

LINUX FORMAT

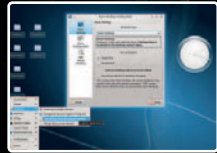
Главное в мире Linux

Декабрь 2012 № 12 (164)

Linux в CERN

Бравый пингвин участвует в розыске бозона Хиггса

Android Раскурочим фотокамеру
Arduino Как пройти в библиотеку
Erlang Упражнения в многозадачности



Razor-qt

» Обезжиренная альтернатива
тяжеловесному KDE **с. 46**



Общественная жизнь

« Правительство вкладывается в ядро Linux
года этак с 2000 »

Гуннар Хеллесон: поборник Свободного ПО **с. 42**

Мониторинг Syslog-ng

» Следите за событиями:
что творится в вашем ПК

Безопасность PacketFence

» Гоните шпионов/пришельцев от домашней сети

Многостаночность Поделитесь компом

» Личное кибер-кафе – кейлогеров просим не беспокоиться

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

Linux  center
www.linuxcenter.ru

ПЛЮС!
Основы
дарвинизма
с. 58

Также в номере...

Вывести мораль

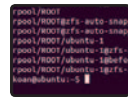
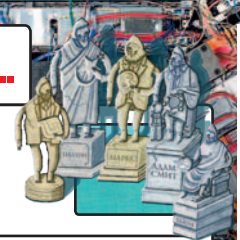
Оправдайте свою манию величия
от того, что вы линуксоид **с. 50**

Построим файлы

Попробуйте ZFS, последнее слово
в файловых системах **с. 54**

Дистрибутив для детей

Кто-нибудь, ну пожалуйста, вспомните о детях! **с. 28**



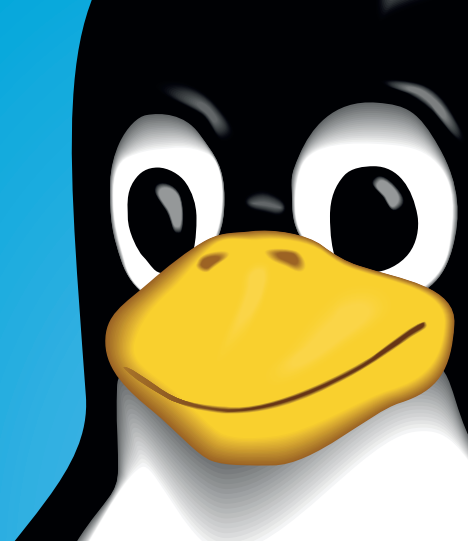
Испробуйте свеженький
Plus: SolusOS, Zorin и более tar!




Raspberry Pi:
» Превратите Pi в ZX81
и сыграйте в Hunt The Targoid **с. 70**



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 дружественных вариантов ОС GNU/Linux и прикладных решений на базе свободного ПО;
- абонентскую поддержку вашей сети;
- обучение сотрудников вашей компании.

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

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

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

Red Hat Enterprise Linux

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

Самая популярная в мире Linux платформа для бизнеса

Обеспечивает высокую производительность, надежность, масштабируемость и безопасность

Сертифицирована ведущими производителями оборудования и разработчиками ПО



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

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

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



ГНУ/Линуксцентр — Linux-эксперт для вашего бизнеса

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

Red Hat — ведущий серверный дистрибутив Linux

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

Специальное предложение!

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



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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



Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

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

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



Кто мы

В этом месяце мы посетили CERN. Что бы вы сделали с Большим адронным коллайдером, когда наконец-то обнаружится бозон Хиггса?



«Место силы»

» Во многих традиционных верованиях есть понятие «место силы»: это место, где духи предков и самой природы наиболее активно помогают местному шаману. В ИТ про духов сказать трудно, но места, где создались наиболее благоприятные условия для развития технологии, несомненно существуют. Если говорить о Европе, то такое место находится в Швейцарии вблизи Женевы и известно даже людям, далеким от науки. Это ЦЕРН (CERN, Conseil Européen pour la Recherche Nucléaire, Европейский совет по ядерным исследованиям).

С момента учреждения ЦЕРН в 1954 году участники его интернациональной команды совершили немало блестящих открытий в физике и существенно продвинули вперед информационные технологии, хотя последнее никак не являлось их основной задачей. То, что ЦЕРН лидирует по разработкам в области высокопроизводительных вычислений и Grid-технологий, не удивительно. То, что он стал одним из пионеров Интернет в Старом Свете – тоже понятно. А вот изобретение Тимом Бернерсом-Ли [Sir Timothy John “Tim” Berners-Lee] Всемирной паутины стало классическим примером «случайной революции»: рядовой проект, направленный на облегчение публикации результатов исследований внутри самого ЦЕРН, перевернул весь мир, открыв Эпоху Интернет. 30 апреля 1993 года было объявлено, что спецификациями URI, HTTP и HTML можно пользоваться совершенно свободно. Как и принято поступать с результатами фундаментальных исследований.

Кирилл Степанов
Главный редактор
» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Вопросы распространения: sales@linuxformat.ru
Web-сайт: www.linuxformat.ru

» Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
» Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112

 <p>Гэри Уокер Сунул бы в него футболиста Терри, политика Осборна и дуэт Jedward, перевел ручку на 11, отошел и злорадно захохотал.</p>	 <p>Эндрю Грегори Я бы употребил мощь бозона Хиггса на поиски чайника Бертрана Рассела.</p>	 <p>Эфраим Эрнандес-Мендоса Установил бы его как декоративную деталь ручейка в моем саду.</p>
 <p>Бен Эверард Подключил бы его к моему Arduino и завоевал весь мир.</p>	 <p>Маянк Шарма Телепортировался бы с его помощью в мозг Майка Сондерса и установил бы над ним контроль.</p>	 <p>Джонатан Робертс Переделал бы его в обширную пещеру и хранил бы там свои сисадминские миллионы и бочонки батского пива.</p>
 <p>Майк Сондерс О, Маянк Шарма – это голова...</p>	 <p>Валентин Синицын Раз уж механизм генерации массы теперь известен, пусть поищут эффективный способ похудения.</p>	 <p>Ник Вейч Демонтировал бы его, а на освободившейся площади организовал концерты Кристины Агилеры. Прибыльно!</p>
 <p>Сюзан Линтон Поискала бы, нет ли у него передачи заднего хода.</p>	 <p>Шашанк Шарма Превратил бы его в масштабную модель кольцевой дороги в Дерби – это моя любимая городская дорожная система.</p>	 <p>Нейл Ботвик Ой-йиййй-йиййийй...</p>

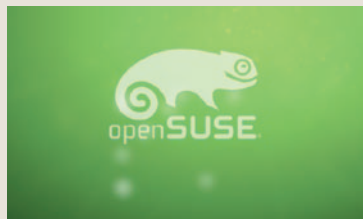
Содержание

Спим и видим, как построить свой БАК из мусорных коробок...

Обзоры

OpenSUSE 12.2 14

Полюбопытствуйте, что приготовил для вас последний из дистрибутивов на базе KDE.



› Скорее на стр. 14 — ради буйного пиршества зеленого. Истинно немецкого.

Qubes 16

Мы разобрались, готов ли этот супер-безопасный дистрибутив для широкого потребления.

Amarok 2.6 18

Некогда уважаемый музыкальный проигрыватель KDE с очередным релизом вернулся в новом обличье. Покорит ли он высоту, однажды взятую превосходной версией 1.4?



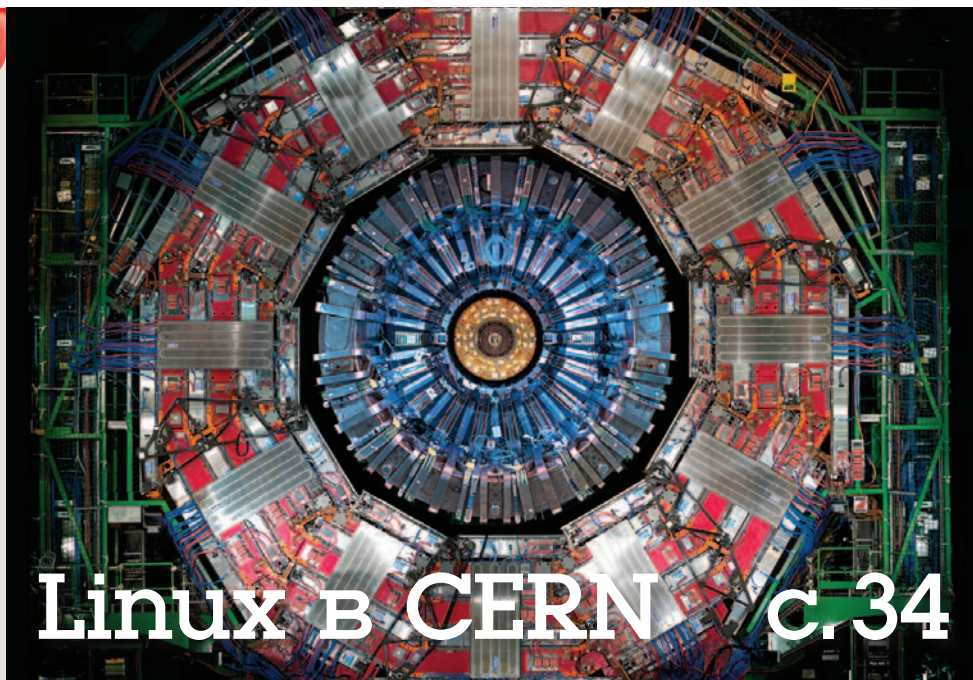
› Amarok 2.6: достойное пристанище музыкальных альбомов и подкастов.

Highscreen Yummy Duo 19

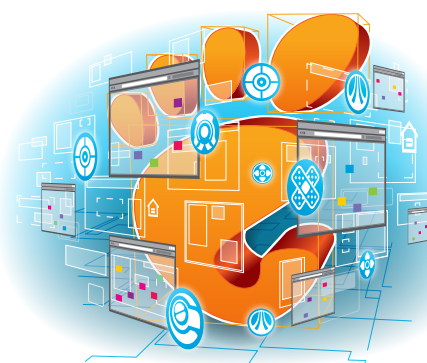
Российский смартфон с двумя SIM-картами — и отверткой в нагрузку. Для тех, кто постоянно в движении!



› Габариты смартфона помогают ему комфортно укладываться в руке.



Сравнение: Дистрибутивы для детей с. 28



Что за штука — Gnome OS? с. 62

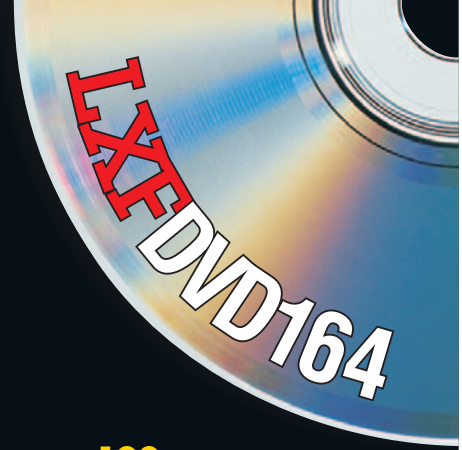
Люди говорят



«Моя задача — держать правительство в курсе происходящего в мире Open Source»

Гуннар Хеллесон из Red Hat — о своей отличной работе с. 42

На вашем бесплатном DVD



OpenSUSE 12.2

» Возвращение хамелеона

Zorin Core 6

» Заманим виндузеров на Linux

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

Ищите в этом номере



Razor-qt 46

В полку легковесных рабочих столов прибыло.

Философия открытого кода 50

Linux, коммунизм, капитализм. Обсудим подробнее.

А также дарвинизм 58

Посмотрим, что затаилось внутри яблока.



Пропустили номер?

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



Учебники

Хакерство Raspberry Pi 70

Запритесь изнутри, задерните занавески, отключите телефон и предайтесь радостям эмуляции золотого века игр 1980-х.

Безопасность Packet Fence 74

Защитите свою сеть от атак при помощи этой системы контроля за сетевым доступом. Все байты будут под контролем.

Администрирование Системные журналы 78

Пробираемся в таинственные недра вашей Linux-системы в надежде раскрыть ее мрачные секреты.

Электроника Arduino 80

Создайте библиотеки для автоматизации задач, и ваш следующий проект запрограммируется намного быстрее.

Языки программирования Erlang 84

Многозадачность требует практики – одной голый теорией сыт не будешь. Итак, попрактикуемся.

Пользовательский контроль Многоместные системы 88

Как потихоньку играть в Minecraft, пока ваши детки мирно смотрят онлайн-видео, и все на одном компьютере.

Постоянные рубрики

Новости 4

Состоялась SECR-2012, началось бета-тестирование Steam для Linux, Ubuntu сменил схему разработки, а планшетики вышли на свободу.

Новости Android 20

Устройства Nexus размножаются, а KDE перебирается под Android.

Android 22

Джюльетта Кемп поможет вашим приложениям увидеть мир через глазок камеры... встроенной в мобильный телефон.

Сравнение 28

Обеспечьте своим отпрыскам безопасное погружение в увлекательный мир Linux.

Что за штука 62

Gnome OS: один из самых спорных рабочих столов Linux становится операционной системой. Вот это да!

Рубрика сисадмина 64

Тандем Mondo и Mindi выручит вас при катаклизме на жестком диске. А также: продолжаем путь к звездам в системном администрировании.

Ответы 92

ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!

Нейл Ботвик отвечает на вопросы про спасательные DVD, установку VOIP-телефона и не только.

Hotpicks 96

Отведайте горяченького: лучшие в мире новинки свободного ПО.

Диск Linux Format 102

Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 104

У вас есть шансы воскресить опечатки былых времен.

Школа LXF 108

Одноклассники всех стран, соединяйтесь! На сайте, построенном вами лично, для начала – про известного писателя.

Через месяц 112

Хитроумные штуки с Raspberry Pi – плюс патенты, Bitcoin и прочее.



ГЛАВНОЕ Разработчики собрались » Долгожданный клиент » Ubuntu каждый день
» Плазма и планшеты

КОНФЕРЕНЦИЯ РАЗРАБОТЧИКОВ

Software Engineering Conference in Russia



» Рубрику готовил
АРТЕМ ЗОРИН

Независимая международная конференция «Разработка ПО» прошла в Москве.

С 31 октября по 2 ноября в Москве прошла VIII Независимая международная конференция «Разработка ПО» – Central & Eastern European Software Engineering Conference in Russia (CEE-SECR 2012).

Как и раньше, местом проведения стал Центр новых технологий «Digital October». Участников было более 700 из 15 стран мира – программисты, инженеры по качеству, архитекторы, аналитики, научные сотрудники, руководители разных уровней. Оживленные беседы шли не только в залах, где порой не хватало мест для всех желающих, но и в коридорах Центра.

Два зала были оборудованы камерами, вещавшими происходящее в Сеть, так что аудитория конференции увеличилась еще на 5000 человек, посетивших сайт мероприятия. Было проведено более ста докладов, панельных дискуссий и мастер-классов. Доклады шли одновременно в четырех залах и на площадке OpenSpace. Это позволило слушателям выбирать между пятью выступлениями.

Организаторы утверждают, что конференция этого года можно по праву назвать самой насыщенной за всю историю CEE-SECR.

Судя по отзывам участников, наибольший интерес вызвали дискуссии и доклады на следующие темы: «Сохранится ли профессия программиста?», «Типичные ошибки стартапов», «Обучение коду в мире онлайн», «Как финансировать компанию, не продав душу», «Моделирование и облачные вычисления».

Подавляющее большинство аудитории составили руководители среднего звена в возрасте 25–35 лет, а также программисты. Скудно не было никому.

Отмечено, что сегодня в России складываются благоприятные условия для технологических стартапов: есть положительные сдвиги в государстве, которое как минимум не мешает начинающим предпринимателям. Но последним надо во многом менять свой менталитет: активнее развивать бизнес, не закликиваться на проблемах и придерживаться взгляда «стакан наполовину полон, а не наполовину пуст». Одной из типичных ошибок начинающего российского бизнеса в области разработки ПО названа переоценка роли инженеров в ущерб бизнес-функциям: организации маркетинга, продаж и т. д. И очевиден недостаток наукоемких проектов – здесь необходимо активное участие государства в стимулировании и финансировании научных разработок.

Церемония закрытия конференции ознаменовалась вручением Премии Бертрама Мейера за лучший исследовательский доклад (Бертран Мейер – создатель языка Эйфель, член Программного комитета SECR и заведующий кафедрами в ETH и НИУ ИТМО). Ее получили В. Трифанов и Д. Цителов из компании DevExperts с докладом «Динамическое обнаружение гонок в многопоточных Java-программах».



» **Наибольший интерес участников конференции вызвала дискуссия, посвященная стартапам.**

Конференция состоялась благодаря поддержке Российской венчурной компании, Epam Systems, Intel, T-Systems, Microsoft, Luxoft, Deutsche Bank, First Line Software, EMC, Oracle, Jet Brains, Parallels, Microfocus, Infostroy. Некоммерческое партнерство РУССОФТ второй год подряд проводит ежегодное собрание в рамках конференции; активно поддержали конференцию и АП КИТ, а также Agile Russia, РАСПО, РАЭК, ISDEF, OMG, Cloud Standards Customer Council и многие другие. Информационным партнером Конференции, среди прочих, выступил и журнал *Linux Format*.

Презентации, видеозаписи и фотографии доступны на сайте www.secr.ru.

О конференции «Разработка ПО/CEE-SECR»

Разработка ПО/CEE-SECR – ежегодная независимая конференция, в которой участвуют разработчики ПО, исследователи, инженеры-практики, лидеры общественного мнения, предприниматели. За годы работы конференция завоевала заслуженный авторитет и традиционно поддерживается мировыми лидерами индустрии высоких технологий.

В Программный комитет SECR входят эксперты мирового уровня, представители ИТ-индустрии и научно-исследовательских организаций. Среди них – профессор СПбГУ и гендиректор Ланит-Терком А. Терехов, академик РАН В. Иванников, гендиректор Российской венчурной компании И. Агамирзян, соучредитель и главный архитектор CUSTIS М. Цепков.

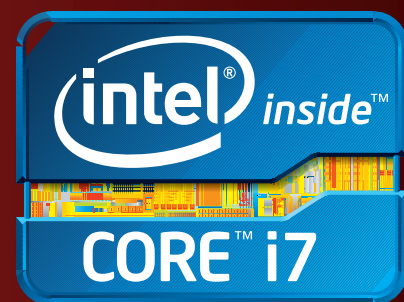
lenovo® FOR
THOSE
WHO DO.™*



КОГДА КАЖДАЯ СЕКУНДА
НА СЧЕТУ!

THINKPAD X230

- Новый процессор Intel® Core™ i7
- Технологии ThinkVantage® Active Protection System™



Intel, логотип Intel, Intel Inside, Intel Core, Core Inside являются товарными знаками корпорации Intel на территории США и других стран. *Для тех, кто действует. Реклама.

LINUX СТАНЕТ ИГРОВОЙ ПЛАТФОРМОЙ

Началось бета-тестирование Steam для Linux

Важное событие в экосистеме Linux в 2012 году состоялось благодаря совместным усилиям Valve и Canonical.

7 ноября этого года произошло событие, знаменательное для дальнейшего развития Linux как настольной системы: компания Valve начала бета-тестирование Linux-версии клиента сервиса доставки игр Steam. Этого события с нетерпением ждали почти все линуксоиды, даже, пожалуй, сильнее, чем выхода очередной версии Ubuntu, и оно состоялось. Разработка Steam для Linux велась более двух лет, а разговоры о том, что клиент вот-вот выйдет, велись и того дольше.

На момент выхода данного номера нашего журнала для оценки работы сервиса предоставлена возможность бесплатной установки игры *Team Fortress 2* для Linux. Кроме того, в сервис Steam добавлены еще две игры, работающие в Linux. Всего в настоящее время 26 игр помечены в каталоге Steam, как поддерживающие Linux. Стоимость Linux-версий игр варьируется от 49 до 399 рублей. Steam для Linux доступен

для тестирования в сборках для Ubuntu 12.04 и 12.10. Linux-версия клиента примечательна поддержкой нового режима Big Picture, созданного для использования на телевизорах.

Бета-версия Steam для Linux поставляется только среди ограниченного круга утвержденных тестеров, однако в Интернете уже появились инструкции по неофициальной установке, позволяющие запустить Steam на компьютерах пользователей, не вовлеченных в программу бета-тестирования Valve.

Судя по первым отзывам русских пользователей, Steam для Linux пока сыроват и не отвечает всем требованиям, но на стадии бета это простительно. В целом, мировое сообщество Linux-пользователей положительно оценило даже бета-версию.

А вот на проходившем в конце октября в Дании Саммите разработчиков Ubuntu (Ubuntu Developer Summit) Дрю Блисс

[Drew Bliss] из компании Valve Software, известной своими воистину эпическими играми – например, *Half Life*, заявил, что Linux более жизнеспособна как игровая платформа, нежели недавно вышедшая на рынок Windows 8. Как и предполагалось, причиной, по которой Valve начала портировать Steam на Linux, стала практически прямая конкуренция с онлайн-мага-

«Этого события с нетерпением ждали почти все линуксоиды.»

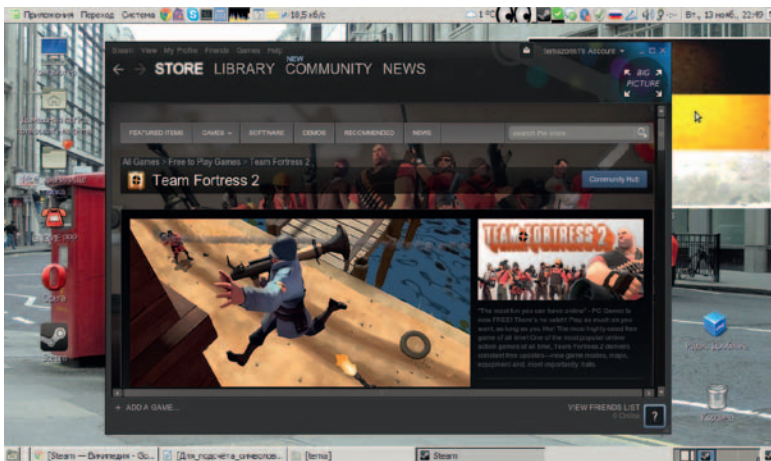
зином Microsoft. Именно поэтому Valve выходит на рынок, еще не занятый ни одной популярной платформой распространения цифрового контента.

По поводу Steam для Linux Блисс, в частности, сказал, что Steam отлично работает на Ubuntu, а многие игровые студии заинтересовались в портировании игр на Linux. Совместная работа с Canonical была очень успешной; Ubuntu был выбран потому, что это самый популярный среди пользователей Linux дистрибутив, имеющий за спиной поддержку коммерческой компании; в Linux есть все технологии, необходимые для игр – *OpenGL*, *PulseAudio*, *OpenAL* и поддержка ввода; игры на основе нового еще не выпущенного движка Source будут выпущены в первую очередь для Linux; Valve не будет заниматься вопросами защиты игр от копирования – этот вопрос оставлен на усмотрение разработчиков.

Кроме этого, на конференции обсуждался вопрос создания отдельной команды разработчиков, которые будут работать над развитием Ubuntu как игровой платформы. На сервисе Launchpad уже создана страница этой команды. В качестве создателя группы выступил Даниэль ван Вугт [Daniel van Vugt], работающий в Canonical над *Compiz* и *Unity*. Кроме того, в группу вошли мейнтейнеры *Compiz*, три инженера из компании *Unity 3D*, развивающей одноименный игровой движок, и около десяти заинтересованных разработчиков.

Заметим, что локомотивом превращения Linux в игровую платформу стала команда разработчиков Ubuntu.

➤ Главное окно Steam для Linux манит своей аутентичностью — никакого Wine и CrossOver больше не потребуется.



Немного истории

Steam – служба цифровой дистрибуции (говоря по-русски – распространения) игр и цифрового контента на ПК пользователя через Интернет, – появилась еще в далеком 1999 году. Изначально это был WON (World Opponent Network) – более узкоспециализированный сервис, не являвшийся системой распространения данных по Интернету и принадлежавший небезызвестной игровой компании Sierra, купленной впоследствии Valve. Свое нынешнее название сервис получил в 2003

году, после коренного пересмотра Valve политики распространения цифрового контента.

В конце нулевых Valve не раз заявляла, что выпустит клиент Steam для Linux, однако дальше заявлений, которые живо обсуждались в сообществе, дело не шло; но выяснилось, что это были не пустые слова. Сейчас у сервиса сотни тысяч пользователей во всем мире, в основном под Windows, но ситуация изменится, верят в Valve, благодаря поддержке Linux и Mac OS X.

softline®



Services Software Cloud

ИТ-архитектура вашего бизнеса



ДИСТРИБУТИВ ЗА СУТКИ

Шаттлворт меняет процесс разработки

Ubuntu переживает очередные революционные преобразования. На этот раз меняется способ выхода «сюрпризов» и бета-версий.

Ubuntu развивается быстрее любого дистрибутива Linux, во многом благодаря смелым и неоднозначным решениям его создателя – Марка Шаттлворта [Mark Shuttleworth].

На прошедшем в Копенгагене саммите разработчиков Ubuntu (Ubuntu Developer Summit) был одобрен план модернизации процесса разработки Ubuntu 13.04, согласно которому команда Canonical отказывается от формирования отдельных альфа-версий дистрибутива, а перед релизом будет только один бета-выпуск. Вместо альфа-версий пользователям будет доступна серия постоянно формируемых «снапшотов». Финальная «полировка» релиза перед выпуском будет проводиться путем более широкого распространения бета-версии.

По мнению разработчиков, такой подход позволит обеспечить непрерывный процесс разработки, сгладив стадии тестирования и оценки качества. Разработка будет разбита на двухнедельные циклы, в рамках которых будет производиться непрерывная интеграция готовых к использованию новшеств. Помимо этого, планируется значительно расширить использование автоматизированных тестов для постоянной оценки качества дистрибутива.

Формирование тестовых ISO-образов будет производиться ежедневно, но подобные снапшоты будут готовиться и проверяться более внимательно, чтобы не допустить их поставки с неработающим инсталлятором. При этом по мере развития формируемого выпуска дистрибутива командам тестировщиков и добровольцам будет предлагаться обратить внимание на проверку работы тех или иных новшеств, интегрированных в рамках очередного цикла разработки.

Все это справедливо только для основной ветки дистрибутива Ubuntu. Все дополнительные сборки, такие как Kubuntu, Xubuntu и Lubuntu, будут развиваться в рамках собственных циклов подготовки релизов, самостоятельно выбирая время заморозки, методы тестирования и расписание выпуска промежуточных версий.

И уже 7 ноября сего года в рамках принятого решения начался выпуск ежеднев-

ных тестовых сборок Ubuntu 13.04. В настоящее время развитие Ubuntu 13.04 находится на самом начальном этапе, поэтому в представленных сборках отсутствуют заметные изменения, кроме обновлений версий некоторых пакетов.

Нелишним будет отметить, что развитие Ubuntu как дистрибутива № 1 в мире происходит скачкообразно, а не эволюционно, и в этом большая заслуга Марка Шаттлворта. Революционные изменения происходят с завидной частотой, что может вызывать как одобрение, так и осуждение.

В конце октября Марк объявил в своем блоге об изменении процесса разработки новых функций для Ubuntu. Если раньше все новшества анонсировались еще на этапе зарождения идеи, что частенько приводило к излишней критике со стороны пользователей, то отныне Canonical будет представлять новшества, потенциально способные спровоцировать шквал

«Развитие Ubuntu происходит скачкообразно, а не эволюционно.»

критики, уже после полной готовности к использованию. Причем новшества будут готовы к тестированию задолго до релиза, что даст возможность учесть выявленные недочеты.

По словам Шаттлворта, уход от предварительных публичных обсуждений вызван желанием избежать искажения и размытия изначальных идей, до того как они будут реализованы в виде работающего

В заботе о приватности

В конце октября некоммерческая правозащитная организация Electronic Frontier Foundation (EFF) опубликовала заявление, в котором призвала компанию Canonical отключить по умолчанию функцию показа контекстной рекламы при вводе поисковых запросов в панели Dash и устранить серьезные нарушения конфиденциальности пользователя, всплывшие в результате анализа применяемых методов взаимодействия со внешними сервисами. EFF считает недопустимым по умолчанию, не предупредив пользователя, осуществлять действия, способствующие утечке частных данных о его предпочтениях. Вместо навязывания сервиса предлагалось после первого входа в систему выводить специальное окно, где и уточнить, согласен ли пользователь на передачу результатов запросов внешним службам при поиске в Dash. Canonical реализовала предложение немедленно.

прототипа. К категориям улучшений, которые не стоит афишировать раньше времени, отнесены такие вещи, как web-сервисы, дизайн, приватность и аналитика. Все скрытно созданные наработки, как и обычные новшества, будут распространяться в исходных текстах под свободными лицензиями. Изменение справедливо только для проектов, развиваемых внутри самой Canonical и сотрудниками компании; при этом ограничения носят характер рекомендаций и не требуют подписания соглашения о неразглашении. Более того, в разработке внутренних проектов Canonical смогут принимать участие и представители сообщества, но такое участие будет организовано на индивидуальной основе, путем отправки личных приглашений. Общий процесс разработки дистрибутива, как и прежде, остается полностью открытым и прозрачным.

Примером скрытых улучшений является появившаяся в Ubuntu 12.10 линза с предложением о покупке цифрового контента, которая была представлена уже в готовом виде. В ответ на конструктивную критику указанная возможность была доработана – в частности, были учтены пожелания по обеспечению приватности и добавлена функция отключения обращения к внешним сервисам. Таким образом, в процессе подготовки версии Ubuntu 13.04 пользователей ожидает серия сюрпризов.

➤ **Главное событие в жизни разработчиков Ubuntu — это Саммит, на котором принимаются судьбоносные решения.**



НАЙДИ РАБОТУ ЛЕГКО!

на www.hh.ru



САЙТ РАЗРЕШЕН ДЛЯ ПОСЕЩЕНИЯ БЕЗ ОГРАНИЧЕНИЯ ВОЗРАСТА

hh **ru**
HeadHunter

Выбирай из более чем 200 000 вакансий

ЧИСТО ОТКРЫТОЕ ЖЕЛЕЗО

Дешевых планшетников прибыло

Настойчивые американцы не оставляют попыток сделать дешево и хорошо.

Небольшая американская компания из штата Флорида, выпускающая компьютеры и мобильные устройства на основе свободных прошивок и Linux, объявила в середине ноября о выпуске серии недорогих устройств для настоящих ценителей Linux. Устройства будут отличаться предустановкой сразу двух ОС – Android 4.1 и окружения на базе развиваемой сообществом KDE оболочки Plasma Active. Выпуск устройств планируется наладить уже в начале 2013 года. Сейчас в форме краудфандинга организован сбор предзаказов, который поможет обеспечить начальное финансирование производства. Отмечается, что оборудование уже находится на стадии готовых прототипов.

В планах компании – выпуск трех устройств:

- » семидюймового планшета PengPod713 (разрешение экрана 800 × 480 пикселей) по цене 120 долларов с 512 МБ ОЗУ, 4 ГБ Flash и аккумулятором на 3300 мАч;
- » десятидюймового планшета PengPod1000 (разрешение экрана 1024 × 600 пикселей) за 185 долларов с 1 ГБ ОЗУ, 8 ГБ Flash и аккумулятором 6000 мАч;
- » неттопа PengBox по цене 110 долларов с 1 ГБ ОЗУ и 4 ГБ Flash;
- » нетбука PengBook по цене 160 долларов с 10-дюймовым экраном, с 1 ГБ ОЗУ и 8 ГБ Flash.

Все устройства будут основаны на платформе Allwinner A10 (это система на кри-

сталле, разработанная компанией Allwinner Technology Co. Ltd., находящейся в городе Чжухай, в Китае) с процессором 1,2 ГГц Cortex A8 ARM, графическим процессором MALI400MP с поддержкой OpenGL ES 2.0, акселератором декодирования видео с разрешением до 2160p. Железки будут оснащены портом USB 2.0, HDMI-портом, слотом SATA-II, разъемом Ethernet, Wi-Fi и web-камерой. В качестве аксессуара будет поставляться кожаный чехол с USB-клавиатурой, превращающий планшеты в подобие нетбука.

Прошивка для устройств основана на подготовленном проекте Linaro системном образе, базирующемся на пакетной базе Ubuntu. Кроме оболочки Plasma Active, в состав включен медиа-центр XBMC и набор типовых приложений, таких как офисный пакет *Libre Office* и браузер *Chromium*. Тестовые образы используемой прошивки уже доступны для загрузки и оценки.

Такие устройства будут полезны и интересны, прежде всего, энтузиастам и настоящим «фанатам», утверждают сами создатели на сайте компании PengPod: <http://pengpod.com>. Отметим, что это не первая попытка создать планшетный компьютер на основе Plasma Active. Компания Make Play Live планировала выпустить планшетный ПК еще в мае этого года, однако из-за проблем с производителем комплектующих выпуск был отложен на неопределенный срок. **LXF**



» Сегодня трудно кого-то удивить дешевым и открытым планшетом, особенно после фактического провала Vivaldi.

Новости короткой строкой

» После финансового аудита Mozilla Foundation, инициированного еще в 2008 году, компания выплатит 1,5 млн долларов налоговому ведомству США. Источник: <https://blog.lizardwrangler.com>

» 9 ноября 2012 года исполнилось 8 лет браузеру *Mozilla Firefox*, ставшему иконой в мире СПО. Источник: <http://www-archive.mozilla.org>

» Намеченное на 5 ноября 2012 года судебное заседание в рамках разбирательства Apple против Motorola, в котором Motorola обвинялась в злоупотреблении FRAND-патентами на ключевые беспроводные технологии, отклонено судом на стадии анализа. Источник: <http://www.groklaw.net>

» Линус Торвалдс вернулся на KDE. До этого он пытался остановиться на окружении *Xfce*, когда его не устроили изменения в *Gnome 3*. Источник: <https://plus.google.com>

» Компания Nokia прекратила доработку и расширение функциональности ОС Symbian. Платформа переведена в режим сопровождения, производится только исправление ошибок. Источник: <http://www.developer.nokia.com>

» Управляющий совет Фонда Apache придал офисному пакету *OpenOffice* статус первичного проекта Apache, после полутора лет нахождения проекта в инкубаторе Apache. Источник: <https://globenewswire.com>

» Российский системный интегратор КРОК совместно со своей образовательной компанией КРОК ОК провели с 27 по 30 ноября в Москве олимпиаду для студентов и молодых специалистов, посвященную администрированию Linux. Источник: <http://www.crocok.ru/olympiad/>

» Raspberry Pi поменял конфигурацию – с 16 октября 2012 года будет поставляться модифицированная модель B, с 512 МБ ОЗУ вместо ранее предлагаемых 256 МБ. Цена устройства остается прежней, 35 долларов. Источник: <http://www.raspberrypi.org>

» На прошедшей в начале ноября в Барселоне конференции LinuxCon Europe компания AMD подтвердила информацию о закрытии подразделения Operating System Research Center (OSRC) в Дрездене, в рамках программы по сокращению 15% персонала с целью оптимизации расходов компании. Источник: <http://www.h-online.com>

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Юстируем шрифты

Какому-то памяты времена, когда на шрифты в Иксах трудно было смотреть без слез (в прямом смысле). Потом к Иксам научились прикручивать шрифты из Windows, оказавшиеся, по недосмотру Microsoft'a, свободно доступными при соблюдении неких условий. Да и начали появляться качественные шрифты, уже по-настоящему свободные. И тут выяснилось, что качество само по себе не гарантирует «смотрибельности» на экране: не менее важны условия рендеринга.

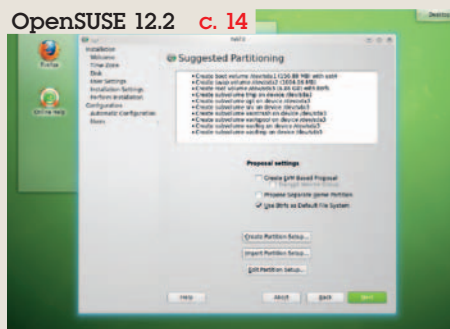
Пошли эксперименты с опциями сборки библиотек поддержки вывода шрифтов – сначала уменьшечки, потом дело поставили на промышленную основу в Ubuntu. В большинстве дистрибутивов Linux из «первой десятки» шрифты приобрели пристойные очертания «из коробки». Казалось бы, чего еще желать? Оказалось, есть чего.

Одним из первых проектов по «улучшению» шрифтов был *infinality.net*; в его рамках разрабатывались патчи для поддержки субпиксельного рендеринга. Их применение было делом вкуса, да и нужда в них помаленьку отпадала, хотя они есть в репозиториях большинства дистрибутивов. Но один из результатов проекта остается интересен тем, кто к шрифтам «нервно дышит».

Это – пакет *fontconfig-infinality*. Сам он ничего не «улучшает». Но – позволяет выбрать стиль рендеринга, скажем, командой *infinality-ctl setstyle*. Стили – *otladdonly*, *linux*, *infinality*, *osx*, *osx2*, *win7*, *win98*, *winxp* – различаются параметрами хинтинга. Какой краше – не скажу: смотрите сами...
alv@posix.ru

Сегодня мы рассматриваем:

- OpenSUSE 12.2** 14
Самый истинно немецкий из всех дистрибутивов, OpenSUSE по-прежнему предоставляет лучшие шансы положить лапы на безупречно интегрированный рабочий стол KDE. А также выдает желающим стать сисадминами едва ли не самые продвинутые инструменты настройки системы за пределами CERN/NASA.
- Qubes** 16
Этот экспериментальный дистрибутив Linux избрал нетривиальный подход к обеспечению безопасности системы: все пользовательские функции разнесены по разным виртуальным машинам, чтобы приложения работали изолированно друг от друга. И это настолько хитроумно придумано, что нам осталось только в восхищении покачивать головами и улыбаться.
- Amarok 2.6** 18
Теперь еще и со встроенным транскодированием, *Amarok* является медиа-проигрывателем для настоящих мужчин с широким образом мыслей. Вы можете раскидать элементы интерфейса так, чтобы они вписались в ваш экран, а он все равно будет выуживать слова песен, картинки и прочие метаданные из Сети.
- Highscreen Yummy Duo** 19
Смартфон от российской компании Highscreen радует приемлемой ценой и двумя SIM-картами – явное преимущество для любителей путешествовать, поддерживая связь с домом без разорительной оплаты роуминга. Правда, вставлять эти SIM-карты непросто – придется прихватить с собой еще и отвертку. Зато работают они безотказно и постоянно находятся в активном режиме ожидания.



» Хамелеоны вообще-то умеют менять цвет, почему же талисман OpenSUSE вечнозеленый?



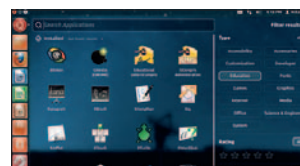
» Неустанный напев *In Dulci Jubilo* досрочно наводит нас на мысли о Рождестве.

