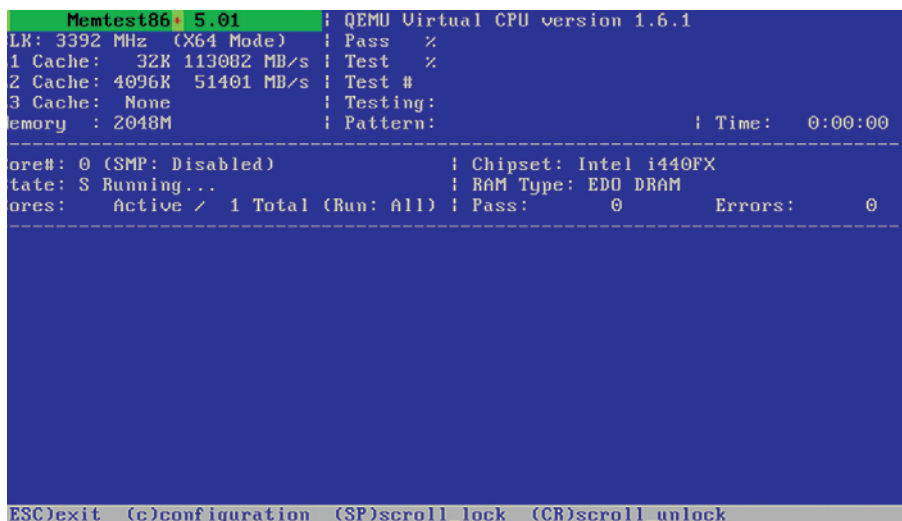
Эти команды размонтируют том, сделают новый снимок и выполнят инкрементальную отправку, при которой пересылаются только изменения с момента предыдущего снимка, поэтому она гораздо быстрее. Повторите эти действия для каждого тома в пуле, и все готово. Если Вы задали атрибут `mountpoint` для любого из томов, измените или снимите его для старого пула, в противном случае оба тома будут пытаться смонтироваться в одну точку монтирования.

Так как ZFS принимает и отправляет данные через канал, то благодаря SSH новый и старый тома могут даже находиться на разных компьютерах.

```
zfs send -R pool/vol | ssh othercomputer zfs receive otherpool/othervol
```

## 6 Разрушение диска

**У меня серьезная проблема с маминим ноутбуком. Он довольно старый, и она жаловалась, что Windows тормозит. Я дал**



➤ Свободно доступная программа `memtest86` выполняет проверку оперативной памяти на предмет наличия неисправностей. Если Errors не равно нулю, у вас есть проблемы.

ноутбуку новую жизнь, установив Ubuntu в конфигурации с двойной загрузкой. С тех пор мама счастлива и пользуется Linux уже несколько лет. У ноутбука нет PAE, поэтому я застрял на одном из ранних релизов — возможно, *Natty Narwhal*. Проблемы начались, когда ноутбук завис при загрузке, и его пришлось принудительно выключить. После перезагрузки открывалась командная строка «восстановления grub», а при загрузке с Live DVD Mint не удавалось определить тип файловой системы единственного раздела с Linux! *Smartctl* даже при полном сканировании ошибок не нашла, при запуске `fsck` нашлось несколько сотен, если не тысяч ошибок, после чего с раздела исчезло все, кроме папки `lost+found`!

Теперь я установил Mint 13, и он ведет себя хорошо, но я не имею ни малейшего понятия о том, что было причиной этой проблемы.

**Роб Пирмен (Rob Pearman)**

SMART отслеживает физические параметры диска: порченные блоки, температуру и т.д. `Fsck` работает на уровне файловой системы. Эта ситуация больше похожа на повреждение файловой системы, когда на диск были записаны неверные данные, а не на проблемы с диском. Неисправность самого диска обязательно проявилась бы при проверке `smartctl`. Так у нас появляется вопрос: было ли зависание системы в самом начале причины или следствием повреждения файловой системы? Скорее всего, причиной. Что-то заставило компьютер зависнуть, а после его выключения файловая система осталась в некорректном состоянии, которое не смогло исправить даже `fsck`.

Может ли это произойти снова? Определенно может. Можно ли этого избежать? Отчасти, если принять должные меры.

Лучше всего было бы избежать зависаний в будущем. Так как Ваш ноутбук не юн, я бы начал с аппаратных причин зависания. Запустите расширенный тест `memtest86`, загрузившись с последнего DVD `Linux Format`. Еще одна возможная причина — перегрев. Если Вы сможете открыть корпус

ноутбука, то наверняка увидите на вентиляторах и радиаторах много пыли. Уберите ее, и компьютер будет нагреваться меньше. Проверьте и батарею, хотя это менее вероятно.

Если компьютер завис, сделайте все возможное, чтобы избежать жесткого выключения. Даже когда рабочий стол заблокирован и не отвечает, ядро напрямую следит за комбинацией клавиш "Magic SysRq" и может отреагировать на нее. Нажмите Alt и SysRq (она же Print Screen) и затем нажмите следующие клавиши по одной: R, E, I, S, U, B. Существуют разные мнемонические выражения, чтобы ее запомнить; мое любимое — "Reboot Even If System Utterly Broken [Перезагрузиться, даже если система полностью вышла из строя]", но проще всего запомнить, что это BUSIER наоборот.

R сбрасывает драйвер клавиатуры, E отправляет сигнал TERM всем процессам, I отправляет сигнал KILL, S синхронизирует все смонтированные файловые системы, а U размонтирует их. Две последние операции позволяют избежать повреждения файловой системы. В перезагружает компьютер.

## 7 GPT или DOS?

**В** У меня компьютер с этой новой штукой UEFI. Я хочу переустановить ОС (Mint) и собирался перейти на GPT, так как прочитал, что схема разбиения на разделы MS-DOS сегодня уже устарела. Во всех руководствах, которые я могу найти в сети, речь идет о дисках больше 2 ТБ.

**У меня диск всего в 500 ГБ. Есть ли причина, по которой не стоит использовать GPT на дисках размером меньше 2 ТБ?**

Purplepenguin, с форумов

**О** Все нет. В руководствах речь идет о дисках, больших 2 ТБ, потому что для них нужна специальная процедура. На этих дисках используются блоки по 4К вместо предыдущего стандарта в 512 байт. Большие блоки дают лучшую производительность, но и обращаться с ними нужно по-другому.

Чтобы окончательно не сбить с толку старые версии Windows размером блока, который в 8 раз больше того, к чему они привыкли, в прошивках создается видимость размера блока 512 байт. Затем это вызывает проблемы в программах форматирования диска Linux, которые пытаются делать все правильно и не выравнивают границы файловой системы для больших блоков, что снижает производительность.

Решение — выровнять все файловые системы по 4К. В последних версиях утилит для работы с диском в Linux так происходит по умолчанию.

Лично я пользуюсь GPT везде — это позволяет избежать многих ограничений и недостатков разбиения диска DOS. Таблицы разделов DOS гораздо более подвержены повреждениям, а GPT не только хранит всю информацию о разделах вместе в непрерывном блоке, но и хранит резервную копию этого блока в других местах диска. **Linux**

## Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, поскольку проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства.

Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) — она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файле, который вы сможете приложить к своему письму.

Не уступающий в удобстве альтернативный вариант — *lshw* (<http://ezix.org/project/wiki/Hardware-LiSter>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл **system.txt** к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
```

```
lspci >>system.txt
```

```
lspci -vv >>system.txt
```



## Часто задаваемые вопросы

### FISH

**› Это имеет какое-то отношение к надоедливим постам на форумах Linux Format?**

Нет, это тема серьезная, полезная и не имеющая прямого отношения к форумам.

**› Тогда что это?**

Это неточное сокращение фразы Files transferred over SHell protocol [файлы, передаваемые по протоколу Shell]". Ну как, удалось увидеть рыбку?

**› Не слишком интересно. А зачем это нужно?**

Если в двух словах, это позволяет вам работать с файлами на удаленном компьютере только с помощью SSH (или RSH). На удаленном компьютере не нужно никакое специальное ПО, только SSH и стандартные команды оболочки. На локальном компьютере для доступа

к этим файлам нужна программа с поддержкой FISH.

**› Где мне найти такую программу?**

Скорее всего, она у вас уже есть. *Midnight Commander* поддерживает FISH. Если вы наберете в *Midnight Commander*

```
cd /#sh:user@hostname/some/path
```

— будет показано содержимое каталога **/some/path** на удаленном компьютере. Если опустить часть **user@**, будет использовано текущее имя пользователя.

**› Значит, с FISH кто попал может прочесть файлы на моем компьютере?**

Нет, потому что используется SSH. Для входа в систему нужно ввести пароль или настроить аутентификацию с помощью ключей. В любом случае, вам придется подтверждать свою подлинность.

**› Все это хорошо, но Midnight Commander немного старомоден, не так ли? Не предусмотрена ли подходящая графическая программа?**

Вы пользуетесь KDE? Если да, откройте *Konqueror* (или мини-консоль, по Alt+F2) и перейдите по адресу