Сравнение: Дистрибутивы для детей с. 28

DoudouLinux



Edubuntu



LinuxKidX



Qimo



Sugar on a Stick



В Linux можно научиться многому, и было бы эгоистично держать это исключительно при себе. Мы подобрали вам пять лучших дистрибутивов для развития незрелых детских умов – пусть малолетки растут над собой.

OpenSUSE 12.2

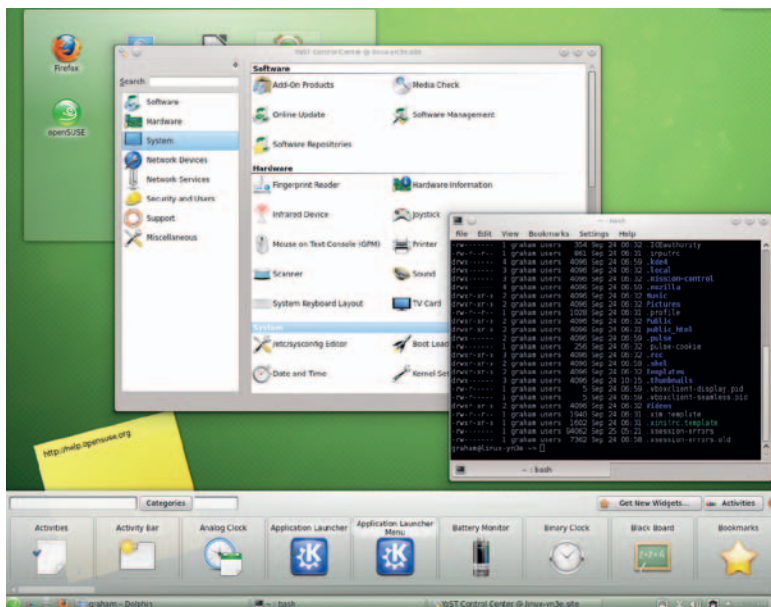
Этот релиз заставил себя подождать, но, несмотря на запоздание, он все же сумел местами вызвать улыбку на лице **Грэма Моррисона**.

Вкратце

» Один из главных дистрибутивов Linux, с поддержкой на два цикла релизов и два месяца. См. также: **Kubuntu** и **Fedora**.

Не будем припоминать задержки, омрачившие появление этого релиза. Такое случается и с лучшими дистрибутивами. Но первое, что бросается в глаза после установки стандартного рабочего стола KDE – что это версия 4.8, а не последняя 4.9. Разница вроде и невелика, но многие, скорее всего, предпочтут вместо OpenSUSE дистрибутив с KDE 4.9. А жаль, ведь OpenSUSE – один из последних великих дистрибутивов KDE.

Хорошо бы, конечно, кто-нибудь решил убрать типовую синюю подсветку KDE, которая абсолютно не сочетается с зеленым фоном OpenSUSE, и избавился от засилья Комнат, портящих вид любого KDE. Мало кому понадобится фотоконната, при наличии тех же *Gwenview* и *Digikam*. Мы бы также посоветовали команде OpenSUSE быть посмелее насчет конфигурации KDE, установив, например, некоторые наиболее популярные виджеты и добавив пару альтернативных тем и готовых обоев. Ведь этим и примечателен



» OpenSUSE, как прежде, великолепен в облике KDE, который можно легко обновить до версии 4.9, набрав всего пару команд.

«Хорошо бы кто-нибудь убрал типовую синюю подсветку KDE.»

KDE, а новички навряд ли знают об их существовании. И все же, благодаря тому впечатлению и ощущению от использования, что оставляет заказная установка OpenSUSE и инструменты настройки, это

по-прежнему одна из наиболее органичных версий KDE, независимо от конфигурации по умолчанию. Понравилось, к примеру, что экраны приложений, таких как *GIMP* или *LibreOffice*, оформили в фирменном зеленом цвете SUSE, придав цельность рабочему столу. Этим OpenSUSE и силен. Установка с live-носителя, который вы найдете и на приложенном к нынешнему выпуску DVD, очень проста. Как и прежде, есть горестное предупреждение о необходимости иметь больше 1 Гб сво-

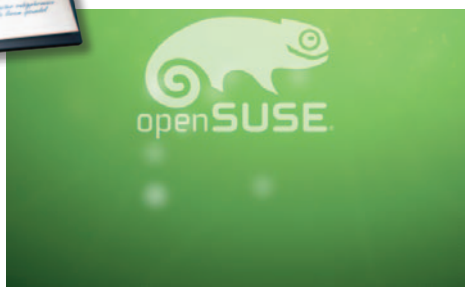
бодного места на диске, хотя готовой альтернативы не предлагается, и до сих пор приходится принимать нудное лицензионное соглашение. Зато мы оценили возможность создавать отдельный домашний раздел на диске одним щелчком, а также установку на базе LVM.

Мы установили версии и с KDE, и с Gnome, и протестировали их в настольной версии и на виртуальной машине. Последняя работала безупречно, благодаря новой системе 3D-рендеринга, позволяющей использовать композитный режим Gnome в виртуальной среде. Однако некоторые проблемы со стандартной установкой все же возникли.

Часть из них всплыла при первом обновлении системы: во-первых, связь с главным сервером не было, а красивого варианта отката не предусмотрено. Нет, например, опции «игнорировать все», на случай, если пакеты нельзя установить. Во-вторых, система пожаловалась, что *PackageKit* уже запущен. Немного терпения – и все эти проблемы разрешились, но новичков они застопорят.

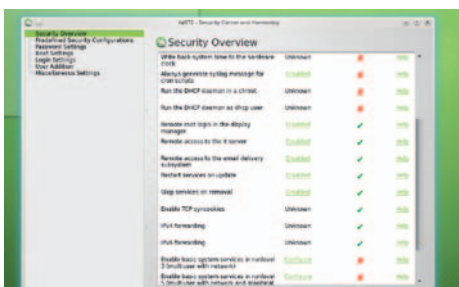
При первой загрузке появляется новое меню *Grub 2* и загрузчик *Plymouth*. Оба вместе придают графической части загрузки профессиональный лоск, помогая вам добраться до рабочего стола

Свойства новскидку



Новый облик установки

Анимация в виде движущихся по экрану огней во время загрузки напомнила нам огни из *Близких контактов третьей степени*.



Настройки безопасности

Уровень безопасности по умолчанию рядовым пользователям покажется чрезмерным, но для защиты окружения это благоразумно.

без всякого намека на командную строку или сбои. В релизе присутствует множество улучшений производительности, в основном с фоновыми обновлениями пакетов, но здесь OpenSUSE идет практически голова к голове с первой beta-версией Kubuntu 12.10.

По сравнению с нашим рабочим столом KDE в Arch, этот более медленный, но мы списываем это на отсутствие фоновых процессов в Arch. Там мы, к примеру, отказались от *Akonadi* и оставили только поисковый демон *Strigi*, запускающийся время от времени. На современной технике проблем бы не было ни с тем, ни с другим, но если вы хотите установить OpenSUSE на нетбук, об этом стоит задуматься. Если в меру подсократить часть наиболее капризных элементов KDE, он будет прекрасно работать, даже при включенных композитных эффектах.

Установка Gnome

Рабочий стол Gnome мы установили параллельно с KDE не только затем, чтобы полюбоваться на *GTK+* в зеленом цвете, но и чтобы проверить, приложила ли команда столько же усилий, чтобы модернизировать и другой рабочий стол. Однако, помимо цвета, в основном конфигурация осталась прежней. И это отнюдь не плохо. Gnome 3.4 – отличный релиз, и первый, по которому видно, что Gnome вновь оживает.

В этой версии сюрпризов нет, помимо добавления *YaST GTK* в окно Администрирование системы и возможности отключения питания из пользовательского меню, без нажатия Alt. Может, OpenSUSE и не станет дистрибутивом номер один для гномофилов, но этот рабочий стол хорошо интегрирован и выучит вас, если в вашей компании используют OpenSUSE, а вы предпочитаете именно Gnome, а не KDE. Традиционно, на любителя и вездесущие инструменты настройки в OpenSUSE. Долгие годы они нам все же нравились, позволяя смешать самые раз-



» Рабочий стол Gnome предстал в стандартном варианте, за исключением зеленого фона.

Версии ПО

- » Ядро 3.4.6
- » Qt 4.8.2
- » GIMP 2.8.0
- » Firefox 14
- » Grub 2.0.0
- » Gnome 3.4.4
- » KDE 4.8.4
- » Xfce 4.10.0
- » LibreOffice 3.5.4
- » Thunderbird 14
- » GCC 4.7

нородные компоненты любого рабочего окружения, будь то управление сетью, параметры пользователей или установка обновлений, и собрать их в один набор инструментов. Но большинство рабочих столов теперь имеют собственные инструменты настройки, отдельные и интегрированные, превращающие *YaST* в своеобразный атавизм из времен Mandrake и Red Hat Linux.

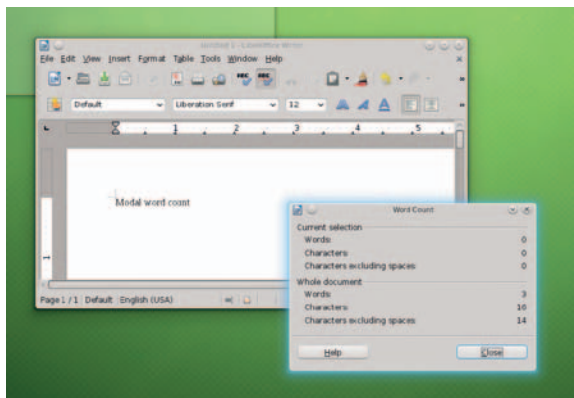
YaST из прошлого

Хуже всего то, что *YaST* ничуть не изменился. Это монолитная программа, разобраться в которой сложно, и нет никаких значительных новых функций. Было бы здорово, если бы в нем появились, например, управление виртуализацией или контейнерами, настройка удаленного рабочего стола, параллельно с мониторингом состояния ЦП и системы. По крайней мере, это сделало бы OpenSUSE хорошей альтернативой для установки на семейные машины или машины друзей. Вместо этого, интерфейс управления ПО выглядит запутанным, заставляя ломать голову над выбором правильного режима мета-пакетов при установке Gnome.

Модуль настройки сети избыточен, ведь это делает рабочий стол, а межсетевой экран проще настроить при помощи внешнего приложения. Для сисадминов выполнять все действия в одной консоли всегда было удобно. Но нам не кажется, что это и есть основная аудитория скачиваемого OpenSUSE, так что пора бы либо отправить

YaST в отставку, либо преобразовать его в некую web/XML-модель, чтобы от него была польза.

OpenSUSE сохранил свое величие, со здоровым уклоном в KDE. Стандартный набор приложений выглядит оправданным, особенно *Firefox* на рабочем столе KDE, как и прекрасная версия *LibreOffice*. Система стабильна, и вы также можете рассчитывать на множество обновлений и поддержку от сообщества OpenSUSE. Обидно, конечно, что этот релиз не богат на новшества. Но в мире, где многие другие дистрибутивы так отчаянно борются с последствиями своих нововведений, OpenSUSE остается оплотом надежности, и это хорошо. **LXF**



» LibreOffice все так же отстает на одну версию, так что все 1120+ слов в этом обзоре подсчитаны старой утилитой *Writer*.

LINUX **Вердикт**
FORMAT

OpenSUSE 12.2

Разработчик: OpenSUSE Project
 Сайт: www.opensuse.org
 Лицензия: GPL

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

» Мы презираем национальные стереотипы, но этот немецкий дистрибутив – основательный, стабильный и профессиональный релиз.

Рейтинг 7/10

Qubes 1

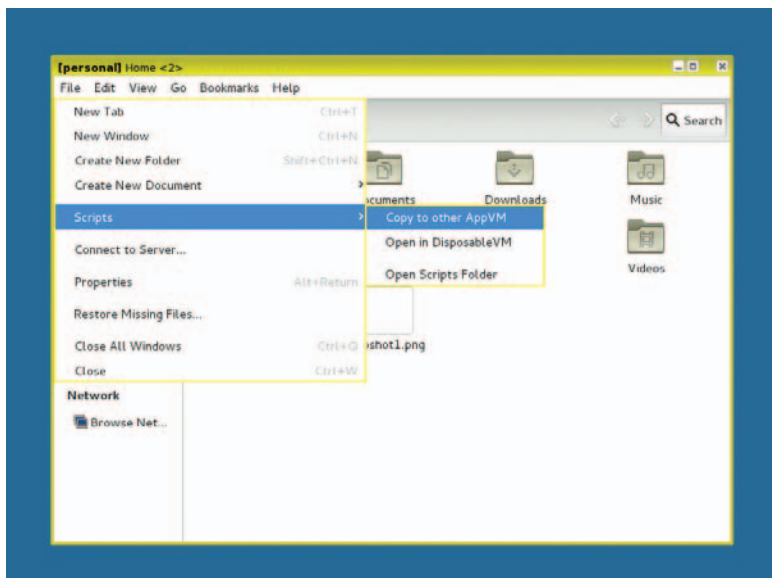
Новый дистрибутив обещает непробиваемую защиту, разделив вашу систему на независимые домены. **Бен Эверард** наводит справки.

Вкратце

» Новый дистрибутив для супер-безопасной работы.

Qubes – это новый дистрибутив с упором на безопасность рабочего окружения. Традиционно, для этого в Linux использовались скромные, хорошо испытанные приложения, устанавливаемые поверх системы и жестко контролируемые пользователем. Qubes реализует совершенно иной подход. Он разделяет вашу систему на несколько доменов (более известных как виртуальные машины, VM). Поскольку все они практически изолированы друг от друга, вторжение в одну (вируса, трояна или хакера) не повлияет на остальные. По сути, атакующий даже не узнает об их существовании.

Спусковым механизмом разработки послужила недавно обнаруженная уязвимость Java. Ситуация задела многих сисадминов: замечено удаленное выполнение кода в важнейшем приложении, а поставщик не торопится это исправлять. Для нынешних многопользовательских,



» Перекидывать файлы между доменами, на которые Qubes делит вашу систему, легко, но безопасность будет на вашей совести.

«Спусковым механизмом разработки послужила уязвимость Java.»

многозадачных ОС это напрямую ведет к тому, что выполнив этот код, злоумышленник получит контроль над машиной. Поэтому с уязвимостью Java оказалось непросто справиться.

Частная территория

Подход Qubes помог бы смягчить угрозу. Будь это отдельный домен, уязвимое при-

ложение Java можно было бы просто отсоединить от остальной системы, а значит, остановить атаку. Конечно, могут пострадать данные или приложения, находящиеся в том же домене, и здесь важно правильно настроить систему. Qubes – это лишь основа для формирования доменов, а не операционная система внутри них. Это просто виртуальные машины,строенные так же, как VM в *VirtualBox* или *Qemu*. А в качестве ОС по умолчанию используется Fedora, и лучшей защиты внутри каждого домена не получится.

Если вы выберете стандартную установку, у вас будет 4 домена: Банк, Личное, Непроверенное и Работа. Можете сократить или увеличить их число, в зависимости от своих потребностей, включив дополнительные виртуальные машины для непредвиденных ситуаций.

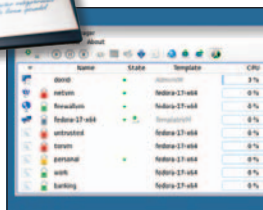
При запуске приложения оно появляется на рабочем столе, как и в любом другом дистрибутиве. Единственное отличие – цветная рамка окна, указывающая, в каком домене оно выполняется. Несмотря на всю эту магию безопасности, творящуюся за кадром, по ощущениям это вполне обычный рабочий стол Linux. Два очевидных исключения – функции копирования и вставки, а также управление файлами.

Чтобы установить приложения в домены, необходимы шаблоны VM. Таким об-

разом можно обновлять базовую систему, используемую виртуальными машинами. Выбор К-меню > Приложения > Шаблон: Fedora-17-x64 > Fedora-17-x64: Добавить/Удалить программы запустит программный менеджер *PackageKit*. Выполнив нужное действие, необходимо выйти из шаблона VM и из домена, чтобы его привязать. Можно сделать это правым щелчком по соответствующей вкладке в Менеджере доменов Qubes. После его закрытия вы можете запускать приложение из командной строки в соответствующем домене. Для некоторых приложений можно установить ярлык через К-меню, выбрав опцию Добавить ярлыки в подменю нужного домена, но, как выяснилось, это возможно не всегда. Утилиты для создания программного менеджера ярлыков здесь нет.

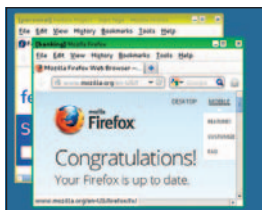
Так выглядит установка приложений, принадлежащих всем доменам, но бывают случаи, когда нужно ограничиться только определенными виртуальными машинами. К примеру, вы хотите установить тестовую версию вашего web-браузера в один из доменов, сохранив более раннюю и надежную в домене Банк. В этом случае вам понадобится создать новый шаблон для добавления программ. Не забывайте, что чем больше шаблонов вы создаете, тем больше вам потом придется обновлять. Как правило, буфер обмена для каждого

Свойства новскидку



Менеджер доменов

С помощью этого инструмента Qubes можно следить за всеми запущенными доменами.



Выделение цветом

Каждое окно имеет цветную рамку, показывающую, к какому домену оно принадлежит.

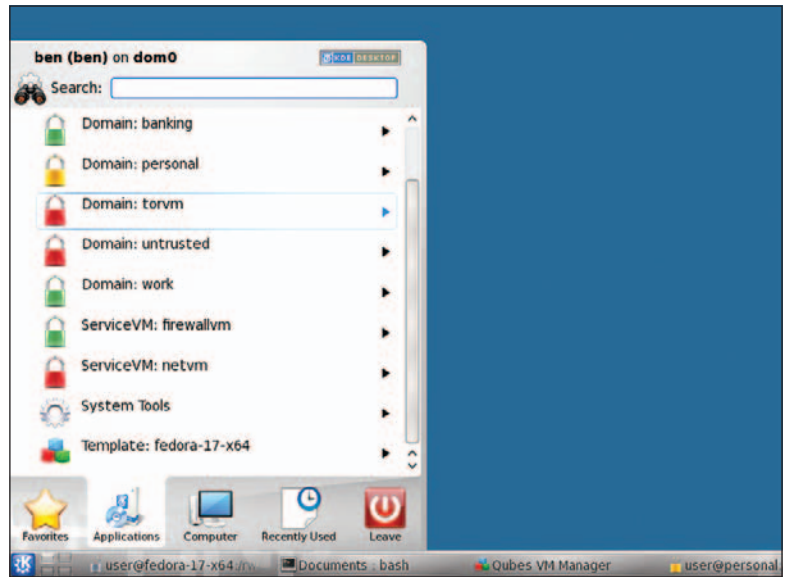
домена свой. Однако при желании можно переместить его содержимое в общий буфер, используя Ctrl+Shift+c. А затем вставить в любой домен при помощи Ctrl+Shift+v.

В каждом домене своя файловая система, и вы не можете перейти из одной в другую. К счастью, существует скрипт, позволяющий копировать файлы отовсюду. Хотя стоит предупредить, что безопасность в этом случае – на совести пользователя. Общее правило в том, что копировать следует из более надежных доменов в менее надежные. Для переноса файлов, откройте файловый менеджер и пройдите Файл > Скрипты > Копировать в другой домен приложений [AppVM] (см. первый рисунок). Для успешного завершения нужно, чтобы оба домена были запущены (можно воспользоваться Менеджером доменов Qubes). Как только перенос выполнен, файлы появятся в `~/incoming/from-<domain>/`.

Домены – не единственный класс виртуальных машин в составе Qubes; имеются также другие, выполняющие се-

«Qubes рановато внедрять в сферах повышенной секретности.»

тевые функции. В целях безопасности, системная ОС (Dom0) не имеет выхода в Интернет. Эту функцию выполняет сетевой домен [NetVM], а затем передает информацию защитному домену [FirewallVM], который фильтрует ее, прежде чем она поступит в другие домены. По крайней мере, таков стандартный путь. Виртуальную сеть можно заменить, установив по отдельности проводной или беспроводной доступ, или же настроив некоторым доменам выход через Tor. Хороший пост одного из разработчиков, описываю-



➤ Привязать приложение к конкретной машине можно при помощи К-меню.

щий этот процесс, можно найти на <http://theinvisiblethings.blogspot.be/>

Как вы можете догадаться, виртуализация ведет к потерям производительности. Мы проводили тестирование на машине с двухъядерным процессором, частотой 2,1 ГГц и 4 ГБ памяти, и она слегка притормаживала. Полноэкранное видео немного подрагивало – не катастрофично для системы, но лишая ожидаемого ощущения безупречности. Пожалуй, это минимальный предел возможностей оборудования, чтобы использовать Qubes.

Ставка на будущее

В Qubes вложена уйма труда, гораздо больше, чем во многие другие дистрибутивы, и это хороший вклад в стабильность Linux. Главные идеи, лежащие в основе этой системы – идеи безопасности – хороши, и разработкой занималась весьма уважаемая команда.

Однако на данном этапе Qubes рановато внедрять в сферах повышенной секретности – он еще слишком нов. Слишком мало людей успели его опробовать и найти слабые стороны. С другой стороны, дистрибутивы Linux общего назначения достаточно надежны, и многим покажется, что эта безопасность не стоит претерпеваемых неудобств. Возможно, в будущем Qubes упростит пользователям освоение, и эта ситуация изменится. Любой из тех, кто использует один и тот же компьютер и для повседневного использования, и для более конфиденциальных операций, может обойтись установкой отдельных Linux-систем на разных зашифрованных дисках. Единственное преимущество Qube в этом случае – возможность работать параллельно и в защищенной, и в открытой среде.

В будущем Qubes может также способствовать укреплению корпоративной безопасности, по-прежнему позволяя сотрудникам использовать компьютер в личных целях. Хотя пока что процесс установки трудноват даже для рядового пользователя, не говоря уж о службах техподдержки, работающих со всей организацией.

Так что пока Qubes можно охарактеризовать как достойное воплощение идеи, полезное некоторым людям в определенных ситуациях, когда им приходится запускать небезопасное приложение в безопасных окружениях. Возможно, придет время, и эти принципы будут усвоены главными дистрибутивами. Например, специальный браузер в виртуальной машине, предназначенный только для выполнения банковских операций, может оказаться весьма неплохой и легко реализуемой задумкой.

Короче говоря, мы в LXF пока что на Qubes не перейдем, но будем пристально следить за его развитием. **LXF**



➤ За завесой виртуализированной магии скрывается рабочий стол KDE.

LINUX Вердикт
FORMAT

Qubes

Разработчик: Invisible Things Lab
Сайт: www.qubes-os.org
Цена: Бесплатно

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

» Лучшая ОС для обеспечения безопасности изоляцией, но пока не готова для широкого применения.

Рейтинг 8/10

Amarok 2.6

Маяк Шарма вновь обращается к любимому музыкальному плееру KDE и выясняет, сумел ли он достичь прекрасной формы версии 1.4.

Вкратце

» Многофункциональный кроссплатформенный музыкальный плеер. См. также: *Banshee*, *Clementine*.

Что за год стал 2012-й для всего, что есть в KDE – начиная с самого рабочего стола, теперь доступного на разных устройствах, до его многофункциональных кросс-платформенных приложений! Хотя *Amarok* не входит в состав *KDE Software Compilation*, позитивная динамика сказалась и на нем.

Последние пару лет *Amarok* пытался добраться до планки, некогда заданной его лучшим релизом 1.4, и на сей раз разработчики пошли верным путем. Если вы раньше не пользовались *Amarok*, знайте, что он не для тех, кто хочет проиграть пару треков. *Amarok* – для подлинных меломанов, которым нужно создавать собственную музыкальную коллекцию и работать с файлами на переносных плеерах.

Новички, остерегайтесь

Новички могут испугаться интерфейса *Amarok*. Вообще-то он был постоянным источником нареканий. И хотя разработчики пытались сгладить недостатки, до сих пор требуется некоторая доработка, чтобы он стал по-настоящему удобным. Например, часть текста, появляющегося при первой установке, по-прежнему обрывается, и на некоторых панелях все еще присутствует горизонтальная прокрутка, несмотря на полноэкранный интерфейс.

Amarok использует 3 панели, отображающих источники воспроизведения, Контекстное окно и текущий плей-лист. Помимо локальных файлов, плеер умеет проигрывать содержимое других ресурсов, включая музыкальные интернет-сервисы, такие как *Last.fm*, и подкасты. К стати



» Немного доработки, и *Amarok* смог бы конкурировать с элегантнейшими музыкальными приложениями.

о подкастах: в *Amarok* нет специального источника с готовым списком, он просто предоставляет текстовое поле для добавления RSS-канала или Atom. А составить список можно во вкладке Подкасты, расположенной под Интернет-источниками. В *Amarok* представлен большой лист интернет-источников, из которых он может проигрывать музыку, включая платные, вроде магазина MP3 от Amazon и *Magnature.com*, и бесплатные, наподобие *Jamendo.com* и *LibreVox.org* с контентом по лицензиям Creative Commons.

Внешний вид плеера можно настроить под себя: изменить размеры панелей, переставить их или скрыть. На устройствах с маленьким экраном можно оптимизировать занимаемое пространство, наложив их друг на друга в виде вкладок.

Умник

Amarok может воспроизводить любую музыку с любого ресурса. Он также может подключаться к различным устройствам при помощи USB и MTP, и к iPod через *libgpod*. Библиотеки поддержки для iPod и iPhone были полностью обновлены, но устройства iOS 5 все еще не поддерживаются из-за ограничений библиотеки.

Главное новшество этого релиза – расширение возможностей конвертации файлов при переносе на другие устройства. В прежних версиях при добавлении в коллекцию файла с внешнего устройства *Amarok* позволял кодировать этот трек в другой формат. Теперь можно делать то же и при обратном действии. Новички

оценят, что в диалоге Транскодирования предлагаются готовые настройки, например, битрейт в зависимости от выбранного кодека. Мелким упущением видятся расположение и размер панели, отображающей ход транскодирования, а также отсутствие уведомления о завершении.

Amarok 2.6 прекрасно смотрится на KDE и должен быть доступен в репозиториях вашего дистрибутива. Проверьте, чтобы у вас также установлен пакет *ffmpeg*, для конвертации. Еще вам понадобится пакет *phonon-backend-gstreamer*, перенаправляющий запросы от приложения *Phonon* на модуль *GStreamer*, отвечающий за воспроизведение. А поскольку этот плеер еще и бережлив к системным ресурсам, он вполне способен обольстить новичков. LXF

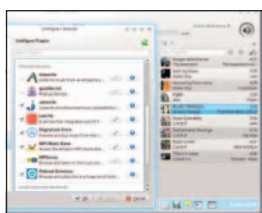


Свойства новскидку



Транскодирование

Теперь пользователи могут транскодировать треки до их копирования на или со своих плееров.



Интернет-источники

Вы можете покупать музыку и слушать ряд интернет-сервисов и подкастов прямо в *Amarok*.

LINUX FORMAT Вердикт

Amarok 2.6

Разработчик: The Amarok Team
Сайт: amarok.kde.org
Цена: Бесплатно по GPL

Функциональность	9/10
Производительность	9/10
Удобство использования	5/10
Документация	7/10

» Передовой музыкальный плеер, улучшенный достаточно, чтобы понравиться даже самым строгим критикам.

Рейтинг 7/10

Highscreen Yummy Duo

Артем Зорин гадает, является ли флагманом новый флагманский «двухсимочный» смартфон от Highscreen.

Вкратце

» Стильный отечественный смартфон китайской сборки, с поддержкой двух SIM-карт, гораздо дешевле зарубежных аналогов. Для тех, кому бренд — не главное.

Спецификации

- » Процессор MT6573 650 МГц, ОЗУ 512 МБ, флэш-память 200 МБ (доступно 178 МБ)
- » Android 2.3
- » TN-дисплей, 4,3", 480×800, «мультитач»
- » GSM, GPRS, 3G, Bluetooth, Wi-Fi, GPS, FM-радио
- » Акселерометр, компас, датчики приближения и освещенности
- » Слот microSD
- » Камера 5 Мп, автофокус, светодиодная вспышка
- » Батарея Li-ion на 1500 мА·ч

Российская компания Highscreen выпускает неплохие смартфоны, и в модельном ряду этого производителя есть настоящие флагманы. Мы рассмотрим один из них — «двухсимочный» смартфон Highscreen Yummy Duo.

Внешний вид

Внешне он схож с HTC One X. Большой 4,3-дюймовый экран занял почти весь «фасад». Но аппарат довольно тонок — всего 10,5 мм, это отличный показатель для двухсимочных телефонов: у ближайших конкурентов, Gigabyte GSmart G1345 и Fly IQ260 BlackBird, толщина 12–13 мм.

Габариты хорошо собранного и со всех сторон обтекаемого корпуса позволяют легко держать его и брать с поверхности стола. Вес — 110 г. Материал — приятный на ощупь мягкий пластик — не даст телефону выскользнуть из рук. Цвет — черный; лишь металлическая полоса по периметру корпуса разбавляет строгость. На наш вкус, пластик смартфона выглядит чуть дешево. Но и цена невысока — от 6990 руб.

Недостаток дизайна — неудобное расположение слотов для SIM-карт, каковые не сменить, не открыв заднюю крышку аппарата (например, отверткой); вдобавок надо вынуть батарею. Производителю стоило подумать над этим! Внешние органы управления, кнопка Вкл/Выкл/Сон и регулятор громкости, уютно расположились на правом и левом торцах соответственно. Они удобны и не продавливаются.

Железо

Работает все без проблем — быстро и надежно, хотя процессор по нынешним временам слабоват, а объем ОЗУ невелик.

Приложения и ОС работали без тормозов, лишь изредка рушились некоторые системные процессы, что можно списать на инженерный экземпляр; надеемся, что в серии эти недостатки будут исправлены.

Сенсорный экран иногда не реагирует на прикосновения, но это можно списать на начинку ПО. Отзывчивость и яркость хороши, как и время автономной работы: в смешанном режиме (звонки, SMS, Интернет через Wi-Fi) телефон проработал более 7 часов — что вполне прилично.

В комплект входит проводная стереогарнитура, подключаемая к стандартному 3,5-мм Jack-разъему на верхнем торце.

Звук почти оглушает; не советуем выставлять максимум громкости, иначе напугаете прохожих. Но громкий и чистый звук вовсе не значит качественный: низких частот нет, а высокие слушать устаешь очень быстро. Зато оповещение владельца о входящем звонке или SMS не упустишь.

Камера, несмотря на внушительную матрицу, выдавала снимки не самого высокого класса. Вспышка с автофокусом работала через раз, что тоже не радует. Видео... ну, для телефона нормально.

Софт

Yummy Duo шустро работает под отживающей свой век версией ОС Android 2.3.5 Gingerbread. Производитель не вмешивался в лаконичный интерфейс системы (кроме телефонной части): семь горизонтально прокручиваемых рабочих столов и длинное общее вертикальное меню с несортируемыми программами.

За функции телефона Yummy заслужил 5 баллов: прекрасно держит сигнал сети и быстрее многих моделей восстанавливает связь после потери. Радиомодуль один, но обе SIM-карты — в постоянно активном режиме ожидания, и обе сети мигают. Сетки работают стабильно и бесперебойно. Остальным модулям связи — тоже 5: Wi-Fi «ловится» даже в плохую погоду на улице, а 3G, Bluetooth и GPS не сбоят.

Набор предустановленного ПО скромный, но достаточен, чтобы включить и начать работу. Ни одна из программ не вызвала ошибок, не рухнула, не заставила смартфон перезагрузиться, как бывало на младших моделях Highscreen. Да и Google play никто не отменял... Единственный минус программной начинки аппарата — отсутствие новых прошивок от производителя.

Производительность

Понятно, что по производительности Yummy не назвать «топовым» на фоне «флагманов» с комплектующими поновее. Заоблачных результатов его «бенчмаркинг» не показал.

Правда, пользователь может увеличить объем предустановленной памяти, вставив в соответствующий слот карту памяти типа microSD объемом до 32 Гб (в комплекте поставляется 4-Гб карта).



» Притягательность этого телефона раскрывается не сразу. Нужно с ним пожить и понять, как он хорош при своей цене.

Выводы

Highscreen Yummy Duo хорош доступностью для большинства. Он берет качеством сборки, отличным телефонным блоком, долгой работой от аккумулятора, «двухсимочностью», большим и качественным экраном; и все это — не по цене подержанных «Жигулей». На его недостатки «эстетсы» могут смотреть пренебрежительно, но это ничуть не умаляет всех достоинств этого замечательного смартфона, а их больше, чем недостатков. Его можно посоветовать купить всем, кто проводит много времени в командировках: две симки им очень пригодятся. **LXF**

LINUX FORMAT **Вердикт**

Highscreen Yummy Duo

Разработчик: Highscreen
 Сайт: <http://highscreen.org>
 Цена: от 6990 руб.

Функциональность	9/10
Производительность	7/10
Удобство использования	8/10
Оправданность цены	10/10

» Хороший, добротный телефон для тех, кто не гонится за производительностью и модой.

Рейтинг 8/10



ВОТ ЭТО ПИРОЖКИ

Самое вкусное – как всегда, от Google

Google анонсировала Android 4.2 Key Lime Pie и серию новых устройств семейства Nexus.

Компания Google официально анонсировала мобильную платформу Android 4.2 Key Lime Pie [лаймовый пирог] и первые мобильные устройства на ее основе. Выпуск Android 4.2 продолжает развитие ветки 4.x и предлагает пользователям целую серию интересных улучшений, таких как поддержка нового метода ввода в экранной клавиатуре и многопользовательский режим. Кроме того, выполнен ряд оптимизаций, направленных на увеличение отзывчивости интерфейса и скорости работы платформы.

Из основных улучшений можно отметить уже упомянутый многопользовательский режим, который позволяет организовать совместное использование планшета разными людьми без пересечения на уровне данных и настроек. Режим подразумевает наличие в системе нескольких независимых пользователей с индивидуальными настройками всех приложений, раздельным доступом к данным и возможностью оформления домашнего экрана на свой вкус.

В экранной клавиатуре появилась штатная реализация метода набора неразрывными жестами по экрану, когда ввод слов осуществляется путем скольжения от одной буквы к другой, без отрыва пальца от экрана. Использование интеллектуальных методов оценки вероятности выбора той или иной буквы в сочетании с проверкой по словарю позволило заметно поднять эффективность и скорость ввода, при этом не требуя от пользователя строгого сосредоточения внимания, на какую букву нажать.

Интересная особенность добавлена в приложение для работы с камерой. Это функция Photo Sphere для создания панорамных снимков окружающего пользователя пространства в стиле Google Street View. В итоге формируется виртуальная сфера, в центре которой находится пользователь. Созданные фотосферы можно



» Лаймовый пирог дал название новой версии Android 4.2 от Google.

просматривать локально, отправлять знакомым через Google+ или размещать для общего обозрения в Google Maps.

Появилась также возможность передавать изображения с экрана телефона на телевизор без проводов, при условии, что ТВ поддерживает протокол беспроводной передачи данных Miracast и оборудован Wi-Fi.

Значительно улучшена работа функции Android Beam, позволяющей легко обмениваться фотографиями, ссылкой

«Android 4.2 предлагает целую серию интересных улучшений.»

ми, контактами, приложениями и видеороликами с находящимся рядом Android-устройством, поддерживающим NFC. Достаточно приблизить аппараты бок о бок и перетащить пальцем нужный элемент в сторону рядом расположенного аппарата.

В целом, Android 4.2 удался. Что же касается популярности и степени распространенности, то тут дело обстоит несколько иначе: две трети устройств до сих пор работают на устаревшей вер-

сии Android 2.3, остальная часть использует более свежие версии 4.0 и 4.1. Причина такого отставания производителей в том, что они банально не успевают за Google – слишком уж часто выпускает эта богоспасаемая компания версии своей мобильной платформы; что, с одной стороны хорошо, потому что всегда есть шанс обогнать конкурентов, но с другой стороны, погоня за Google изматывает вендоров, и из-за этого, в частности, происходит фрагментация кода Android.

Сейчас сложно сказать, когда и сколько именно устройств будут работать на Android 4.2, кроме тех, что уже представила сама компания Google – это смартфон Nexus 4 (произведен в сотрудничестве с компанией LG и оснащен 4,7-дюймовым экраном, 4-ядерным процессором Snapdragon S4 Pro и 2 ГБ ОЗУ, стоит от 299 до 349 долларов), и планшеты Nexus 10 (создан совместно с компанией Samsung, поставляется с десятидюймовым экраном с рекордно высоким разрешением 2560×1600 точек; планшет будет поставляться в вариантах с 16 и 32 ГБ встроенной памяти и двуядерным процессором A15; стоимость модели с 16 ГБ встроенной памяти составит 399 долларов, с 32 ГБ – 499 долларов) и Nexus 7 (выпущен компанией ASUS и отличается от прошлой модели увеличением размера памяти и доступностью варианта с 3G-модемом; стоимость модели с 16 ГБ встроенной памяти составит 199 долларов, с 32 ГБ – 249 долларов, с 3G-модемом и 32 ГБ памяти – 299 долларов).

Стоимость аналогичных по характеристикам планшетов от других производителей выше в полтора, а то и в два раза.

Одно из указанных устройств в порядке комплимента было бесплатно отправлено почтой создателю Linux, и Торвальдс давно уже резвится с новеньким планшетником.

Исходные коды и SDK Android 4.2 уже опубликованы.

ИНСТРУМЕНТАРИЙ СТАЛ МОБИЛЬНЫМ

Qt Creator 2.6 и порт Qt на Android

Скоро великолепие рабочего стола KDE переключается на мобильные устройства.

Компания Digia выпустила релиз IDE Qt Creator 2.6, для создания кроссплатформенных приложений на базе библиотеки Qt – как классических программ на языке C++, так и на языке QML, применяющем JavaScript для определения сценариев, а структура и параметры элементов интерфейса задаются CSS-подобными блоками. Исходники IDE лицензированы по LGPL.

Также объявлено о включении в состав Qt Project проекта Necessita. Это набор средств разработки и портирования Qt-приложений для Android. Изначально проект развивался группой энтузиастов при поддержке сообщества KDE. Отныне он переходит под эгиду сообщества Qt Project при финансовой поддержке Digia;

» **«Пишите меньше кода. Создавайте больше. Развертывайте везде». Девиз Qt теперь будет справедливым и для разработчиков Android.**



наработки Necessita позволяют построить Qt 5 для Android (сейчас поддерживается Qt 4.8) и за 2013 год перевести Android в разряд первичных платформ для Qt.

Процесс подготовки, сборки и отладки программ для Android интегрирован в IDE Qt Creator. На телефоне все Qt-приложения используют один набор разделяемых библиотек Qt, что дает большую экономию па-

мяти при одновременном запуске нескольких Qt-программ. Единоразово собранная как стандартный арк-пакет Qt-программа может далее применяться в различных версиях Android, не требуя пересборки при выходе новых версий. Для установки и поддержки Qt-библиотек в системе Android создана программа *Ministro* – это сервис для Android, обеспечивающий наличие в системе актуального для текущей версии набора Qt-модулей. При отсутствии требуемого модуля он загружается с Сети.