```
fish://user@hostname/some/path
```

С помощью волшебства **kio\_slaves** в *Konqueror* будет загружен удаленный каталог. Разбив окно на два, можно легко сравнивать и работать с локальной и удаленной копией каталога.

**› И я могу использовать ее для работы с файлами в Midnight Commander или Konqueror, да?**

Не только — работой с файлами дело не исчерпывается. Любая программа KDE, имеющая дело с файлами, может заниматься этим через FISH. Хотите изменить файл на веб-сервере на другом конце земного шара? Просто скомануйте

```
kate fish://me@my.server/iffypage-name.html
```

чтобы открыть файл в *Kate*, и нажмите Save [Сохранить] для отправки файла обратно на сервер.

**› Выходит, с помощью FISH можно управлять файлами в учетных записях таких систем удаленного копирования, как Rsync.net или Strongspace?**

К сожалению, нет. FISH работает только в том случае, когда вы можете войти в систему и выполнить базовый набор команд оболочки. В таких сервисах полный доступ по SSH запрещен, поэтому FISH работать не будет, однако есть альтернатива. SSHFS — файловая система FUSE, которая позволяет монтировать такую файловую систему с учетной записью, на котором разрешено только копирование файлов (командой **scp** или **rsync**), но тогда вам обязательно придется монтировать удаленную файловую систему, даже если нужен всего один файл.



# LXF HotPicks



**Ричард Смедли**

Активист FOSS и (вернувшийся к нам) давний автор LXF, Ричард вынюхивает под слоем опавших листьев в Интернете ценнейшие трюфели.

Emacspeak » PMW » JaxEdit » Drawpile » Git-info-bar » RedNotebook » ExMplayer » Warzone 2100 » Intricacy » PCGen » Clonezilla Live

## Аудиосреда рабочего стола

# Emacspeak

Версия: 39.0 Сайт: <http://emacspeak.sourceforge.net>

**E**macspeak — аудиопакет рабочего стола, построенный на Emacs. Этот доступ к вашей ОС «втемную» предназначен не только для слепых и слабовидящих пользователей. Аудио весьма удобно для страдающих дислексией и прочими расстройствами восприятия, и предлагает массу вариантов применения многостаночникам, вынужденным пребывать в отвлекающей среде. Например, мы пытались искать путь по указаниям наладонника через наушники и интерфейс Emacspeak к Google Maps.

Emacspeak 39.0 (Big Dog) — у всех релизов собачьи имена, начиная с 7.0 Labrador — теснее интегрировался с Chrome и Firefox (W3M, браузер Emacs, отлично работает с сайтами, предоставляющими

текст, хотя с сайтами на Ajax — похуже) и улучшил работу с Сетью, а также совместимость с Emacs 24.3.

Он также совместим с Emacs23, и уже сейчас должны иметься пакеты для вашего дистрибутива — поскольку большинство дистрибутивов с Emacspeak дружат. Его установка — всего лишь щелчок или apt-get, а вся работа с пакетами tcl и зависимостями emacspeak-espeak-server идет автоматически. Запустив его, вы получаете приветствие, заканчивающееся

```
/usr/share/emacs/site-lisp/emacspeak/sounds/cartoon-22k-mo
alarm.wav fill-object.wav no-answer.wav
alert-user.wav full.wav off.wav
ask-question.wav help.wav on.wav
ask-short-question.wav item.wav open-object.w
button.wav large-movement.wav paragraph.wav
center.wav left.wav progress.wav
close-object.wav mark-object.wav quit.wav
define-theme.el modified-object.wav right.wav
delete-object.wav n-answer.wav save-object.w
deselect-object.wav new-mail.wav scroll.wav
ellipses.wav news.wav search-hit.wa

/usr/share/emacs/site-lisp/emacspeak/sounds/chimes-stereo
alarm.wav fill-object.wav no-answer.wav
alert-user.wav help.wav off.wav
ask-question.wav item.wav on.wav
ask-short-question.wav large-movement.wav open-object.w
button.wav left.wav paragraph.wav
center.wav mark-object.wav progress.wav
close-object.wav modified-object.wav right.wav
complete.wav n-answer.wav save-object.w
define-theme.el network-down.wav scroll.wav
delete-object.wav network-up.wav search-hit.wa
```

» Не только голос: здесь есть звуковые сигналы для множества событий и для подсказок по навигации.

словами: «полностью готов к работе, и все мои схемы функционируют идеально»; затем последует звуковое воспроизведение всего, что вы печатаете, и каждой строки, на которую вы наводите курсор.

Тем, кто не привык к чтению в быстром темпе текстов и кода, придется к этому приноровляться, но если вам знаком Emacs, вы скоро восхититесь степенью интеграции с ним Emacspeak. Он читает сообщения состояния, говорит, на какую глубину вы погрузились в буфер, когда вы переключаетесь на него, и отлично справляется практически со всеми важными и неважными режимами и с внешними программами вроде MPlayer. Он даже работает с Tetris, встроенным в Emacs.

Открытие меню и некоторые другие действия сопровождаются звуковыми сигналами, которые также имеются для некоторых визуальных отметок, например, значков и пространственных взаимоотношений. Это аудиоформатирование (включающее полную поддержку Aural CSS от W3C), создает полноценный интерфейс для Web, чата, почты и вообще всего, что делает Emacs.

Чтение заметок по релизу — тоже огромное удовольствие, ибо они отличаются тонким юмором и четкостью, присущими стилю T.V. Raman, главного разработчика, который столь же узнаваем и в его блоге.

## «Вы восхититесь степени интеграции Emacspeak.»

## Исследуем интерфейс Emacspeak

### Tetris

Да, Emacspeak интегрируется даже с режимом Emacs Tetris, хотя это требует огромной сосредоточенности.

### Voice [Голос]

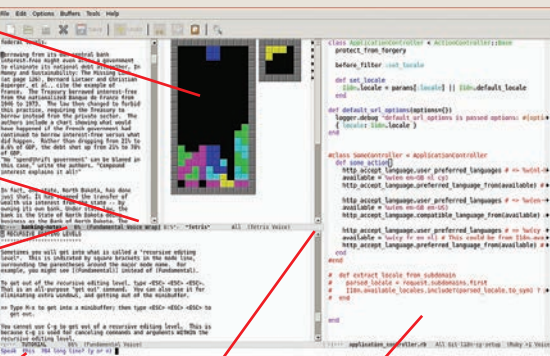
Бывалые пользователи Emacs быстро найдут новый режим Voice.

### Состояние [Status]

Когда вы переключаетесь в другой фрейм или буфер, Emacspeak зачитает сообщения строки состояния, в том числе о том, далеко ли вы прокрутили документ.

### Длинные строки

Emacspeak спросит у вас, прежде чем приступить к чтению длинных фрагментов текста.



### Окна

Фреймы Emacs пригодятся пользователям с любым качеством зрения для удобной организации рабочего пространства.

### Ruby

Охвачены все языковые режимы. Python'истам понравится, что Emacspeak сообщает о количестве отступов в каждой строке.

## Музыкальный наборщик

## PMW

Версия: 4.2.7 Сайт: [www.quercite.com/pmw.html](http://www.quercite.com/pmw.html)

Постоянные читатели видели и увидят на этих страницах немало программ, которые ведут начало со времен Amiga. *Philip's Music Writer (PMW)* принадлежит к другой части истории домашнего компьютера: изначально он был написан для Acorn RISC OS (тогда у него было менее успешное название *PMS*).

Долгое время он был доступен для платформ \*nix под GPL, и его по-прежнему разрабатывает Филип Хэйзел [Philip Hazel], более известный пользователям Linux как автор почтовика *Exim*. Релизы этого прочного, как скала, ПО выходят лишь пару раз в год с мелкими отладками, доработками кода и добавлениями небольших своеобразных функций — в данном случае, стало можно растянуть «или (немного) сплющить» множественные паузы.

Мы распаковали tar-архив и привычно не встретили никаких трудностей в обычной процедуре `compile` и `make`. В отличие от некоторых наших горячих новинок,

*PMW* сопровождается map-страницей и 180-страничным руководством; нет нужды упоминать, что набрано оно исключительно изящно.

В пакет включено также несколько текстовых файлов. Попробуйте запустить один из них в *PMW* —