Итак, Qt получает шанс стать еще более популярной средой разработки – для нового класса программистов-универсалов, пишущих и для настольных, и для мобильных устройств. Разработчики избавятся от громоздкого API для Android от Google.

ТЕПЕРЬ И НА ПЛАНШЕТАХ

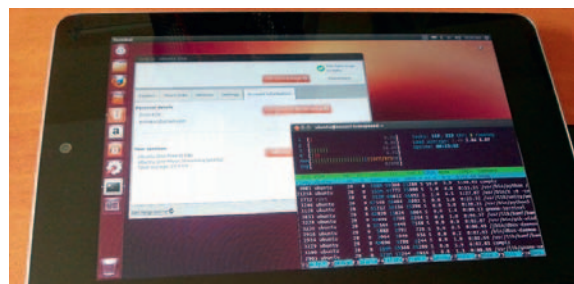
Ubuntu портирован на Nexus 7

Проект интересен для программиста и по сути бесполезен для пользователя.

В октябре вышла новая модель нетбука Chromebook от Google, по цене 249 долларов, с 11,6-дюймовым (1366×768) экраном и весом всего в 1,1 кг. Заряда батареи хватает на 6,5 часов. Устройство создано в партнерстве с компанией Samsung и позиционируется как компьютер для всех. В отличие от прежних устройств с ChromeOS, Chromebook базируется на архитектуре ARM, а не x86.

Нетбук оснащен 2-ядерным процессором Samsung Exynos 5 1,7 ГГц (ARMv7), 2 ГБ ОЗУ, 16 ГБ SSD, Wi-Fi 802.11 a/b/g/n, web-камерой, портами USB 3.0 и 2.0, портом HDMI, Bluetooth 3.0, свежей ChromeOS и приложениями для интеграции с сервисами Google (Gmail, Drive, Search, Maps, YouTube, Play и Google+). Пользователям на 2 года бесплатно выдается 100 ГБ в облачном хранилище Google Drive. Заказ доступен пока только в США и Великобритании.

Буквально через пару дней Виктор Палау [Victor Palau], директор по коммерческому инжинирингу компании Canonical, опубликовал демо-ролик о развитии интерфейса Ubuntu для планшетников, описывающий работу оболочки Unity на Nexus 7 с управлением через сенсорный



» **Ubuntu на Nexus – как это «по-гиковски», и как бесполезно! Но очень занимательно...**

экран. А Джоно Бэкон [Jono Bacon], менеджер Canonical по работе с сообществом, объявил о выдаче на тестирование экспериментальной сборки Ubuntu 12.10 для Nexus 7. Сборка поставляется с интегрированной экранной клавиатурой. Пользовательское окружение пока идентично оболочке для настольных ПК. Для желающих попробовать подготовлена инструкция по разблокировке загрузчика планшета и установке Ubuntu. Инсталлятор опубликован в PPA-репозитории *ubuntu-nexus7*; он автоматизирует загрузку свежего образа Ubuntu и перепрошивку планшета.

Планы разработчиков по улучшению Ubuntu для планшетов амбициозны: сократить потребление памяти и время запуска

приложений пользователя, увеличить эффективность управления энергопотреблением, время работы от аккумулятора и производительность Unity; повысить удобство с сенсорного экрана... и пр., и пр.

Да, интересно, но что-то шепчет: путевого в этой затее мало. Ведь Android на мобильных устройствах объективно лучше и быстрее, чем Ubuntu, «заточенный», наоборот, под настольные ПК. Хочется сказать талантливой команде Ubuntu: не тратьте время зря, лучше совершенствуйте свое – настольную версию Ubuntu. А мобильные устройства оставьте программистам Google.

Впрочем, можно предположить, что разработчики Ubuntu просто хотят отработать технологию создания программ для сенсорных экранов, либо оттачивают свое ремесло на разных классах устройств. Конечно, фанатикам стоит попробовать установить Ubuntu на свой новенький Google Nexus 7. Но трудно представить себе светскую львицу занимающейся перепрошивкой своего Chromebook или планшета. А ведь именно на них – пользователей без серьезного опыта – нацелены обе компании, и Google, и Canonical. LXF



Делаем снимки

Не нравится программа для камеры в своем телефоне? Есть идея программы для работы с фото? Джульетта Кемп знакомит с Android Camera API.



Наш эксперт

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

В подавляющем большинстве телефонов с Android есть встроенные камеры, и с развитием технологий они становятся все удобнее. Программ, которые так или иначе работают с камерой, великое множество, и предоставляемое Android API максимально упрощает эту задачу.

Если вам нужен всего-то один снимок, возможно, проще воспользоваться Намерением [Intent] для активации встроен-

ной камеры (см. врезку внизу), но если вам нужно больше рычагов управления, можно написать собственное Занятие [Activity] Camera.

На нашем уроке мы создадим Занятие, которое просто использует камеру, но, разумеется, ничто не мешает создать приложение с другим главным Занятием, где заодно будет и Занятие для камеры, вызываемое при необходимости.

Подготовка камеры

Прежде чем писать код, следует объявить требования для камеры в **AndroidManifest.xml**:

```
<manifest ....>
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-feature android:name="android.hardware.camera" />
```

Если вы не собираетесь сохранять изображения, **WRITE_EXTERNAL_STORAGE** не понадобится (но так как мы позже это сделаем, то добавим ее сразу). Директива **uses-feature** означает, что для установки приложения в телефоне обязательно должна быть камера. Если камера – только часть вашей программы, и пользователи прекрасно смогут работать с программой и без нее, добавьте **android:required="false"** к этой строке. Также можно задать характеристики камеры; более подробно они описаны в одном из следующих разделов.

Теперь можно продолжить с кодом первоначальной настройки, в котором мы впервые получаем экземпляр объекта камеры. Метод **onCreate()** в **MyCameraActivity** должен выглядеть так:

```
@Override public void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
if (!checkCameraExists(this)) {
Toast.makeText(this, "Увы, камеры нет!", Toast.LENGTH_LONG);
finish();
}
camera = getCameraInstance();
}
```

Один из параметров манифеста требует обязательного наличия камеры на устройстве, но ее наличие стоит проверить и здесь. Метод **checkCameraExists()** прост:

```
private boolean checkCameraExists(Context c) {
```

```
if (c.getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA)) {
return true;
} else {
return false;
}
}
```

Метод **getCameraInstance()** ненамного сложнее:

```
private Camera getCameraInstance() {
Camera c = null;
try {
c = Camera.open();
} catch (Exception e) {
Toast.makeText(this, "Увы, камеры нет!", Toast.LENGTH_LONG);
Log.e(TAG, "Нет камеры: исключение " + e.getMessage());
e.printStackTrace();
finish();
}
return c;
}
```

Метод **Camera.open()** обращается к первой камере на задней поверхности устройства.

Камер на устройствах Android может быть несколько; для доступа к конкретной камере воспользуйтесь методом **Camera.open(int cameraId)**. Метод **Camera.getNumberOfCameras()** вернет количество камер устройства, а метод **Camera.getCameraInfo()** – информацию о заданной камере. Для большинства задач нам подойдет первая камера на задней поверхности устройства.

В следующих разделах мы займемся просмотром изображения с камеры и созданием снимка. Но перед этим нужно сделать еще одну важную вещь: закончив работу с камерой, освободите ее.

Иначе другие процессы не смогут получить доступ к камере – это плохая практика; кроме того, это обеспокоит пользователей.

Создаем быстрый снимок

Если нужно сделать быстрый снимок, пригодится Намерение:

```
{
    Intent i = new Intent(MediaStore.
ACTION_IMAGE_CAPTURE);
    fileUri =
getOutputMediaFileUri(MEDIA_TYPE_IMAGE);
    i.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
    startActivityForResult(i,
CAPTURE_IMAGE_ACTIVITY_REQ);
}
protected Uri getOutputMediaFileUri(int type) {
    // См. код в основном тексте для
getOutputMediaFile()
    return Uri.fromFile(getOutputMediaFile(type));
}
```

Поместите первый фрагмент кода в подходящее место; можно связать его с кнопкой, поместив в метод **onClick()** (как в коде на нашем DVD) или сделав пунктом меню. В этом фрагменте кода создается новое Намерение – намерение по умол-

чанию для создания снимка – и оно связывается с URI, указывающим, где сохранить результирующее изображение. Затем мы запускаем Намерение для запуска Занятия. Нам также нужно обработать результат:

```
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
    if (requestCode == CAPTURE_IMAGE_ACTIVITY_
REQ) {
        if (resultCode == RESULT_OK) {
            if (data == null) {
                // Здесь замечена ошибка! Снимок должен
сохраниться в fileUri
                Toast.makeText(this, "Снимок успешно
сохранен",
                Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(this, "Снимок успешно
сохранен: " + data.getData(),
                Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

```
// Здесь вы можете сделать что-нибудь еще...
} else if (resultCode == RESULT_CANCELED){
    // Пользователь отменил операцию; ничего
не делаем
} else {
    Toast.makeText(this, "Отказ вызова захвата
изображения!",
    Toast.LENGTH_LONG).show();
}
}
```

На практике вы, наверное, захотите как-то обработать изображение. В некоторых устройствах замечена ошибка, связанная с сохранением изображения посредством ПО камеры. URI изображения должен быть возвращен вместе с Намерением. Однако иногда возвращается пустое Намерение, хотя файл был сохранен правильно. Чтобы обойти эту проблему, сохраните URI изображения, который вы отправляли с Намерением, и снова воспользуйтесь им при получении успешного результата.

Внесем нужные изменения в метод **onPause()**:

```
@Override protected void onPause() {
    super.onPause();
    releaseCamera();
}
private void releaseCamera() {
    if (camera != null) {
        camera.release();
        camera = null;
    }
}
```

@Override protected void onResume() {

```
    if (camera == null) {
        camera.getCameraInstance();
    }
    super.onResume();
}
```

Когда мы создадим код просмотра изображения с камеры, данный метод придется несколько усложнить, но мы займемся этим потом. А пока вам важнее всего помнить, что следует обязательно освобождать камеру, когда наше приложение приостанавливается.

Просмотр изображения с камеры

С помощью Camera API можно сделать снимок так, что пользователь ни о чем не узнает, но в подавляющем большинстве случаев сначала нужно показать изображение с камеры, а затем пользователь нажмет на кнопку. Для этого создадим собственный класс CameraPreview, унаследованный от SurfaceView:

```
public class CameraPreview extends SurfaceView implements
SurfaceHolder.Callback {
    private static final String TAG = "CameraPreview";
    private SurfaceHolder sh;
    private Camera camera;
    public CameraPreview(Context context, Camera cm) {
        super(context);
        camera = cm;
        sh = getHolder();
        sh.addCallback(this);
        // устаревший, но требуемый pre-3.0
        sh.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
    }
    public void surfaceCreated(SurfaceHolder holder) {
        try {
```

```
            camera.setPreviewDisplay(holder);
            camera.startPreview();
        } catch (IOException e) {
            Log.e(TAG, "Ошибка установки предпросмотра: " +
e.getMessage());
            e.printStackTrace();
        }
    }
    public void surfaceChanged(SurfaceHolder holder, int format, int
width, int height) {
        if (sh.getSurface() == null) {
            // Нет поверхности предпросмотра!
            return;
        }
        // Остановка предпросмотра до внесения изменений.
        try {
            camera.stopPreview();
        } catch (Exception e) {
            // Попытка пресечь несуществующий предпросмотр
        }
    }
}
```

Скорая помощь

При повороте камеры в положение Пейзаж ее, кажется, можно не сбрасывать, и следует делать это только при переключении на Портрет (переключение камеры при повороте на Пейзаж вызовет «падение» при записи видео).

»

```
try {
    camera.setPreviewDisplay(sh);
    camera.startPreview();
} catch (Exception e) {
    Log.e(TAG, "Ошибка при рестарте предпросмотра: " +
        e.getMessage());
    e.printStackTrace();
}
}

public void surfaceDestroyed(SurfaceHolder holder) {
    // Занятие следит за освобождением предпросмотра
    камеры
}
}
```

Этот класс унаследован от SurfaceView, осуществляющего рендеринг гораздо быстрее, чем стандартный View (и отрисовка может выполняться фоновыми нитями, чего не допускает View), что идеально для быстрой отрисовки, которая необходима нам для просмотра изображения с камеры.

Для доступа к поверхности изображения нам понадобится также реализовать интерфейс SurfaceHolder.Callback. Здесь необходимы методы **surfaceCreated()**, **surfaceChanged()** и **surfaceDestroyed()**.

Мы получаем SurfaceHolder (с помощью метода родительского класса), затем добавим к ней Callback, который проинформирует клиента о любых изменениях с поверхностью.

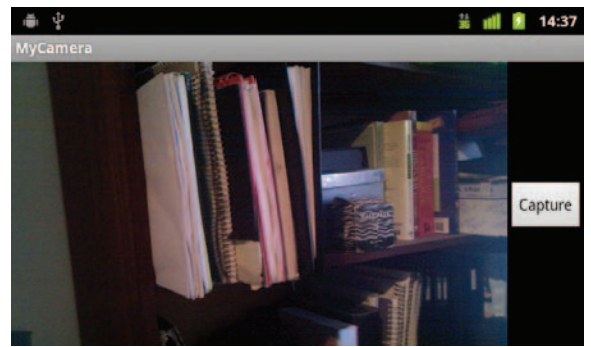
Наша версия **surfaceChanged()** ничего не меняет. Желая ввести изменения, делайте это между остановкой и запуском предпросмотра. Одна из ситуаций, когда это может пригодиться – об-

Скорая помощь

Чтобы удачнее расположить компоненты, можно задать разные файлы **main.xml** в **res/layout-land** (для пейзажа) и **res/layout-port** (для портрета). Пример есть на DVD.



➤ Несколько деревьев в режиме Портрета.



➤ Простейший предпросмотр изображения с камеры (это моя книжная полка).

работка изменения положения устройства (камеры и экрана) в пространстве.

В устройствах с Android версии до 2.2 предпросмотр автоматически включался как Пейзаж, и для максимальной совместимости проще всего принудительно установить режим Пейзажа, добавив атрибут к элементу activity в **AndroidManifest.xml**:

```
<activity [ ... ] android:screenOrientation="landscape" >
```

Однако в версиях с 2.2 и выше это можно убрать, взамен задав положение просмотра согласно положению устройства. Добавьте такие строки в метод **surfaceChanged()** в **CameraPreview.java** между вызовами **stopPreview()** и **startPreview()**:

```
if (getResources().getConfiguration().orientation == Configuration.
    ORIENTATION_PORTRAIT) {
    camera.setDisplayOrientation(90);
}
}
```

Не забудьте удалить настройку пейзажа из **AndroidManifest.xml**, и вы увидите, что предпросмотр поворачивается вслед за поворотом камеры.

Создав класс CameraPreview, добавьте метод **setUpLayout()**, вызываемый из метода **onCreate()** в **MyCameraActivity**.

Здесь он выделен в отдельный метод, чтобы проще приостанавливать/возобновлять программу; подробности см. ниже.

```
private void setUpLayout() {
    setContentView(R.layout.main);
    preview = new CameraPreview(this, camera);
    FrameLayout frame = (FrameLayout) findViewById(R.
        id.camera_preview);
    frame.addView(preview);
}
}
```

Также нужно добавить кое-что в файл **main.xml**:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/
    res/android"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <FrameLayout
        android:id="@+id/camera_preview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
    />
    <Button
        android:id="@+id/button_capture"
        android:text="Capture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```


Скорая помощь

На эмуляторе есть «камера», но она ничего не показывает (только пустой экран). Для проверки работы предпросмотра понадобится настоящее устройство.

```
android:layout_gravity="center"
```


Кодом кнопки мы займемся в следующем разделе. Скомпилируйте и запустите программу, и на экране должно появиться изображение с камеры с кнопкой Capture [Сделать снимок] рядом, хотя и при нажатии кнопки пока ничего не произойдет.

Наконец, ранее я говорила, что после написания кода просмотра методы **onPause()** и **onResume()** нужно немного усложнить. Если просто освободить камеру, то при возобновлении приложения старый класс предпросмотра тоже попытается обратиться к объекту камеры, который у него был (и который теперь освобожден), и вызовет исключение.

Чтобы это исправить, нужно освобождать объект при приостановке приложения, а при возобновлении – создавать новый:

```
protected void onPause() {
    releaseCamera();
    super.onPause();
}
```

```
}
private void releaseCamera() {
    if (camera != null) {
        camera.stopPreview();
        camera.release();
        camera = null;
        preview = null;
    }
}
protected void onResume() {
    if (camera == null) {
        camera = getCameraInstance();
        setUpLayout();
    }
    super.onResume();
}
```

Теперь вы сможете выйти из программы и затем перезапустить ее без ошибок.

Создание снимка

У нас есть кнопка Capture; теперь нужно связать с ней какие-нибудь действия. Добавьте следующий код в **setUpLayout()**:

```
Button captureButton = (Button) findViewById(R.id.button_capture);
captureButton.setOnClickListener(
    new View.OnClickListener() {
        public void onClick(View v) {
            getImage();
        }
    }
);
```

Метод **getImage()** выглядит так:

```
protected static final int MEDIA_TYPE_IMAGE = 0;
protected static final int MEDIA_TYPE_VIDEO = 1;
private void getImage() {
    PictureCallback picture = new PictureCallback() {
        public void onPictureTaken(byte[] data, Camera cam) {
            File picFile = getOutputMediaFile(MEDIA_TYPE_IMAGE);
            if (picFile == null) {
                Log.e(TAG, "Ошибка при создании медиа-файла; верны ли разрешения на запись?");
                return;
            }
            try {
                FileOutputStream fos = new FileOutputStream(picFile);
                fos.write(data);
                fos.close();
            } catch (FileNotFoundException e) {
                Log.e(TAG, "Файл не найден: " + e.getMessage());
                e.printStackTrace();
            } catch (IOException e) {
                Log.e(TAG, "Ошибка I/O файла: " + e.getMessage());
                e.printStackTrace();
            }
        }
    };
    camera.takePicture(null, null, picture);
}
private File getOutputMediaFile(int type) {
```

```
File directory = new File(Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_PICTURES),
    getPackageName());
if (!directory.exists()) {
    if (!directory.mkdirs()) {
        Log.e(TAG, "Не удалось создать каталог для хранения.");
        return null;
    }
}
String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
File file;
if (type == MEDIA_TYPE_IMAGE) {
    file = new File(directory.getPath() + File.separator + "IMG_" +
        timeStamp + ".jpg");
} else if (type == MEDIA_TYPE_VIDEO) {
    file = new File(directory.getPath() + File.separator + "VID_" +
        timeStamp + ".mp4");
} else {
    return null;
}
return file;
}
```

В методе **getOutputMediaFile()** мы сохраняем фотографии в публичном каталоге на внешнем устройстве (SD-карте). Это означает, что если удалить приложение, фотографии останутся. Если вы предпочли бы их удалить, сохраняйте их в каталоге, полученном с помощью **Context.getExternalFilesDir()**. Однако учтите, что пользователи могут не ожидать того, что их фотографии будут удалены при удалении программы работы с камерой! Впрочем, при некоторых обстоятельствах такое имеет смысл – тщательно продумайте это для своего приложения.

Интерфейс **PictureCallback** предоставляет доступ к данным изображения из программы для работы с фотографиями; здесь мы получаем файл, куда нужно записать данные (что по сути занимает большую часть кода!), и записываем их в этот файл.

Настроив **PictureCallback**, мы сообщаем камере, что нужно снять фотографию с помощью встроенного метода **takePicture()**,

»

и передаем эти данные PictureCallback при вызове этого метода (т.е. при нажатии кнопки Capture).

При попытке запустить программу сейчас вы увидите, что после создания снимка предварительный просмотр «зависает» и больше не работает. Чтобы это исправить, добавьте вызов `camera.startPreview()` после создания снимка.

Однако у нас остается другая проблема: пока изображение сохраняется в файл, предпросмотр будет «заморожен».

Чтобы это исправить, сохраним фотографию в фоновом режиме, создав AsyncTask:

```
private void getImage() {
    PictureCallback picture = new PictureCallback() {
        public void onPictureTaken(byte[] data, Camera cam) {
            new SaveImageTask().execute(data);
            camera.startPreview();
        }
    };
    camera.takePicture(null, null, picture);
}

class SaveImageTask extends AsyncTask<byte[], String, String> {
    @Override protected String doInBackground(byte[]... data) {
```

```
File picFile = getOutputMediaFile(MEDIA_TYPE_IMAGE);
if (picFile == null) {
    Log.e(TAG, "Ошибка при создании медиа-файла; верны ли разрешения на запись?");
    return null;
}
try {
    // блоки try/catch, как сделано выше...
}
return null;
}
```

Я рассказывала о нем в предыдущей статье; по сути это вспомогательный класс, от которого можно унаследоваться, чтобы легко запустить процесс в фоновой нити (обычно все, что запускается в коде, работает в одной основной нити. Помните, что интерфейс Android небезопасен с точки зрения нитей, поэтому выполняйте в фоне только задачи, не обновляющие графический интерфейс!). В данном случае мы просто выносим в фон код сохранения файла, чтобы повторно перезапустить просмотр сразу после создания снимка.

Съемка видео

Камерой можно снимать не только фотографии, но и видео. Если вы хотите это делать, сначала установите право `android.permission.RECORD_AUDIO uses-permission` в `AndroidManifest.xml`.

Съемка видео сложнее съемки фотографий, и для нее нужно выполнить следующие действия в указанном порядке:

1. Разблокировать камеру.
2. Создать новый объект `MediaRecorder` и добавить в него камеру.
3. Задать источники аудио и видео.
4. Задать профиль с настройками и форматом кодирования видео, начиная с API 8 (для устройств в Android версии менее 2.2 это делается вручную).
5. Задать выходной файл.
6. Включить предпросмотр.
7. Подготовить `MediaRecorder`.
8. Начать запись.

Следующий код выполняет шаги 1–7:

```
private MediaRecorder mr;
protected boolean prepareForVideoRecording() {
    camera.unlock();
    mr = new MediaRecorder();
    mr.setCamera(camera);
    mr.setAudioSource(MediaRecorder.AudioSource.CAMCORDER);
    mr.setVideoSource(MediaRecorder.VideoSource.CAMERA);
    mr.setProfile(CamcorderProfile.QUALITY_HIGH);
    mr.setOutputFile(getOutputMediaFile(MEDIA_TYPE_VIDEO).toString());
    mr.setPreviewDisplay(preview.getHolder().getSurface());
    try {
        mr.prepare();
    } catch (IllegalStateException e) {
        Log.e(TAG, "IllegalStateException при подготовке MediaRecorder " + e.getMessage());
        e.printStackTrace();
        releaseMediaRecorder();
        return false;
    }
```

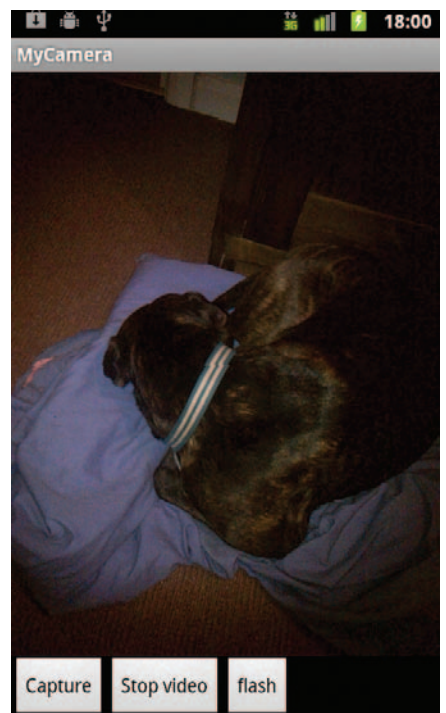
```
} catch (IOException e) {
    Log.e(TAG, "IOException при подготовке MediaRecorder " + e.getMessage());
    e.printStackTrace();
    releaseMediaRecorder();
    return false;
}
return true;
}
```

Затем можно настроить кнопку для остановки и записи воспроизведения:

```
private void setUpVideoButton() {
    videoButton = new Button(this);
    setUpButton(videoButton, "Старт видео");
    videoButton.setOnClickListener(
        new View.OnClickListener() {
            public void onClick(View v) {
                if (isRecording) {
                    mr.stop();
                    releaseMediaRecorder();
                    camera.lock();
                    videoButton.setText("Старт видео");
                    isRecording = false;
                } else {
                    if (prepareForVideoRecording()) {
                        mr.start();
                        videoButton.setText("Стоп видео");
                        isRecording = true;
                    } else {
                        // Что-то не в порядке! Освободите камеру
                        releaseMediaRecorder();
                        Toast.makeText(MyCameraActivity.this, "Извините: видео не стартует", Toast.LENGTH_LONG).show();
                    }
                }
            }
        }
    );
}
```

```
}
);
}
```

Тогда вы сможете запускать и останавливать запись одной кнопкой. Задайте все настройки, затем запустите `MediaRecorder`, и готово. Обратите внимание, что файл просто сохраняется на внешний носитель; чтобы сделать с ним еще что-нибудь, нужно дописать код.



➤ Съемка моей спящей собаки на видео, с постоянной подсветкой [Torch].

Возможности камеры и эффекты

В камерах современных смартфонов может быть масса дополнительных возможностей – вспышка, баланс белого, режим съемки быстро движущихся объектов, даже распознавание лиц (поддерживается только в последнем релизе Android). Чтобы узнать обо всех доступных вам возможностях, изучите API. Для выяснения, какие возможности доступны на вашем устройстве, можно воспользоваться классом `Camera.Parameters`. Узнав, какие параметры/возможности вам доступны, можете работать с ними единообразно. Здесь я включу вспышку; этот код легко расширить на другие возможности. Мы создадим метод `setUpFlash()`, который вызывается из `setUpLayout()`:

```
private void setUpFlash() {
    final Camera.Parameters params = camera.getParameters();
    final List<String> flashList = params.getSupportedFlashModes();
    if (flashList == null) {
        // Вспышки нет!
        return;
    }
    final CharSequence[] flashCS = flashList.toArray(new
    CharSequence[flashList.size()]);
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Выберите тип вспышки");
    builder.setSingleChoiceItems(flashCS, -1,
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            params.setFlashMode(flashList.get(which));
            camera.setParameters(params);
            dialog.dismiss();
        }
    });
    final AlertDialog alert = builder.create();
    alert.show();
}
```

Сперва проверьте, есть ли на этой камере вспышка, получив `Camera.Parameters` для этой камеры и поддерживаемые режимы вспышки из параметров. Если режимов нет, мы возвращаемся, ничего не делая.

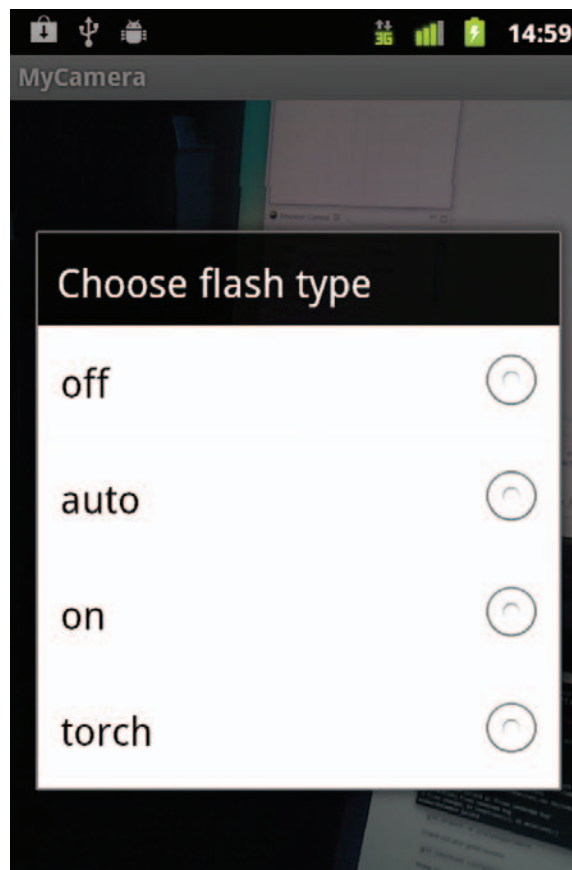
Чтобы включить вспышку, используем `AlertDialog` (можно исключить вспышку, использовать и что-то другое, на ваше предпочтение). Диалогам `AlertDialog` нужен массив `CharSequence` для получения списка компонентов, и нам нужно создать его из списков режимов вспышки. `setSingleChoiceItems()` создает `AlertDialog` с радиокнопками; если вам нужен список без радиокнопок, воспользуйтесь `setItems()`.

Наконец, когда один из элементов списка выбран, мы задаем соответствующий режим вспышки (учтите, что для получения

нужного объекта нам нужно вернуться к `flashList`). Чтобы изменения вступили в силу, нужно вызвать `camera.setParameters()`; без этого параметры не будут установлены и режим вспышки не изменится, пока вы не сделаете снимок. После выбора параметра мы закрываем диалог для возврата в главное окно.

Создайте и покажите диалоговое окно, и все готово! Если скомпилировать и запустить программу, вы увидите, что окно появляется сразу при запуске программы, а это не то, что нам нужно. Во избежание этого, добавьте кнопку программно (см. врезку).

Чтобы улучшить внешний вид программы, используйте графическую кнопку вместо текстовой; кроме того, есть еще масса возможностей, с которыми можно поэкспериментировать при желании. Остается только вопрос, что делать со всеми этими снимками... **LXF**



➤ Доступные режимы вспышки. Постоянная подсветка [Torch] — хорошее решение.

Динамическое/программное расположение компонентов

Информацию о расположении компонентов лучше всего хранить в XML. Но иногда нам необходимо делать это динамически или программно – как в нашем случае, когда кнопка Flash должна появиться, только если у камеры есть вспышка. Добавьте следующий код в конец метода `setUpFlash()`:

```
LinearLayout lin = (LinearLayout) findViewById(R.
id.linearlayout);
Button flashButton = new Button(this);
```

```
flashButton.setText("Flash");
flashButton.setLayoutParams(new
LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT));
lin.addView(flashButton);
flashButton.setOnClickListener{
    new View.OnClickListener() {
        public void onClick(View v) {
            alert.show();
        }
    }
}
```