```
pmw -includefont testfile
```

и взгляните на результат (см. рисунок сверху справа). Если вы не встраивали шрифты с помощью `-includefont`, вам придется делать ссылку на шрифты из директории шрифтов *GhostScript*, чтобы вы могли видеть их на экране и чтобы они использовались *ps2pdf*, если вы создаете файл PDF.

Опции печати включают два изображения A5 на листе A4, и то же для A4 на A3,



Отличный набор партитуры из простых текстовых файлов. Заломы кривых — это артефакты экрана: распечатываются они превосходно.

а также разнообразные опции для печати страниц в виде книги или брошюры. Включен также базовый вывод MIDI для проверочного прослушивания по завершении сочинения.

Некоторые музыканты слепо доверяют *LilyPond*, с ее прекрасной партитурой благодаря разметке на базе TeX, а некоторые, наоборот, волосы на себе рвут из-за сложностей формата ввода *LilyPond*. Мы бы рекомендовали дать *PMW* шанс, но жаль, что другие музыкальные программы, типа *MuseScore*, не способны выдавать результат в формате *PMW*. В *PMW* также отсутствуют фильтры ввода для MusicXML, хотя вывод *PMW* в MIDI имеет некоторые перспективы в плане экспорта — возможно, сообществу стоит вникнуть?

«Его разрабатывает Филип Хэйзел, автор почтовика Exim.»

## Редактор LaTeX

## JaxEdit

Версия: 0.30 Сайт: <http://jaxedit.com>

Кстати о наборе (вот примерно так мы и планируем HotPicks...). Многим известно, что *TeX* и *LaTeX* несравненны в подготовке математических формул к печати, и их продолжают изучать студенты-математики во многих университетах Великобритании, но применять их для обычных документов уже не спешат.

А зря, надо сказать; и поэтому мы рады видеть инструмент, который помогает быстро взглянуть на вывод ваших файлов *LaTeX*, когда вы или собираете их, или пытаетесь освоить этот прекраснейший из языков набора. *JaxEdit 0.30* предлагает новый интерфейс и отличную раскладку с быстрым откликом для мобильных устройств, что означает шансы овладеть навыками работы в *TeX* где угодно.

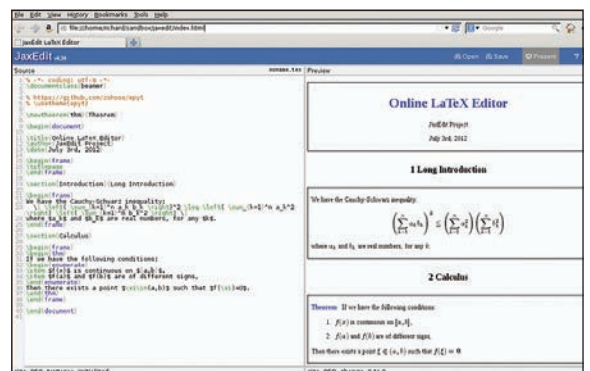
Мы скачали файл ZIP, распаковали его подборку файлов JavaScript и открыли страницу `index.html` в браузере (*Chrome* превосходно с этим справился; мы также

попробовали онлайн-версию на [jaxedit.com](http://jaxedit.com) в *Firefox*). Перед нами предстало классическое двухпанельное отображение исходника с выделением синтаксиса слева и отформатированным результатом справа, и мы приступили к изучению.

## Отличная презентация

Щелкните по Present наверху, и вы окажетесь в режиме Presentation, где каждый фрейм исходника — это кадр в браузере; и выглядят они намного лучше отформатированными и более четкими, чем типичный результат в *PowerPoint* (прости, *LibreOffice Impress*, к тебе это тоже относится). Попытки работы с нашими собственными файлами *LaTeX* открыли нам несколько

«Помогает освоить LaTeX, прекраснейший из языков набора.»



Отлично работает со встроенными примерами, но пока что доступны не все команды.

полезных команд, пока что не учтенных в форматировании, но ведь это еще только версия 0.30 — мы уверены, что нас ждет гораздо лучшее форматирование, а поскольку *JaxEdit* написан на JavaScript, в него очень легко добавлять собственные усовершенствования, коли охота.

Если вы хотите познакомиться с ним, не устанавливая редактор, просто щелкните по Start на сайте проекта, и получите в скачивании те же готовые примеры, которые покажут вам все достопримечательности *TeX*.

Программа для совместного рисования

# DrawPile

Версия: 0.7.2 Сайт: <https://github.com/callaa/>

SourceForge завален покинутыми проектами свободного ПО. Иногда они чисто по Дарвину проигрывают в борьбе за выживание, уступив лучшему коду, но иногда разработка случайно прекращается по иным причинам. И мир лишается хорошей программы. Крайне редка ситуация, чтобы программа вернулась к активной разработке после пятилетнего перерыва — а вот с *DrawPile* так и произошло.

*DrawPile* недавно не только воскресла, но и получила «полностью переписанный сетевой протокол и сервер, и... поддержку уровней [и] файловый формат OpenRaster».

За месяц после перезапуска последовало еще два основных релиза от Калле Лаакконена [Calle Laakkonen], что указывало на значительную активность, поэтому мы отложили наши карандаши и подключили планшет USB, желая увидеть, что же умеет *DrawPile*. Перед запуском `tar` и `cmake` понадобятся еще `zlib` и `qt5`. В Ubuntu могло создать проблему

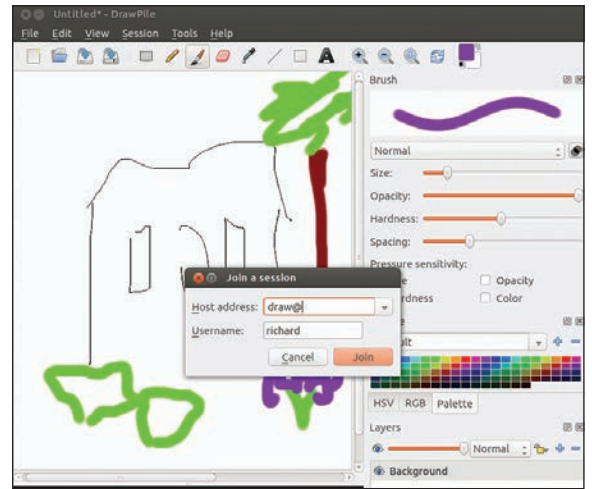
отсутствие *Qt 5.1*, но добавьте `getdeb.net` в `apt/sources.list`, и вы получите версию для 5.0.2 в Ubuntu.

Как видите, художественный редактор был занят, и с наброском, которого устыдился бы даже дошкольник, пришлось вылезти мне; но зато было с чем приступить к знакомству с интерфейсом.

Более серьезные художники заметят большое разнообразие кистей, а дальнейшая разведка интерфейса выявит плавное рисование и множество режимов смешивания, чувствительность к давлению для планшетов и довольно милые функции слоев, известные по *GIMP*. Есть также и бонус — возможность указать, каким пользователям доступен какой слой.

Доступ для нескольких пользователей обеспечивает встроенный сервер —

**«Сервер позволяет работать над рисунками друг друга.»**



► Пусть вы и неважный рисовальщик — вы обретаете силу, сотрудничая с другими в выработке идей.

он позволяет делать заметки и сотрудничать над рисунками как в офисе, так и за его пределами. Понятно, что тут полно как общественно полезных, так и самых дурацких областей применения, но *DrawPile* также весьма удобна для обмена идеями о продукте или даже о графике для своего нового приложения, чтобы программисты, работающие над клиентской стороной, смогли оценить ее вид в интерфейсе.

Плагин оболочки

# Git-info-bar

Версия: 1.2.0 Сайт: <http://bit.ly/1maqW1U>

Э тот удобный плагин оболочки изначально был написан для *Bash*, но адаптирован для других оболочек, и в последних релизах разработчики упорно трудятся над отладкой и проблемами кросс-платформенности, чтобы расширить применение приложения. Оно размещено — что и неудивительно — на GitHub; мы клонировали репозиторий, затем запустили скрипт `install.sh` (предусмотрен также и `uninstall.sh` — пять баллов им за это), и запустили

```
~/bashrc
```

и вот появилась наша информационная строка *Git*. Она сидит прямо над подсказкой с долларом, сообщая, в какой вы ветви и какова контрольная сумма, а также о статусе изменений и задумок (при наличии таковых).

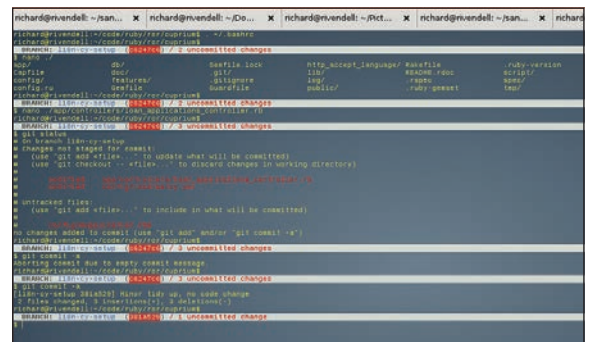
Если у вас есть неотправленные изменения, то контрольная сумма отображается красным, вместе с сообщением о числе

изменений, тоже красным. Этот цвет также употребляется при работе с главной веткой. Наличие строки посередине экрана, если именно там находится окно приглашения *Bash*, сначала несколько обескураживает, но в этом есть смысл по мере накопления изменений, чтобы вы могли оглянуться на визуальную пунктуацию вашей последней работы в своем репозитории.

## Удобство в действии

Выход в директорию за пределами репозитория *Git* приводит к тому, что отображение исчезает, но стоит нам вернуться в репозиторий, как возвращается и строка состояния. Вот такие небольшие штрихи

**«Программу не назывешь потрясающей, но это удобная утилита.»**



► *Git-info-bar* информирует вас о том, что происходит в *Git*, но не лезет к вам под ноги.

и помогают программе постепенно завоевывать вашу симпатию.

Эту программу не назовешь потрясающей, но это удобная утилита, которой неплохо обзавестись, если вы занимаетесь разработкой из командной строки, вызывая редактор по необходимости вместо того, чтобы, скажем, постоянно сидеть в *Emacs*. Написанная на сочетании скрипта оболочки и Perl, она являет собой хороший пример небольшой утилиты старого стиля для \*nix, которая по-прежнему используется в современных средах GitHub и написания web-приложений. В стадии разработки — версия 2.0, которая позволит индивидуально настраивать вид отображения, через файл `config`.



Программа ведения дневника и заметок

# RedNotebook

Версия: 1.8.0 Сайт: <http://rednotebook.sourceforge.net>

**Н**е перепутайте ее с книгой Пола Остера [Paul Auster] с тем же названием: *RedNotebook* [англ. Красная Тетрадь] — это графический дневник или журнал, который отслеживает ваши мысли и заметки ежедневно.

Пакеты свежего релиза имеются для самых популярных дистрибутивов. Она написана на Python с интерфейсом GTK+, и нас впечатлила ее скорость по сравнению и с настольной, и с онлайн-альтернативами. Частично причина тут в простом интерфейсе пользователя, который не мешает вам собирать свои мысли и записывать их, а вдобавок — весьма мощные функции находятся на расстоянии всего пары щелчков от вас.

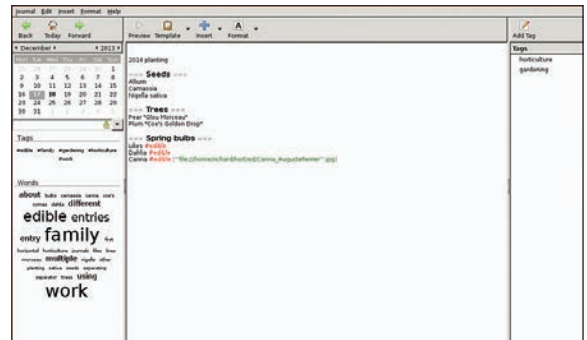
Введите текст в простом формате wiki и переберитесь из режима Edit в Preview, чтобы увидеть результат. Дневники хранятся в виде текста в `~/rednotebook/data`. В базе данных необходимости нет, и директория `~/rednotebook` содержит также файлы настройки, журнал и папку для настраиваемых шаблонов.

Ссылки распознаются автоматически, и можно добавлять рисунки и файлы. Сохранение тоже автоматическое — прямо как в настоящем дневнике на бумаге! Использование одиночных кавычек — двух для вставки в строку и трех на отдельной строке для целого абзаца — позволит ввести код для цвета шрифта, или, например, для вывода формул *LaTeX*.

## Как на бумаге

Им можно пользоваться для чего угодно, что вам нужно будет записать: пометки о встречах, рецепты, предписания, сайты, наблюдения, музыку, дневниковые записи, стихи — всего, что мгновенно распознается по комбинации определенных пользователем тэгов; причем контент допускает поиск.

**«Сохранение тоже автоматическое — как в дневнике на бумаге!»**



➤ При добавлении текста в заметку или дневник, ваши записи автоматически изменят облако тэгов в дневнике.

Если у вас были проблемы с другими приложениями дневника или заметок, возьмите *RedNotebook* на тест-драйв. Экспорт журналов доступен в формате PDF, HTML, *LaTeX* или простого текста, или как резервная копия в архив ZIP.

Если вам не нужна синхронизация *EverNote* для нескольких устройств (мы уверены, что читатели *LXF* могут преспокойно сделать зеркала текстовых файлов *RedNotebook* на всех своих ПК, телефонах и планшетах), вы поймете, почему его соперник из области свободного ПО завоевал столько преданных поклонников; хотя вам в жизни не удастся отобрать у данного автора режим org из *Emacs*.

GUI к MPlayer

# ExMplayer

Версия: 3.2.0 Сайт: <http://exmplayer.sourceforge.net>

**Е**xMplayer (*Extended MPlayer*) — это нечто большее, чем обычный GUI для *MPlayer*. Он добавляет поиск по эскизам и инструменты — программу нарезки, аудиоэкстрактор и аудиокодек — а также кучу кодеков. Он накрывает не только файлы DVD (VOB) и VCD (MPG, DAT) и пару контейнеров и форматов сжатия, но и еще 416 видеокодеков и 199 аудио.

В версии 3.0 появилось воспроизведение 3D, а в 3.2 улучшено управление экраном и сделаны мелкие поправки. 3D-совместимость распространяется не только на разные очки и форматы ввода, но также позволяет смотреть 3D-фильмы как обычные 2D.

Уйма кодеков и 3D — дело хорошее, но повседневное использование делают удобным именно всякие симпатичные дополнения: например, появление эскизов по мере перемещения мыши по оси времени фильма.

Можно скачать tar-архив исходника или взять его в GitHub, но в Ubuntu есть собственный репозиторий *ExMplayer*, который будет работать и с совместимыми дистрибутивами. Ранее он именовался *rphMPFE*; новое и (немного) более разумное название появилось в версии 2.0 вместе с улучшением отображения поиска, который стал своего рода уникальной фишкой *ExMplayer*.

Возможно, вы также сочтете — как и мы — что аудиоэкстрактор и аудиокодек удобнее, чем более утонченный, но отдельный инструмент. Приятным дополнением стал потоковый браузер — он очень удобен для таких визуальных

**«3D-совместимость позволяет смотреть 3D-фильмы как 2D.»**



➤ Ну есть у вас видеоплеер, но слабо ему справиться с 416 кодеками и всеми типами 3D!

медиа, как фильмы — который находится в View > Flow Browser. Он задуман для медиа-центров с большим экраном и мышью со скроллингом, подключенной по Bluetooth, однако на ноутбуке тоже выглядит и работает неплохо. Правда, на более старых машинах с меньшим ОЗУ уже похуже.

Меню *ExMplayer* показались нам довольно логичными по структуре; большинство требуемых функций доступны через GUI. И, в дополнение к солидному числу функций, имеются также фильтры, поддержка плагинов, усилитель громкости и режим Family Friendly.

## HotGames Развлекательные приложения

Гибрид стрелялки и тактики

## Warzone 2100

Версия: 3.1.1 Сайт: <http://wz2100.net>

**М**ы знаем, что читатели LXF любят открывать огонь — по крайней мере, виртуальный: правила пожарной безопасности нередко запрещают тестировать на рабочем месте свой бомбовоз-квадрикоптер. И мы также знаем, что вы любите применять свои «маленькие серые клеточки», чтобы порвать врага в клочья хитрым и стратегически выверенным способом. Тогда что может быть лучше RTS, которая требует стратегического и тактического планирования?

Запустив ее, вы услышите киношный саундтрек, изготовленный для более мощной акустической системы, чем на вашем ноутбуке. Проприетарная *Warzone 2100* перешла под лицензию GPL в конце 2004 года, но еще четыре года ушло на выяснение всяческих вопросов с лицензией саундтрека и фильма,