```
}
);
```

Этот код почти не требует пояснений; мы получаем родительский объект `LinearLayout` из XML, создаем кнопку и добавляем ее в `LinearLayout`. Также показано, как программно задать высоту и ширину кнопки. Все остальные атрибуты кнопки тоже можно задать программно.

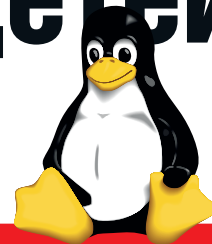
Наконец, `OnClickListener` показывает диалоговое окно `AlertDialog` вспышки при нажатии на кнопку.

Сравнение

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

Дистрибутивы для детей

Linux задуман не только для взрослых — дети тоже его любят. Майк Сондерс ищет лучший дистрибутив для малышей...



Про наш тест...

Многие функции, которые мы проверяли в «обычных» дистрибутивах, здесь не особо важны: например, работа системы пакетов или качество утилит настройки. Детям это неинтересно, и взрослому администратору машины не придется охотиться за самыми свежими обновлениями или возиться с настройками ядра. Идеальный дистрибутив должен установиться без особых хлопот, а потом вы отдадите его на растерзание детям, и чтобы он при этом не ломался и не требовал вашего постоянного дежурства.

Две важнейших функции дистрибутива для детей — это интерфейс и набор программ. Рабочий стол должен быть прост в использовании, и у детей не должно возникать проблем с поиском любимых игр и приложений. И, аналогично, набор программ тоже должен быть хорошим: если все они одинаковы по тематике, или нет должного баланса между игрой и обучением, ребенок скоро утратит к ним интерес.



Наша подборка

- » DoudouLinux
- » Edubuntu
- » LinuxKidX
- » Qimo
- » Sugar on a Stick

Учение с увлечением [Edutainment] — одно из жутких словосочетаний, придуманных теми, кто считает себя чересчур крутым для обычного выражения «обучающая игра». Игры тем ценнее, если они содержат обучающие и развивающие элементы, и со времен *Mario is Missing* у нас были игры, затрагивавшие историю, искусство, математику и другие предметы. Заставить детей все это учить — непростая задача, но если привлечь к ее решению интересных персонажей и веселую музыку, даже лютые ненавистники школы смогут накопить более полезную информацию, чем

сведения о расположении скрытых звезд в *Mario Galaxy*. Linux издавна был очагом образовательных программ, вроде столь могучего инструмента, как *GCompris*, объединявших простые игры с задачами, направленными на обучение. И венчает все это ОС, чья открытость и простота индивидуальной настройки делает ее идеальной для юных пользователей. Любой может

создать интерфейс пользователя и подобрать подходящие программы, нацелив дистрибутив на определенную возрастную категорию. Linux действительно подходит всем, независимо от возраста, и в нем уйма программ, разработанных для реальной помощи детским мозгам. Итак, в этом месяце мы рассмотрим дистрибутивы, нацеленные на юных пользователей.

«Linux издавна был очагом образовательных программ, вроде могучего GCompris.»

Установка

Насколько быстро и легко его запустить?

Велика вероятность, что вы решите установить дистрибутив сами, а не отдадите это на откуп своему ребенку, поэтому здесь особое дружелюбие к малолеткам роли не играет. Однако вам вряд ли захочется погрязать в излишних сложностях. Например, многим ли подросткам нужен SSH-доступ к своим машинам? Если вы делаете установку на компьютер, где ОС уже есть, и вам нужна двойная загрузка, не мешает ли достойный инструмент разбиения диска.

ISO DVD Edubuntu «весит» 2,6 ГБ, и, подобно большинству производных Ubuntu, позволяет загрузить рабочий стол и произвести установку, или установить его прямо из меню загрузки. Интерфейс по умолчанию – Unity, а резервная опция – Gnome. Программа установки – стандартная программа Ubuntu, с хорошими инструментами для разбиения диска, но в нее включен ряд дополнений, например, интеграция LTSP (Linux Terminal Server Project). LTSP предназначен для работы в классе, и он позволит вам поставить Edubuntu на мощном компьютере и подключить к нему более слабые

машины в качестве тонких клиентов. После установки будет занято 4,5 ГБ на жестком диске и 230 МБ ОЗУ при свежей загрузке. DoudouLinux грузится также и в live-режиме, но у него нет даже намека на графическую программу установки. По словам разработчиков, «одна из основных целей DoudouLinux – предоставление детям возможности использовать компьютер без риска повредить уже хранящиеся на нем данные. Поэтому установка DoudouLinux никогда не будет совершаться одним щелчком мыши». Итак, она скрыта, и вам придется пошарить по сайту, чтобы найти ее. Не радует, что установка на жесткий диск производится только с версии для USB-брелка, а не с CD.

Qimo – еще одна производная Ubuntu, поставляемая на 700-МБ CD и имеющая стандартную программу установки Ubuntu; она спросит у вас о вашем местоположении, предложит разбить диск на разделы и так далее. На стадии копирования файлов появится несколько полезных подсказок, и установка займет 2,1 ГБ на жестком диске и 250 МБ ОЗУ при чистом запуске. Sugar on a Stick имеет другую



» Sugar on a Stick идет с хорошо протестированной программой установки от Fedora, Anaconda.

основу – Fedora 17, и хотя он загружается в весьма необычный интерфейс Sugar, вы можете запустить Anaconda, установщик Red Hat, открыв терминал, переключившись в root и введя liveinst. Дистрибутив использует 2,1 ГБ места на жестком диске и 151 МБ ОЗУ при первом запуске.

LinuxKidX – дистрибутив на базе Slackware, также загружаемый в режиме Live. Программа установки скрыта в разделе меню Development – и до ужаса недолетана. Там куча текстовых ошибок, приходится вручную монтировать жесткий диск для установки и вводить путь к программе установки, что осложнено раскладкой клавиатуры по умолчанию для бразильского варианта португальского языка. Пользоваться этим практически невозможно.

Вердикт

Sugar on a Stick ★★★★★
 Edubuntu ★★★★★
 Qimo ★★★★★
 DoudouLinux ★★★★★
 LinuxKidX ★★★★★
 LinuxKidX ★★★★★

» Anaconda в Sugar – самая гибкая.

Первые впечатления

Выглядит ли дистрибутив дружелюбным – или враждебным?

О деталях интерфейса мы поговорим позднее, а здесь рассмотрим первые впечатления – как выглядит дистрибутив после установки и начального запуска. DoudouLinux сопровождает стадии загрузки музыкой

и открывает вам яркое и забавное меню, предлагая пять программ и несколько режимов интерфейса. Здесь имеются крупные кнопки выключения и перезагрузки. Некоторое разочарование испытываешь при завершении программ из этого режима, поскольку X-сервер перезапускается, высвечивая при этом текст командной строки.

Qimo также отличается яркостью и выразительностью. При загрузке рабочего стола внизу появляется огромная панель с семью кнопками, обеспечивающими доступ к основным программам дистрибутива. Традиционная панель задач размещается сверху, что выглядит по-компьютерному, но все же

намного лучше, чем Edubuntu. Видите ли, в Edubuntu у нас есть постоянно и неизбежно только Ubuntu. Конечно, обои рабочего стола отличаются, и есть несколько косметических украшений, но раскладка, шрифты и все остальное – вплоть до барабанной дроби на экране приглашения – Ubuntu в чистом виде. Это заставляет задуматься: а что же, собственно, такого уж особо детского в этом дистрибутиве? Да и яркостью и приветливостью он не грешит.

LinuxKidX пестрит яркими красками и мультяшными пингвинами, придающими ему очень жизнерадостный вид. На рабочем столе множество значков, хотя в общем и целом все выглядит в стиле ретро. А вот Sugar отличается крайним минимализмом – только графика, минимум текста или обычных элементов пользовательского интерфейса. Последний просторен и явно нацелен на концентрацию на отдельных задачах.



» DoudouLinux после загрузки: яркий, разноцветный, и сразу выделяет основные программы.

Вердикт

DoudouLinux ★★★★★
 Qimo ★★★★★
 LinuxKidX ★★★★★
 Sugar on a Stick ★★★★★
 Edubuntu ★★★★★

» У Doudou самый дружелюбный интерфейс.

Интерфейс пользователя

Трудны ли в них нахождение и запуск программ, а также навигация?

Детей не особо волнует управление многочисленными файлами, или вложенность меню приложений, или программы запуска. Для них главное – приложения. Найдя те, что им понравятся, они просто хотят запускать их как можно быстрее. Однако при изобилии приложений дистрибутиву нужна некая способность к их категоризации или просмотру, чтобы он не просто демонстрировал вам один-единственный экран с крошечными значками.

Поэтому особое внимание в нашем тестировании дистрибутивов мы уделяли следующему вопросу: сколько усилий потребует поиск программы, и есть ли риск того, что ребенок по ошибке запустит что-то неподходящее?

И подходит ли интерфейс для разных возрастных категорий, или детям постарше будет казаться, что все чересчур «разжевано» и разработчики снисходят к их умственному уровню, принимая их за недоразвитых?

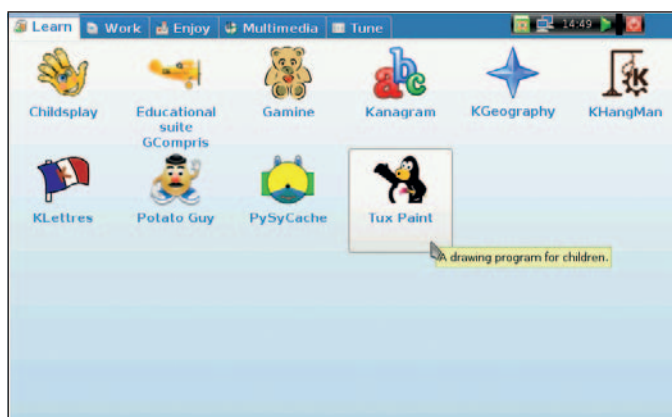
DoudouLinux ★★★★★

Итак, мы куда-то попали! На предыдущей странице вы видели интерфейс Doudou по умолчанию, и он идеален для самых маленьких. Однако Doudou предназначен для детей от 2 до 12 лет, и школьники сочтут его слишком примитивным.

К счастью, предусмотрена Полная версия интерфейса [Whole], которая предоставляет доступ во вкладках к разным категориям, очень сильно напоминая старый GUI на Eee PC от ASUS.

Наверху имеется небольшая панель для отладки сетевых настроек, изменения громкости звука и выключения.

Это практически идеально соответствует нашим ожиданиям, а разделение интерфейса на обычный и полный означает, что дистрибутив пригоден для детей разных возрастных групп, при этом подростки будут чувствовать себя более продвинутыми.



Edubuntu ★☆☆☆☆

Это Unity. Мы не говорим, что Unity всегда плох – на самом деле мы знаем, что многим он очень нравится. Но в дистрибутиве, предназначенном для юной аудитории, этот выбор абсолютно ужасен.

Представьте, например, что ваш ребенок решил посмотреть обучающие приложения: для этого нужно нажать на значок Dash наверху, а затем на крошечный значок без какого бы то ни было текстового пояснения, с линейкой и ручкой внизу – он отобразит, что одни приложения установлены, а другие нет. И тогда вы нажимаете на Installed, чтобы отобразился огромный список всего в алфавитном порядке. Он опять-таки ужасен, и вы нажимаете на выпадающий список Filter и выбираете категорию Education. И, наконец, измучившись до предела, вы просматриваете установленные обучающие программы. Полный и безоговорочный бардак – и такое приходится делать каждый раз. Ну разве не бред?

Индивидуальная настройка

До какой степени они поддаются переделке?

Интерфейс Sugar особенно не переделашь, но имеется диалоговое окно, где можно изменить настройки сети и язык. Зато индивидуальный подбор программ превосходен: через браузер можно зайти на страницу Sugar Activities и одним щелчком установить сотни игр и инструментов, созданных специально для Sugar. У Doudou возможности настройки интерфейса также ограничены, причем сознательно, чтобы дети не могли потерять какие-то программы

или учинить беспорядок. Doudou включает *DansGuardian* для фильтрации сети, по умолчанию настроенный на блокировку порносайтов. Основан Doudou на старом Debian 5 – но можно добавлять программы из репозитория.

Edubuntu позволяет перетаскивать новые приложения на левую панель. Можно сменить довольно нелепые программы по умолчанию на специальное ПО для обучения, или фон и тему – обычным способом Unity, но нельзя по-простому заблоки-

ровать систему, чтобы не дать детишкам в ней насвинячить. FAQ Edubuntu просто перечисляет клавиши *GConf*, с которыми можно работать, но это вряд ли назовешь разумным решением для большинства.

Qimo позволяет настроить *Xfce* и установить пакеты из несвежих репозиториях Ubuntu 10.04, а LinuxKidX предлагает индивидуальную настройку через KDE. Но придется рыться в пакетах Slackware аж 2009 года, и знакомиться с инструментами пакетов, чтобы установить добавочное ПО.

Вердикт

Sugar on a Stick ★★★★★
 DoudouLinux ★★★★★
 Edubuntu ★★★★★
 Qimo ★★★★★
 LinuxKidX ★★★★★

» Здесь особо не отличился никто.



Qimo ★★★★★★

Qimo GUI основан на Xfce, и предлагает довольно типовую двухпанельную раскладку, с док-панелью внизу и стандартной панелью задач с системным лотком наверху. Если вам знаком Xfce, никаких сюрпризов вы здесь не найдете, и даже пользователи Windows смогут достаточно быстро разобраться с такой раскладкой. Чтобы немного адаптировать все это для детей, разработчики использовали симпатичную фоновую картинку с эскимосом (откуда и происходит название Qimo).

Нашим единственным огорчением было отсутствие информативных подсказок для пунктов в доке; было бы полезно дать пользователям разузнать о программе, прежде чем запускать ее. В общем и целом, Qimo удается совместить дизайн с крупными кнопками, удобными для детей, с более традиционной панелью, удобной для родителей при настройке установленного дистрибутива.



LinuxKidX ★★★☆☆

Интерфейс здесь основан на KDE, и большинство настроек, делающих его дружелюбным к детям, чисто косметические. Однако это KDE 3.5, и на данный момент он несколько устарел; но дистрибутив не обновлялся с 2009 года, так что удивляться нечему.

Очень удобно наличие программ запуска приложений на рабочем столе, потому что меню довольно сильно перегружены. Но главная проблема интерфейса – раскладка клавиатуры по умолчанию, которая ни с того ни с сего оказывается бразильским вариантом португальского (дело в том, что это родной язык разработчиков) – в результате некоторые символы находятся не на месте. Чтобы это исправить, перейдите в меню К > Control Center > Regional & Accessibility > Keyboard Layout. Совершенно дурацкая ситуация, ведь язык-то дистрибутива – английский! Хоть бы меню обеспечили при загрузке дистрибутива...



Sugar on a Stick ★★★☆☆

Он совершенно не похож ни на что из виденного нами ранее. Sugar – это интерфейс, разработанный для проекта OLPC (One Laptop Per Child – Ноутбук каждому ребенку), цель которого – обеспечение развивающихся стран компьютерами по низкой цене. В Sugar очень мало текста, и запуск основных программ осуществляется разбросанными кругом значками. Он отличается минимализмом, и в нем нет ничего лишнего; правда, хотелось бы большей красочности. Все программы запускаются в полноэкранном режиме, с крупными значками повсюду. Наведя курсор мыши на угол, вы активируете другой режим отображения для просмотра других учетных записей пользователя и журнала недавно запущенных программ и действий.

На привыкание к нему нужно некоторое время – особенно когда начинают происходить всякие странные вещи: например, по всему экрану появляются выделенные значки, причем без всяких видимых причин.

Документация

Есть ли в дистрибутиве удобная справочная система?

Edubuntu включает *Ubuntu Desktop Guide*, с довольно ценной информацией по использованию и управлению системой. И есть указатели на сайт дистрибутива с документацией по имеющимся приложениям. Но Doudou в этой области идет куда дальше – на его сайте богатый выбор руководств, хотя хромает перевод (уцелели французские слова) и часть разделов слегка устарела. Дистрибутив предоставляет и собственное руководство на 160 страницах в формате

PDF, доступное через Полный интерфейс; оно порой злоупотребляет подробностями, зато охватывает все наличные приложения. Высший балл за старание.

В Qimo документации по сути нет; есть два пункта Help в главном меню, причем первый ссылается на несуществующую версию локально сохраненной документации Xubuntu, а второй – на руководства для рабочего стола Xfce. Похожая ситуация – в LinuxKidX: здесь тоже нет ничего, кроме общих документов KDE. Мы не ждем особо

толстых справочников – хватило бы файла **welcome.pdf** с перечислением достойных программ, имеющихся в дистрибутиве, чтобы заинтересовать детей (да и родителей). Отличная документация Sugar включена в интерфейс – правда, она уделяет основное внимание ноутбуку OLPC, и некоторые моменты не относятся к настольной системе. Однако руководства по использованию программ богаты экранными снимками, поэтому создают ощущение продуктивности и внимания к пользователю.

Вердикт

DoudouLinux ★★★★★

Sugar on a Stick ★★★★★

Edubuntu ★★

LinuxKidX ★★★★★

Qimo ★★★★★

» Руководство пользователя Doudou просто превосходно.

Обучающие приложения

Какие образовательные программы в них имеются?

Edubuntu сумел уместить самую большую подборку; звездой ее выступает *GCompris*. Для тех, кто о таком не слышал: это подборка игр и обучающих программ, существующая с 2000 года. Она очень красочная, и кое-где слишком детская, но своей цели достигает: в ней есть программы, обучающие работе с компьютером, математике, точным наукам, географии, чтению, искусству и многому другому. Подборка весьма объемиста, но, по нашему мнению, она обязана присутствовать в любом дистрибутиве для детей.

Кроме того, в Edubuntu есть *Celestia* (исследование космоса), трио инструментов *Tux* (Математика, Рисование и Обучение печатанию), а также *Parley*, для увеличения

словарного запаса. Есть продвинутые инструменты, такие, как *Lybniz*, для построения графиков математических функций. Поскольку Edubuntu – крупнейший из дистрибутивов Сравнения, и единственный, которому требуется DVD, а не CD, мы предполагали, что его подборка окажется самой крупной, и он нас не разочаровал.

Doudou умудряется предложить весьма солидный ассортимент в своем относительно небольшом объеме, с мощным *GCompris*, подкрепленным *Childsplay*. Это еще одна зрелая и солидная подборка обучающих инструментов, нацеленная в основном на дошкольников. В ней есть программы для развития памяти, головоломки и игры – это хорошая альтернатива *GCompris* для детей младшего возраста.

Кроме того, вы найдете ряд обучающих игр KDE (*Kanagram*, *KHangman* и *KLettres*), а также *Tux Paint* и *PySyCache* (работа с мышью). Большинство этих программ предназначено младшей части аудитории дистрибутива, заявленный возрастной диапазон которого – от 2 до 12 лет, но есть и более продвинутые программы, например, превосходная *KGeography*. Увы, для астрономических программ места не нашлось.

Sugar практикует в корне иной подход, предлагая свой набор программ (именуемых действиями – activities). Их немного, но они разработаны специально под уникальный интерфейс. Так, есть программа для развития памяти, объединяющая элементы математики с лунным браузером, программа рисования черепашки и среда редактирования Python. Звучит довольно амбициозно для детей, но если 10-летний ребенок сможет состряпать пару игр на высокоуровневом языке, то почему бы и нет? Наша главная жалоба – то, что задачи выглядят простецкими по сравнению с *GCompris*.

В Qimo небольшая, но хорошая подборка – *GCompris*, *Childsplay*, *Tux Paint*, *Tux Math*, *Tux Typing* и *Laby*; последнее – весьма любопытно, оно знакомит детей с программированием по мере того, как они помогают муравью выбраться из лабиринта. LinuxKidX имеет типовой набор из *GCompris*, *Childsplay* и *Tux Paint*, но также и всю гамму из своего обучающего пакета KDE.



» *GCompris* – флагман обучающих программ Linux – содержит 100 ярких задач.

Вердикт

Edubuntu ★★★★★
 DoudouLinux ★★★★★
 LinuxKidX ★★★★★
 Qimo ★★★★★
 Sugar on a Stick ★★★★★

» В этой категории лидирует Edubuntu.

Игры

Есть ли у детишек шансы поразвлечься?

Возможно, некоторым родителям не по душе сама мысль об играх в обучающем дистрибутиве. Разве это не противоречит самой идее? В общем-то да, но в более крупных дистрибутивах для детей неплохо дать им возможность немного отвлечься в процессе обучения.

Qimo и LinuxKidX в этой категории явно не лидируют; единственная отдельно взятая игра, включенная в них, это шахматы. Конечно, в *GCompris* есть игры, но коли дистрибутив предназначен для юной аудитории, то следует ожидать большего количества игр на диске. Даже при нехватке места можно было включить больше на-

стояльных игр. В Sugar тоже мало что есть, кроме поиска выхода из лабиринта, но зато более 130 игр в сети становятся доступны по установке одним щелчком, и это несколько улучшает ситуацию. Edubuntu имеет небольшую, но достойную подборку игр, в том числе *Atomix*, *Blinken*, *gbrainy*, *Potato guy*, *Ri-li* и *Sudoku*.

Однако всем им утер нос Doudou. Его подборка игр разделена на категории, и во вкладке Arcade вы найдете *Frozen Bubble* (ура!), *Circus Linux*, *Help Hannah's Horse* (разновидность *Pac Man*) и *Robots*.

В Adventure находятся *Pingus*, *SuperTux* и *Abe's Amazing Adventure*, а раздел Sports содержит *Foobillard*, *Super Tux Kart*



» Онлайн-страница Sugar, Activities, позволяет загрузить более 130 игр.

и *Kolf*. А еще есть настольные игры, логические игры, карточные игры, и еще многое другое.

Это действительно весьма впечатляющий и обширный выбор, особенно принимаемая во внимание ограниченный объем CD и то, что он уже содержит два полнофункциональных образовательных пакета.

Вердикт

DoudouLinux ★★★★★
 Edubuntu ★★★★★
 Sugar on a Stick ★★★★★
 LinuxKidX ★★★★★
 Qimo ★★★★★

» Doudou – явный победитель этого раздела.

Дистрибутивы для детей

Вердикт

Итак, кто же победил в этой битве? Знаете, прежде чем сесть и увязнуть на много часов в этих дистрибутивах, мы предположили, что всех опередит Edubuntu. В конце концов, это самый известный дистрибутив для детей, и за его плечами стоит Ubuntu. Это ли не формула победы? Как оказалось – нет, по следующим причинам. Unity делает работу в нем ужасной (преимущество быстрого запуска приложения при наборе его имени здесь не работает, если ребенок хочет просмотреть имеющиеся программы), и помимо интеграции LTSP нет никаких причин выносить его в отдельный дистрибутив.

Практически все можно найти в обычной установке Ubuntu и скрипте *Bash* из 20 строк для отладки части настроек и замены части пакетов. Он разработан скорее для классов, чем для домашнего использования, как остальные дистрибутивы, и можно было бы многое еще сде-

лать. Поэтому чемпионом стал Doudou, с меньшим выбором программ, но зато с превосходным интерфейсом, большим объемом документации и отличной подборкой игр. Он очень продуман, и создается впечатление, что разработчики понимали, что нужно детям. Местами он вычурен и нацелен на несмышленишей, но Полный интерфейс исправляет эту ситуацию. Для детей от 2 до 8 лет мы рекомендуем его.

Если ваш ребенок старше, он или она, возможно, сочтут более привлекательными программы в LinuxKidX. Главная проблема здесь в том, что дистрибутив этот изрядно устарел, а поскольку он основан на Slackware, поиск новых программ может стать делом нелегким. Проблемы с раскладкой клавиатуры и хромая программа установки тоже не способствуют улучшению ситуации.

Qimo неплох – это славный компактный дистрибутив,



» Doudou победил по нескольким причинам; одна из них — отличная документация, которая так и тянет вас поизучать имеющихся в нем программ.

прямо как если бы Edubuntu сделали немного тщательнее и для домашнего использования. Визуально он соответствует основам Ubuntu/Debian, но выбор программ бедноват, а документации не хватает катастрофически.

И еще есть Sugar... он нам очень нравится, несмотря даже на ощущение экспериментальности. Хотя это не совсем то, что дети ожидают увидеть на своем настольном компьютере, его стоит попробовать хотя бы ради превосходных репозиторий программ в Интернете – мы там обнаружили настоящие шедевры.

«Чемпион — Doudou, с превосходным интерфейсом и отличной подборкой игр.»

I DoudouLinux ★★★★★
 Сайт: www.doudoulinux.org Версия 1.2
 » Дружелюбный, хорошо документированный, с отличными играми и инструментами.

IV Sugar on a Stick ★★★★★
 Сайт: <http://bit.ly/1236lv> Версия Основан на F17
 » Новаторский интерфейс требует некоторого привыкания, но он того стоит.

II Qimo ★★★★★
 Сайт: www.qimo4kids.com Версия 2.0
 » Достойная попытка на базе Ubuntu, но не хватает приложений и документации.

V Edubuntu ★★★★★
 Сайт: www.edubuntu.org Версия 12.04
 » Обычный Ubuntu, только в него добавили LTSP и изменили некоторые пакеты.

III LinuxKidX ★★★★★
 Сайт: <http://linuxkidx.blogspot.com> Версия 2.0
 » Ныне несколько устарел, но все же лучший дистрибутив для детей постарше.

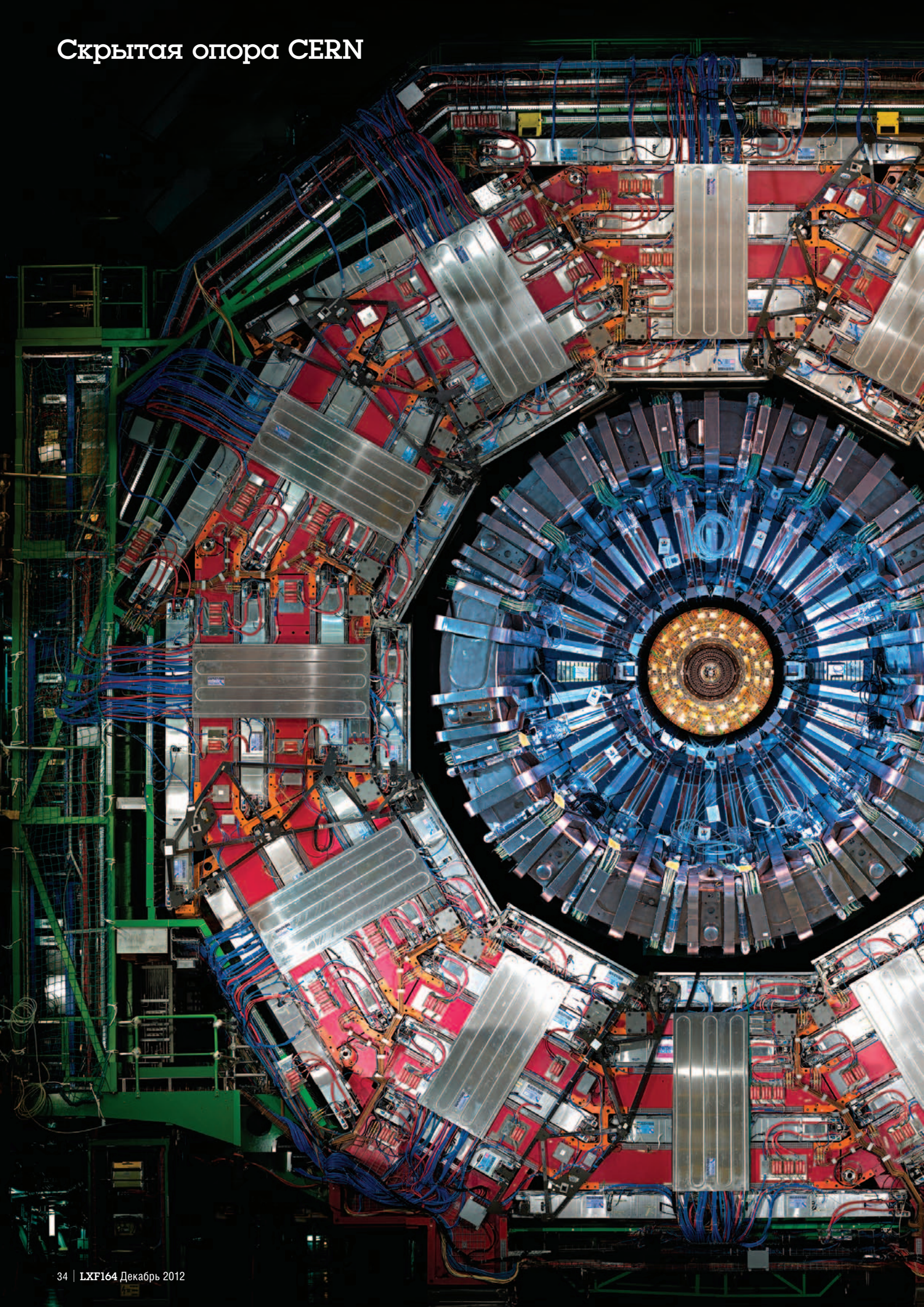
Обратная связь
 А что думаете вы? Вы знакомы с этими дистрибутивами – возможно, у вас сложилось иное, отличное от нашего, впечатление? lx.f.letters@futurenet.co.uk

Рассмотрите также...

Дистрибутивов для детей не так уж и много. У Foresight была разновидность с обучающими программами, но ее забросили. Kiddix (www.kiddix-computing.com) выглядит весьма многообещающим, но он коммерческий, стоит \$10. Будь у него внешний лоск и глубокая интеграция, то тут и говорить бы не о чем, но оказалось, что он практически полностью состоит из свободных

программ, которые легко найти самим. Однако на рынке по-прежнему вакантны два места: первое – это дистрибутив, подобный Doudou, но учитывающий интересы более широкой возрастной группы. Представьте себе, что он поставляется на DVD, и при загрузке можно указать возраст и получить соответствующий интерфейс и набор программ. А на втором представьте себе дистри-

бутив для изучения самого Linux, со множеством небольших руководств, всплывающих подсказок на рабочем столе, и т. д. Это стало бы отличной тестовой площадкой, знакомящей детей с тем, как работает ОС – мы не имеем в виду настройки *sysctl* и прочие низкоуровневые сложности, мы говорим о структуре директории, где хранятся персональные файлы, и т. п. **LXF**



Скрытая опора CERN

Адам Оксфорд выясняет, как Linux помогает охотиться за бозоном Хиггса.

Обычный день доктора Фабрицио Сальваторе [Fabrizio Salvatore] складывается примерно таким образом: кофе, короткий брифинг с двумя кандидатами наук, работающими на его кафедре в университете Сассекса [University of Sussex], в зеленом кампусе Брайтона. После этого – немного более продолжительная встреча с парой аспирантов, занимающихся исследованиями, а потом – обычные преподавательские дела и обязанности. Весьма типичный академический день.

Если же д-р Сальваторе уложит время на собственное исследование, он начнет с того, что загрузит программу под названием ROOT в свой ноутбук с Ubuntu. ROOT – излюбленная программная среда для физики высоких энергий (HEP) и анализа частиц. Раз в неделю он проводит конференцию по телефону со своими сотрудниками по исследованию по всему миру.

В один действительно прекрасный день – возможно, это случится единственный раз за всю его жизнь – он подпишет бумагу об обнаружении почти стопроцентного доказательства существования бозона Хиггса, так называемой частицы Бога, которая придает остальным частицам массу и, таким образом, создает реальность именно в том виде, в котором она существует.

Д-р Сальваторе – один из 3300 ученых, работающих в ATLAS, проекте, включающем создание, обработку и анализ данных от Большого адронного коллайдера (БАК) во всемирно-

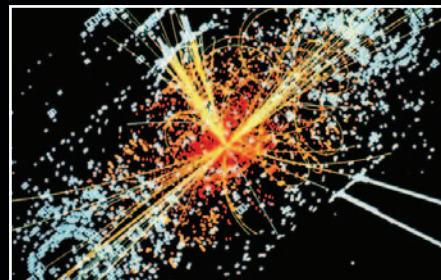
известной швейцарской лаборатории, CERN. «Когда я впервые попал в CERN в 1994 году, – говорит д-р Сальваторе, – это было мое первое пребывание за границей. И моя работа здесь – а длилась она всего месяц – убедила меня в том, что я хочу работать над чем-то, связанным с ней. Я знал, что буду работать над кандидатской в области физики частиц, и знал, что хочу заниматься этим именно здесь».

В июне этого года ученые из ATLAS и их коллеги из CMS, другого эксперимента CERN, объявили о том, что они обнаружили вероятное доказательство существования бозона Хиггса, важной субатомной частицы, чье существование предполагалось чисто теоретически на протяжении полвека, но которой никто не наблюдал. Эту новость окрестили «одним из самых важных научных открытий всех времен», и сделал это не кто иной, как секс-сим-

«Невероятное явление столкновения протонов лишь начало работы.»

вол научной сферы, профессор Брайан Кокс [Brian Cox]. И – отложим в сторону такую мелочь, как само создание БАК – Хиггс был обнаружен почти целиком и полностью с помощью Linux. На самом деле, многие из тех ученых, с которыми мы беседовали, сказали, что иначе сделать это было бы невозможно.

Интерес общественности к работе CERN неувидителен. Невозможно остаться равно-



Имитация того, как могут выглядеть данные столкновения бозона Хиггса.

душным при разговорах об ускорителях частиц, квантовой механике и воссоздании того состояния, в котором пребывала Вселенная в самом начале времен, даже если вы не понимаете того, что на самом деле означают понятия супер-симметрии и элементарной частицы. Внутри коллайдера протоны направляются друг на друга со скоростью, близкой к скорости света. Они пролетают порядка 17 миль замкнутого пространства под Франко-Швейцарскими Альпами, и субатомные осколки их столкновений фиксируются одним или несколькими из семи детекторов, размещенных по периметру кольца. ATLAS и CMS – только два из них.

И невероятное физическое явление столкновения двух протонов – это лишь начало напряженной работы; именно то, что происходит потом, требует наличия одного из крупнейших в мире компьютерных проектов с открытым кодом.

БАК – ЭТО МНОГО ДАННЫХ

Эксперименты или «события» в БАК дают массу информации. Даже после отбрасывания 90 % данных, полученных его сенсорами, по изначальным приблизительным подсчетам, для хранения данных от коллайдера потребуется порядка 15 петабайт ежегодно. В 2011 году коллайдер предоставил около 23 петабайт данных для анализа, а в 2012-м ожидается, что эта цифра вырастет до 30 ПБ, то есть в два раза превзойдет изначально планируемый показатель. Этой зимой ускоритель закроется на 20 месяцев на ремонт и модернизацию, что приведет к еще большему увеличению объема данных, полученных в результате экспериментов.

Вот эти цифры и есть продукт: тестируя результаты, физики не работают исключительно с последними данными. Событие внутри коллайдера никогда не рассматривается изо-



Компьютерную сеть распределенных вычислений Великобритании, GridPP, можно просматривать через разные порталы.

лированно, но всегда считается частью целого. Сбор данных от БАК, их распределение и тестирование – колоссальная задача. «Самое главное испытание для нас при компьютерной обработке данных, – объясняет Ян Берд [Ian Bird], руководитель проекта Worldwide LHC Computing Grid, – это поиск ресурсов, ко-

«С точки зрения технологий и ПО мы в чистом виде мастерская Linux.»

торые понадобятся для эксперимента в будущем, потому что и объем данных, и требования к их обработке будут все время повышаться, а экономическая ситуация такова, что финансирование сильно не увеличится».

Проводя параллель, можно сказать, что Google ежедневно обрабатывает около 25 ПБ. Но при этом Google не анализирует каждый пиксель в каждой букве каждого слова, помещаемого в архивы, из-за того, что в нем зарегистрирована подпись основной частицы, которую покамест никто не наблюдал.

«То, чем мы занимаемся, не похоже на то, чем занимаются остальные, – говорит Берд. – Скажем, скачав видео, вы получите немало данных, но большинство, скорее всего, просто посмотрит десять первых файлов, и их можно сохранить в кэше, чтобы ускорить обработку. Наша проблема в том, что мы работаем с колоссальными наборами данных, и физикам нужен весь набор целиком. Им нужны не первые четыре гигабайта этого набора, а все

2,5 ПБ – так же, как и тысяче других исследователей. И вы не можете использовать обычную коммерческую сеть передачи данных для решения этой проблемы».

Пауло Калафиура [Paulo Calafiura] – главный архитектор экспериментального ПО ATLAS, и он работает в этом проекте с 2001 года. Тогда

ощущение его масштабности многое изменило.

«Мы были на передовой “больших данных”, – говорит Калафиура. – Когда я впервые сказал, что у нас будет 10 ПБ данных в год,

у народа челюсти отвисли. В наши дни Google или Facebook могут без особых усилий сделать это в любом из своих центров обработки данных. Однако в науке мы по-прежнему остаемся лидерами по объему данных». Будучи по специальности программистом, Калафиура имеет богатый опыт работы в области физики. До начала работы в ATLAS он помог писать GAUDI Framework – это фундамент большинства приложений физики высоких энергий, особенно используемых в CERN. Идея GAUDI заключалась в создании общей платформы для физических исследований, чтобы облегчить сотрудничество ученым всего мира.

До GAUDI, по словам Калафиуры, большинство программ для анализа обычно писались по мере надобности на языке FORTRAN. Благодаря переходу на объектно-ориентированную среду для сбора данных, моделирования и анализа с использованием C++, команда, в которой работал Калафиура, заложила основы широкомасштабного глобального сотрудничества на основе работы CERN.

«Среда GAUDI определено является мультиплатформенной, – говорит Калафиура. –

Что такое бозон Хиггса?

Нас всех очень взволновало сообщение ATLAS и CMS о том, что они обнаружили частицу, которую никто ранее не наблюдал и которая выглядит, как бозон Хиггса, если бы он существовал, но почему?

«В физике частиц, – говорит профессор Бьяндо Лучини [Biagio Lucini] из Университета Суонси [University of Swansea] – он еще и пишет для LXF! – частицы удобно считать стационарными возмущениями поля. Если поле – мембрана на барабанах, то частица – результат удара по ней палочкой. Частицы – это отклонения от нормальной величины поля, которая для большинства типов частиц составляет ноль. Частица Хиггса отличается от прочих, т.к. соответствующее ей поле имеет величину, отличную от нуля. Это запускает т.н. механизм Хиггса – по Стандартной модели физики частиц, это феномен, придающий другим частицам массу».

Само поле было названо в честь профессора Питера Хиггса [Peter Higgs], предположившего его

существование в 1964 г. Поле делает это, нарушая симметрию внутри слабого электрического поля других частиц, и, таким образом, создавая материю в реальности в том виде, в котором мы ее знаем.

Для тех, чье знание квантовой физики сводится в той или иной степени к коту Шредингера: нарушение симметрии – это пункт, в котором ящик открывается и кот обнаруживается живым или мертвым. Без поля Хиггса или без чего-либо подобного ему не могли бы сформироваться более крупные частицы – или объекты вроде нас с вами.

Бозон Хиггса – нестабильная частица, которая, существуя она, распалась бы на частицы помельче почти сразу после отделения от столкнувшихся протонов. В результате данные с сенсоров анализируются на предмет наличия структур, указывающих, вместо самой частицы, на наличие частиц, на которые она могла распасться – скажем, на два протона, или Z- и W-бозоны.



Gridmap в EGI отображает статус разных сетей, масштабируемых логическими CPU.

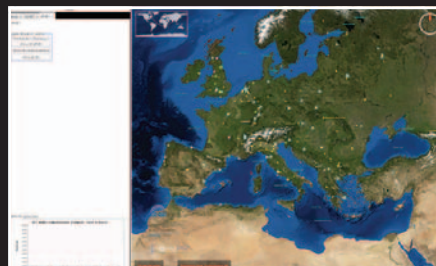


Сначала программа ATLAS поддерживалась рядом платформ Unix, а GAUDI поддерживала – и поддерживает – Windows. Примерно в 2005 году мы отключили сборку Solaris [из-за отсутствия интереса], а до этого большая часть оборудования работала на HP-UX. Но серверы перешли на Linux, и все были довольны».

«В данный момент, – продолжает Калафиура, – с точки зрения настоящих компьютерных технологий и разработки прикладных программ мы являемся в чистом виде мастерской Linux. Растет число тех, кто ратует за MacOS, но работа ведется на самой лучшей основе».

По словам Калафиуры, ноутбуки Apple все чаще встречаются среди участников конференций по физике высоких энергий, но в них очень редко можно встретить кнопку Пуск [Start]. «В своей душе архитектора, – признает он, – я не очень доволен тем, что мы работаем исключительно на Linux, потому что проще

решать некоторые проблемы, не будучи зависимым от платформы. Но использование только Linux позволяет нам обойти ряд острых углов. Например, мы обсуждали non-POSIX функцию Linux под названием Splice – это канал, где вы не копируете данные, что повышает эффективность нашей обработки данных».



› Для желающих понаблюдать за распределенными вычислениями БАК в реальном времени имеется отличный апплет Java.

Словарь CERN

CERN	European Organisation for Nuclear Research (Conseil Européen pour la Recherche Nucléaire) – Европейская организация ядерных исследований
ATLAS	A Toroidal LHC Apparatus – Тороидальный аппарат БАК
CMS	Compact Muon Solenoid – Компактный мюонный соленоид
TOTEM	Total Elastic and diffractive cross section Measurement – Общее измерение эластичности и дифракционного сечения
ALICE	A Large Ion Collider Experiment – Эксперимент Большого ионного коллайдера
LHCb	LHC beauty – Красота БАК
MoEDAL	Monopole and exotics Detector At the LHC – Детектор медленных частиц
LHCf	LHC Forward
WLCG	Worldwide LHC Computing Grid – Всемирная сеть распределенных вычислений БАК
LHC	Large Hadron Collider – Большой адронный коллайдер (БАК)

Сотрудничество в области открытого кода

Около 10 000 физиков по всему миру работают над проектами, связанными с CERN; две трети из них брошены на крупные эксперименты, ATLAS и CMS.

Анализ данных, произведенных БАК – это дерзание на многих уровнях. Предоставление ученым результатов подразумевает массовую

передачу данных. Затем встает вопрос обработки этих данных. Надежность – тоже важный аспект: если на сервере где-то посреди процесса произойдет сбой после двухсуточной работы, его перезапуск будет стоить всего ранее затраченного драгоценного времени и усилий.

Сфера обработки и хранения данных в CERN обеспечивается мощным проектом распределенных вычислений внутри одной из самых мощных вычислительных сетей в мире, с самым широким охватом. Точная терминология и реализация WLCG разная в 36 странах и 156 организациях, принимающих участие