и тогда стало можно распространять и их тоже.

После (случайного) ядерного Армагеддона вы становитесь участником одной из трех разведгрупп, которые занимаются поисками технологий для восстановления, но на вас нападают другие организации. Игроки лавируют между группами и обнаруживают, что врагами управляет саморазвивающийся компьютерный вирус.

В 3.1.1 намного больше мелких изменений, чем можно ожидать от версии с таким номером. Если последний раз вы устанавливали *Warzone 2100* с диска

**«Все здорово, но пальбу придется сочетать с раздумьями.»**



► Если осмотрительность — лучшая часть доблести, то «отступление со средними потерями» — довольно неплохая отговорка...

LXF90, пора обновиться. Графический движок iViS 3D предоставляет отличный UI, поддерживающий все ожидаемые увеличения масштаба, панорамы и прокрутки поля боя для игрока. Следя за датчиками на предмет вражеских боевых единиц, игроки координируют атаки, и каждый уровень ограничен по времени, что при лимите ресурсов требует стратегической мысли и тактического подхода к сражениям. Короче, все здорово, но пальбу всласть придется сочетать с раздумьями.

## Соревнование по составлению головоломок

## Intricacy

Версия: 0.2.2 Сайт: <http://bit.ly/1i50pvY>

**Е**сли вы любитель решать головоломки, почему бы не попробовать их составлять? *Intricacy* противопоставит ваши умения умениям других игроков, и это задаст тренировку вашему мозгу.

Вдохновение было почерпнуто в солидной многопользовательской игре Джейсона Ропера [Jason Rohrer] со взломами и защитой жилища, *The Castle Doctrine*. Однако в *Intricacy* игроки скорее создают из замков головоломки, а не запирают на них дом: противник решает головоломку, управляя крюком и гаечным ключом, чтобы перемещать части замка и очистить отверстие для болта.

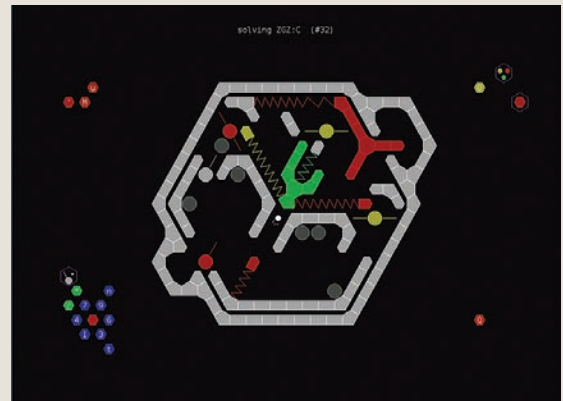
Проталкивание и повороты инструментов сжимают пружину и проталкивают и вытягивают блоки, на которые, в свою очередь, воздействуют другие части механизма замка. Блоки соединены пружинами, стержни закручиваются

и шарики приходят в движение под воздействием высвободившихся сил. Там, где встречаются силы, появляются правила для комбинированного результата.

Встроенный сервер может запускать игры в локальной сети или в Интернет, и вы можете указать свой порт вместо порта по умолчанию 27001. Можете соединиться с официальным сервером или указать иной. Но где бы вы ни искали соперников, только играя против них сможете вы проверить свою храбрость.

*Intricacy* написана на Haskell, в версиях SDL и Curses (и форматом для сохранения уровней является ASCII-art). В этом релизе есть ряд полезных отладок

**«В Intricacy игроки создают из замков головоломки.»**



► Куда повернуть? Стержень? Растянуть пружину? Или передвинуть блок? Каждый замок — это разумный баланс сил.

и мелких поправок, однако там еще есть над чем поработать в плане улучшения дружелюбия к начинающему пользователю, чтобы было легче освоиться.

Пока игра не слишком широко известна, но у нее большой потенциал, и она определенно стоит того, чтобы пробиться через стадии начинающего. Кроме того, это не очень большая программа, и физика игры отлично объяснена. Если вы ищете интересную программу Haskell, чтобы улучшить свои навыки функционального программирования, то вот вам.

## Генератор персонажей ролевок для систем d20

## PCGen

Версия: 6.2.0 RC2 Сайт: <http://pcgen.sourceforge.net>

Д о появления видеоигр каждый, кто жаждал потратить несколько часов на путешествие по фантастическим территориям, вынужден был либо перечитывать Толкиена, либо играть в *Dungeons & Dragons*. D&D выжили, поскольку картинки у вас в голове куда ярче, чем на экране, однако с помощью *PCGen* вы сможете использовать свой ноутбук для создания персонажей (Player Characters, PCs) для любой ролевой игры (RPG) с бросанием кубика

Эта версия *PCGen*, на JRE 1.6 или 1.7, добавляет в процесс создания персонажей множество улучшений, а также множество дополнений в RPG *Pathfinder*.

Большая часть предлагаемых вам для скачивания 60 МБ — это файлы данных для кампаний: запустите файл **pcgen.jar**, и вашим первым заданием будет выбрать одну из этих кампаний. Мы решили воспользоваться пошаговым созданием персонажа, с целью полностью осознать, как это работает.

Возможно, вам придется — а нам так пришлось не раз — обратиться к онлайн-помощи (нажав на F1), когда вы будете искать нужную вкладку для какой-нибудь мрачной характеристики вашего класса персонажа.

Помимо этого и некоторой корявости Java, присущей многим системам Linux, критиковать нечего, поскольку команда *PCGen* продолжает пытаться добавить все разрешенные законом детали RPG. Поддержка сообщества предоставляется через активную группу Yahoo! Group, которая выдает множество запросов на добавление новых функций в новых релизах.



➤ Данные кампаний практически исчерпывающие, но если вам мало, добавьте еще и свои.

## Создание образов и клонирование диска

## Clonezilla Live

Версия: 20131125-saucy Сайт: <http://clonezilla.org>

Разработчики *Clonezilla* весьма активны: младшие релизы у них выходят каждые несколько недель. Хотя обычно это обновления до свежего Debian Sid (нестабильного дистрибутива), добавляющие пакеты из него, они также регулярно вставляют и полезные новые функции.

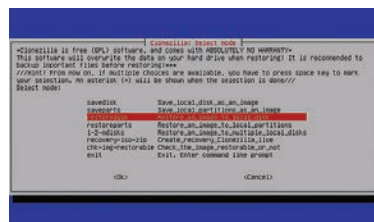
Серверное издание *Clonezilla* удобно для массового клонирования дисков в сети, и в первую очередь требует наличия сервера DRBL; а *Clonezilla Live* позволяет вам запускать клонирование диска на собственной машине без лишней мороки.

Раньше она была основана на Debian Live, но новая версия выбрала за основу Ubuntu 13.10. Вы можете скачать загружаемый ISO — или архивированный пакет для загрузки с USB-брелка; именно так поступили мы.

После загрузки USB-брелка нужно сделать выбор, причем есть опция «загрузки

в ОЗУ», которая позволяет удалить носитель или даже переписать его, хотя boot-to-media почему-то не разрешает создать при этой опции ISO восстановления. Клонирование дисков и их восстановление было беспроblemным, а самое замечательное в *Clonezilla* — это ее хваленая скорость.

В поддерживаемых файловых системах (ext2, ext3, ext4, VMFS, FAT, NTFS и ряд других) она просто копирует используемые блоки. В неподдерживаемых используется **dd** для более медленного посекторного копирования. Очень удобно, что поддерживается еще и LVM2. **LXF**



➤ Предлагаемые *Clonezilla* опции выбора, выделенные цветом, умело привлекают ваше внимание.

## Также вышли

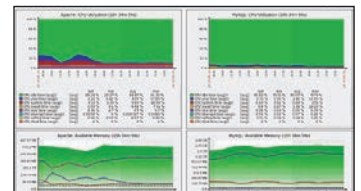
Новые и обновленные программы, тоже достойные внимания...

## ➤ Camera Life 2.6.6

Удобная локальная и удаленная система каталогизации фотографий.  
<https://github.com/fulldecent/cameralife>

## ➤ Zabbix 2.2.1

Программа мониторинга с открытым кодом, уровня предприятия.  
[www.zabbix.com](http://www.zabbix.com)



➤ Если вы не настроите мониторинг как следует, он рухнет.

## ➤ Volumio 1beta

Музыкальный плеер, встраиваемый в Raspberry Pi, Cubox и т.д. на радость меломанам.  
<http://volumio.org>

## ➤ feedparser 5.1.3

Библиотека Python для парсинга всех форматов новостных лент, включая Atom и RSS.  
<http://bit.ly/1gUsMJB>

## ➤ XOWA 0.12.0

Редактируйте Wikipedia, Wiktionary, Wikisource и Wikiquote оффлайн.  
<http://xowa.sourceforge.net/>

## ➤ CWIS 3.0

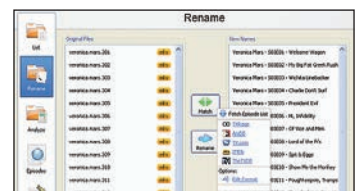
Собирайте, организуйте, делитесь и рекомендуйте информацию по онлайн-ресурсам.  
<https://scout.wisc.edu/cwis>

## ➤ FileBot 3.8

Упорядочивайте и переименовывайте фильмы и телешоу и скачивайте субтитры и т.д.  
[www.filebot.net](http://www.filebot.net)

## ➤ Bluestar DeskLite-E17 Ed

Настольный/мультимедиа-дистрибутив на базе Arch Linux с оптимизированной производительностью.  
<http://bluestarlinux.org>



➤ *FileBot* позаботится о вашей экраномании.

# На диске

Дистрибутивы, приложения, игры, подкасты и всякое-разное...

Лучшее из Интернета, упакованное в 8 ГБ качественного DVD.



## К лучшему

С тех пор, как многие пользователи Ubuntu глянули на Unity и разом отшатнулись, Linux Mint обретает все большую популярность, заслуженную благодаря введению не нового, а, наоборот, старого и удобного, но современным способом. Linux Mint признал, что далеко не всем нужны перемены ради самих перемен, и создал способ работать в прежней манере, но без обветшалых программ, наглядно доказав ростом своей доли рынка, что воля пользователей — важный фактор.

Linux Mint также показал, что, возможно, мы, линуксоиды, вовсе не такие уж передовые, как нам хочется думать. При первом же признаке смены подхода мы дружно вскинули лапки кверху и побежали искать то, к чему привыкли, не дожидаясь выяснения, будет ли в данном случае «новое» означать «лучшее». Это, конечно, сильное обобщение: кому-то Unity, наоборот, пришелся по душе, а кому-то все равно в принципе (таких людей именуют пользователями KDE/Xfce/консоли).

Ну да, новое часто бывает и худшим, но мы, похоже, стали менее авантюристами, чем на заре Linux; или просто повзрослели?

*Neil*

## » Важно ВНИМАНИЕ!

### Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу [disks@linuxformat.ru](mailto:disks@linuxformat.ru) или телефону +7 (812) 309-0686.

## Дистрибутив Linux

# Fedora 20



Red Hat был первым дистрибутивом Linux, который я попробовал — и было это на Amiga 4000. С тех пор Red Hat прошел долгий путь, а его свободная версия от сообщества зашла еще дальше. Fedora сейчас в версии 20, и у нас есть для вас два live ISO на DVD, по одному на каждую среду рабо-

чего стола — GNOME и KDE. Как это уже стало привычным для Linux Format DVD, вы можете скопировать эти ISO на диск или USB-брелок для дальнейшего распространения, или загрузиться с DVD и выбрать Fedora 20 в меню. Обе версии 64-битные; пожалуйста, не пытайтесь запускать их на BBC Model B.

## Дистрибутив Linux

# Linux Mint 16

Linux Mint стал чрезвычайно популярен за последние пару лет. Он и раньше был неплох, но часто рассматривался всего лишь как очередной клон Ubuntu, хотя и из лучших. Введение им рабочих столов Mate и Cinnamon, продолжавших традицию удобства — в противовес Unity и GNOME 3 — все изменило, притянув пользователей, которые не хотели ломать свои привычки.

Согласно философии Mint, пожеланиям сообщества приписывается главный приоритет, и дистрибутив успешно продолжает развитие в том же духе, под руководством своего бессменного вождя Клемана Лефевра [Clement Lefebvre].

В нашей версии Mint 16 собран с рабочим столом Mate плюс самые последние

обновления системы безопасности и отладки. Это 32-битный дистрибутив — просто вставьте диск в дисковод, и все у вас получится. Спасибо, Клеман!





# Новичок в Linux? Начните отсюда!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент *MS Office*?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы приводятся в [Index.html](#) на диске.

## Из журнала

### Subsonic

Настройте свой музыкальный потоковый сервис!

### Код Scratch

Примеры кода на Scratch, плюс ПО для учебника.

### Удаленное журналирование

Попробуйте дополнения к журналам, типа *Graylog2* и *AfterGlow*.

## LXFHotPicks

### Clonezilla

Программа разбиения диска на разделы/клонирования диска для развертывания, резервного копирования и восстановления.

### DrawPile

Программа рисования, позволяющая делиться рисунками с другими в реальном времени.

### Emacspeak

Речевой интерфейс для *Emacs* делает его полноценным звуковым рабочим столом.

### ExMplayer

Воспроизводит любые файлы без внешних кодеков.

### Git-info-bar

Плагин оболочки *Git* для просмотра атрибутов *Git* рабочей копии вашего репозитория.

### Intracacy

Сетевая игра, где вы создаете головоломки для ваших противников.

### JaxEdit

Онлайн-редактор с просмотром *LaTeX*, языка разметки документа.

### PCGen

Генератор персонажей для ролевых.

### PMW

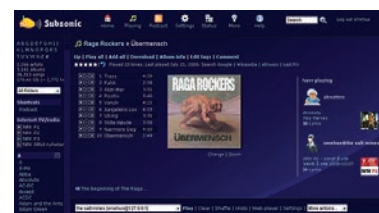
*Phillip's Music Writer* для высококачественной партитуры.

### RedNotebook

Современный дневник с навигацией по календарю, функциями экспорта и облаком слов.

### Warzone2100

Битва за передел мира после практически полного вымирания человечества в результате ядерного удара.



## И еще!

### Системные инструменты

#### Главное

**Checkinstall** Установка *tar*-архива с помощью менеджера пакетов.

**GNU Core Utils** Основные утилиты, обязанные быть в каждой операционной системе.

**Hardinfo** Инструмент тестирования системы.

**Ядро** Исходник последней версии ядра.

**Mentest86+** Проверьте свое ОЗУ на сбои и ошибки.

**Plop** Простой менеджер загрузки для запуска операционных систем.

**RaWrite** Создавайте загрузочные диски в Windows.

**SBM** Независимый от ОС менеджер загрузки с простым в работе интерфейсом.

**Wvdial** Соединяйтесь с Интернетом через телефонный модем.

#### Чтение

#### Книжная полка

**The Cathedral and the Bazaar** [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond], объясняющий преимущества открытой разработки.

**Intro to Linux** Руководство для новичков.

**Linux Kernel in a Nutshell** [Ядро Linux в двух словах] Введение в ядро, написанное магистром хакеров Греггом Кроа-Хартманом [Greg Kroah-Hartman].

**Справочник администратора Debian** Руководство для системных администраторов.

**Словарь Linux** Linux от А до Z.

**Окунитесь в Python** Мастер-класс по этому популярному языку.

**Руководство по созданию скриптов Bourne Shell** Осваивайте скрипты оболочки.

**Расширенное руководство по созданию скриптов Bash** Больше информации для опытных пользователей.

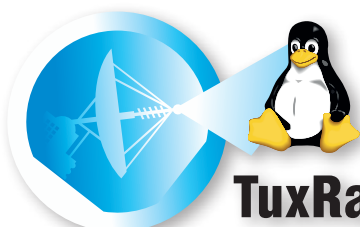
**Руководство сисадмина** Основы работы с системой.



## Подкасты

### Сезон 6 Эпизод 0 — пилот

TuxRadar возвращается — слушайте новый сезон и вводную серию с новыми лицами... в смысле, голосами.



TuxRadar



## Пропустили номер?



Закажите его через сайт [www.linuxformat.ru](http://www.linuxformat.ru) в «ГНУ/Линуксцентр»! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:



**LXF177**  
Декабрь 2013

250 руб.

- » Дрожи, Android Конкуренты давно уж подпирают
- » Держите в курсе Собираем сплетни читалками новостей
- » Rust Язык от взломщиков защитит
- » Кстати о взломщиках LMD на страже системы

**LXFDVD:** openSUSE 13.1 (64-разрядная сборка), GnuM 2013.09, LXLE, OS/4 OpenLinux, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_177/](http://shop.linuxformat.ru/lxf_177/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_177/](http://shop.linuxformat.ru/elxf_177/)



**LXF178/179**  
Январь 2014