в проекте, но по сути это – многоуровневая сеть для доступа и ресурсов. В центре этой сети располагается CERN, или Tier 0 (T0), располагающий свыше 28 000 логических CPU для обеспечения распределенных вычислений. T0 – это место, где генерируются необработанные экспериментальные данные. С T0 соединены – чаще всего 10-Gbps волокон – центры Tier 1 (T1), которые обычно располагаются в крупных национальных лабораториях физики высоких энергий и выступают в роли локальных узлов. В большинстве случаев все T1 отражают одни и те же данные, произведенные в CERN.

Ниже центров T1 располагаются центры обработки данных Tier 2 (T2). Как правило, они расположены в крупных университетах, где имеются необходимые ресурсы, чтобы обеспечить свободное место, стойки и как минимум одного сотрудника для работы с ними. Обычно соединение между центрами T1 и T2 осуществляется через национальную академическую сеть (в Великобритании это JANET), и поэтому они имеют высокую скорость, но не специально выделенный трафик. Локально хранятся не все исследовательские данные – скорее, поднаборы данных, представляющие интерес для определенной организации, будут находиться на постоянном хранении, а остальные данные будут по мере надобности браться в T1.

Учреждения Tier 3 (T3) – это, как правило, менее масштабные университеты и исследовательские центры, которым не требуются отдельно выделенные для работы местные ресурсы, но которые, согласно требованиям, скачивают данные из сети T2 и обеспечивают их обработку в сети распределенных вычислений.



► Д-р Фабрицио Сальваторе — один из 3000 ученых, работающих над ATLAS. He Tardis.

Существует несколько информационных онлайн-панелей, куда каждый может зайти и увидеть состояние сети (самые информативные – <http://dashb-wlwg-transfers.cern.ch/ui/>, <https://netstat.cern.ch/monitoring/network-statistics/visual/?p=ge> и <http://wlwg.web.cern.ch/>). В настоящий момент имеется

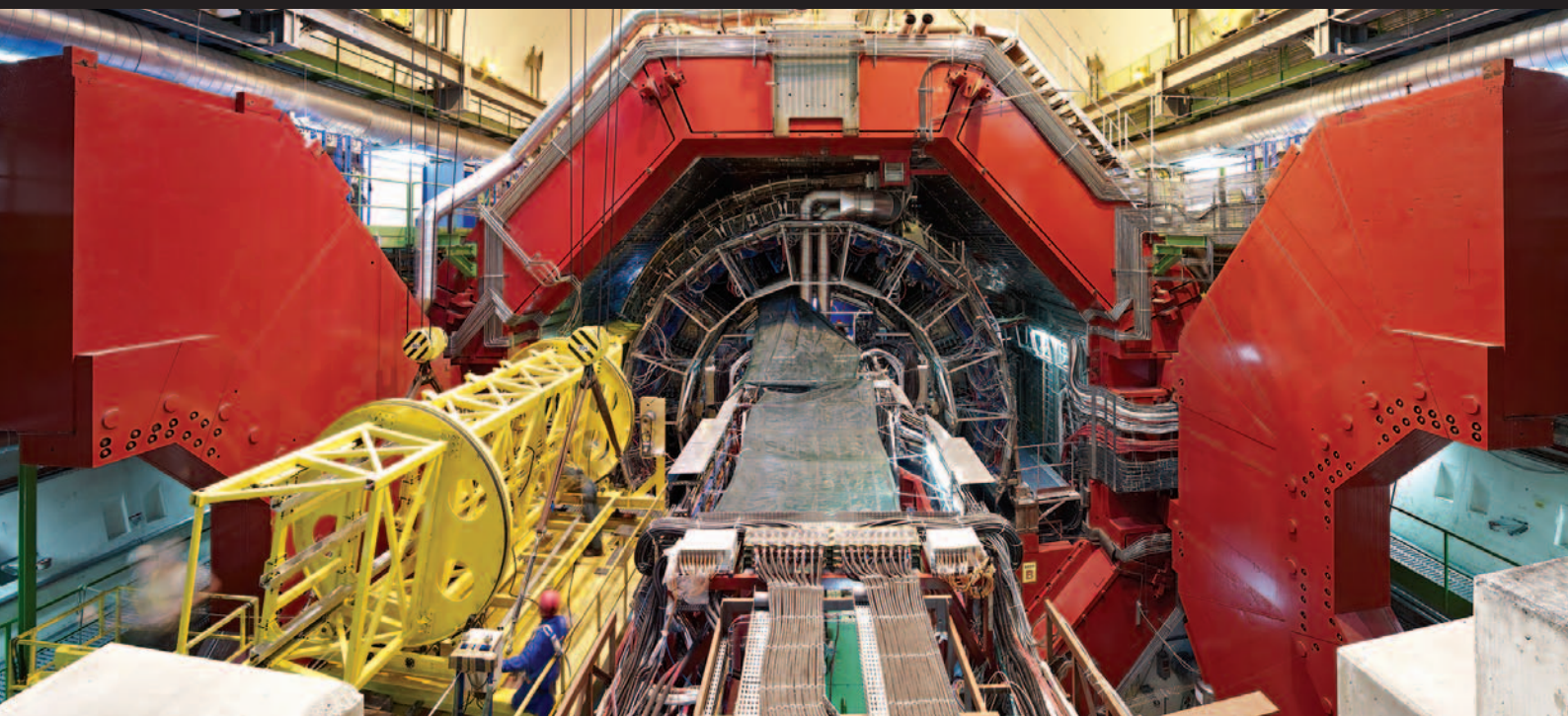
«Крайне важно, что Linux позволяет центрам HEP снизить расходы.»

около 90 000 физических CPU на 345 893 логических ядрах, соединенных с WLCG. Общая емкость хранения составляет 300 ПБ. В Великобритании национальный проект распределенных вычислений WLCG известен как GridPP, и управляет им из Университета Оксфорда Пит Гронбах [Pete Gronbach]. Согласно карте сети – www.gridpp.ac.uk – с GridPP соединено более 37 000 логических ядер CPU, и все это работает как единый ресурс для ученых, занимающихся аналитикой на основе данных CERN.

«Мы постоянно проводим мониторинг системы, убеждаясь, что наши сервисы работают нормально, – объясняет Гронбах. – Если вы потерпели неудачу в каком-либо тесте, это отображается на контрольной панели, и ответственные за мониторинг наблюдают за такими вещами и присваивают таким центрам талон неисправности. Есть протокол, согласно которому центр, чтобы быть частью сети, обязан обеспечивать определенный уровень услуг, и неисправность должна быть устранена за оговоренное время. Все делается на очень профессиональном уровне – возможно, вы не ждали такого от университета. Но это не та вещь, с которой мы можем шутить».

Одно из требований для центра T2 – наличие работника с полной занятостью, чья обязанность – поддерживать ресурсы сети, и еженедельно проводятся видео- или аудиоконференции между сисадминами WLCG, чтобы обеспечить корректную работу всей сети и своевременное обновление ее ресурсов.

Крайне важно, что Linux позволяет центрам HEP снизить расходы, поскольку можно использовать более или менее общие компоненты во всей сети обработки и хранения данных. CERN – огромное общественное вложение, более миллиарда евро ежегодно, и он должен уметь считать деньги. Это также означает, что CERN может поддерживать ПО с открытым кодом, типа *Disk Pool Manager (DPM)*, применяемое для поддержки кластеров хранения. Как и следует ожидать от организации, подарившей человечеству Всемирную паутину, она отлично осознает все преимущества совместной разработки.



«GridPP существует более 10 лет, – говорит Гронбах, – но многие пакетные системы остались теми же: мы используем TORQUE и MAUI, на базе PBS. Один-два центра используют Grid Engine, но для него поддержка не распространена. Такие вещи практически неизменны долгие годы. Другие части ПО, такие, как Computing Element (это элемент, сидящий между входящими задачами пользователя и их отправкой в пакетную систему обработки), уже прошли через множество поколений, и мы обновляем их примерно раз в полгода».

По словам Калафиуры, многоядерная обработка данных стала самым значимым улучшением сети распределенных вычислений.

«Благодаря многоядерной обработке, – говорит он, – мы смогли использовать специфическую хитрость Linux – ответить, скопировать и записать – для запуска восьми или 16 копий одного и того же приложения».

Сейчас основное обновление, происходящее в GridPP – установка Scientific Linux 6 (SL6) и свежей версии Lustre для хранения файловых систем. SL совместно поддерживается CERN и американской лабораторией Fermilab. На момент писания нашей статьи SL6 работает на 99365 компьютерах – это ниже нормы, видимо, из-за летних каникул в большинстве стран северного полушария. <https://www.scientificlinux.org/about/stats/2012/arch.html>

SL6 – ответвление Red Hat Enterprise Linux, выбранное за его невероятную стабильность. Не все исследователи применяют SL, и все инструменты, жизненно важные для работы

Генератор ускорителя

Вопрос о том, сколько на самом деле данных генерируется БАК, может отчасти ввести в заблуждение, и на него трудно дать точный ответ. В той же бесхитростной манере, что и частицы, за которыми ведется наблюдение, большая их часть исчезает в считанные секунды, когда происходит событие.

В команде ATLAS одной из главных ролей д-ра Сальваторе является участие в группе, работающей над «триггерной» программой для эксперимента.

«Эксперимент состоит из двух пучков протонов, которые сталкиваются каждые 25 наносекунд, и из этого взаимодействия что-то должно получиться, – говорит Сальваторе. – И надо решить, представляет это что-то интерес или нет. Все, что у нас есть – это 40-МГц взаимодействия, но у нас нет возможности хранить 40 МГц данных – мы сохраняем порядка 400 Гц. Частота выборки не перево-

дится непосредственно в байты, на принцип заключается в том, что изначальный набор данных может уменьшиться по объему в тысячу раз».

«Есть три уровня программ, – продолжает Сальваторе, – и первый должен работать в несколько первых микросекунд и вплоть до третьего уровня, который решает, представляет все событие интерес или нет. Задача «Триггера» – быстро провести приблизительный анализ данных и убедиться, что сохраняемые данные представляют интерес, а не являются просто еще одним столкновением. То, что не выбирается из 40 МГц, оказывается потерянным. Перед началом эксперимента мы проводили длительное моделирование по методу Монте-Карло, чтобы посмотреть, что может случиться во время события, и иметь возможность настроить триггер и извлечь из данных все по максимуму».

CERN, совместимы с другими дистрибутивами – по-прежнему популярен Ubuntu. Но на всех серверах, соединенных с WLCG, версия SL5 или выше. «Что порадовало за последние 10 лет – это автоматическая установка и администрирование этих систем, – продолжает Гронбах. – Мы можем устанавливать системы, используя Pixiboot, Kickstart, CF Engine или Puppet для быстрой или удаленной переустановки узла, потому что компьютерные центры часто располагаются вне университета».

Унификация оборудования и отлаженное FOSS также облегчает многим центрам выход

онлайн и возможность предложить свои ресурсы сети. В июне Сассекс стал центром T2. Университет присоединился к проекту ATLAS в 2009 году, когда д-р Сальваторе и д-р Антонелла Де Санто [Antonella De Santo] вошли в проект от Университета Роял Холлоуэй [Royal Holloway], еще одного центра T2, создав группу ATLAS. Благодаря их работе Сассекс получил финансирование, так что д-р Сальваторе, д-р Де Санто и кафедра ИТ смогли создать центр обработки данных на 12 стоек, со 100 CPU и 150 ТБ хранимых данных, в зале с естественным охлаждением, затратив 80000 фунтов.

Углубимся в детали

Помимо управления пакетной обработкой данных для тысяч результатов, ежедневно поступающих в WLCG HER, процесс научного исследования экспериментальных данных также требует подхода, основанного на открытом коде. Все участники эксперимента отвечают за постоянное улучшение программы в плане ее способности выделить значительные события среди миллиардов частиц.

Нет ничего невозможного в том, чтобы быть физиком и при этом не разбираться C++ и не знать, как оптимизировать код с помощью компилятора, но это входит в программу обучения в Сассексе и других учреждениях HER.

«Наше сообщество было сообществом открытого кода задолго до того, как оно получило это название, – говорит Берд. – Мы используем коммерческое ПО там, где это нужно и можно, но все программы для анализа наших собственных данных написаны у нас же, пото-

му что для выполнения такой работы больше ничего не существует».

Величина и сложность ATLAS и CMS означает наличие сотен профессионалов в области ПО, работающих над созданием и под-

держкой кода, но, поскольку разные центры занимаются разными частями эксперимента, такие инструменты, как SVN, жизненно важны для управления работой над каждым алгоритмом. «Вы не смогли бы сделать этого в другой



► Вот что видят ученые: сверху – рабочий режим БАК, ниже – интенсивность пучка, затем – позиции TED и TDI и экраны регистрации луча. Для ученых во всем этом есть глубокий смысл.

операционной системе, – говорит д-р Сальваторе. – С инструментами вроде SVN проще работать изнутри ОС, и мы это определенно эксплуатируем, как можно глубже проникая в ОС; в проприетарных же системах ОС скрыта за несколькими уровнями».

Однако управление результатами работы тысяч академиков HEP – это задача, требующая не просто управления версиями. «Здесь присутствует и некая доля социологии, – говорит Калафиура. – При наличии 3000 участников нужны особые процедуры, чтобы выяснить, кто является членом ATLAS, а кто подписывает документ о Хиггсе. Существует система кредитов. Традиционно это либо работа с детектором, либо разработка программ. Если у кого-то есть хороший алгоритм для отслеживания движения частицы по детектору, они тратят несколько лет на совершенствование этого алгоритма и передают его в общий программный репозиторий, и он становится частью официальных инструкций. До 2010 года большая часть вклада в развитие ПО была сделана физиками, и только в области ядра трудились профессионалы. А затем все изменил тот момент, когда начали поступать реальные данные. В течение нескольких последних лет физики используют инструменты,

подготовленные куда меньшим сообществом. Я занимаюсь физикой высоких энергий большую часть своей жизни, и каждый раз, когда эксперимент начинает предоставлять данные, большая часть сообщества сразу теряет всякий интерес к техническому аспекту».

Зачастую исследования начинаются с того, что физики разрабатывают теоретическую базу для расчета результатов события, в котором появится бозон Хиггса. Как только ученый предлагает определенный сценарий, начинаются попытки смоделировать его, используя *Monte Carlo*, программу, работающую на осно-

ве сети распределенных вычислений, которая пытается спрогнозировать, что произойдет при столкновении протонов. Результаты сравниваются с базой данных реальных результатов, чтобы выяснить, кто будет проводить дальнейшие исследования. Событие, в котором выявляется вероятность появления частицы Хиггса, появляется из набора данных, состоящего примерно из квадриллиона столкновений. Есть группы ученых и разработчиков ПО, связанные с каждой частью «основных» компьютерных ресурсов, которые управляют поведением БАК и основным отбором данных.



➤ Это не съемки web-камерой знаменитого пятничного турнира CERN по Quake. Это установка рекорда энергии для БАК в ноябре 2009 года – двойной пучок достиг показателя 1,18 ТэВ.

Будущее

Поскольку в конце года БАК будет отключен на модернизацию, естественно, что в высших эшелонах проекта вычислений шло немало дебатов по поводу того, чем будет заниматься WLCG, чтобы соответствовать возросшим требованиям экспериментов в области физики высоких энергий.

«Когда мы начали всю эту работу, – говорит Берд, – нас беспокоила сеть, получение достаточной скорости передачи и надежность всего этого. Все получилось просто фантастическим и превзошло все ожидания. Поэтому сегодня вопрос звучит так: «Как нам использовать сеть еще лучше?» Мы ищем более разумные способы использования сети, кэширования и передачи данных, когда того потребует работа, вместо попыток выяснить, где они потребуются».

Неизбежно возникает тема облачных вычислений. Нечто вроде EC2, например, непрак-

тично для повседневного использования, из-за высокой стоимости передачи данных с и на серверы Amazon. Однако он может пригодиться для добавочных вычислений во время проведения основных конференций, поскольку на них всегда есть спрос, когда ученые работают над завершением своих статей и уточнением результатов. Большинство кластеров сети предназначены для 90 % использования своих ресурсов при обычных обстоятельствах, и в них недостаточно свободных циклов, которые можно было бы задействовать.

Запуск сети распределенных вычислений

Ян Берд, руководитель проекта CERN WLCG, рассказывает о появлении сети:

«В конце 90-х существовали группы людей, которые собирались и пытались разобраться с тем, какого масштаба вычисления нужны БАК. Были предложены некоторые модели, благодаря которым мы осознали, что у CERN нет собственных возможностей для обеспечения достаточного объема ресурсов для требуемого уровня вычислений и хранения данных. И было создано несколько моделей распределенных вычислений и их работы.

Как раз в тот момент, когда разрабатывались эти модели – около 2000/2001 – появилась эта штука под названием «grid-вычисления», которая вышла из сообщества компьютерных дисциплин, и оказалась неплохим способом реализации модели рас-

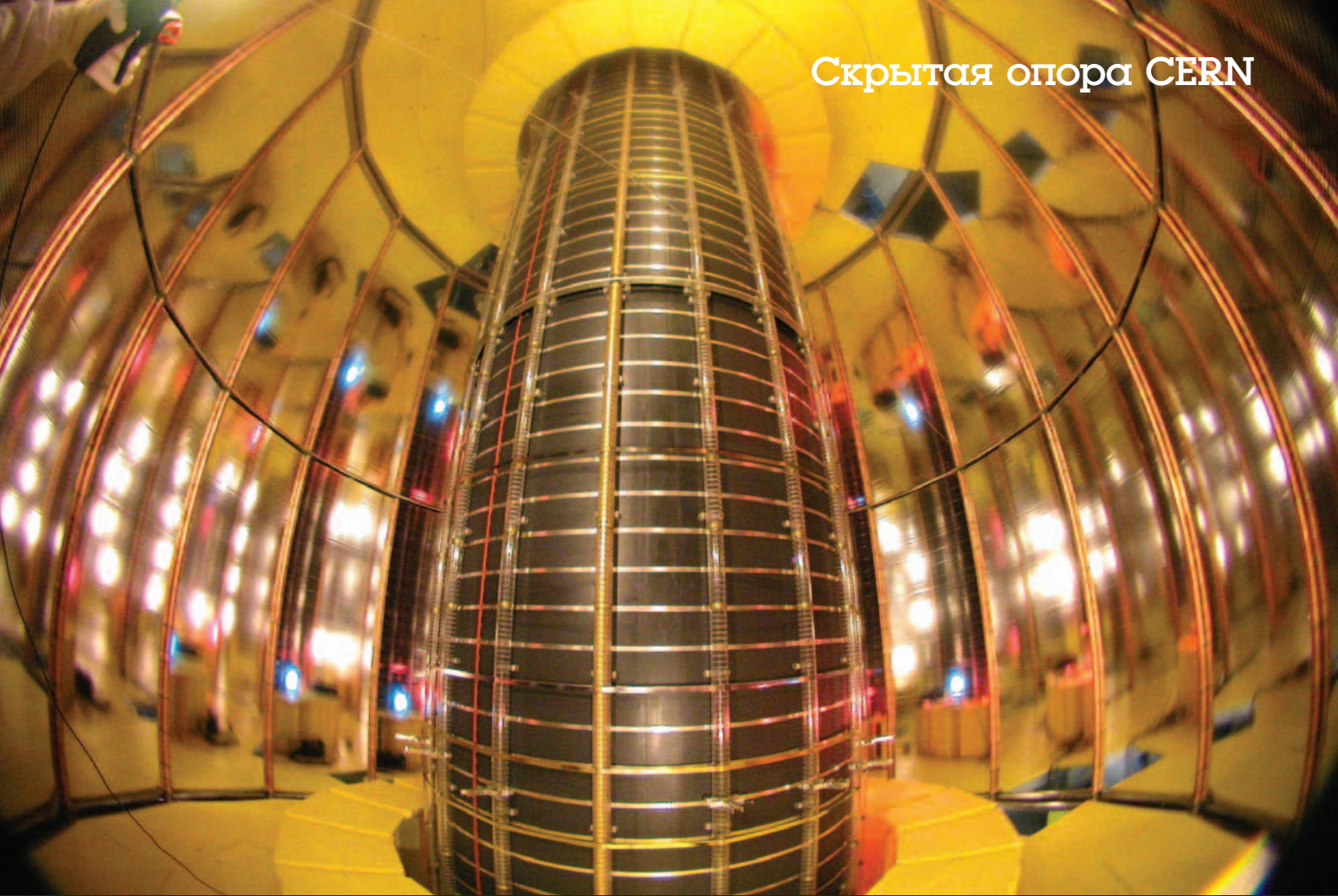
пределенных вычислений, которую предложили мы. Несколько лет мы занимались созданием прототипа, чтобы выяснить, сумеем ли мы добиться таких вычислений, которые будут делать то, что нам требуется».

Это все переросло в проект LCG, который формально появился в 2002 году; примерно в то же время я приступил к работе в CERN, а за этим последовали семь лет разработок и отладки технологии.

Работая с этими технологиями в требуемом для нашей деятельности масштабе, не имеешь права на неудачу, потому что на них было затрачено невероятное количество времени и усилий. Они должны быть надежными и достаточно масштабыми, чтобы соответствовать уровню всех проблем производственного уровня реального мира – и это

на самом деле раздвигало границы того, что на тот момент могло делать ПО. В основном мы начинали с нуля, и у нас ушло на это десять лет. Но использовали мы его «по-настоящему» с самого начала, чтобы проводить моделирование и т.п., и это отлично выучило нас создавать надежные программы.

Я не сказал бы, что у нас были проблемы с тем, чтобы убедить спонсоров в необходимости финансовой поддержки, но, я думаю, они с подозрением отнеслись к такому крупному проекту в области компьютерных технологий, равного которому еще не было. Поэтому нам пришлось проходить через такие же проверки, через которые проходят эксперименты – обычно в области компьютерных технологий такого не случается, но с этим проектом все было именно так».



«С использованием коммерческих облаков есть проблемы – оно обходится значительно дороже, чем мы можем сделать сами, – говорит Берд. – С другой стороны, имеются такие технологии, как OpenStack, которые крайне интересны. Сейчас мы используем пилотный кластер OpenStack, который позволяет нам предлагать разные виды сервисов и дает возможность применять другой способ соединения

центров обработки данных – это не совсем то, что нам нужно, но мы стараемся идти в ногу с технологиями, чтобы не застаиваться, не вязнуть в болоте, занимаясь тем, чем больше никто не занимается».

«Работа CERN будет продолжаться, становясь все требовательнее.»

на порции и отправить каждую из этих порций на отдельный CPU, и ему незачем знать, как решаются другие задачи. Но сама по себе эта задача не параллельная; она не очень хорошо вписывается в обработку данных типа GPU».

Столь же горячей темой для обсуждения среди разработчиков ATLAS является GPGPU. «Некоторые академики, например, в Глазго, рассматривают возможность портирования своего кода в GPGPU, – говорит Гронбах из GridPP, – но вопрос в том, соответствует ли тип анализа, который они делают, данному типу обработки данных. Большая часть нашего кода является тем, что известно под названием

“усложненного параллелизма”, мы можем нарезать события и данные

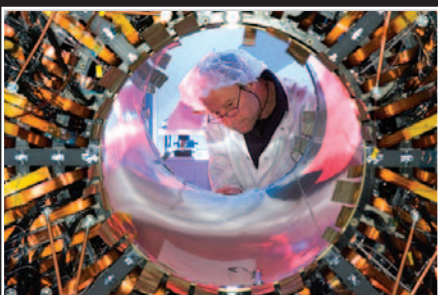
По словам Берда, большой интерес вызывает ожидаемое оборудование Intel Larabee/Xeon Phi. И хотя по поводу развития самой крупной в мире сети распределенных вычислений ведутся оживленные дебаты, можно со всей определенностью сказать, что работа CERN будет продолжаться, и будет становиться все более требовательной каждый год. Существование бозона Хиггса пока не подтверждено, цель Большого адронного коллайдера пока не достигнута и не является единственной задачей, над которой он работает. Это всего лишь

первый крупный анонс множества – будем надеяться – грядущих событий.

Поэтому в следующий раз, любясь звездами и предаваясь философским размышлениям о том, как же все это началось, помните, что все наши недавно полученные знания о жизни, Вселенной и обо всем остальном – это результат труда тысяч ученых и некоего замечательного пингвина. LXF



› Прототипы суперпроводящих магнитов для БАК во время испытания на испытательном полигоне String 2.



› Полупроводниковые цилиндрические трекары для ATLAS монтировались и проходили испытания в Оксфорде.

Открытое правительство

Что общего между ФБР, погодой (по крайней мере, в Америке) и данными по безработице в США? Нам помог разобраться Гуннар Хеллексон.



Послужной список Гуннара Хеллексона [Gunnar Hellekson] необычайно внушительен. Он главный технический стратег Red

Hat по работе с госсектором США, где в его задачи входит продвижение Open Source в правительственных структурах и взаимодействие с производителями систем по реализации подобных заказов. Также он является сопредседателем Open Source for America, организации, выступающей за то, чтобы все ПО, разработанное на деньги налогоплательщиков, стало открытым и доступным любому гражданину. Помимо этого, он состоит членом правлений Military Open Source Working Group, Civic Commons и SIA Software Division. Наконец, это умный малый, вхожий к влиятельным людям, и мы не преминули упомянуть, что один из наших прежних авторов, Джон, нашел отличную работу благодаря Red Hat.

LXF: Во-первых, поддерживает ли правительство США открытое ПО — или считает, что оно отбирает хлеб у Microsoft?

ГХ: Вообще-то там давно используют открытое ПО, еще с 1978. Правительство США финансировало, в том числе, разработку стека BSD TCP/IP; функция эхо-тестирования создавалась в военной исследовательской лаборатории, а где-то годах в 90-х люди заинтересовались и стали присматриваться к Open Source; акт Clinger-Cohen, принятый в 1996 и формулирующий правила развития ИТ в государственных структурах также был инициативой правительства. Вдруг появились правила, а значит,

и необходимость следить за их соблюдением. Пока правил не было, целесообразность Open Source не вызвала сомнений. Когда же они появились, люди стали задумываться: «Стоп, а законно ли это открытое ПО?». Только в 2003-2004 году Министерство обороны, совместно с Административным и бюджетным управлением США, объявило: «Да, открытое ПО законно. Как и любая коммерческая лицензия, — можете не беспокоиться».

Ирония в том, что пока неповоротливые политические механизмы приходили в действие, Министерство энергетики, Минобороны и Управление национальной безопасности открыто предоставляли свой исходный код, совершенно беспрепятственно. Так что говорить о сформированной позиции правительства

О SELINUX

«Выдача SELinux в общий доступ укрепила безопасность страны.»

по поводу открытого ПО не просто неточно, но и невозможно, ведь речь идет о 12 миллионах человек. Кто-то из них — большой сторонник открытого ПО, кто-то — наоборот.

LXF: SELinux, ныне входящий в состав ядра, тоже ведь разработали в Управлении национальной безопасности?

ГХ: Security Enhanced Linux, в 2001 г. И их мотивация ясна, правда? Причин несколько. Во-первых, они хотели освободить себя от необходимости разрабатывать технологию своими

силами. Если бы они сделали ее сами и держали при себе, им бы пришлось самим и заботиться о поддержке, а это дорого. Так что, доверив ее открытому сообществу и добавив в ядро Linux, они получили помощь, что было им на руку.

Еще одна важная сторона этой миссии, о которой многие забывают, состоит в том, что Агентство национальной безопасности также отвечает за сохранность информации внутри страны и за то, чтобы коммерческие продукты были более защищенными. Так что выдача SELinux в общий доступ укрепила безопасность всей страны. В общем, причин предостаточно.

LXF: В чем заключается ваша роль в Red Hat?

Пытаетесь продвигать их продукты в разных правительственных организациях?

ГХ: В том числе и это. Точнее можно сказать так: моя задача — держать правительство в курсе происходящего в мире Open Source, и доводить его инициативы до открытых сообществ.

LXF: Ясно. Нечто вроде специалиста по работе с сообществом из 12 миллионов человек, работающих в правительстве.

ГХ: Скорее ведущего переговоры об освобождении заложников.

LXF: Знакомы ли вы с текущими проектами в Великобритании? С NHS IT, например. Это громадный проект — и черная дыра для денег налогоплательщиков. Как думаете, не в том ли дело, что в правительствах привыкли мыслить глобально?

ГХ: Так, это становится интересным, потому что в США мы как раз стремимся от этого

уйти. Федеральный IT-директор объявил об отмене крупных закупок, поэтому вместо одного контракта на \$500 млн мы заключаем 100 контрактов по 5 млн. И не без причины, не только потому, что это более эффективно и менее рискованно, но и потому, что нынешняя система финансирования отстает от современных темпов развития технологий. В Министерстве обороны при подготовке программы высокого уровня дается 48 месяцев на то, чтобы от инициализации дойти до выработки требований. То есть вы еще не определились с ценовым предложением, не объявили тендер, зато потратили 4 года, разрабатывая требования. А за эти 4 года изменился весь мир! Запускать подобные IT-проекты просто нецелесообразно.

Итак, в 2012 году, согласно законопроекту об ассигнованиях для МО, Конгресс предписал определить новые закупочные стратегии, специально для информационных технологий, чтобы решить эту проблему. В частности, речь шла о постоянном взаимодействии с пользо-

вателем, итеративном, эволюционном подходе... что формирует желаемый образ подвижного IT-проекта. Именно эта модель была впоследствии реализована в правительстве — с более итеративным подходом и проектами, поделенными на малые фрагменты.

LXF: Один из наших министров уже высказывал мнение, что именно так и следует дробить проекты, но дальше слов дело пока не пошло.

ГХ: Ну, насколько мне известно, с начала 2000-х правительство Великобритании заключило ряд очень крупных и долгосрочных контрактов, соглашений на 10 лет, например. Речь идет, в частности, о Минобороны, попросту отдавшем большую часть своей IT-структуры на откуп консорциуму из пяти компаний... как же он назывался? Fijistu Siemens, кажется, так. И вы получаете то, за что платили, верно? Помимо огромных стартовых вложений, еще и то, что правительство начисто лишилось права голоса, ведь любая их инициатива теперь выльется в необходимость новых выплат то-

му же консорциуму. Именно этому противостоит подвижная IT-сфера: она стимулирует не только постоянное развитие, но и конкуренцию, на каждом этапе.

LXF: Не кажется ли вам, что это подвижное и более адаптивное развитие... не думаете ли вы, что это и есть главное преимущество использования Open Source в крупных правительственных проектах, по сравнению с теми же открытыми форматами файлов?

ГХ: Конечно, ведь что важнее? Открытый код или открытые стандарты? По-моему, и те, и другие направлены на решение разных проблем. Создавая открытый стандарт, вы создаете рынок. Вы даете многим людям возможность выполнять одну и ту же задачу. Допустим, я использую стандарт IMAP для электронной почты и имею возможность обращаться к любым серверам IMAP, при этом не меняя клиент, ведь с IMAP я могу сделать так, чтобы серверы конкурировали друг с другом, что снизит мои затраты. А имея открытый

»

код, я становлюсь поставщиком в первой инстанции, ну или последней инстанции, и выбор всегда за мной. И даже если у вас есть открытый стандарт, но используется он только в вашей компании, вы все равно ограничены. А Open Source предоставляет свободу. Можно использовать код сам по себе, можно нанять умного компьютерщика, который будет его поддерживать.

LXF: А как насчет того аргумента, что свободное ПО вытесняет коммерческое и лишает компании заработка?

ГХ: В правительстве это обсуждалось... Акт Clinger-Cohen, принятый в 90х, сформулировал приоритет использования коммерческого ПО в госструктурах, имея в виду, что оно должно разрабатываться вовне. Дабы гарантировать, что правительству не придется своими силами делать то, что гораздо эффективнее решается частным сектором.

Так что правила гласят: прежде чем нечто создавать, проверь, не создал ли это кто-то до тебя. И когда в правительстве начинают писать ПО, это автоматически воспринимается как дублирование. Но, безусловно, в некоторых случаях это совершенно оправдано. *Assistiva* – хороший этому пример, поскольку в нем есть функции, которых прежде не было нигде. То, как там реализованы алгоритмы, как организовано хранение документов, возможность доступа к данным на уровне пользователя, уникальные функции проекта *Assistiva*, и правительство поступило совершенно верно, сделав этот код открытым. А раз это открытый код – значит, это коммерческий продукт, поскольку он выпускается под коммерческой лицензией, лицензией Apache.

Именно так он и существует. Беспокойство по поводу вытеснения тех или иных продуктов оправдано, но по факту речь идет о программах, которые правительству уже не принадлежат.

LXF: Да и на самом деле, вы скорее не вытеснили частное предприятие, а создали рынок услуг поддержки.

ГХ: Есть такая компания, SURFL, которая несколько месяцев назад заключила первый этап финансирования, чтобы стать Red Hat для *Assistiva*. Вот и живой пример технологического обмена с правительством. И я считаю, что ситуация вытеснения имеет место, но нужно различать программы, принадлежащие правительству, и программы, созданные им же, но являющиеся открытым кодом; это разные вещи.

LXF: Еще одним преимуществом Open Source традиционно считается более низкая цена, потому что не нужно платить за лицензию. Это важный фактор, или на уровне правительства он не имеет значения? Ведь я могу представить, во сколько обойдется нанять достаточное число разработчиков.

ГХ: Написание ПО стоит дорого, поддержка – еще дороже. Но код мы потратили уже очень много.

Цена, зачастую, важный аргумент в пользу Open Source, во всех упомянутых вами отношениях. Оно часто дешевле. Но я всегда предостерегаю людей от утверждения, что это всегда так, ведь хотя процесс разработки открытого ПО и имеет некоторые преимущества, содержание проекта может стоить очень дорого. Или продвижение на IT-рынок. Так что, если речь об экономии денег, важнее всего знать, на что именно вы хотите их потратить. Объем затрат на открытый проект определяется его спецификой.

Как уже было сказано, Open Source имеет ряд свойств второго порядка, являющихся его преимуществами. Как то: вы всегда можете конкурировать за право поддержки; всегда можете исправить ошибки; и, главное, только в этом случае вы обладаете всем спектром инновационных возможностей.

LXF: Которых у вас нет, если вы связаны 10-летним контрактом с CarGemini.

ГХ: Мы смотрим на это так: возьмите своего любимого поставщика ПО и обведите кружком его раз-

дело с таким ПО и поставщиками ПО, на которых работает максимум специалистов.

LXF: Как идут дела у Red Hat?

ГХ: Да замечательно, вы же в курсе сегодняшних рейтингов в прессе. Я слышал про третье место, про 12-е место, но мы одни из немногих компаний-разработчиков с годовым доходом более \$1 млрд.

LXF: Ну да, в этом вопросе была доля шутки.

Пару номеров назад у нас была заглавная статья под названием «Как заработать миллиард долларов способом Red Hat».

ГХ: О да, я ее видел!

LXF: Надеюсь, ваши специалисты по авторскому праву не возмутились, что мы использовали логотип без разрешения. И еще одна реклама Red Hat: вы, ребята, оплатили одному из наших авторов курс обучения, со сдачей экзамена на сертификат Red Hat. Пройдя курс, он через пару недель нашел новую работу и отправился покорять новые высоты. Нам уже было не по карману его удержать.

ГХ: Это презабавно. Для нас.

LXF: Ну, мы и сами за него рады.

ГХ: Для меня самое замечательное в Red Hat – это то, что мне нравится там работать, вот уже семь лет. Никогда и ничего меня не увлекло больше, чем OpenShift. Знаю, вам это трудно понять, но открытая платформа для сервера поможет моим клиентам в корне пересмотреть свои стратегии вложений в ПО. Дать им возможность полностью контролировать развитие своей технологии, с этой точки зрения, значит рассматривать платформу как сервер. Я планирую серию постов об этом в своем блоге. Это большое дело, и очень увлекательное.

ОБ ОТКРЫТОМ И ПРОПРИЕТАРНОМ КОДЕ
«Надежнее — ПО, на которое работает максимум специалистов.»

работчиков. Где в мире будет больше одаренных людей: внутри кружка или вне его? И так с любой организацией. Если вы сотрудничаете с проприетарным поставщиком, качество вашего ПО определяется тем, сколько классных специалистов они могут нанять, и это, по-моему, не лучшее соотношение, с учетом рисков. Гораздо надежнее иметь



LXF: Это как-то связано с OpenStack?

ГХ: OpenStack – это основа. Он предоставит вам виртуальную машину; а OpenShift позволит вам сказать: «Мне нужна среда Python, среда PHP и еще WordPress поверх». И затем он автоматически создаст то, что мы называем картриджами: для Mongo, PHP, Perl, Ruby, Java... где все конструктивные блоки уже скомпонованы и защищены. И вы даже не следите за самой виртуализацией, даже не думаете, где все это запускается. Как разработчик, единственное, с чем вы работаете, это Git. Вы пишете код, кидаете его в Git, и он запускается в среде. Это действительно круто.

LXF: Прямо как в той новой штуковине Ubuntu, Juju, облачном сервисе, который они запустили ранее в этом году.

ГХ: Два дня назад я был на совещании, где обсуждали Juju, и все еще пытаюсь сообразить, что же это такое. Juju решает схожую проблему, но с другого конца. В OpenShift все по-другому. Juju создан на благо сисадминов; а OpenShift позволяет создавать Linux-контейнеры и запускать приложения внутри них. Каждый из них изолирован, как тюремная камера, и никто ни с кем не взаимодействует. У нас он работал на EC2: запускал виртуальные машины, затем контейнеры внутри них, и в итоге, на одной системе могло работать до 400 человек. OpenShift позволяет распределить контроль.

А деньги мы зарабатываем тем, что 3 картриджа вы можете завести бесплатно, но если вам нужно больше, если вам нужны инструменты управления и прочее, вы регистрируетесь и платите за дополнительное место. Этим интересуется ВМФ, авиация...

LXF: А вы знаете, что Linux используется в одном из телеуправляемых кораблей в США?

ГХ: Ага. На Firescout. На самом деле, это проект нас повеселил. Если восстановить хронологию использования Open Source в правительстве, вы увидите, что первые точки на этой линии появились еще в 1978, и вплоть до 2012 года было множество направлений, требующих содействия и внедрения Open Source. И потом вдруг заключается 26-миллионный контракт на установку Linux на Firescout, что не так уж существенно, правда? На том же совещании был вопрос: «Когда же, наконец, прекратится обсуждение Open Source в правительстве?» На что я ответил: «Каждое такое обсуждение ставит нас в проигрышное положение». Мы хотим, чтобы Open Source не привлекал внимание. Он должен быть частью инфраструктуры. И если кого-то удивляет, что Linux используется в правительстве... так он, знаете ли, давно используется. Правительство вкладывается в разработку ядра Linux по крайней мере года этак с 2000. И смешно слышать, как кто-то

говорит: «О, а на Firescout стоит Linux». Как будто это перевернет все в GPL. Нет! Это не тактизм – это контракт.

LXF: А есть какие-то особо значимые примеры использования RedHat в правительстве США?

ГХ: Конечно. Федеральное агентство авиации,

же пользователь Red Hat. Каждую неделю правительство публикует данные по занятости. Даже пятиминутная задержка может привести к коллапсу на Уолл-стрит. И они используют Linux. Бюро регистрации патентов и товарных знаков, что еще... национальная метеорологическая служба, так что любой прогноз погоды берется из системы на Linux. О, еще же ФБР, и это довольно забавно. Все данные по отпечаткам и биографии, все это хранится в машинах на... Да у них, по-моему, задействованы все продукты Red Hat. Это 16 миллионов записей, и каждая проверка сведений, каждый отпечаток, вся биометрия... они же сняли с вас отпечатки пальцев на границе, так? Ну вот, эти данные были отправлены в центр, где все это хранится, и в течение 15 секунд вернули ответ, что вы чисты. И все это Red Hat. **LXF**

**ОБ ОТКРЫТОМ КОДЕ В ПРАВИТЕЛЬСТВЕ
«Правительство вкладывается в разработку ядра Linux года этак с 2000.»**

их система управления трафиком основана на Linux, с 2001 года. Так что каждый раз, когда вы совершаете перелет в США, этим управляет наша система. Бюро переписи в Штатах то-



Появился новый рабочий стол, дружелюбный к ресурсам памяти, и это дает пищу для размышлений команде KDE. Майк Сондерс все объяснит...

Razor-qt:

Легкая альтернатива KDE

Поклонники Gnome недавно вздохнули с облегчением. «Что?! — воскликнете вы. — Да у пользователей Gnome тяжелые времена! Поклонников старого рабочего стола поверг в уныние капитальный пересмотр интерфейса, а сторонников Gnome 3 постоянно ругают».

Все это правда. Но, видите ли, пользователи Gnome, которым очень нравился старый дизайн 2.x и которые терпеть не могли новый, получили шанс обходного маневра после выхода Gnome 3: они могли перейти на Xfce. Там их ожидал зрелый полнофункциональный рабочий стол на базе Gtk, подобный тому Gnome 2, который они так хорошо знали и любили; правда, с рядом отличий. Либо они могли попробовать LXDE, другой рабочий стол на основе Gtk. Так что, как видите, (невероятно/волшебнo/крайне — ненужное удалить) смешные решения команды Gnome не привели к концу света.

Обратимся к поклонникам KDE, которые ощущали себя вполне комфортно в серии 3.x.

Когда появилась серия 4.x, которая выбросила большую часть кодовой базы и реализовала некоторые основные функции, этим пользователям KDE было совершенно некуда деться. У них не было ни аварийного выхода в виде Xfce, ни похожей среды рабочего стола на базе Qt, в которой они могли бы работать со своими любимыми приложениями Qt и KDE, не используя самого рабочего стола (конечно, поч-

«Он имеет много функций и достаточно солиден как постоянный рабочий стол.»

ти любую программу Linux можно запустить на любом рабочем столе или в любом менеджере окон, но суть не в этом. Просто приложения Gtk лучше всего смотрятся и работают в среде Gtk/Gnome, а приложения Qt чувствуют себя наиболее комфортно в Qt/KDE). И так, в линейке рабочих столов Linux образовалась огромная дыра для тех, кто любит такие при-

ложения Qt высшего уровня, как Scribus и VLC, но кому при этом нужно нечто более легкое и простое, чем ужасающий монстр, в который превратился KDE.

Загляните в Razor-qt (<http://razor-qt.org/>). Он описывает сам себя как «продвинутую, простую в использовании и быструю среду рабочего стола на технологиях Qt». Razor-qt пока находится на стадии разработки, и его самый последний релиз на момент написания этой статьи был 0.4.1. Несмотря на это, он уже имеет много функций и достаточно солиден, чтобы сгодиться как постоянный рабочий стол. Лучшее в нем то, что он намного меньше грузит ОЗУ,

чем KDE. Посмотрите на данные по использованию памяти в Kubuntu 11.10 после загрузки:

С загрузкой KDE: 315 МБ

С загрузкой Razor-qt: 126 МБ

То есть, просто переключившись на другой рабочий стол, вы экономите почти 200 МБ ОЗУ. Razor-qt к тому же значительно проще в работе, чем KDE.

Установка рабочего стола

Два рабочих стола с поразительно несхожим происхождением.

Поскольку Razor-qt – довольно новый проект, многие дистрибутивы пока что не включили его в свои репозитории. И все же лучше сначала заглянуть туда, потому что использование родных для вашего дистрибутива пакетов всегда является самым простым способом сделать что-то. Как и другие среды рабочего стола, Razor-qt состоит из множества пакетов, но должен быть один базовый или мета-пакет **razor-qt**, который присоединяет к себе остальные.

Если вы используете Ubuntu, можно установить рабочий стол через PPA – Personal Package Archive, который отделен от официальных репозиториях, но остается отличным источником проверенных и надежных пакетов. Просто здорово, что там есть пакеты для многих более ранних версий Ubuntu, до версии 10.04. Так что если у вас старый компьютер с винтажной версией Ubuntu 2010 года и вы хотите его разогнать, его легко оснастить Razor-qt.

Чтобы установить его на Ubuntu, введите следующие команды в окне терминала:

```
sudo add-apt-repository ppa:razor-qt
sudo apt-get update
sudo apt-get install razorqt
```

Первая сообщит Ubuntu, что мы хотим использовать пакеты из собственного репозитория Razor-qt. Затем мы обновляем список локально хранимых пакетов и загружаем файлы Razor-qt. Если вы используете Kubuntu, т.е. у вас уже установлено большинство зависимостей, таких, как Qt, то загрузить надо около 15 MB.

Если ваш дистрибутив не имеет отношения к Qt, как, например, просто Ubuntu, вам предложат загрузить дополнительные зависимости.

А если вы используете Fedora, CentOS или OpenSUSE, вы можете найти репозитории здесь: <http://download.opensuse.org/repositories/X11:/Qt/Desktop>. Более подробные инструкции и информацию по другим дистрибутивам имеются на <http://razor-qt.org/>. Если вам не повезло, и Razor-qt не заработал на выбранном вами дистрибутиве, можете попробовать Ubuntu Razor-qt Remix (<http://www.emanueleincremona.biz/index.php/U-R-R/>).

Это версия Ubuntu с высокой степенью индивидуальной настройки и с Razor-qt в качестве рабочего стола по умолчанию.

И, разумеется, есть возможность компиляции из исходника, в качестве исключительно крайнего средства (или большого удовольствия, если вам нравятся подобные вещи). Razor-qt GitHub wiki на <http://tinyurl.com/cd4lgrc> предлагает исчерпывающую информацию, в том числе полезные ссылки на список зависимостей. Исходный код находится на LXF DVD.

Установив рабочий стол, завершите текущую сессию рабочего стола, и вы сможете выбрать Razor-qt в окне приглашения. Через несколько секунд (он грузится очень быстро, особенно по сравнению с KDE и Gnome) вы увидите рабочий стол; итак, давайте-ка приступим к знакомству с ним...



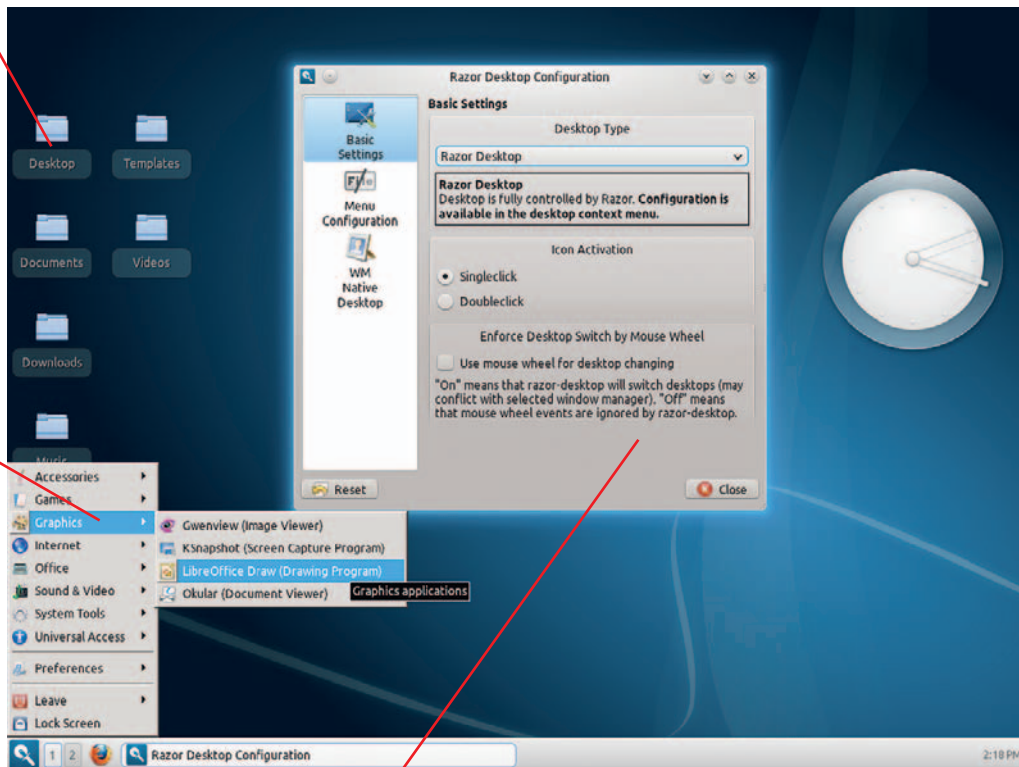
Исследуем интерфейс

Рабочий стол

По умолчанию вы, возможно, ничего здесь не увидите (если у вас ничего нет в директории `~/Desktop`), потому что Razor-qt практикует исключительно необычный способ разделения рабочего стола. Вы можете выделять области для отображения виджетов, один из которых может быть отображением директории – на данном экране снимке это наша домашняя директория. На следующей странице мы расскажем вам, как это настраивается.

Панель

Раскладка панели Razor-qt довольно типичная: в дальней левой части размещается меню для запуска приложений, за которым следует переключатель виртуального рабочего стола. Далее идет область, отведенная под быстрый запуск часто используемых приложений, и переключатель задач. Затем, в дальней правой части вы можете рассмотреть системный лоток (где размещаются значки работающих приложений) и часы. Можете изменить размер и длину панели, щелкнув правой кнопкой по пустому пространству и выбрав `Configure`; чтобы переместить ее, щелкните правой кнопкой и перейдите в `Show This panel` в подменю.



Инструменты настройки

Здесь есть множество мелких утилит, которые находятся в основном меню в `Preferences` >

Razor-qt settings. Например, можно выбрать, открывать ли папки однократным или двойным щелчком, и дать менеджеру окон полный

контроль над рабочим столом. У Razor-qt нет собственного менеджера окон, но он отлично работает с KWin и Openbox.

Настройка

Нет предела совершенству.

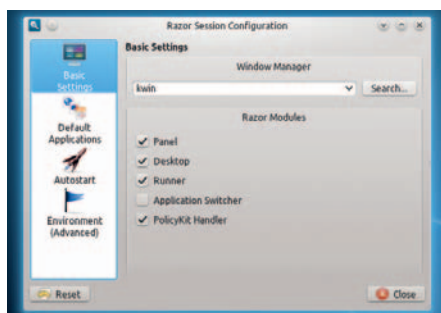
Настройки по умолчанию Razor-qt очень неплохи для повседневного использования, но вы, без сомнения, захотите довести их до совершенства. Давайте начнем с рабочего стола: как уже упоминалось, он поделен на области, где можно размещать виджеты (или плагины, как их здесь именуют). Щелкните правой кнопкой по пустому пространству на рабочем столе и выберите Edit Desktop во всплывающем меню. Включится режим редактирования, как станет ясно из крупного текста, появляющегося на фоне (см. снимок экрана). В настройке по умолчанию Razor-qt имеет две панели: слева – отображение файловой системы Filesystem view, указывающая на директорию `~/Desktop`, а справа – часы.

Обе настраиваются через контекстные меню, вызываемые щелчком правой кнопкой мыши. Например, щелкните по Filesystem view, и затем Configure Plugin. Теперь можно выбрать, какое содержимое директории будет отображаться внутри панели. Эта функция очень удобна: вы получаете большой объем экранного пространства, которое можете использовать, отведя часть под музыкальные файлы, другую часть – под фотоколлекцию, еще одну – под важные рабочие файлы, и т.д. Гибкость рабочего стола на столь ранней стадии разработки вызывает в нас большое желание проследить, что же будет с ним дальше.

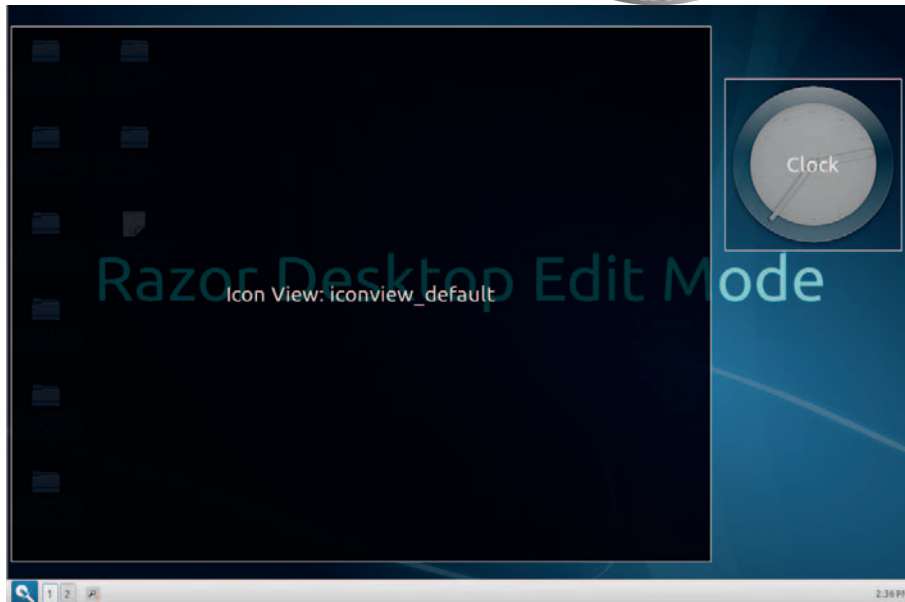
За пределами Filesystem view и часов не особо много плагинов, о которых стоит рассказывать, но мы ожидаем, что в грядущих релизах их будет больше. Завершив настройку, снова щелкните по пустому пространству рабочего стола правой кнопкой и выйдите из окна Edit Desktop. Обратите внимание, что через это меню Context вы также можете изменить настройки фона рабочего стола.

Поиграем с панелью

Все значки на панели можно расположить в другом порядке. Щелкните правой кнопкой по пустому месту на панели, и во всплывающем меню перейдите в Plugins. Здесь вы увидите список всех значков и прочего содержимого панели, и сможете переместить или удалить каждый из этих пунктов. Для многих плагинов вы также увидите



► Хотите сменить менеджер окон? Загляните в панель Session Configuration.



► В режиме Edit можно изменять размер областей рабочего стола, отведенных под виджеты.

добавочные опции настройки – например, в меню плагина Task Manager можно увеличить окно, чтобы настроить его внешний вид (скажем, чтобы отображались значки и текст или только значки).

Добавить новые пункты на панель очень легко. Щелкните правой кнопкой по пустому месту, перейдите в Plugins, а затем – в Add Plugins. Здесь вы увидите список имеющихся плагинов с описаниями. Добавление кнопки Quick Launch – дело довольно хитрое, потому что, щелкнув по кнопке Add, вы, скорее всего, не увидите никаких изменений на панели. Будет даже прискорбно, если вы решите, что эта функция в данном релизе не работает...

Однако на самом деле происходит вот что: в левой части панели появляется крошечное пустое пространство размером в три пикселя, рядом с кнопкой Main Menu (которая теперь сместится немного вправо). Это и есть ваша новая кнопка Quick Launch, однако попробуйте щелкнуть по ней правой кнопкой мыши – это весело. Чтобы превратить ее в нормальную кнопку, пригодную для использования, откройте основное меню, найдите нужное вам приложение и перетащите его значок на то крошечное пустое место, которое вы создали. И тогда оно волшебным образом увеличится в размерах и станет функциональной кнопкой. Нам нравится здесь возможность drag-and-drop, но было бы намного лучше, если бы пустая кнопка Quick Launch была отмечена вопросительным знаком или чем-то другим, чтобы привлечь внимание.

В Main Menu > Preferences > Razor-qt settings > Razor Session Configurator вы увидите панель настройки, которая позволит вам выбрать другой менеджер окон (если он у вас установлен) и мо-

дули, вызываемые при запуске рабочего стола. Также вы сможете определить приложения по умолчанию для работы в сети и терминале. Пока в Razor-qt нет собственных приложений (хотя они должны появиться – загляните в интервью разработчиков), и вам придется что-то выбрать.

Есть два способа это сделать. Если ваш компьютер имеет большой объем памяти, а производительность – не проблема, и вы выбираете Razor-qt только потому, что вам не нравится KDE, можно просто выбрать приложения KDE в качестве набора инструментов по умолчанию. То есть использовать *KWrite* как основной текстовый редактор, *Konsole* – как эмулятор терминала, и т.д. Тогда вы получите набор приложений, которые выглядят и работают, как в KDE, но без всяких функциональных осложнений, присутствующих в KDE.

Однако если вы используете Razor-qt, чтобы полностью избавиться от присутствующих в KDE проблем с потреблением памяти, то вы также захотите избежать использования программ KDE, потому что даже самая легкая из них может запустить множество фоновых сервисов KDE. Наилучшим выходом будет выбор исключительно приложений Qt – то есть программ, использующих только инструментарий, а не полный пакет рабочего стола. Вот что мы рекомендуем:

Файловый менеджер: qtfm (www.qtfm.org)
Терминал: qterminal (<https://github.com/qterminal>)
Текстовый редактор: JuffEd (<http://www.juffed.com>)
Web-браузер: QupZilla (www.qupzilla.com)
Программа для мгновенного обмена сообщениями: qutim (www.qutim.org)
Музыкальный плеер: Qmmp (<http://qmmp.ylsoftware.com>)

Слово – разработчикам

Своим видением делится дуэт, стоящий за Razor-qt.

Петр Ванек [Petr Vanek] и Александр Соколов [Alexander Sokoloff] – двое основных разработчиков среды рабочего стола. Мы решили узнать от них, как появился проект и каких еще приятных сюрпризов нам стоит ждать.

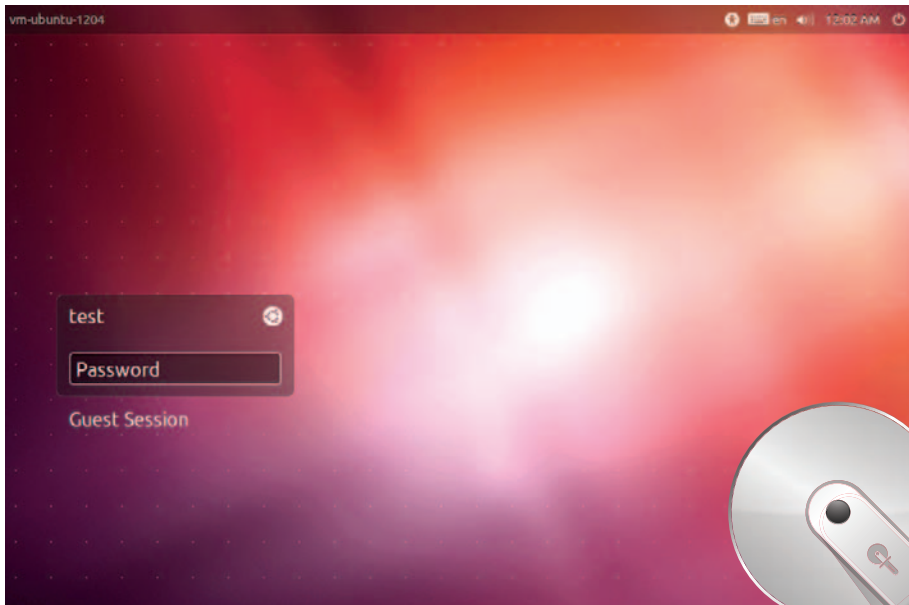
LXF: Откуда появилась идея Razor-qt? Вас не устраивало направление развития KDE, или его потребление памяти, или просто так захотелось?

ПВ: Раньше я использовал на своем офисном компьютере *Openbox* (дома – KDE), и меня он почти устраивал. Но приходилось использовать в *Openbox* сторонние компоненты для его расширения. А на работе я применял очень маленький набор инструментов – большинство их было основано на *Qt*, и меня увлек поиск решения на базе *Qt*. Я испробовал разные исторические подходы (менеджер окон *Antico*, *Antico-deluxe*, и другие ответвления), но все они были в той или иной степени непригодны либо к использованию, либо к компиляции. Потом я нашел старый код Razor-qt. Он по своему дизайну был сыроват, но не перегружен. И было ощущение, что многие ищут как раз нечто похожее. Я не люблю сравнивать Razor-qt с KDE. Я по-прежнему считаю, что KDE – самая лучшая среда рабочего стола. И предлагаю людям ее использовать. Но есть некоторые области, где она не годится. Альтернатива – Razor-qt.

АС: Пользователи области *Gtk* имеют несколько альтернатив *Gnome*. Пользователи KDE не могли похвастаться тем же. Мы решили исправить это.

LXF: Вы планируете создавать собственные компоненты, типа менеджера окон и файлового менеджера?

ПВ: Я ленив, и, полагаю, это одна из самых ценных моих черт. Я также согласен со стратегией «разделяй и властвуй», поэтому если я нахожу какое-то внешнее решение своей проблемы, я использую его. На сегодняшний день мы тесно контактируем с разными проектами, и обсуждаем функции и решения. Всегда лучше иметь особую команду или разработчика в одном спецприложении для наших ограниченных ресурсов. Некоторые «основные» приложения, такие, как *Andromeda*, *JuffEd*, *Qterminal*, *Qupzilla*, тестируются для Razor-qt, и проблемы исправляются там, где это нужно (в том числе инструменты *Qt*, *X11*, *XDG*, и т.д.).



➤ Грядущий Razor-qt 0.5 включит поддержку LightDM (как в Ubuntu), легковесного окна приглашения.

LXF: Далеко ли вы планируете зайти с библиотеками Razor-qt? Они предназначены исключительно тем, кто пишет плагины и виджеты, или вы думаете о полномерных приложениях Razor-qt?

ПВ: Мы уже много раз говорили об этом. На данный момент мы не хотим разрабатывать приложения исключительно для Razor, за исключением инструментов рабочего стола. Мне, например, представляется не слишком разумным разрабатывать текстовый редактор с зависимостями Razor, потому что у *Qt* многоплатформенная природа. С другой стороны, я могу себе представить GUI сетевого менеджера, использующий библиотеки Razor, потому что он будет решать только проблемы, касающиеся Razor.

В любом случае, все, что мы делаем потенциально полезного для общественного использования, распространяется в виде отдельной библиотеки или пакета – например, библиотека *QtXDG*.

АС: Наборы инструментов разделяют мир Linux. У нас есть множество приложений, похожих, как близнецы. Их единственное различие – наборы инструментов, которыми они пользуются. У нас нет желания увеличивать энтропию, поэтому библио-

теки Razor-qt на данный момент не являются наборами инструментов для сторонних приложений.

LXF: Каковы планы на грядущие релизы, и когда ожидать релиза 1.0?

ПВ: «Когда будет готов». Это пока что очень молодой проект. И мы все еще находимся в процессе изменений некоторых основных внутренних моментов. С другой стороны, Razor-qt сейчас уже отвечает моим личным потребностям. Однако я понимаю, что у других могут быть куда большие потребности и ожидания.

LXF: Какая помощь вам нужнее всего – что могут сделать наши читатели для участия в проекте?

ПВ: Мы будем благодарны за любую. Судя по всему, у нас сложилась более нетривиальная ситуация, чем в других проектах с открытым кодом. У нас много переводчиков, много создателей пакетов, пользователей. Но мала команда разработки.

Да, у проекта есть основные разработчики и отличные участники, но мы будем благодарны всем, кто захочет помочь. Помните: кодирование для нового проекта – это большое удовольствие! **LXF**

Разработчики, разработчики, разработчики...

Если вы пишете программы в *Qt*, вам не придется что-то менять ради уверенности в том, что ваши программы отлично подойдут для нового рабочего стола. Однако есть библиотека, которая поможет вам добавить специфичные для Razor-qt функции в ваше приложение, улучшая тем самым его интеграцию.

Список классов API имеется на <http://razor-qt.org/develop/docs/classes.html> – большинство функций для разработчиков, пишущих плагины для панели.

Прочтите краткое руководство по применению простого плагина «Hello, world» на <http://tinyurl.com/92bsa58>. Не сказать, чтобы здесь кипела дея-

тельность по части сторонних плагинов, но мы надеемся, что разработчики заинтересуются и вольются, и скоро у нас будет лучшее из обоих миров: минималистская, с низкими издержками среда рабочего стола в основных пакетах, с массой функций, доступных из других источников.



Несправедливость
гораздо более выгодна,
чем справедливость

Индивидуальные
творения отдельных наций
становятся общей собственностью...
из многочисленных национальных
и местных литератур
складывается мировая
литература

Все для себя
и ничего для других людей,
нажется, во все времена
было подлой максимой
для хозяев человечества

Будь философом —
но при всей
твоей философии,
оставайся человеком

Множество величайших
вещей человек добился не в итоге
сознательно направленной мысли,
и еще менее — в итоге объединенных
усилий отдельных личностей,
но в процессе, в котором отдельная
личность играет роль, полностью ею
никогда не осознаваемую

Философия свободного ПО

Джонатан Робертс изучает место Linux в мире этических, экономических и религиозных воззрений.

Как мы недавно обнаружили в TuxRadar, люди очень любят спорить. Мы поинтересовались (сами — держа язык за зубами), как воспринимают Linux: марксистским или капиталистическим.

Полученные ответы немало нас повеселили, но многие из них были весьма продуманны и заставили и нас задуматься о том, как Linux и движение свободного ПО соотносятся с широкими философскими, экономическими, религиозными и этическими воззрениями

и спорами, занимающими человечество на протяжении веков. Глядя на то, как сам Линус Торвалдс втянулся в столь праздную болтовню, что показало его недавнее интервью на BBC, мы решили, что было бы интересно продолжить беседу. Мы взглянем на Linux и движение СПО со стороны, изучая их под ви-

дом продолжающихся дебатов, и рассмотрим несколько теорий и возможность их применения к нашей любимой ОС.

Сначала предупредим: нам в Linux и свободном ПО важнее всего то, что это — практическая реальность: просто здорово, что все это работает, что это свободно и что люди могут получить огромное удовольствие, используя и создавая его — а некоторые при этом умудряются еще и нажиться. Все остальное — просто гарнир, так что не стоит особо расстраиваться из-за того, что вы читаете!

**«Нам важнее всего
в Linux и свободном ПО
то, что это реальность.»**

Что такое теория игр?

В этой статье мы немного поговорим о теории игр, но вы можете не знать, в чем она заключается. Определение в Wikipedia не очень помогает разобраться с этим: оно гласит, что это изучение конфликта и сотрудничества между ответственными за принятие рациональных решений лицами посредством математической модели. Что же это означает на самом деле?

Хороший способ разобраться с тем, что это означает – рассмотреть пример, игру под названием «дилемма заключенного». Стенфордская энциклопедия философии описывает эту игру так: «Предположим, полиция арестовала двоих участников вооруженного грабежа. К сожалению, у нее недостаточно улики, чтобы убедить присяжных. Однако

улик хватает, чтобы посадить обоих арестованных на два года за кражу автомобиля, использованного для бегства с места преступления. Главный следователь делает следующее предложение каждому из арестованных: если ты сознаешься в ограблении и сдашь поделницу, причем она не сознается, то ты свободен, а она получит десять лет. А если вы оба сознаетесь, то получите по пять лет каждый. Ну, а если никто из вас не сознается, вы получите по два года за кражу машины».

Из этого описания игры понятно, что если первый арестованный рационально обдумает ситуацию – то есть попытается понять, какой вариант минимизирует их срок заключения – то их мыслительный процесс может выражаться таким образом:

» Если второй человек молчит, то я должен его сдать. Это простой выбор между свободой и двумя годами тюрьмы.

» Если второй сознается и сдает меня, то я, опять же, должен сдать его. Это простой выбор между десятью и пятью годами тюрьмы.

Рационально рассуждая, становится ясно, что единственный выход для каждого подозреваемого – сдать другого. Что бы ни делал второй, это гарантирует наилучший исход.

Похожий анализ применим ко множеству всяких ситуаций, в том числе к эволюции языка, юриспруденции и многим другим концепциям. Это мощный инструмент с массой возможностей для применения в современном мире.



» Линус Торвальдс ничуть не стесняется вступать в дебаты.

концов, если у вас будет мой код, и вы продолжите улучшать свой, вы получите передо мной преимущество – вам удастся делать меньше работы, но при этом иметь лучший результат – а люди эгоистичны!

Похоже, даже Платон как минимум подозревал, так же, как и Торвальдс, что мир вряд ли изменится, если кто-то скажет: «Давайте все дружно споем “Кумбайя”, собравшись вокруг костра, и сделаем этот мир лучше».

«Зайцы» и безопасность

Брюс Шнайер [Bruce Schneier] обращается к той же проблеме в своей последней книге «Лжецы и посторонние», что доказывает актуальность этого спора для мира технологии, как изнутри, так и за его пределами.

В книге он описывает игру «Ястребы и голуби», из теории игр. Суть ее в том, что в дикой популяции, где пищевые ресурсы ограничены и все стремятся получить к ним доступ, птицы делятся на голубей и ястребов. Ястребы агрессивны и готовы драться за еду: если встретятся два ястреба, они будут драться, и в результате один получит пищу, а другой – раны; возможно, смертельные. Голуби пассивны, и встречаясь около еды, они стараются поделиться между собой. Если голубь встретит ястреба, то ястреб совершенно точно получит

пищу, потому что голубь предпочтет уступить.

Хотя из этой игры можно сделать множество выводов, самое важное наблюдение, сделанное Шнайером,

заключается в том, что при любом сценарии в выборке всегда будет по меньшей мере несколько ястребов. Если бы изначально население на 100 % состояло из голубей, некоторые из них миглом смекнули бы, что они получат для себя больше пищи, если поведут себя, как ястребы, очень мало при этом рискуя нарваться на других ястребов, поскольку голубей большинство.

Раз уж мы упомянули интервью Линуса Торвальдса для BBC, давайте с него и начнем. Линус сказал: «...открытый код реально работает только тогда, когда им занимаются по своим эгоистическим причинам... в целом собственность по GPLv2 являет собой очень простую модель “ты – мне, я – тебе”: я вам отдам свои улучшения, если вы пообещаете отдать свои».

В наблюдении Торвальдса интересно то, что оно связано с дискуссиями в области философии, этики, биологии, психологии и даже математики, и уходит корнями аж в сочинения Платона (как минимум!). В своем диалоге «Государство» Платон обсуждает правосудие и мораль, и размышляет о том, являются ли они по своему происхождению социальными структурами или неким абстрактным благом.

Рассуждая таким образом, Главкон, один из героев диалога, предлагает идею волшебного перстня, которое делает того, кто его носит, невидимым. Он предполагает, что и законопослушный, и законопослушный владелец такого кольца будет вести себя одинаково: брать что-то на рынке, проникать

в дома и «возлечь» с приглянувшимся человеком или убить своего врага. Он говорит: «Если человек, овладевший такою властью, не пожелает когда-либо поступить несправедливо и не притронется к чужому имуществу, он всем, кто это заметит, покажется в высшей степени жалким и неразумным», поскольку «несправедливость гораздо более выгодна, чем спра-

«Наблюдение Торвальдса уходит корнями аж в сочинения Платона.»

ведливость» [цит. по: Платон, собр. соч. в 4-х т., изд-во «Мысль», 1994, – прим. ред.].

Какой пессимистичный взгляд на людей! Согласно вы с Главконом или нет, очевидно, что это точка зрения Линуса: без социальных ограничений вроде GPLv2 я не смогу быть уверен в том, что, поделившись с вами своими улучшениями кода, я, в свою очередь, получу от вас ваши. Зачем вам это делать? В конце

Конечно, по мере роста популяции ястребов наступит момент, когда это начнет оказывать дурные последствия на всю популяцию. Голуби начнут страдать от недостатка пищи и вымирать, уступая в конфликтах за еду, а ястребы начнут stalkиваться друг с другом чаще, и для них тоже возрастет риск гибели в результате этих стычек.

Ладно, довольно о ястребах и голубях. Какое отношение они имеют к свободному ПО и GPL? Ну, вывод здесь такой: не позволяйте GPL «быть эгоистами», по словам Торвальдса, мы оказались бы в такой ситуации, когда чересчур много ястребов хватало бы код, ничего не возвращая взамен и постепенно уменьшая доверие и сотрудничество, и в итоге популяция программистов открытого кода вымерла бы.

В остальной части книги Шнайер предлагает различные «механизмы безопасности», способные помочь нам доверять действиям других людей, создавая возможность сотрудничества даже в том случае, если мы не можем доверять (эгоистичным) мотивам других.

Шнайер указывает на такие вещи, как закон, эволюция зеркальных нейронов, и т.д., и GPL тоже можно понимать в этом смысле, т.е. как механизм безопасности, созданный для обеспечения доверия и сотрудничества. И еще он очень разумен.

Свободное ПО и экономика

Помимо того, что это просто интересное тематическое исследование для всех, кто заинтересован в сотрудниче-

стве, свободное ПО привлекает пристальное внимание из-за своего сходства с различными экономическими системами.

Хорошим примером будет Билл Гейтс, который в 2005 году сказал: «В наши дни завелся особый вид коммунистов, которые под разными предлогами хотят отменить поощрение для... разработчиков программ».

Конечно, есть вероятность, что Гейтса куда меньше интересовал этот вопрос с экономической точки зрения, и он куда больше был озабочен тем, чтобы отпугнуть испытывающие любовь к свободному рынку капиталистические американские фирмы от применения свободного ПО; это замечание всплывает настолько часто, что на него стоит обратить внимание.

Во-первых, стоит отметить, что свободное ПО имеет мало общего с советским коммунизмом, ключевыми понятиями в котором было централизованное планирование и огромное полицейское государство с исправительными лагерями и принудительным трудом. Те, кто давно знаком со свободным ПО, знают, что централизованное планирование — вещь

«Знакомые с СПО знают: централизованное планирование — редкость.»

редкая, если вообще возможная: рост количества форматов пакетов, дистрибутивов, офисных пакетов, сред рабочего стола, сетевых и почтовых серверов — достаточное тому свидетельство. Более того, никого не принуждают работать над свободным ПО или применять его. Фактически, поскольку все наши форматы данных реализованы в открытом коде, любой может заново реализовать их в конкурирующей программе, и глазом не моргнув.

Многие выдвигали эти аргументы, чтобы доказать — к немалому огорчению Гейтса, как нам представляется — что свободное ПО имеет с советским коммунизмом меньше

общего, чем многие проприетарные компании. Такие компании, как Apple и Microsoft, известны своим планированием сверху вниз, и их даже хвалят за него; они также печально известны тем, что вынуждают людей использовать их пакеты программ и оборудования посредством применения по умолчанию закрытых проприетарных форматов данных, которые конкурирующие программы не могут нормально реализовать.

Марксизм

Если у свободного ПО мало общего с советским коммунизмом, возможно, у него больше общего с марксизмом. Одна из основных идей этого мировоззрения — та, что, владея средствами производства, будь то оборудование, знания или нечто другое, высшие классы эксплуатируют низшие; не владея средствами производства, рабочие обязаны «добровольно» работать за зарплату, чтобы быть в состоянии купить все необходимое для выживания: жилье, одежду, еду и развлечения.

На деле они не могут выбирать, работать им или нет, и у них никогда не будет излишков зарплаты или доли в распределении прибыли. Одна из самых живучих идей Маркса заключалась в его надежде на то, что эта ситуация может измениться, если рабочие получат свободу, в бесклассовом обществе со совместной собственностью на средства производства.

Поскольку в современном мире одним из главных средств производства является компьютер, свободное ПО очень неплохо вписывается в систему Маркса. Код находится в совместной собственности. Все могут

жест измениться, если рабочие получают свободу, в бесклассовом обществе со совместной собственностью на средства производства.

Поскольку в современном мире одним из главных средств производства является компьютер, свободное ПО очень неплохо вписывается в систему Маркса. Код находится в совместной собственности. Все могут



Эбен Моглен (Eben Moglen), работавший над GPL с самого начала, и зачинщик Freedom Box (общественный проект по разработке, проектированию и продвижению персональных серверов, на которых запущено свободное ПО, — прим. пер.) полагает, что СПО играет жизненно важную роль в борьбе за свободу идей.

Свободное ПО или открытый код?

Одним из классических споров внутри и вокруг мира свободного ПО и ПО с открытым кодом является вопрос о том, как же его называть. Некоторые говорят исключительно «свободное ПО», некоторые — исключительно «открытый код», некоторые используют то одно, то другое название, а есть и группа, которая выдвигает странные выражения, сочетающие и то, и другое одновременно, например free/libre open source software (FLOSS).

Те, кто глубоко вовлечен в движение свободного ПО, определенно считают, что название важно, и пишут целые сочинения на данную тему; то же самое можно сказать и об активистах движения открытого кода, поскольку они нашли вре-

мя, чтобы предложить новое название и описание, с целью дистанцироваться от движения свободного ПО. Но так ли это важно на самом деле?

После этой статьи мы полагаем, что название важно. Как стало понятно после рассмотрения множества тем, которые мы применяли в той или иной степени к свободному ПО и ПО с открытым кодом, философию и политику можно применить к чему угодно, и люди увидят в этом то, что им захочется. Без ясного и понятного названия, которое будет выражать то, что, по нашему убеждению, мы делаем, кто-то сможет интерпретировать наши действия и работу в ином свете, нежели мы изначально предполагали.



залась доступна, если не сказать, достигнута. Web-серверы не предназначены только для владельцев средств производства, потому что код является свободным, и любой может при желании поделиться любым культурным достижением.

Может быть, это только слова, а может быть, и средство создания всемирной децентрализованной валюты, подобной Bitcoin, или планы использования машин для создания вашего собственного небольшого городка, как в Global Village Construction Set [платформа для создания сельскохозяйственных машин с от-

общества и урегулировать и минимизировать конфликты начинают привычки и традиции. Однако, в отличие от Смита и Хайека, Хьюм полагал, что людские страсти не исчерпываются жаждой прибыли, и порядок и привычки могут появиться из любви к иным вещам.

Какое отношение это имеет к свободному ПО? Ну, это же очевидно, правда? Свободное ПО – пример стихийного порядка в том виде, как его понимал Хьюм. Поскольку оно не приносит прибыли его разработчикам и распространяется бесплатно, в нем отсутствуют ценовые колебания; в сообществах свобод-

ного ПО люди свободно сотрудничают и общаются для создания ПО, которое общество в целом считает ценным.

Однако имеются определенные сигналы, оказывающие влияние на выбор того, над чем будут работать создатели проекта. Например, если пользователи некой программы найдут лучший эквивалент, они, очевидно, перейдут на него. Разработчики, не желая работать над никому не нужной программой, с тем же успехом могут поискать новые проекты, которые люди сочтут полезными.

Таким образом, даже без ценовых сигналов разработчики свободного ПО направляют свои усилия в области, приносящие наибольшую пользу наибольшему числу людей, что и будет большим благом для общества в целом. **LXF**

«Свободное ПО, похоже, действительно поддержал бы Маркс.»

свободно читать его, изучать его, делиться им и, в конце концов, перекомпилировать его. В таком виде невозможно принуждение рабочих теми, кто располагает в классовой системе выше, поскольку в любой момент они могут принять решение использовать средство производства, код, для собственных нужд.

Свобода мысли

Эбен Моглен оспорил пагубность воздействия, оказываемого совместной собственностью на код, на наше общество в своем вступительном выступлении на Wizards of OS 3 под названием «Мысли свободны: свободное ПО и борьба за свободу мысли».

Он заявил, что «защищать невежество значит защищать рабство» (он отлично умеет находить нужные слова!). Его идея заключается в том, что без знания экономики, без знания технических наук, культуры – всего того, что движет этот мир – низшие классы не имеют ни малейшей надежды на улучшение своего положения и ни малейшей надежды получить в собственность средства производства. Заслуга свободного ПО – как и свободного оборудования, свободной культуры и всех свобод – в высвобождении средств, благодаря которым свобода мысли и информации ока-

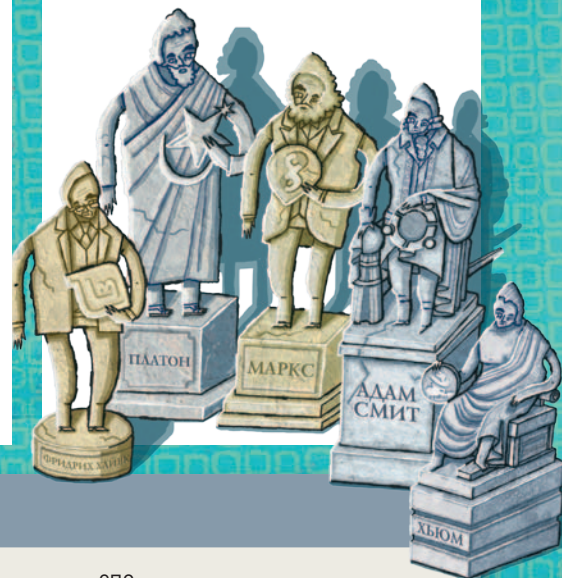
крытым кодом, – прим. пер.]. И важно то, что все это стало возможным благодаря общей собственности на код.

Стихийный порядок

Свободное ПО, пожалуй, действительно поддержал бы Маркс, но вы будете удивлены, узнав, что можно очень солидно обосновать возможность для него стать моделью свободного рынка, столь милого сердцу капиталиста и столь ненавистного марксистам и противникам глобализации по всему миру. Ну, пусть не свободного рынка, но по крайней мере одной из его основных идей – стихийного регулирования.

Одна из основных идей свободного рынка заключается в том, что, словно повинувшись невидимой руке, колебания цен координируют усилия отдельных лиц на благо общества. Эта идея очень похожа на идеи Адама Смита [Adam Smith] и Фридриха фон Хайека [Friedrich von Hayek], использовавших термин «стихийный порядок» для ее описания, но на самом деле она берет начало у Дэвида Хьюма [David Hume], одного из величайших философов шотландского просвещения.

Хьюм полагал, что при отсутствии центрального управления организовывать жизнь



Что еще почитать

Многие тезисы этой статьи являют собой только наброски, предоставляя лишь весьма общее описание весьма сложных и тонких идей.

В результате было бы неплохо почитать об этих идеях поподробнее. Чтобы помочь вам в этом, предлагаем несколько указателей на ресурсы, которые помогли в написании этой статьи и могут стать первым шагом в изучении темы.

» **Янис Варуфакис** [Yanis Varoufakis], экономист компании Valve's (игровая компания), написал превосходный материал о стихийном

порядке на своем рабочем месте: <http://blogs.valvesoftware.com/economics/why-valve-or-what-do-we-need-corporations-for-and-how-does-valves-management-structure-fit-into-todays-corporate-world>

» **Брюс Шнайер**, «Лжецы и посторонние [Liars and Outsiders]» – захватывающая книга и прекрасный стартовый пункт для знакомства с теорией игр и эволюционной этикой.

» **Персональная страница Эбена Моглена** дает ссылки на его выступления и презентации

на тему СПО; многие из них опираются на марксистскую теорию. <http://moglen.law.columbia.edu>

» **Платон, «Республика»**. Полная бесплатная версия работы Платона доступна на MIT <http://classics.mit.edu/Plato/republic.html> [или <http://www.lib.rus.ec/s/32877>, – прим. ред.]

» **Стенфордская энциклопедия философии** [Stanford's Encyclopedia of Philosophy] предоставляет отличные ознакомительные статьи (достаточно подробные) по множеству упомянутых и других тем. <http://plato.stanford.edu>

ZFS on Linux: Вопросы истории

Алексей Федорчук рассказывает о перспективной системе размещения данных и предпосылках ее возникновения.

Эта статья посвящена истории ZFS – универсальной системы размещения данных, интегрирующей в себе собственно файловую систему и технологию управления дисковыми массивами и логическими томами.

Общее введение

Одна из главнейших задач при работе на компьютере – манипулирование данными: создание, модификация, копирование, перемещение и так далее. И тут первое – это организация их размещения. Это понятие включает широкий круг частных вопросов: схемы дисковой разметки, управления дисковыми массивами и логическими томами, файловые системы и их монтирование в файловую иерархию. Они тесно связаны между собой, но традиционно решаются каждая с помощью собственного инструментария.

Однако в последние годы в Linux'e получили распространение интегрированные системы размещения данных, объединяющие в себе и файловые системы, и задачи управления массивами и томами, и даже, частично, задачи разметки дисков. Такие системы, как мы увидим из исторического обзора, существовали очень давно – со времен доисторического UNIX'a, но были они проприетарными. ZFS же, разработанная фирмой Sun для своей ОС Solaris, ныне распространяется свободно, под лицензией CDDL. Благодаря чему была портирована на FreeBSD, а в последние годы нативно поддерживается и в Linux'e. Именно ZFS on Linux и будет героиней нашего романа, и не только в силу своих несравненных достоинств.

А во-вторых, развитие проекта ZFS on Linux блестяще демонстрирует торжество инженерного разума над юридической заумью. И потому являет собой просто замечательный литературный сюжет, мимо которого не в силах пройти ни один сочинитель в жанре технологической новеллы. И начать этот сюжет надо издалека.

Дисковая разметка

Говорят, что во времена далекие, теперь почти былинные, файловых систем не было: информация на носители записывалась побитно, без всякой организации в именованные ее наборы. Впрочем, такой способ записи данных применялся и много позднее – например, при резервном копировании на стриммерные ленты. Можно обходиться без файловых систем и при записи на стандартные блочные устройства – винчестеры, SSD, компакт-диски.

Однако в большинстве случаев данные на носителях блочного типа организуются в виде файлов, а файлы объединяются в файловые системы – плоские, как в древнем DOS'e, древовидные, как во всех UNIX-подобных операционках, или, так сказать, «многодревные», как в Windows. Каковые могут быть созданы непосредственно на носителе как raw-устройстве, но обычно накладываются на дисковые разделы.

До недавнего времени в Linux'e применялась разметка в MS-DOS-стиле, предполагающая возможность разбиения диска на че-

тыре раздела, называемых первичными [primary partitions]; один из них может быть определен как расширенный раздел [extended partition], внутри которого по «матрешечному» принципу можно создать логические разделы, максимальным числом до 63.

Разметка в MS-DOS-стиле преобладает в дистрибутивах Linux'a и по сей день. Однако все большее распространение получает разметка в GPT-стиле. Среди ее преимуществ – возможность создания на диске до 128 абсолютно равноправных (то есть не разделяющихся на физические и логические) разделов. А в случае использования винчестеров «продвинутого» формата [Advanced Format] и SSD, размер блоков которых равен 4 КБ, она обеспечивает оптимальное выравнивание границ разделов.

Исторически сложилось так, что одному разделу соответствовала одна файловая система. Соответственно, и выходить за границы несущего их устройства файловые системы не могли. И если требовалось работать более чем с одной файловой системой на одном физическом накопителе (а в UNIX-подобных ОС это почти всегда так), то был необходим тщательный расчет дискового пространства для каждой из них: ошибки в расчетах влекли весьма неприятные последствия, вплоть до необходимости пере разбиения диска и переустановки ОС вообще.

Правда, дисковые разделы могут не только разделяться, но и объединяться в программные массивы или в группы томов, о которых мы сейчас и поговорим.

Массивы и логические тома

Задача объединения носителей информации особенно актуальна при использовании нескольких физических накопителей, и особенно при их добавлении в работающую систему. В элементарном исполнении это делалось просто (по крайней мере, в UNIX-подобных ОС): второй (новый) накопитель просто размечался по соответствующей для данной ОС схеме, на нем создавалась новая файловая система определенного типа, которая монтировалась в общую файловую иерархию. Однако выход за границы существующего раздела и диска для файловой системы был по-прежнему невозможен.

Для решения задачи объединения физических носителей в единое логическое устройство и «размазывания» по ним файловых систем традиционно используется два основных способа: RAID (Redundant Array of Independent Disks – избыточный массив независимых дисков) и LVM (Logical Volume Manager – менеджер логических томов).

RAID'ы существуют трех видов – аппаратные, квази-аппаратные (так называемые Fake RAID) и чисто программные (Soft RAID). Первые дороги и на десктопах почти не встречаются; работа вторых под Linux'ом часто проблематична, так что речь пойдет в основном о третьих. Впрочем, с точки зрения логики это роли почти не играет.

Логически в любом из RAID'ов несколько дисков (а в Soft RAID – и дисковых разделов) могут просто слиться воедино (Linear

RAID), при записи на них может осуществляться расщепление данных [stripping], что приводит к ускорению дисковых операций (RAID Level 0); на объединенных разделах можно создать различные формы избыточности, обеспечивающей восстановление данных при отказах дисков. Из таких избыточных массивов чаще всего используется полное дублирование (RAID Level 1, он же mirror) или избыточность за счет контрольной суммы (RAID Level 5). Наконец, возможно и совмещение стриппинга с дублированием.

RAID любого типа и уровня может разбиваться (и обычно разбивается) на разделы, которые уже несут на себе файловые системы. И, таким образом, позволяют размещать их на нескольких физических устройствах. Однако они не решают второй проблемы размещения данных – необходимости расчета потребного для них дискового пространства и его перераспределения при необходимости.

Этим целям служит технология LVM, объединяющая физические носители в группы логических томов, разделяемых на собственно логические тома, которые, в свою очередь, разбиваются на экстенды – объединения физических блоков дисковых устройств. Логические тома предстают перед операционной системой как обычные разделы, каждый из которых может нести свою файловую систему. При этом технология LVM дает возможность при необходимости перераспределять физическое пространство носителей между ними посредством добавления или отнятия экстендов на лету, не только без перерасметки дисков, но и без перезапуска системы.

Как и RAID Level 0, технология LVM может обеспечить стриппинг данных между физическими томами с целью повышения быстродействия файловых операций. А в сочетании с Soft RAID позволяет и создавать массивы с полной (зеркалирование) или частичной (за счет контрольных сумм) избыточностью, повышающей надежность.

Таким образом, LVM выполняет оба поставленных условия: слияние дискового пространства, в том числе и вновь подключаемых накопителей, и возможность его перераспределения между существующими файловыми системами, да еще и с бонусом в качестве повышения быстродействия. Комбинация же LVM и Soft RAID позволяет и повысить надежность. Казалось бы, чего еще не хватает для счастья?

А не хватает для счастья простоты: если установить LVM позволяет инсталлятор почти любого современного дистрибутива, то управление логическими томами и по сей день задача не из самых тривиальных. Да к тому же мы забыли о файловых системах, без которых ни RAID, ни LVM к использованию не пригодны. Так что на очереди к рассмотрению – они.

Файловые системы

Как известно еще с советских атеистических времен, Господь Бог, создавая человека, хотел сделать его умным, честным и партийным. Но оказалось, что даже он, при всем своем всемогуществе, не смог ему дать больше двух качеств вместе.

Аналогично и с файловыми системами: разработчики хотели бы видеть их быстрыми, надежными и простыми в обращении. Давайте посмотрим, удалось ли им превзойти Господа.

В UNIX-подобных системах требование быстродействия удовлетворяется, во-первых, оптимизированным расположением каталогов, метаданных и данных файлов на физических носителях. Но во-вторых и главных – кэшированием записи.

Думаю, каждого, кто начинал знакомство с Linux'ом во времена безраздельного господства файловой системы ext2fs, поражала быстрота выполнения всех файловых операций, обусловленная их асинхронностью – то есть кэшированием данных

и метаданных. Обратная сторона медали – отказ системы по любой причине влек за собой тяжкие последствия, вплоть до полного ее разрушения. Но и даже когда до полной катастрофы дело не доходило, отказы (например, по питанию) вызывали за собой долгую и нудную процедуру проверки целостности файловой системы.

Были разработаны различные механизмы решения этой проблемы. Однако основным в Linux стало так называемое журналирование, когда сведения о файловых операциях записываются в специальный файл журнала до того, как эти операции будут фактически выполнены. Это дает возможность после любого сбоя «откатить» файловую систему до последнего непротиворечивого состояния. Обратной стороной чего, как обычно, является снижение быстродействия – различное для отдельных файловых систем и видов файловых операций.

Правда, с точки зрения простоты использования ни в одну из файловых систем Linux'a бросить камень рука не подымается: создание и монтиро-

вание их никаких трудностей не сулит. Так что требование «партийности» выполняется, пожалуй, при всех соотношениях «ума» и «честности». Но эта ситуация сохраняется, пока мы не начинаем комбинировать «ум, честность и партийность» файловых систем с аналогичными качествами систем управления RAID'ами или с LVM. В результате чего получаем:

- » либо быстрое и простое решение на основе RAID Level 0, не блестящее надежностью;
- » либо надежное решение без ощутимой потери быстродействия на основе одного из RAID с избыточностью, не являющееся, однако, эталоном простоты;
- » либо, наконец, относительно надежное и потенциально быстрое решение при использовании технологии LVM – однако о простоте здесь можно забыть сразу.

Причем все эти решения – многоуровневые. И очевидно, что удлинение «цепочки» уровней в любом случае приводит к снижению надежности: чем больше в ней звеньев, тем вероятней отказ всей цепи.

И тут-то и возникает вопрос: а нельзя ли уменьшить количество уровней, сделать систему более «плоской»? И системы размещения данных, в том числе и ZFS – попытка ответа на него.

Из истории систем размещения

Не в интересах правды, а истины ради нужно заметить, что ZFS была отнюдь не первой комплексной системой размещения данных – хотя ее исторические предшественницы также именовались просто файловыми системами.

Первой из таких предшественниц была, видимо, файловая система Veritas (или VxFS), разработанная фирмой Veritas Software и представленная миру в 1991 году. Она же претендует на звание первой в истории мироздания журналируемой файловой системы. Хотя, насколько мне известно, JFS – эпоним всех журналируемых ФС – в своей реализации для AIX появилась в 1990 году, так что вопрос приоритета остается не вполне ясным.

VxFS является основной файловой системой в HP UX, работает также во всех ныне живущих проприетарных UNIX'ах и теоретически может использоваться в Linux'е. Однако о практических примерах последнего я не слышал: VxFS является системой проприетарной и весьма дорогой.

VxFS тесно интегрирована с менеджером логических томов – VxVM. Благодаря чему в ней возможно изменение (в любую сторону) размера файловой системы «на лету», включение различных режимов использования томов – стриппинг данных, их зеркалирование, а также комбинации того и другого, создание

«Основным в Linux стало так называемое журналирование.»

»

избыточных массивов по типу RAID Level 5, изменение внутренней организации данных без остановки работы. Все это позволяет VxFS (в сочетании с VxVM) претендовать на звание комплексной системы размещения данных.

Впрочем, не меньше к тому оснований было и у AdvFS – файловой системы, разработанной к 1993 году фирмой DEC для своего проприетарного варианта UNIX, именовавшегося сначала OSF/1, затем Digital UNIX, и завершившего свою жизнь под именем Tru64 UNIX. Судьба ее была печальной. Снискав заслуженное признание на своей родной платформе DEC Alpha под управлением указанной ОС, она после покупки DEC фирмой Compaq оказалась в загоне. А после того, как Compaq, в свою очередь, был поглощен фирмой Hewlett Packard, использовавшей для своего UNIX'a на платформах HP PA и Itanium только что упомянутую VxFS, AdvFS оказалась совсем не при делах.

В результате HP сделала щедрый дар сообществу свободного софта вообще и Linux-сообществу в особенности: в середине 2008 года исходники файловой системы AdvFS были открыты под лицензией GPLv2 – ради максимальной совместимости с ядром Linux. С предложением использовать их в качестве богатой технологической базы для этой ОС. Правда, оговорка, что сама HP не заинтересована в дальнейшем развитии AdvFS заставляла вспомнить народную присказку: «Возьми, небоже, что мне не гоже».

Да и предложение несколько запоздало: как мы скоро увидим, к тому времени интенсивно развивались и ZFS, и btrfs.

Однако, помимо исходников, HP предоставила также доступ ко всей документации – благодаря чему об AdvFS при желании можно узнать больше, чем о любой другой проприетарной файловой системе для UNIX-подобных операционок. Это избавляет меня от необходимости описания особенностей AdvFS. Замечу только, что среди них мы увидим все черты развитой комплексной системы размещения данных. Те самые, с которыми ознакомимся, когда дело дойдет наконец до рассмотрения устройства ZFS. Но для начала перейдем к обзору уже ее истории.

Начало истории ZFS

Разработчики ZFS поставили себе честолюбивую цель: создать систему хранения данных, которая отвечала бы всем трем критериям идеала. Разработка ее проводилась в компании Sun Microsystems, командой под руководством Джеффа Бонвика [Jeff Bonwick] и Мэттью Аренса [Matthew Ahrens]. Первоначально название ZFS рассматривалось как аббревиатура от Zettabyte File System, но быстро стало просто условным именованием. Его можно интерпретировать как последнюю точку в развитии файловых систем вообще. И в последующем мы увидим: это недалеко от истины.

Результаты работы над ZFS были представлены миру в августе 2004 года. А в 2006 году она была включена в штатный состав OS Solaris 10 (релиз 6/06). То есть, подобно своим предшественникам, она также была проприетарным продуктом. И пользователям свободных UNIX-подобных систем поначалу от ее существования было ни холодно, ни жарко. Однако период камерного существования ZFS продолжался недолго – уже в ноябре 2005 года, то есть до включения в Solaris, ее поддержка была интегрирована в открытый ее вариант, OpenSolaris. Ибо она основывалась на том же ядре SunOS 5, что и коммерческий прототип.

Исходники ZFS распространяются, как и собственно OpenSolaris, под лицензией CDDL (Common Development and Distribution License). Эта лицензия, базирующаяся на Mozilla Public License (MPL), не влияет на общую лицензию проекта, в со-

став который включены CDDL-компоненты. И потому оказывается совместимой с большинством свободных лицензий. За исключением... какой? Правильно, GPL во всех ее проявлениях.

Разумеется, ZFS была задействована в клонах OpenSolaris, таких, как BeleniX, SchilliX и, в первую голову, в Nexenta OS. Правда, последняя развивалась в направлении коммерческой системы хранения данных, а о числе пользователей остальных можно было только гадать.

Некоторое время ZFS была доступна пользователям Macintosh'a – в Mac OS X Leopard от осени 2007 года. Правда, ходившие перед ее выходом слухи, что она будет там файловой системой по умолчанию, оказались несколько преувеличенными: поддержка ZFS оказалась опциональной и лишь в режиме «только для чтения».

А в последующих версиях семейства кошачьих вообще исчезла и, видимо, уже не возродится.

Так что для широких народных масс ZFS по прежнему оставалась недоступной.

Пока... пока ее не портировали под FreeBSD в 2007 году, и официально не включили ее поддержку в 7-ю версию этой ОС, вышедшую в начале 2008 года. В чем, как и в дальнейшем ее развитии, основная заслуга принадлежит Павлу-Якубу Давидеку [Pawel Jakub Dawidek] и Ивану Ворасу [Ivan Voras]. Правда, до недавнего времени ZFS нельзя было задействовать при установке FreeBSD средствами ее штатного инсталлятора и конфигуратора *sysinstall*. Однако это без труда можно было осуществить в дальнейшем руками. В том числе и разместить на ZFS корень файловой иерархии.

С самого начала поддержки ZFS во FreeBSD появилась и возможность задействовать ее, что называется, «искаропки», в десктоп-ориентированном клоне последней – PC-BSD. А с переходом FreeBSD, начиная с версии 9.0, на новую программу установки – *BSDInstall*, эта функция распространилась и на материнскую систему.

Успех ZFS во FreeBSD, где она стала если не главной файловой системой, то добилась равноправия с UFS2, послужил примером для других BSD-систем. Так, ныне ZFS поддерживается в NetBSD – эта работа была начата Оливером Голдом [Oliver Gould] летом 2007 года в рамках акции Google Summer of Code. А в 2009 году Адам Хамсик [Adam Hamsik] интегрировал ее код в ядро NetBSD. Правда, насколько я понимаю, использование ZFS в этой операционке рекомендуется только в экспериментальных целях.

Наконец, одно время в списках рассылки DragonFlyBSD активно обсуждался вопрос о портировании ZFS и на эту ОС. Потом, правда, разговоры эти стихли – вероятно, в связи с активной разработкой файловой системы Hammer, обладающей во многом аналогичными возможностями. Однако, учитывая легкость адаптации к DragonFlyBSD любых сторонних файловых систем, можно не сомневаться, что поддержка ZFS на уровне обмена данными будет включена в нее тогда и если (или если тогда), когда (и если) это кому-то понадобится.

Таким образом, пользователям большинства BSD-систем ZFS или уже доступна как нативная, или может стать доступной в ближайшее время.

Из истории юриспруденции

А что же Linux, спросите вы меня? Как обстоит дело с поддержкой ZFS в самой массовой из свободных UNIX-подобных операционных систем нашего времени? А вот с Linux'ом все оказывается гораздо сложнее. Ибо не зря поминали мы выше лицензию CDDL. Которая сама по себе очень даже свободная, и не накладывает почти никаких ограничений на распространение защищаемых ею программ.

«Разработчики ZFS поставили себе честолюбивую цель.»

В частности, не запрещает CDDL и коммерческого распространения производных продуктов в виде бинарников, без открытия исходных текстов. Как известно, не накладывает такого ограничения и лицензия BSD, почему включение кода поддержки ZFS в любые BSD-системы и проходит юридически безболезненно, как мы только что видели на примере FreeBSD.

А вот с лицензией GPL обеих актуальных версий (v2 и v3) CDDL входит в диалектическое противоречие. Ибо любые продукты, производные от программ под GPL, вне зависимости от формы распространения, должны сопровождаться исходными текстами. Что делает юридически невозможным включение кода поддержки ZFS непосредственно в ядро Linux, распространяемое, как известно, на условиях GPLv2.

Кроме того, невозможность включения в ядро Linux кода поддержки ZFS объясняется тем, что GPL требует распространения всех основанных на ней продуктов под GPL же, тогда как CDDL – сохранения ее для «своих» компонентов.

Правда, часть кода ZFS была открыта под GPL с тем, чтобы соответствующий патч можно было включить в загрузчик *Grub*. Это обеспечило возможность загрузки Open Solaris непосредственно с ZFS-раздела. Однако оказалось недостаточным для полноценной реализации этой системы, которую можно было бы распространять под данной лицензией.

Впрочем, не будучи юристом, ломать голову над лицензионными вопросами не буду, и моим читателям не советую, ибо понять это все равно невозможно. А достаточно лишь запомнить, что всеми резонными и юридически подкованными людьми признано, что поддержки ZFS в ядре Linux быть не может.

Таким образом, сложилась абсурдная, с точки зрения здравого смысла, ситуация: два программных продукта под свободными лицензиями (обсуждать вопрос, какая из них «свободней другой», мы сейчас не будем), созданные друг для друга, как Huggies и... э-ээ... место пониже спины (дальнейшие события показали, что технических сложностей при портировании ZFS на Linux практически нет), невозможно было использовать в составе одного проекта. По крайней мере, для законопослушных граждан, чтущих... нет, не только уголовный кодекс, но и принципы свободного программного обеспечения.

И, разумеется, здравомыслящие люди попытались эту ситуацию разрешить. И первая такая попытка была предпринята еще в 2006 году в рамках Google Summer of Code. Основывалась она на поддержке ZFS через FUSE (Filesystem in Userspace). Поскольку модуль FUSE работает как пользовательское приложение, необходимости во включении кода ZFS в ядро Linux нет, что снимает все юридические вопросы. Однако встают вопросы другие – производительности и устойчивости.

Проект ZFS-FUSE развивается по сей день, хотя и не очень быстрыми темпами. Правда, находясь в стадии хронической бета-версии, он до сих пор рассматривается как сугубо экспериментальный. Да и в любом случае в таком виде ZFS выполнять свои функции – быть надежным хранилищем данных большого объема – скорее всего, не сможет.

Так что ZFS-FUSE нельзя считать кардинальным решением вопроса с этой системой размещения данных в Linux. А на то, что в его ядро будет встроена собственная реализация ZFS, рассчитывать не приходится.

Появление героини

И тем не менее, решение этой проблемы нашлось – и решение столь же изящное, сколь и очевидное. Его предложил весной 2010 года Брайан Белендорф [Brian Behlendorf], некогда один из основ-

ных разработчиков web-сервера *Apache*. Он создал модуль поддержки ZFS, который собирается и может распространяться отдельно от ядра, сохраняя прародительскую лицензию CDDL. А поскольку последняя, как уже говорилось, является лицензией «пофайловой», этим самым обходится антагонистическое противоречие – запрет на распространение продуктов, в которых смешан код, лицензируемый под CDDL и GPL.

На базе разработки Брайана возникло сразу два проекта. Первый осуществлялся индийской компанией KQ Infotech, которой уже в сентябре 2010 года удалось выпустить работоспособный, пригодный для тестирования Linux-ядра с реализацией файловой системы ZFS. А в январе следующего, 2011, года появилась финальная его версия, доступная тогда в исходниках и в виде двоичных пакетов для Fedora 14, RHEL6, Ubuntu 10.04 и 10.10.

Однако весной того же года KQ Infotech была куплена фирмой STEC, занимающейся производством SSD-накопителей, каковых, впрочем, в наших палестинах никто не видел. И работы по дальнейшему развитию нативной поддержки ZFS были свернуты. Хотя исходники модуля и сопутствующих компонентов до сих пор доступны, последнее их обновление происходило более года назад. А информации о дальнейшей судьбе проекта с тех пор не появлялось.

Однако сам Брайан продолжал свою работу – вместе с сотрудниками Ливерморской национальной лаборатории, каковая, будучи в подчинении Министерства энергетики США, занимается не только вопросами ядерного оружия (эвфемизмы вроде Минсредмаша в ходу не только в бывшем Советском Союзе), но и разработкой суперкомпьютеров. В результате скоро возник проект ZFS on Linux – <http://zfsonlinux.org>, в рамках которого модуль поддержки ZFS и сопутствующие утилиты поддержки, пор-

тированные из Solaris – так называемый SPL (Solaris Porting Layer), были доведены до ума, и к началу 2011 года стали пригодны для использования в экспериментальном режиме. А к настоящему времени,

несмотря на формальное сохранение статуса release candidate, порт ZFS on Linux можно считать готовым к практическому применению.

Правда, майнтейнеры основных дистрибутивов не торопились включать поддержку ZFS в свои системы даже в качестве дополнительных неофициальных пакетов. Подозреваю, что не столько из косности и лени, сколько из-за очередной сложности: видимо, по все тем же лицензионным ограничениям модули *zfs* и *spl* приходится привязывать к фиксированной версии (и даже конкретной сборке) ядра Linux. Что при регулярных, даже корректирующих, обновлениях последнего требует и их пересборки.

Тем не менее, разработчики проекта воплотили результаты своей работы в виде дополнительного (так называемого PPA) репозитория для Ubuntu. А также сочинили подробные инструкции по собственноручной сборке пакетов в форматах RPM и Deb (ссылки можно найти на странице проекта).

Достаточно подробно включение ZFS описано в Gentoo Wiki. А майнтейнеры ее клона, дистрибутива Sabayon, прославившиеся своей склонностью к экспериментам, включили поддержку ZFS почти «искаропки»: соответствующие модули подгружаются при старте с LiveDVD и могут быть опробованы в «живом» режиме. Хотя штатного способа установки системы на ZFS в инсталляторе этого дистрибутива, все из-за тех же юридических заковок, и не предусмотрено.

Дистрибутив openSUSE не сподобился попасть в список «инструментируемых» на сайте проекта.

Так что ныне нет ни малейших препятствий к использованию ZFS в openSUSE, о чем и пойдет речь в следующей статье. **LXF**

«Порт ZFS on Linux можно считать готовым к применению.»

Darwin: Вид происхождения

Свободная ОС от несвободной компании? Нет, Роман Ярыженко вовсе не шутит!



Наш эксперт

Романа Ярыженко хлебом не корми – дай попробовать какую-нибудь экзотическую (и не очень) ОС. Желательно открытую.

Darwin – это не тот, который выдумал теорию эволюции, а тот, на котором основана проприетарная Mac OS X. Читатели, возможно, удивятся: с чего это **LXF** рассказывает о проприетарной ОС? Загвоздка тут в том, что Darwin как раз таки свободная и открытая ОС – причем в полном соответствии со смыслом, вкладываемым в эти слова FSF и OSI.

История

Полагаю, что историю Darwin надо начать с ухода Стива Джобса из Apple. Мы не можем знать точно, из-за чего он ушел, но итогом стало то, что он переманил в новую фирму некоего Ави Теваняна [Avie Tevanian], который работал над микроядром Mach и был одним из его основных разработчиков (концепция Mach разработана в Университете Карнеги-Меллона). Переманил его Джобс, конечно, не просто так, а для разработки ОС, которая в дальнейшем получит название NeXTStep.

Для разработки новой ОС было решено использовать объектно-ориентированные технологии. Неведомо почему был выбран Objective-C, а не C++. Некоторые источники показывают, что C++ к моменту начала работы над NeXTStep еще не был создан, но они вступают в противоречие с другими источниками, поэтому можно предположить, что новоявленная компания просто выбрала тот язык, который старше и больше похож на C – основной язык разработки ОС того времени. NeXT Computers лицензировала Obj-C в 1988.

Язык этот на тот момент был довольно «продвинутом» – он, например, включал ООП (причем ООП реализовано через сообщения, как в SmallTalk, несколько непохоже на то, к чему мы привыкли сейчас), динамическую типизацию, а в современной версии еще и сбор мусора.

Ядро же NeXTStep было гибридным (несмотря на то, что некоторые источники называют данную ОС микроядерной, это не так) и написанным на чистом C. Разработчики взяли лучшие черты микроядра Mach, такие как модульную архитектуру и ядра BSD (скорость и сетевой стек).

Чарльз Дарвин: вокруг света

Рассматривать установку будем на примере VMWare Server 1.0 – у автора отсутствует процессор с виртуализацией, а без нее многие современные виртуальные машины работают очень медленно. Тип виртуальной машины – FreeBSD, жесткий диск IDE; указываем ISO-образ, загружаемся с него... поехали!

» **Шаг 1** Выбираем жесткий диск для установки. С этим, думается, проблем не возникнет ни у кого.

» **Шаг 2** Выбор типа разметки жесткого диска. Тут мы, как это ни странно, выбираем ручную, ибо автоматическая, сколько ни пытались, не хотела работать. Нас выбросит в *fdisk*. Набираем **auto hfs** (автоматическая разметка диска с одним разделом HFS+) и... скорее всего, получаем ошибку вида “malloc:*** Deallocation of a pointer not malloced...”. Еще раз набираем **auto hfs** – теперь уже этого сообщения не будет. Записываем MBR командой **w**, затем выходим из *fdisk* – **quit**. Теперь нас попросят выбрать корневой раздел – честно говоря, непонятно, зачем надо вводить его целиком, когда можно было бы выбирать из имеющихся путем нажатия соответствующей цифры – но это уже вопрос к разработчикам программы установки. Забегая вперед, скажем, что это далеко не единственная ошибка в ней. Далее у нас попросят подтвердить: действительно ли мы хотим совершить «чистую» установку? Отвечаем – и снова вопрос: имя тома; здесь можно написать что угодно. После ответа, как водится, ждем <Enter>. Оба-на! Плишет “error: nknown filesystem type”. Уходим на перезагрузку – заново с образа. Замечаем, что на первом этапе появился пункт 3 – “Use existing partition”. Его и выбираем. Опять попросят указать корневой раздел. После этого будет вопрос: какой тип файловой системы использовать? Вопрос этот выглядит, по меньшей мере, странным – есть же таблица разделов, оттуда и берите – но теперь хотя бы понятна причина перезагрузки: установочная программа (скрипт?) попыталась подмонтировать раздел без ФС, и, естественно, вылезла вышеуказанная ошибка. С нас снова спросят подтверждение и имя тома – если нам еще не надоело, отвечаем на них, и – ура! – началось копирование файлов. Это может

Что нам надо

Пакеты, необходимые для компиляции MC версии 4.6.1:

- » Pkg-config 0.25
- » Gettext 0.18.1.1
- » Glib 2.6.4
- » slang 2.1.4
- » pcre 8.20

Основные вехи истории NeXTStep/Darwin

- » **1989** Вышла бета-версия NeXTStep на NeXT Cube – уникальном для своего времени компьютере на основе процессора Motorola 68030. В сентябре же появилась версия 1.0. Базировалась она на Mach 2.5 и 4.3BSD. GUI же было основано на PostScript.
- » **1990** Версия 2.0 включила такие новшества, как поддержку приводов CD-ROM и цветных мониторов, динамически загружаемые драйвера и многое другое.
- » **1991** Вышла версия 2.1.

- » **1992** Версия 3.0. В этом же году на выставке NeXTWORLD Expo была представлена версия для PC, стоимостью чуть меньше 1000 долларов.
- » **1993** В мае и октябре были выпущены версии 3.1 и 3.2 соответственно.
- » **1994** NeXT и Sun выпустили открытые спецификации OpenStep, включая несколько API и фреймворков. На их основе можно было написать собственную реализацию платформы, что и было сделано позднее.

- » **1995** Выпущена последняя версия NeXTStep, 3.3. Далее она выходила под другим названием – OpenStep/Mach.
- » **1997** Apple купила NeXT и на основе NeXTStep (точнее, OpenStep/Mach) началась разработка проекта Rhapsody.
- » **2000** Создано ответвление Rhapsody – Darwin. Тот самый Darwin, о котором идет речь в нашей статье. Историю релизов Darwin при желании несложно найти в Сети.

занять несколько мин... ой. Журнал-то про *nix, зачем здесь повторять сообщение Windows? Но, как бы то ни было, это довольно длительный процесс, и во время него можно, к примеру, съесть пирожное.

» **Шаг 3** Копирование завершено. Нас спросят пароль root (дважды – в этом смысле все нормально), потом спросят имя домена, а затем выбор из 3 пунктов: создать нового пользователя, перезагрузиться и оболочка. Ну, пользователя можно создать и потом, оболочка порождается глючная, поэтому выбираем пункт 2 – Reboot. Перезагружаемся... и судорожно вспоминаем, был ли вопрос про загрузчик. Выясняется, что не было. Да-да! Его надо ставить ручками.

» **Шаг 4** Снова загружаемся с образа. Только теперь необходимо работать в оболочке, поэтому набираем **shell**. Вошли. Набираем первую команду –

```
fdisk -f /usr/standalone/i386/boot0 -u -y /dev/rdisk0
```

Что эта команда делает? Делает она следующее: ставит первую часть загрузчика в MBR (без таблицы разделов – опция -u) без запроса подтверждения – опция -y.

» **Шаг 6** После этой команды даем команду

```
dd if=/usr/standalone/i386/boot1h of=/dev/rdisk0s1
```

для установки загрузочного сектора раздела HFS+.

» **Шаг 7** Монтируем раздел –

```
mount_hfs /dev/disk0s1 /mnt
```

и копируем файл –

```
cp /usr/standalone/i386/boot /mnt/
```

(собственно, сам загрузчик). Отмонтируем – **umount /mnt** – и перезагрузимся – **shutdown -r now**. Установка завершена.

Первое знакомство

Загрузка может пойти нормально, а может и зависнуть (“Still waiting for root device”) – мы не смогли выяснить, с чем это связано. Во втором случае следует перезагрузиться – иногда несколько раз. После успешной загрузки видим приглашение входа в систему и ошибки ненастроенного mDNSResponder. Переводим строку и входим как root. Для начала хотелось бы порекомендовать скомпилировать старый добрый *MC* – конечно, можно обойтись и без него, но с ним удобнее.

С компиляцией придется повозиться: необходимо мало того что скачать как собственно *MC*, так и все необходимые зависимости, но еще и передать их на виртуальную машину. А каким образом? Сеть-то не настроена. Единственный более-менее доступный путь – использовать ISO-образ.

После того, как создадим и подмонтируем образ диска (на всякий случай – порядок действий: в каталоге **/Volumes** создать папку **cdrom**, потом смонтировать диск командой **mount_cd9660 /dev/disk1s0 /Volumes/cdrom**. Естественно, образ уже должен быть подготовлен для использования в виртуальной машине. Специфика Darwin...), необходимо скопировать все файлы в домашний каталог. Команда копирования стандартная; единственное, что надо отметить – Darwin странно отображает файлы на ISO9660, поэтому копировать лучше по маске, а разархивировать с использованием автодополнения по клавише <Tab>, благо оно имеется.

Да! Перед началом компиляции выполните следующие команды:

```
export MACOSX_DEPLOYMENT_TARGET=10.4
```

– устанавливает минимальную версию ОС;

```
export PATH=$PATH:/usr/local/bin
```

– понятно что;

```
export LIBDIR=/usr/local/lib
```

– тоже понятно, но для тех, кому непонятно – указывается путь, куда при компиляции будут складываться библиотеки.

Собственно компиляция зависимостей происходит долго и нудно – распаковка (**tar xzvf имя_архива**), переход в каталог, **configure**, **make**, **make install**, распаковка, переход в каталог...



» Так выглядел компьютер с прародителем Darwin – NeXTStep.

важно лишь, в каком порядке – а он следующий: **pcrc**, **pkg-config**, **gettext**, **slang** и, наконец, **glib**.

Зависимости скомпилировались – переходим к собственно *mc*. Тут действуем почти так же – но **configure** необходимо выполнить с опцией **--with-screen=slang**. И после выполнения **make install** будет нам счастье – скомпилированный *mc* под Darwin

Но счастье редко бывает полным – по неизвестным причинам в текстовом режиме в ОС от Apple не работают функциональные клавиши. По слухам, они работают в графическом терминале – но компиляция X-сервера и какого-либо оконного менеджера выходит за рамки этой статьи. А что же делать с функциональными клавишами? Придется использовать фокус – вместо функциональных клавиш <F1>–<F10> *mc* позволяет использовать комбинацию <Esc>+цифра от 1 до 0. Не самый удобный способ, да, но с этим ничего не поделаешь.

Пройдемся немного по дереву каталогов – оно, дерево, сильно отличается от привычного нам. И не стоит забывать, что Darwin – «десктопная ОС без десктопа», если можно так выразиться, поэтому некоторые элементы там присутствуют лишь как заглушки. И так, зрим в корень... и вот какие каталоги мы там видим.

» **Каталог /Trashes** как раз и является заглушкой – в Mac OS X он выполняет функцию корзины для суперпользователя.

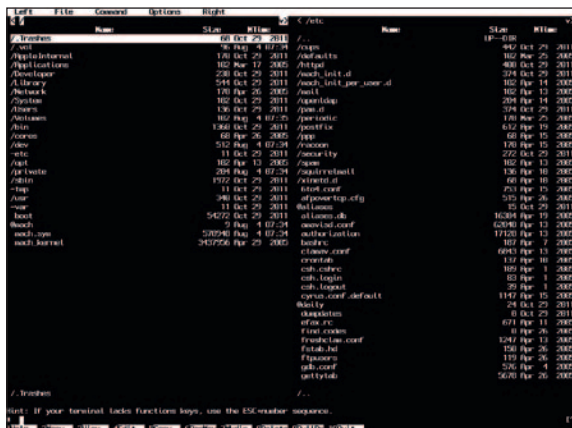
» **Каталог /vol** – загадочный каталог, где находится еще один (или более) каталог с цифровым именем. Но если посмотреть вывод команды **mount**, становится понятным, что к оному каталогу подмонтирована псевдофайловая система **volfs**, назначение которой из названия, впрочем, не становится менее загадочным. Данная файловая система предназначена для доступа к файлам не по имени, а по inode. Это было необходимо для стыковки POSIX