250 руб.

- » Навалимся толпой Революция в экономике?
- » Начинающим Взлетная полоса в небо Linux
- » Q15 Знаменитая платформа разработки
- » Штампует сайты Да, это реально легко — если с Drupal

**LXFDVD:** Fedora 20 (64-разрядная сборка), Elementary OS Luna, PinguOS, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_178-179/](http://shop.linuxformat.ru/lxf_178-179/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_178-179/](http://shop.linuxformat.ru/elxf_178-179/)



**LXF180**  
Февраль 2014

250 руб.

- » Запах мяты Mint, симпатия пользователей
- » Закину на облако Сервисы хранения данных
- » OpenStreetMap Карту мира — своими руками
- » Платы для Raspberry Pi Расширяем компьютер-крошку

**LXFDVD:** Linux Mint 16, Clonezilla 2.2.1, openSUSE 13.1, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_180/](http://shop.linuxformat.ru/lxf_180/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_180/](http://shop.linuxformat.ru/elxf_180/)

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через [www.linuxformat.ru/subscribe/](http://www.linuxformat.ru/subscribe/) или [www.linuxcenter.ru](http://www.linuxcenter.ru), получают электронную версию в подарок! На сайте [shop.linuxformat.ru](http://shop.linuxformat.ru) вы также сможете приобрести предыдущие выпуски LXF (если успеете).

Подписывайтесь на сайте [www.linuxformat.ru/subscribe/](http://www.linuxformat.ru/subscribe/)

Телефоны отдела подписки:

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



## Linux Format ВКонтакте

Вступайте в нашу  
группу [vk.com/linuxform](http://vk.com/linuxform)

На странице LXF ВКонтакте вы найдете:

- » Новости о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

## Информация о диске

### Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

### Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

### Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

### Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя\_программы-1.0.1.i386.rpm** — вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя\_программы-1.0.1.i386.deb** — такой же пакет, но уже для Debian;
- » **имя\_программы-1.0.1.tar.gz** — обычно это исходный код;
- » **имя\_программы-1.0.1.tgz** — тот же файл, что и выше этажом по списку: «tgz» — это сокращение от «tar.gz»;
- » **имя\_программы-1.0.1.tar.bz2** — тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя\_программы-1.0.1.src.rpm** — также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя\_программы-1.0.1.i386.FC4.RPM** — двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя\_программы-1.0.1.ppc.Suse9.rpm** — двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя\_программы-devel-1.0.1.i386.rpm** — версия для разработчиков.

### Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

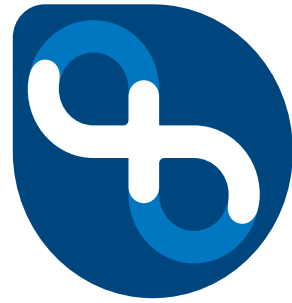
**Внимательно прочтите это перед тем, как использовать LXF DVD!**

**А ТАКЖЕ: X2Go, BlueZ 5, Perl 5.18 и другие обновления**

# fedora 20

Ядро 3.11 » Gnome 3.10 » Gimp 2.0 » KDE Plasma Workspaces 4.11

32-разрядная сборка



**А ТАКЖЕ: Emacspeak 39.0, RedNotebook 1.8.0, 10 книг о Linux и многое другое...**



# Mint 16 Mate



# fedora 20 Live

64-разрядные сборки с рабочими столами Gnome и KDE

Март 2014  
**LXF DVD 181**

**LINUX**  
FORMAT

Март 2014  
**LXF DVD 181**

**LINUX**  
FORMAT

## Сторона 1

### ДИСТРИБУТИВЫ

**Fedora 20** Обновленная версия популярного дистрибутива. Установочный DVD, 32-разрядная сборка (загрузка с LXFDVD)

**Minit 16** с рабочим столом Mate, 32-разрядная сборка (загрузка с LXFDVD)

## Сторона 2

### ДИСТРИБУТИВЫ

**Fedora 20** Обновленная версия популярного дистрибутива. Установочный DVD, 32-разрядная сборка (загрузка с LXFDVD)

### ДОКУМЕНТАЦИЯ — 10 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

**Bash Scripting** Подробное руководство по программированию на Bash

**Bourne Shell Scripting** Начальное руководство по программированию на Bash

**Cathedral Bazaar** Классический текст Эрика Раймонда [Eric S Raymond] «Сбор и базар»

**The Debian Administrator's Handbook** Руководство администратора, написанное разработчиками Debian

**Dive Into Python** Учебник по программированию на Python

**Emacspeak 39.0** Аудиосреда рабочего стола

**ExMPlayer 3.2.0 GUI** к MPlayer

**Git Info-bar 1.2.0** Плагин оболочки для работы с Git

**Intracacy 0.2.2** Игра-соревнование по составлению головоломок

**JaxEdit 0.30** Редактор LaTeX

**PCgen 6.2.0 RC2** генератор персонажей ролевок для систем d20

**PIWM 4.2.7** Редактор музыкальной партитуры

**RedNotebook 1.8.0** Программа ведения дневника и заметок

**Warzone 21003.11** Игра, гибрид стрелялки и тактики

**Intro to Linux** Начальное руководство по Linux

**Linux Dictionary** Словарь Linux, объясняющий специфическую терминологию

**Linux Kernel in a Nutshell** Описание ядра Linux, созданное одним из его выдающихся разработчиков — Греггом Краа-Хартманом [Greg Kraah-Hartman]

**System Administrators Guide** Руководство по базовому администрированию Linux

**GNU Tools Summary** Руководство по работе в командной строке и обзор основных утилит GNU

### НОТРИСКИ

**Clonezilla 20140114-saucy** Live CD для работы с дисковыми разделами, 32-разрядная сборка (ISO-образ)

**Drawpile 0.7.2** Программа для совместного рисования

**Пожалуйста, перестаньте использовать с диском диск, озвученный в журнале на стр. 109!**

**КОММЕНТАРИЙ** Присылайте ваши пожелания и предложения по электронной почте: [info@linuxformat.ru](mailto:info@linuxformat.ru)

**ДЕФЕКТИВНЫЕ ДИСКИ** В маловероятном случае обнаружения дефектов на данной диске, обращайтесь, пожалуйста, по адресу [disk@linuxformat.ru](mailto: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 валидированный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. Использование редактора Linux Format не в состоянии принести на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных нашей программой или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с нашего сайта: [www.linuxformat.ru](http://www.linuxformat.ru)

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

## Создание установочных дисков при помощи cдrecord

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.





UNIXFORUM.org



Есть вопросы?  
Задавайте!

<http://unixforum.org/>

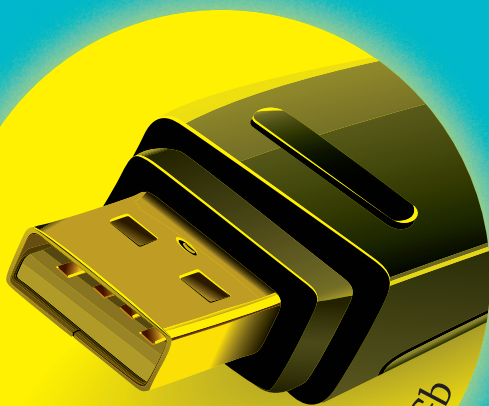
## Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств  
в разных городах России  
позволит вам оптимизировать  
процессы логистики и доставки товара

ПОДРОБНЕЕ О ПАРТНЕРСКОЙ ПРОГРАММЕ:  
[WWW.LINUXCENTER.RU/PARTNER/](http://WWW.LINUXCENTER.RU/PARTNER/)



## ДИСТРИБУТИВЫ GNU/LINUX НА USB FLASH



Linux Mint 16



Ubuntu 13.10

**А ТАКЖЕ**  
версии для юриди-  
ческих лиц —  
с лицензионным  
договором  
присоединения

USB Flash 8 Gb

**495** рублей

[www.linuxcenter.ru/shop/distros/usb/](http://www.linuxcenter.ru/shop/distros/usb/)



UnixEducationCenter

ОБУЧЕНИЕ И СЕРТИФИКАЦИЯ ОС LINUX



ПО ВСЕМ ВОЗМОЖНЫМ УРОВНЯМ:

**RHCSA** – RED HAT CERTIFIED SYSTEM ADMINISTRATOR

**RHCE** – RED HAT CERTIFIED ENGINEER

**RHCA** – RED HAT CERTIFIED ARCHITECT

[www.unixedu.ru](http://www.unixedu.ru)

+7 (812) 611-15-75    [mail@unixedu.ru](mailto:mail@unixedu.ru)

Санкт-Петербург, Черноморский пер., д.4

ГНУ/Линуксцентр  
приглашает на работу!



**ВАКАНСИЯ:** Разработчик систем электронного документооборота

**ТРЕБОВАНИЯ:**

- » Опыт внедрения и сопровождения систем электронного документооборота на базе Alfresco
- » Умение описать бизнес-процесс
- » Знание SQL, понимание принципов построения и функционирования баз данных
- » Знание СЭД, отличных от Alfresco, приветствуется

**ОБЯЗАННОСТИ:**

- » Разработка систем с нуля, сопровождение
- » Настройка системы в соответствии с требованиями бизнеса
- » Написание инструкций для конечных пользователей

**ПОДРОБНЕЕ:** [www.linuxcenter.ru/vacancy/](http://www.linuxcenter.ru/vacancy/)

## Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



[WWW.LINUXCENTER.RU/SHOP/GIFTS/](http://WWW.LINUXCENTER.RU/SHOP/GIFTS/)

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

**РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ****Главный редактор**Кирилл Степанов [info@linuxformat.ru](mailto:info@linuxformat.ru)**Литературный и выпускающий редактор**

Елена Толстякова

**Переводчики**

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова, Ирина Шулакова

**Редактор диска**

Кирилл Степанов

**Верстка, допечатная подготовка**

Сергей Рогожников

**Технический директор**

Денис Филиппов

**Директор по рекламе**Владимир Савельев [advert@linuxformat.ru](mailto:advert@linuxformat.ru)**Генеральный директор**

Павел Фролов

**Учредители**

Частные лица

**Издатель**

ЗАО «Мезон.Ру»

**Отпечатано в типографии ООО «ЛД-ПРИНТ»**

196644, Санкт-Петербург, Колпинский р-н, пос. Саперный, территория предприятия «Балтика», д. б/н, лит. Ф  
Тел. (812) 462-8383, e-mail: [office@ldprint.ru](mailto:office@ldprint.ru)  
Заказ 10227

**РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ****Редактор** Нейл Мор [Neil Mohr] [neil.mohr@futurenet.com](mailto:neil.mohr@futurenet.com)**Заместитель редактора** Мэтью Хэнсон [Matthew Hanson][matthew.hanson@futurenet.com](mailto:matthew.hanson@futurenet.com)**Выпускающий редактор** Крис Торнетт [Chris Thornett][chris.thornett@futurenet.com](mailto:chris.thornett@futurenet.com)**Художественный редактор** Эфраин Эрнандес-Мендоза[\[Efrain Hernandez-Mendoza\] efrain.hernandez-mendoza@futurenet.com](mailto:efrain.hernandez-mendoza@futurenet.com)**ПОДГОТОВКА МАТЕРИАЛОВ**

Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Мэтт Хэнсон [Matt Hanson], Дэвид Хейвард [David Hayward], Ричард Хиллсли [Richard Hillesley], Джульетта Кемп [Juliet Kemp], Джон Лэйн [John Lane], Нейл Мор [Neil Mohr], Грэм Моррисон [Graham Morrison], Лес Паундер [Les Pounder], Маянк Шарма [Mayank Sharma], Ричард Смедли [Richard Smedley], Евгений Балдин, Алексей Федорчук, Александр Хрюкин, Игорь Штомпель

**Художественный ассистент** Карен Льюис [Karen Lewis]**Иллюстрации** Шейн Коллинж [Shane Collinge], Саймон Миддлвик [Simon Middleweek], Элли Волтон [Elly Walton]**КОНТАКТНАЯ ИНФОРМАЦИЯ****UK:** Linux Format, 30 Monmouth Street, Bath BA1 2BWТел. +44 01225 442244, email: [linuxformat@futurenet.com](mailto:linuxformat@futurenet.com)**РОССИЯ:****Санкт-Петербург (редакция):**

пр. Медиков, 5, корп. 7

Тел. +7 (812) 309-0686

**Представительство в Москве:**

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-4954

По вопросам сотрудничества, партнерства, оптовых закупок:

[partner@linuxcenter.ru](mailto:partner@linuxcenter.ru)

**Авторские права:** статьи, переведенные из английского издания Linux Format, являются собственностью или лицензируются Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

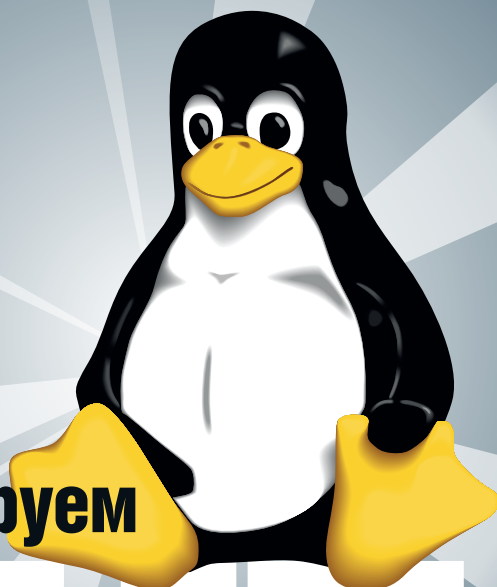
Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает исключительное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на диски — CD или DVD, поставляемые вместе с журналом, если не было оговорено иное.

**Ограничение ответственности:** используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux — зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

"GNU/Linux" заменяется на "Linux" в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>

## Заблокируем

# АНБ

## В апрельском номере

# Заблокируем АНБ

Вам омерзительна мысль, что за вами следят, как за преступником? Вам нечего прятать? Какая разница! У вас есть право на личную жизнь, и мы покажем, как его защитить.

## Обезопасим устройство Android

Заприте маленького зеленого человечка: пусть ваше устройство на Android тоже защитится от шпионов. Предлагаем полное руководство.

## Лучшие пускachi приложений

Как попасть с рабочего стола прямоком в любимое приложение с помощью нашей подборки лучших инструментов для этой цели.

## Фабрика Raspberry Pi

Недалече от Кардиффа есть местечко, где воплощают мечты. Но колдовство тут ни при чем.

Содержание будущих выпусков может меняться — вдруг к нам придерется АНБ...

# Новое поколение средств защиты

## Межсетевые экраны ССПТ, не имеющие IP-адреса

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

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

Защита для высокоскоростных корпоративных сетей Ethernet 100/1000 Мбит/с

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



## Назначение устройства

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

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

Дополнительное устройство защиты для:

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

Москва  
+7 (499)

271-49-54

Санкт-Петербург  
+7 (812)

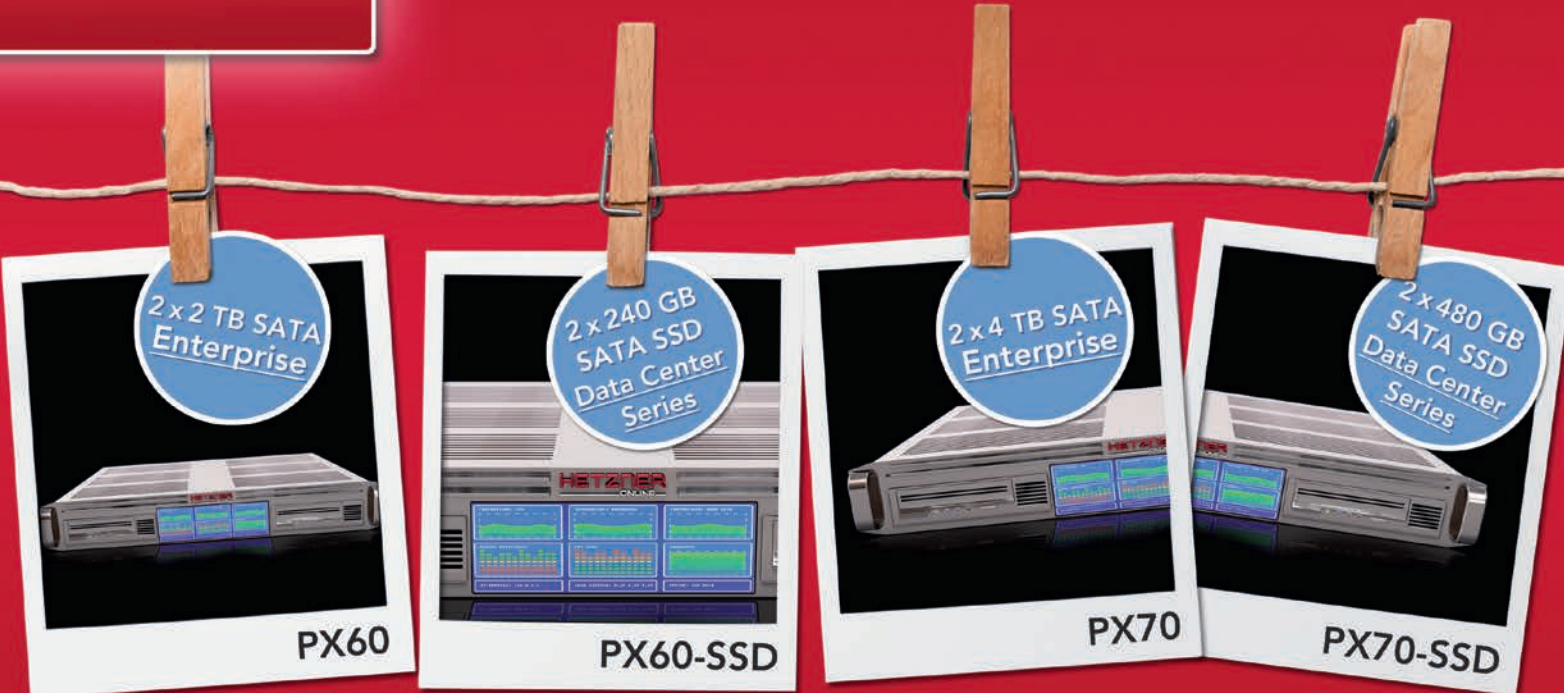
309-06-86

Linux-эксперт для вашего бизнеса. [www.linuxcenter.ru](http://www.linuxcenter.ru)

Linux  center

# СВЕЖИЕ РАЗРАБОТКИ!

## Новинка!



### ВЫДЕЛЕННЫЙ СЕРВЕР PX60

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 2 TB 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 TB Трафик\*

## 2600

рублей в месяц + Установка 3700 рублей

### ВЫДЕЛЕННЫЙ СЕРВЕР PX60-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 240 GB 6 Гбит/с SATA SSD Data Center Series
- 20 TB Трафик\*

## 3000

рублей в месяц + Установка 3700 рублей

### ВЫДЕЛЕННЫЙ СЕРВЕР PX70

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 4 TB 6 Гбит/с SATA 7200 об/м Software-RAID 1 Enterprise класс
- 20 TB Трафик\*

## 3000

рублей в месяц + Установка 3700 рублей

### ВЫДЕЛЕННЫЙ СЕРВЕР PX70-SSD

- Intel®Xeon® E3-1270 v3
- 32 ГБ ECC RAM
- 2 x 480 GB 6 Гбит/с SATA SSD Data Center Series
- 30 TB Трафик\*

## 3700

рублей в месяц + Установка 3700 рублей

#### ВКЛЮЧЕНО:

- Операционная система Linux
- Подключение 1 Гбит порт
- Гарантировано 200 Мбит/с
- IPv6 подсеть (/64)
- Без минимального контракта



Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выбирайте более чистое будущее вместе с Hetzner Online!

[RU.HETZNER.COM](http://RU.HETZNER.COM)



\* Нет платы за превышение. При превышении 20 TB/месяц (PX60, PX60-SSD, PX70), 30 TB/месяц (PX70-SSD) скорость соединения ограничивается (подсчёт ведётся по исходящему трафику, входящий и внутренний трафик не учитывается). Опционально можно снять ограничение, подтвердив оплату 80 рублей за каждый дополнительный TB.