```
Uncompressing AppleFileSystemDriver-5.root.tar.bz2
Uncompressing AppleFlashNVRAM-105.4.0.root.tar.bz2
Uncompressing AppleGMACEthernet-140.3.5.root.tar.bz2
Uncompressing AppleGPIO-118.4.0.root.tar.bz2
Uncompressing AppleGenericPCATA-5.root.tar.bz2
Uncompressing AppleGracklePCI-1.2.root.tar.bz2
Uncompressing AppleHWSensor-137.0.4.root.tar.bz2
Uncompressing AppleHeathrow-121.0.1.root.tar.bz2
Uncompressing AppleI2C-345.0.2.root.tar.bz2
Uncompressing AppleI2S-100.0.2.root.tar.bz2
Uncompressing AppleI386GenericPlatform-5.root.tar.bz2
Uncompressing AppleI386PCI-6.root.tar.bz2
Uncompressing AppleIntel8255x-18.root.tar.bz2
Uncompressing AppleIntelPIIXATA-17.root.tar.bz2
Uncompressing AppleK2SATA-104.3.2.root.tar.bz2
Uncompressing AppleK2SATARoot-104.3.1.root.tar.bz2
Uncompressing AppleKauaiATA-110.3.1.root.tar.bz2
Uncompressing AppleKeyLargo-162.0.2.root.tar.bz2
Uncompressing AppleKeypress-103.0.2.root.tar.bz2
Uncompressing AppleKiwiATA-102.3.1.root.tar.bz2
Uncompressing AppleKiwiRoot-102.3.1.root.tar.bz2
Uncompressing AppleMPIC-1.2.3.root.tar.bz2
Uncompressing AppleMIRRSupport-4.root.tar.bz2
Uncompressing AppleMacRiscPCI-3.0.4.root.tar.bz2
Uncompressing AppleMediaBay-1.0.1.root.tar.bz2
Uncompressing AppleOnboardAudio-250.2.17.root.tar.bz2
Uncompressing AppleOnboardPCATA-2.root.tar.bz2
Uncompressing ApplePCCard16ATA-111.0.1.root.tar.bz2
Uncompressing ApplePCCardATA-103.3.1.root.tar.bz2
Uncompressing ApplePS2Controller-8.root.tar.bz2
```

»

» Разметка диска.

➤ Старый добрый т.с. Синего фона, конечно, немного не хватает...



и Carbon API, который ныне (2012 год) считается уже устаревшим, но в версии Mac OS X, основанной на описываемом в данном разделе Darwin 8.0.1, был вполне себе живым.

- **/AppleInternal** – содержит некоторые заголовочные файлы.
- **/Applications** – опять же, предназначен для OS X. Туда обычно устанавливаются несистемные приложения с GUI, разработанные специально для ОС от Apple. Интересно, что в Darwin присутствует одно такое приложение, хотя GUI там и в помине нет – соответственно, даже и пытаться его запустить не стоит.
- **/Developer** содержит документацию, примеры исходных кодов и шаблоны проектов модулей ядра и расширений ядра – разница между ними будет пояснена чуть позже.
- **/Library** содержит некоторые важные конфигурационные файлы – в частности, файл **/Library/Preferences/SystemConfiguration/com.apple.Boot.plist**, необходимый для загрузчика, и некоторые логи.
- **/Network** – по всей видимости, служит для монтирования сетевых файловых систем.
- **/System** содержит подкаталог **Library**, ближайшим аналогом которого в Linux является **/lib**. В нем присутствуют фреймворки приложений, модули и расширения ядра, модули Perl...
- **/Users** содержит домашние каталоги пользователей и в этом смысле совершенно идентичен **/home**.
- **/Volumes** аналогичен **/mnt** и **/media**. Каталоги, созданные в нем пользователем, существуют до следующей перезагрузки – учтите это, когда будете в следующий раз монтировать компакт-диск.
- **/bin**, **/dev**, **/sbin**, **/usr**, **/opt** – объяснять не будем, и так все понятно. А вот каталоги **/etc**, **/tmp** и **/var** на самом деле являются символическими ссылками на соответствующие каталоги в **/private**.

Если уж заглянули в каталог **/etc**, давайте в файле **profile** исправим переменную **PATH**, добавив в число путей **/usr/local/bin**, и добавим переменные **LIBDIR** и **MACOSX_DEPLOYMENT_TARGET** (значения см. ранее) – на случай, если читателю захочется поупражняться в компиляции чего-либо POSIX-совместимого. В принципе, можно даже попытаться скомпилировать свежие версии ядра и базовых утилит, буде читатель окажется жестким извращенцем – автор этих строк не рискнул поставить подобный эксперимент.

Архитектурные особенности

Как и древние римляне, начнем с яйца – то есть с ядра. Ядро у ОС от Apple, как уже сказано, основано на микроядре Mach. Но «основано на микроядре» вовсе не означает, что архитектура микроядерная: архитектура гибридная, и условно состоит из трех слоев. ➤ В самом низу ядра лежит слой Mach, который отвечает за HAL (не стоит искать данного термина в документации, мы его исполь-

зуем для упрощения), управление процессорами, вытесняющую многозадачность, управление памятью, внутриядерный IPC, поддержку «мягкого» реального времени и, наконец, за отладку ядра – в качестве отладчика на отлаживающей машине используется GDB.

➤ Слой BSD лежит выше и включает реализацию процессов и сигналов, базовую модель безопасности, API POSIX, системные вызовы, поддержку сети и многое другое.

➤ Слой I/O Kit – это фреймворк, позволяющий разработчикам использовать ограниченный набор возможностей C++ для объектно-ориентированного программирования драйверов.

В принципе, эти слои не настолько четко разделены, поэтому некоторые могут еще что-то выделить, а некоторые, наоборот, предпочтут что-то убрать; но общая картина такова.

Xnu (так называется это ядро; расшифровывается же эта аббревиатура как... да-да, Xnu is not Unix – Столлмену впоору обвинять Apple в плагиате) позволяет подключать к себе расширения – так называемые kernel extensions, в единственном числе kext. Расширения могут состоять из одного или более модулей ядра. Модуль ядра (Kmod) является минимальной единицей кода, которую можно загрузить в ядро.

В Darwin используется формат исполняемых файлов Mach-O и поддерживается технология, основанная на нем – мультиархитектурные исполняемые файлы. Отметим, что Apple применяла ее для переноса OS X с архитектуры PowerPC на Intel.

Теперь рассмотрим такие специфические особенности Darwin, как «пакеты [bundles]» и фреймворки (примечание: здесь и далее речь может пойти о вещах,

которые не поддерживаются в версии Darwin 8.0.1 – сия версия 2005 года, а сейчас уже, почитай, 2012 заканчивается скоро). «Пакетом» в данном контексте называется набор

связанных ресурсов, в число которых могут входить исполняемые файлы, изображения, аудио, заголовочные файлы и даже другие пакеты. Ясное дело, что такой «пакет» – понятие абстрактное, а посему приведем конкретный пример – фреймворк. Фреймворк – набор, состоящий из одной или более динамических библиотек с заголовочными файлами, документацией, локализацией и сопутствующими программами для облегчения работы. Давайте рассмотрим некоторые фреймворки, входящие в состав ОС от Apple.

Уже упоминавшийся ранее I/O Kit – фреймворк, состоящий из двух частей: режима пользователя и режима ядра. Часть пользовательского режима отвечает за простой доступ к оборудованию из приложений, а часть режима ядра – за базовую иерархию классов и поддержку C++. Мы уже отмечали, что C++ в I/O Kit несколько ограничен. Заключается это в отсутствии таких вещей, как шаблоны, исключения, RTTI и множественное наследование. Тем не менее, это несколько не мешает разработчикам писать драйверы.

CoreFoundation – фреймворк, обеспечивающий базовые возможности для программ, написанных на Objective-C, такие как разбор XML-файлов, доступ к ресурсам через URL, управление списками свойств и многие другие.

Для облегчения работы с сетью предназначен фреймворк CFNetwork. Он делает удобной работу с сокетами, создание SSL/TLS-соединений, разрешение имен... Разработчик желает работать с HTTP или FTP? Никаких проблем – за это также отвечает данный фреймворк.

Нельзя не упомянуть и о файловой системе HFS+. Особенно-стями она по современным меркам не отличается (разве что поддержка любых символов Юникода в имени файла, за исключением символа NUL), но все же перечислим ее возможности. Это

журналирование, В-деревья для эффективного поиска файлов, прозрачное сжатие...

А как обстоят дела с безопасностью? О, с безопасностью все хорошо, не сомневайтесь! Вот некоторые возможности и технологии, так или иначе относящиеся к ней:

» **ASLR** Технология, размещающая в ОЗУ некоторые критично важные системные данные случайным образом. Это серьезно осложняет эксплуатацию ряда атак, требующих знания их расположения.

» **NX-бит** Запрещает выполнение кода в стеке и в куче.

» **Права доступа** Стандартная модель Unix.

» **ACL** Расширенная система прав доступа; позволяет индивидуально для каждого пользователя назначать права доступа к файлам.

» **Kauth** Мандатная система контроля доступа; аналогична SELinux.

» **Аудит.**

» **Шифрование** раздела подкачки.

» **IPSec** Его реализацию, как, впрочем, и в Linux, стащили из FreeBSD.

» **Подписывание исполняемых файлов** Возможно теоретически.

Необъятное объять нельзя – nemo omnia potest scire – но автор и не ставил такой цели, он ставил цель заинтересовать читателя. Хотелось бы надеяться, что у него это получилось.

Итоги

Впечатление о данной ОС сложились неоднозначные: с одной стороны, интересно посмотреть на внутренности коммерческой ОС от Apple, а с другой... Darwin явно не является системой для настоящего использования. Предпринимались попытки создать дистрибутив на основе этой ОС, но особого успеха они не получили. Может быть, выйдет у кого-нибудь из наших читателей?.. **Linux**

Ссылки по теме

- » <http://www.gnu.org/philosophy/apsl.html>
- » <http://www.opensource.org/licenses/alphabetical>
- » <http://lowendmac.com/orchard/05/next-computer-history.html>
- » <http://www.hitmill.com/programming/cpp/cppHistory.html>
- » http://www.techotopia.com/index.php/The_History_of_Objective-C
- » <http://osxbook.com/book/bonus/ancient/whatismacosx/history.html>
- » http://www.osnews.com/story/3757/Introduction_and_History_of_Darwin/page2/

- » <http://www.morphosppc.com/article/darwin-operating-system-history>
- » <http://www.insanelymac.com/forum/lofiversion/index.php/t57650.html>
- » <http://www.insanelywind.com/msiarhive/viewtopic.php?f=32&t=11459>
- » <http://opensource.apple.com/source/boot/boot-132/1386/boot2/boot.c>
- » <http://hints.macworld.com/comment.php?mode=view&cid=30815>
- » <http://superuser.com/questions/163941/what-is-the-vol-directory-for>
- » <https://developer.apple.com/library/mac/documentation/DeviceDrivers/Conceptual/WritingDeviceDriver/Glossary/Glossary.html>

Подпишись на журнал Linux Format



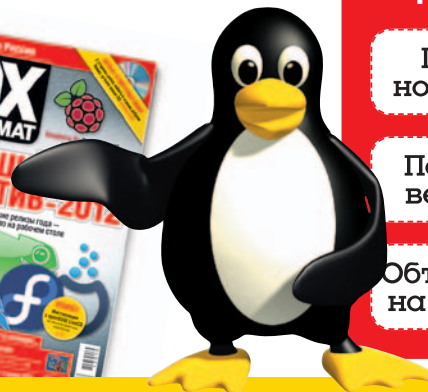
Оформи подписку на 2013 год на печатную версию журнала Linux Format и получи бонусы!

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



Что за штука...

Gnome OS

Самое противоречивое рабочее окружение Linux готовится к новым испытаниям, сообщает **Майк Сондерс**.

В О нет, только не это... ну зачем все превращать в ОС? В прошлом месяце была Firefox OS, и вот опять!

О Ну, это не новое явление. Emacs функционирует как операционная система еще с 1980-х... хотя да, «платформа» – модное словечко в этом году, вот и команда Gnome тоже не пожелала довольствоваться написанием множества библиотек и программ для базовой системы, кото-

рая им не принадлежит. Точнее сказать, они желают заполучить контроль над всей системой взаимодействия Gnome с его пользователями.

В Но ведь это уже так и есть – они проектируют весь интерфейс пользователя целиком.

О Верно, но то, что мы видим – лишь малая толика работы всей системы. Давайте немного отвлечемся и представим, как работает типичный

дистрибутив Linux. В основе всего лежит ядро; оно взаимодействует с оборудованием, запускает программы и распределяет ресурсы между ними.

Далее идет оконная среда – графический слой, разрабатываемый другой группой. Затем – графические инструментари, за которые отвечает третья группа, затем рабочее окружение, и т.д.

Такой распределенный подход был довольно успешен в последнее десятилетие, но когда все

эти команды действуют независимо друг от друга, добиться слаженности системы в целом было очень трудно. Иногда в Linux требуется, чтобы функция реализовывалась и на системном уровне, и на уровне пользовательского интерфейса, и в таком случае привлечь все эти группы к сотрудничеству – все равно что пасти кошек.

Представим такую ситуацию: поставщик услуг широкополосного доступа к Интернет для мобильных телефонов хочет осуществить поддержку Linux. Помимо USB-модема, также потребуются и низкоуровневый компонент (драйвер), и высокоуровневый (программы доступа).

В настоящий момент компании для этого придется перерывать уйму разных дистрибутивов Linux,

единственной формой существования для этого популярного рабочего окружения.

Он предназначен для тестирования и разработки – в качестве эталонной реализации системы Gnome.

В идеале, Gnome OS представит функции, идеи и технологии, которые впоследствии будут усвоены основными дистрибутивами, делая Gnome более стабильным на любом из них. Со временем, разработчики приложений смогут тестировать на Gnome OS свои программы, которые будут столь же легко работать на производных от него.

Сравните это с текущей ситуацией, когда разработчик не может просто написать приложение для Gnome и поделиться им со всеми. Вместо

О И да, и нет. Несмотря на то, что по поводу Gnome 3 было столько споров, и столько теперь уже бывших фанатов Gnome были недовольны теми радикальными изменениями, разработчики Gnome по-прежнему убеждены, что настольный вариант важен. По словам мистера Дэй, «существующие типы устройств – в первую очередь, ноутбуки и рабочие столы – должны оставаться в центре внимания Gnome».

Одновременно продолжается работа над оптимизацией Gnome для сенсорных устройств, и нам интересно, как все сложится здесь.

Как бы там ни было, интересно, что инициатива этого проекта возникла у создателей Gnome именно сейчас, когда так много пользователей просят их решить более насущные проблемы с Gnome 3.

Конечно, многим новый дизайн понравился, но он по-прежнему сеет раздоры в сообществе Linux, вызывая множество нареканий, которыми пестрят все новые сайты и блоги.

Немало давних поклонников Gnome сейчас чувствуют, что их мнение игнорируют, и если бы решение этих проблем вошло в планы сообщества, это бы сняло напряжение.

В Ну, хорошо, допустим, вы меня почти уговорили. Возможно. Когда же мы увидим плоды трудов команды Gnome?

О Первый релиз ожидается в марте 2012. Нам это кажется достойным и выполнимым, и также подразумевает, что на него не будут оттянуты все силы, так что будет время параллельно заниматься проблемами Gnome 3.

Следить за развитием событий можно на <http://live.gnome.org/GnomeOS/Design/Whiteboard>, где представлены разные идеи реализации новой ОС и изменений пользовательского интерфейса. И будьте уверены: как только у команды Gnome будет, что показать, мы осветим это и здесь, в LXF.

В А вдруг из этого начинания ничего не выйдет?

О Не будьте таким пессимистичным! Нужно по крайней мере дать им шанс. Главное – то, что мы получим новые технологии и идеи для улучшения интеграции рабочих окружений. Linux, несомненно, является сокровищницей кода и таланта в мире вычислений, но по сей день существует проблема, как использовать эти ресурсы для создания цельной настольной ОС. Если усилиями команды Gnome станет проще использовать Linux, улучшится взаимодействие дистрибутивов и устранился двойная работа, от этого выиграют все.

Кто знает, даже если сейчас вы возненавидели Gnome 3, быть может, настанет время, когда вы его простите... **LXF**

«Gnome OS станет дистрибутивом Linux, но не вытеснит традиционные.»

пакетных менеджеров, файловых систем, библиотек, интерфейсов и т.п. Это адский труд.

В Разумеется, но таков уж удел разработчика! Кто-то же должен выполнять за нас эту грязную работу?

О Не спорю. Но посмотрите на эту ситуацию с точки зрения пользователя. Вот некий новичок впервые установил Linux и знает, что имеет дело с окружением Gnome. Однажды в Интернете он набредает на крутую новую программу. Хочет ее установить, а доступные пакеты – для другого дистрибутива. И он жалуется: «Почему же я пользуюсь Gnome и не могу установить для него – его же программы?»

Все сводится к тому, что разработчики приложений не могут писать для некоего абстрактного Gnome. Им приходится писать для множества его реализаций, сталкиваясь с таким количеством версий, со столькими внутренними различиями, что эта задача становится просто неподъемной.

Аллан Дэй [Allan Day], дизайнер интерфейса в Gnome, сказал: «В настоящий момент создавать и распространять приложения для Gnome необычайно сложно. Наши API постоянно совершенствуются, а процесс создания приложений остается медленным и раздробленным».

В Чем же тут поможет лишний дистрибутив Linux? Мне кажется, их и без того предостаточно...

О Gnome OS действительно будет дистрибутивом Linux, но он создается не затем, чтобы вытеснить традиционные, и не затем, чтобы стать

этого ему приходится создавать пакеты под разные дистрибутивы, использующие Gnome, с учетом их особенностей и различий, затем добавлять репозиторий и разбираться со всеми вытекающими техническими нюансами.

В Выходит, это будет еще и пакетный менеджер? Да их тоже хоть отбавляй.

О Типа да. Команда Gnome пока не представила на этот счет никаких конкретных спецификаций, но их цель ясна. Печально известно, что донести приложения до пользователей Linux трудно, если только вы не используете дистрибутивы с непрерывным циклом обновления (такие как Arch) и не дружите с поддержкой пакетных менеджеров.

Браться за создание приложений самому – тоже не так просто, как мы описали выше. Если бы пользователи, независимо от дистрибутива, сразу получали доступ к новейшим приложениям Gnome, их создателям тоже было бы куда лучше. При наличии надежных средств разработки и актуальных API писать для Gnome станет значительно легче.

В частности, это сделает Linux более привлекательным и для разработчиков проприетарных приложений, ведь они смогут сконцентрироваться на переносе кода, не тратя силы на миллионы вариаций дистрибутивов и менеджеров управления пакетами.

В Значит ли это, что Gnome отказался от своей навязчивой идеи стать интерфейсом для планшетных компьютеров?



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.

Энтропия в кране

Случайные числа применяются весьма широко – для генерации криптографических ключей, компьютерного моделирования и прочего, не говоря уж о газовых компаниях, которые с их помощью определяют, когда должен (и должен ли) прийти инженер по техобслуживанию. Компьютеры легко генерируют повторяемые псевдослучайные последовательности (это делает библиотечная функция `rand()`), но для получения истинно случайных чисел нужна «энтропия» от какого-либо физического источника случайной активности: от источника радиоактивного распада, бросания игральной кости или наблюдения за временем появления инженера по техобслуживанию.

Запасы энтропии внутри компьютера, как ни странно, невелики. Небольшое количество энтропии ядро получает от некоторых драйверов (например, при перемещении мыши) и делает ее доступной через файл устройства `/dev/random`. Но, запустив `$ hexdump /dev/random` вы увидите всего несколько десятков байт данных; затем программа приостановится, ожидая дальнейших случайных данных (подробно команда описана в `man 4 random`).

Исследователи группы квантовой оптики Национального университета Австралии разработали генератор подлинно случайных данных, на базе наблюдений за квантовыми флуктуациями вакуума. Он генерирует случайные числа со скоростью 5 Гбит/с, и, подключившись к нему через Интернет, вы сможете загрузить свой случайный поток. Это web-сервис стиля JSON, и ряд открытых проектов для Linux предоставляют библиотеки для доступа к его данным из вашего кода.

Детали см. на <http://photonics.anu.edu.au/qoptics/Research/qrng.php>. Сервис с более быстрым доступом – <http://random.org>. На нем есть эмуляции подбрасывания монеты, бросания игральной кости и выбора номера лотереи (и многие другие) по данным атмосферного шума.
chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Mondo и Mindi

Быстро ли вы восстановите систему при сбое жесткого диска? С *Mondo* это делается за несколько минут.

Как вы делаете резервные копии? Для домашних систем отлично работают простые решения, например, запись tar-архива с важными файлами на жесткий диск. Для систем уровня предприятия есть хитроумные централизованные решения, такие как *Amanda*, *Bacula* и *BackupPC*. Но большинство из них не восстанавливают файлы на «голом железе» – если жесткий диск выйдет из строя, то с ними сначала придется переустановить ОС и только потом восстановить содержимое из резервной копии.

Mondo Rescue (www.mondorescue.org) решает эти проблемы, позволяя создать загрузочный диск с выбранными файлами и каталогами для «голового железа». *Mondo* распространяется по лицензии GPL в пакетах для большинства крупных дистрибутивов Linux (Fedora, RHEL, openSUSE, SLES, Mandriva, Mageia, Debian, Ubuntu и Gentoo) и может создавать копии файловых систем различных типов (ext2, ext3, ext4, reiserfs, NTFS и дру-



Первое окно сбора информации при запуске *Mondo* в интерактивном режиме.

гих с дисков IDE, SCSI, LVM, программного и аппаратного RAID) на магнитной ленте, дисках, сетевых дисках или на CD/DVD. Программу можно запустить в интерактивном режиме, и она проведет вас по этапам создания копии и соберет нужную ей информацию, или с командной строки.

Для создания образов на CD *Mondo* применяет сопутствующую программу – *Mindi*, создающую образ по вашему ядру, модулям ядра, библиотекам и утилитам, и при загрузке этого образа мы получим среду, подобную той, резервная копия которой создавалась. Имея CD от *Mindi*, можно выполнить интерактивное восстановление, т.е. восстановить отдельные компоненты архива, или базовое восстановление – тогда диск будет разбит на разделы с нуля и на него будет восстановлена вся файловая система. Можно также клонировать существующую систему.

Как узнать больше

Нажмите на ссылку с документацией на сайте <http://mondorescue.org/>, и вы найдете map-страницы для *mondoarchive*, *mondorestore* и *mindy*. Там же есть и подробное руководство, но оно подразумевает определенное знание командной строки. По ссылке <http://trac.mondorescue.org/browser/devel/mondo-doc/mondo-presentation.odp> есть хорошая презентация.

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

Linux  center

Итак, вы хотите стать сисадмином?

Четвертая часть серии, которая превратит вас из новичка в звезду системного администрирования. На сей раз займемся управлением пакетами.



Месяц назад мы говорили об управлении жесткими дисками серверов Linux. В этом выпуске мы рассмотрим основы управления пакетами, то есть главным образом команды **rpm** и **yum**.

Во всех статьях данной серии мы пользуемся CentOS 6.2. Если вы хотите следовать за нами (что я настоятельно рекомендую), установите CentOS (хотя бы в виртуальную машину) в соответствии с моим описанием в первой части.

У многих команд из обсуждаемых в этом месяце большой объем выходных данных, и на все здесь не хватит места. Если вам нужны подробности, то полный транскрипт многих из сегодняшних команд есть на DVD и на нашем сайте, так что смелее обращайтесь к этому файлу в соответствующих местах.

Великий раскол дистрибутивов

Управление пакетами – наверное, одна из областей, в которых дистрибутивы Linux отличаются друг от друга больше всего; главным образом это отличие между дистрибутивами, в которых используются пакеты и утилиты формата Red Hat (это Red Hat, CentOS и Fedora), и теми, в которых используются пакеты и утилиты формата Debian (это Debian, Ubuntu, Mint и множество других производных). Так как на уроках я пользуюсь CentOS, то сосредоточусь только на утилитах Red Hat. И, желая познакомиться с утилитами Debian, типа **dpkg** и **apt-get**, вы не найдете искомого.

Начнем с начала

Итак, что же такое пакет? По сути, это архив из файлов, которые будут скопированы в систему при установке пакета. Обычно это несколько двоичных файлов, один или несколько файлов настройки, возможно, несколько вспомогательных библиотек и, зачастую, несколько **man**-страниц и другая документация. Пакет также содержит метаданные, в которые входят краткое описание пакета, цифровая подпись и некоторая информация о зависимостях. В системах формата Red Hat пакеты находятся в файлах с расширением **.rpm**.

➤ Для создания разделов с **fdisk** используются однобуквенные команды. Круто? Не очень. Надежно? Да!

Имена этих файлов четко структурированы. Например, разберем имя файла пакета **aide-0.14-3.el6.i686.rpm**. Имя пакета – **aide**, а версия – 0.14 (номер выбирают разработчики пакета). Версия сборки – 3.el6 (номер выбирают создатели пакета), и пакет собран для архитектуры i686 (32-битная архитектура Intel).

Другой пример: пакет **yum-3.2.29-22.el6.centos.noarch.rpm** не привязан к конкретной архитектуре процессора (**noarch**). Такое возможно потому, что данный пакет написан на Python, а это интерпретируемый скриптовый язык.

Прежде чем продолжить, я хотел бы немного поговорить о зависимостях. Зависимости пакета – это программы, которые нужно установить, чтобы пакет работал. Распространенные примеры зависимостей – вспомогательные библиотеки, утилиты командной строки и используемые пакетом внешние сервисы. Устранение отсутствующих зависимостей иногда может стать серьезной проблемой – подробнее мы займемся ими далее. Утилита, используемая для просмотра, установки и обновления пакетов RPM, называется также **rpm**. Изначально это было сокращение от Red Hat Package Manager [менеджер пакетов Red Hat]. Эта команда – мастер на все руки, и параметров у нее даже больше, чем необходимо. Чтобы продемонстрировать ее работу, требуется источник файлов RPM. На установочном DVD CentOS (или на ISO-образе, если система запущена на виртуальной машине) есть каталог **Packages** с более чем 3000 пакетами, поэтому воспользуемся ими.

Во-первых, номер версии пакета можно узнать таким образом:

```
$ cd /media/CentOS_6.2_Final/Packages
$ rpm -qp aide-0.14-3.el6.i686.rpm
aide-0.14-3.el6.i686
```

Она не сообщила нам ничего нового, потому что, как сказано выше, номер версии пакета есть в имени файла.

Содержимое пакета можно просмотреть командой

```
$ rpm -qip aide-0.14-3.el6.i686.rpm
```

Если вам было интересно, что же такое **aide**, вывод команды вам все расскажет. Это средство обнаружения атак на основе проверки целостности файлов (полный вывод команды есть в файле транскрипта).

Список файлов пакета выдаст команда

```
$ rpm -qlp aide-0.14-3.el6.i686.rpm
```

Сборка из исходников

Поскольку исходные коды программ для Linux в основном открыты, весьма заманчиво устанавливать новые программы путем загрузки и компиляции файлов-исходников. На сайтах вроде sourceforge.net есть исходные коды тысяч программ. Но если вы профессиональный администратор, обслуживающий десятки серверов, я советовал бы вам прибегать к этому способу в последнюю очередь: если для необходимой программы нет двоичного пакета или вам обязательно нужна самая свежая версия программы, так как в ней исправлена ошибка или появилась новая

возможность. Почему я против установки из исходников? Потому что тогда на вас ложится большая ответственность – вы должны убедиться, что пакет собран правильно и все зависимости разрешены. Стандартные утилиты управления пакетами ничего об этом не знают, и удалить такой пакет бывает непросто.

Нужно отметить, что есть и смешанный подход: начать с исходного кода (в виде RPM с исходными кодами), затем настроить и собрать его в двоичный пакет RPM, который затем устанавливается обычным образом.

В нем вы увидите один исполняемый файл (`/usr/sbin/aide`), файл настройки (`/etc/aide.conf`) и немного документации, в том числе несколько man-страниц. Также есть файл `/etc/logrotate.d/aide`. Поместив его в каталог `logrotate.d`, мы автоматически настраиваем `logrotate` на ротацию лог-файлов, создаваемых `aide`. Кстати, постоянные читатели этой рубрики, возможно, помнят, как я жаловался на явление, которое назвал «болезнью точка-d». Эта хворь поразила многие системные программы, так что вместо одного файла настройки (например, `/etc/logrotate.conf`) у них возник целый каталог таких (в данном случае, каталог `/etc/logrotate.d`). Ну, а здесь мы видим преимущества этого подхода. Дополнительный файл настройки можно просто перекинуть в должное место, не возясь со скриптами, чтобы добавить запись в существующий файл.

Повышаем запрос

Теперь рассмотрим запрос к уже установленному пакету. Команда выглядит точно так же, не считая того, что нужно опустить опцию `p` и указать только основу имени пакета, а не все имя. Так, чтобы узнать номер нашей версии `Bash`, командуем:

```
$ rpm -q bash
bash-4.1.2-8.el6.centos.i686
# rpm -qi bash
# rpm -ql bash
```

В дополнение к вопросу «какие файлы в этом пакете?» можно спросить «из какого пакета этот файл?» и получить ответ так:

```
$ rpm -qf /bin/ping
iputils-20071127-16.el6.i686
```

Наконец, список всех установленных пакетов вы получите по команде

```
$ rpm -qa
```

Приготовьтесь, он может оказаться весьма длинным!

Ладно, довольно запросов. Установим-ка пакет `aide`. Обратите внимание, что для этого нужно быть администратором:

```
# cd /media/CentOS_6.2_Final/Packages
# rpm -ivh aide-0.14-3.el6.i686.rpm
Preparing... ##### [100%]
1:aide ##### [100%]
```

Программа `rpm` сохраняет информацию об установленных пакетах в `/var/lib/rpm`.

Установка `aide` прошла довольно легко, но так бывает не всегда. Например, попробуем установить `amanda` (утилиту резервного копирования):

```
# rpm -i amanda-2.6.1p2-7.el6.i686.rpm
error: Failed dependencies: xinetd is needed by amanda-2.6.1p2-7.el6.i686
```

[Ошибка зависимостей: `xinetd` требуется для `amanda`.] Здесь установка завершилась неудачно, потому что у `amanda` есть зависимость (программа `xinetd`), которая не установлена. В данном случае это довольно легко исправить – просто установите пакет `xinetd` первым. Но иногда список неразрешенных зависимостей гораздо длиннее, и не всегда понятно, какие пакеты нужно установить для их разрешения. Здесь `rpm` исчерпывает свои возможности.

Знакомьтесь: желтая собака

Что плавно подводит нас к `yum`. Это сокращение от “YellowDog Updater, Modified” [Программа обновления Желтой собаки, модифицированная], и если вы хотите понять, что это значит, разрешаю обратиться к Google. `Yum` – средство управления пакетами более высокого уровня, и у него есть два ключевых отличия от `rpm`. Во-первых, `yum` обычно загружает пакеты из интернет-репозитория (хотя с его помощью можно устанавливать пакеты и с DVD

Удаление пакетов

Для удаления пакета достаточно скомандовать

```
# rpm --erase aide
К счастью, rpm достаточно разумна, чтобы не удалять пакеты, если от них зависят другие установленные пакеты. Поэтому, если попробовать выполнить
# rpm --erase mindi
error: Failed dependencies:
mindi >= 2.0.7 is needed by (installed) mondo-3.0.2-1.rhel6.i386
rpm вполне справедливо откажется удалить mindi, так как тот нужен (установленному) пакету mondo.
```

CentOS). Во-вторых, `yum` разрешает зависимости, автоматически установив все необходимые пакеты. Перейдем прямо к делу и установим с его помощью `amanda`:

```
# yum install amanda
Dependencies Resolved [Зависимости разрешены, установлено]
Installing : 2:xinetd-2.3.14-35.el6_3.i686 1/2
Installing : amanda-2.6.1p2-7.el6.i686 2/2
Complete! [Готово!]
```

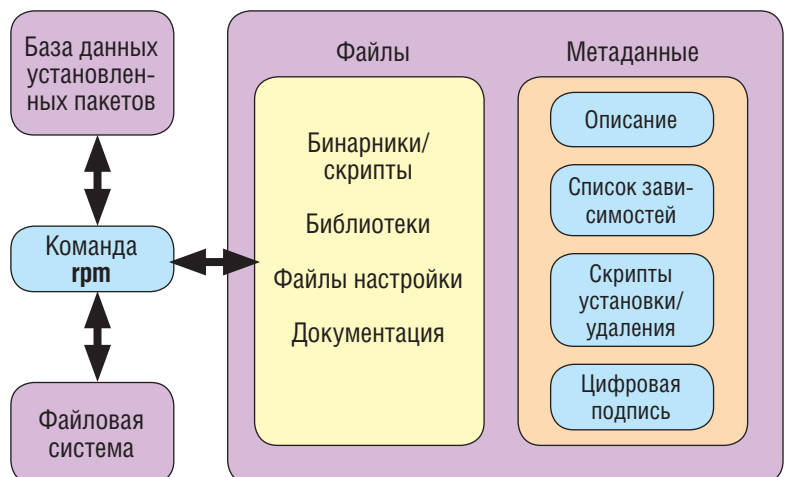
Я существенно сократил вывод, но оставил достаточно, чтобы вы ясно увидели: `yum` автоматически установил `xinetd` для разрешения зависимости. Еще больше упростить установку пакетов нельзя. Однако учтите, что установка пакета для какого-либо сервиса не запускает сам сервис – в этом примере он даже не добавляется в файл настройки `xinetd` (этот подход отличается от используемого в Debian, там при установке сервиса тот обычно приводится в базовое рабочее состояние).

У `yum` есть файл настройки – естественно, в `/etc/yum.conf`. Но вас скорее заинтересуют файлы в каталоге `/etc/yum.repos.d` (опять болезнь «точка-d!»), где задается расположение репозитория. По умолчанию у вас будет файл `CentOS-Base.repo`, с пятью репозиториями (`base`, `updates`, `extras`, `centosplus` и `contrib`), хотя два последние по умолчанию отключены.

Рассмотрим один из разделов этого файла:

```
[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
```

»



» В мире не один формат пакетов, но большинство из них соответствуют этой схеме.

Где узнать больше

Прочтите man-страницы для *rpm* (<http://www.rpm.org/max-rpm/rpm.8.html>) и *yum* (<http://linux.die.net/man/8/yum>). На man-странице *yum.conf* вы узнаете, что можно добавить в файлы *.repo*.

В качестве полного руководства, где есть и инструкции по сборке собственных пакетов RPM, обратите внимание на прекрасную книгу Эрика Фостера-Джонсона [Eric Foster-Johnson] по ссылке http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide – она поможет вам скоротать время долгими зимними вечерами.

```
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

Вместо конкретного адреса репозитория можно указать *yum* список адресов для загрузки в параметре *mirrorlist*. При желании можно указать конкретный URL (параметром *baseurl*), и адреса сторонних репозиториях обычно задаются именно так.

Для данного репозитория настроена проверка цифровых подписей пакетов (и это хорошая идея), а параметр *gpgkey* указывает путь к ключу. Обратите внимание, что это локальный файл, который является частью стандартной установки CentOS, чтобы CentOS по умолчанию знал публичные ключи своих репозиториях. Если вы включите другие репозитории (скоро мы это сделаем), понадобится загрузить дополнительные ключи.

У *yum* можно спросить об известных ей репозиториях следующей командой:

```
# yum repolist
repo id repo name status
base CentOS-6 - Base 4,776
extras CentOS-6 - Extras 4
updates CentOS-6 - Updates 374
repolist: 5,154
```

Сторонние пакеты

В сторонних репозиториях можно найти массу программ для CentOS. Если вы захотите их установить, понадобится добавить файлы *.repo* в */etc/yum.repos.d*, чтобы *yum* знал, где искать репозитории. В качестве примера установим утилиту восстановления системы *Mondo*, о которой я рассказывал ранее. Для начала убедимся, что ее нет в стандартных репозиториях:

```
# yum search mondo
```

```
... < несколько строк вывода опущено > ...
No Matches found [Совпадений не найдено]
```

Итак, нам придется добавить в каталог *yum.repos.d* файл (*mondorescue.repo*) со следующим текстом:

```
[mondorescue]
name=rhel 6 i386 - mondorescue Vanilla Packages
baseurl=ftp://ftp.mondorescue.org/rhel/6/i386
enabled=1
gpgcheck=1
gpgkey=ftp://ftp.mondorescue.org/rhel/6/i386/mondorescue.
pubkey
```

Кстати, я не написал все это сам, а просто загрузил этот файл с сайта *mondorescue*. Обратите внимание, что мы велит *yum* импортировать публичный ключ с сайта *mondorescue*, чтобы можно было проверить цифровые подписи пакетов.

Здесь, вспомнив о безопасности, я должен предупредить вас, что по сути мы здесь даем *yum* право устанавливать пакеты с сайта от имени суперпользователя-root. Если вы не доверяете авторам ПО, не делайте этого! Мало ли какой ущерб могут принести их пакеты? Вообще, если вы работаете в корпоративной среде, проверьте, есть ли в вашей компании политика полного запрета сторонних репозиториях.

С указанным файлом наш поиск стал более плодотворным:

```
# yum search mondo
... < несколько строк вывода опущено > ...
===== N/S Matched: mondo =====
mondo.i386 : MondoRescue is a GPL Disaster Recovery and Cloning Solution
```

Теперь можно установить пакет:

```
# yum install mondo
... < опущено множество строк вывода > ...
Installed:
mondo.i386 0:3.0.2-1.rhel6
Dependency Installed:
afio.i386 0:2.5-1.rhel6
buffer.i386 0:1.19-4.rhel6
mindy.i386 0:2.1.3-1.rhel6
mindy-busybox.i386 0:1.18.5-1.rhel6
syslinux.i686 0:4.02-7.el6
Complete!
```

Как видите, здесь *yum* установил пять дополнительных пакетов для разрешения зависимостей. У *yum* есть и другие козыри, например, поиск, и я советую вам почитать man-страницу.

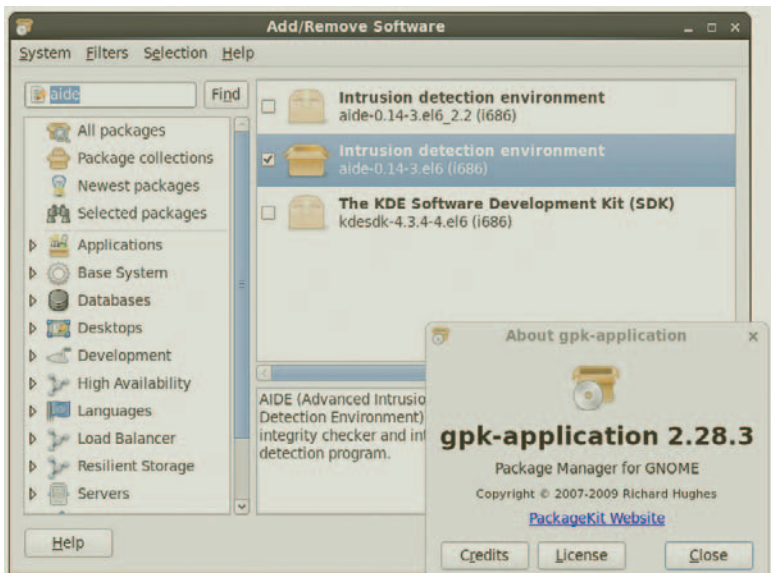
Места у меня осталось только на то, чтобы вновь вернуться к безопасности и рассказать вам еще об одной возможности *rpm*. При указании параметра *--verify* команда сообщит о любых изменениях в файлах пакета с момента установки. Например, выполните команду:

```
# rpm --verify xinetd
Она не выведет ничего: пакет с момента установки не изменился. Зато команда:
$ rpm --verify initscripts
..5...T. c /etc/inittab
```

говорит нам, что у файла */etc/inittab* изменились размер, контрольная сумма MD5 и временная отметка по сравнению с изначально установленным (подробную информацию о выводе команды можно получить, поискав “differs” на man-странице *rpm*). Изменения в файлах настройки после установки – дело обычное, но если вы видите изменения, например, в двоичных файлах системы, это может быть поводом для опасений. Таким образом, эту команду можно использовать как простейшее средство обнаружения вторжений.

В следующем месяце мы дадим нашему серверу настоящий повод для жизни, установив на него несколько сервисов. Увидимся! **LXF**

➤ **Страшитесь командной строки? В менеджере пакетов Gnome можно искать, просматривать, устанавливать и удалять пакеты мышью.**



Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Вживание в Unity

Ничто так не нуждается в исправлении, как чужие привычки.
Сэмюэл Лэнгхорн Клеменс

Наконец я сдался и поставил дома на компьютере Ubuntu с настройками из коробки (до этого я упорно использовал fvwm). На нетбуке он у меня давно, и в принципе меня устраивает. В основном потому, что нетбук нужен разве что для просмотра новостей и почты в момент заварки утреннего чая. Другое дело – рабочий, хоть и домашний, стационарный ПК. Я тут тексты набираю – в частности, этот.

Жить с Ubuntu можно. Что бы ни плели злопыхатели, это вполне себе нормальный дистрибутив GNU/Linux, чего никакому Unity не скрывать. Даже и плюсы есть – правда, не могу вспомнить, какие. Но кое-что бесит. Не понимаю, почему на клик мыши на пустом десктопе нельзя повесить пользовательское меню (мне нужно-то в нем всего с пяток пунктов). Не ахти какой каприз, но попытки выяснить, как это делается, упираются в стену. Всплывающее меню по кнопке Super – интересная затея, но где же поиск по набираемому пути? А еще напрягает Ubuntu One, которая работать стабильно явно не в силах. Ответы на вопросы «Кто виноват?» и «Что делать?» скрыты в тумане, и все чаще хочется не разбираться, а перезапуститься – вдруг самопочинится? Я всеми руками за удобства, но почему они так сложны?

P.S. Любимый геймерами DRM Steam for Linux стал реальностью.

P.P.S. LOR'у стукнуло 14 лет. Он еще молод, и у него все впереди – надо хотя бы школу закончить...

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Играть на Pi 70
1980-е были золотым веком компьютерных игр. Вооружитесь кусачками, собирайте корпусов от старых машинок и оживите эту эпоху вместе с **Дэвидом Хейвардом**.



Отбивать атаки 74
Шашанк Шарма намерен взять приبلудные пакеты в своей сети под контроль раз и навсегда. У него ни один неаутентифицированный байт мимо не проскочит...



Мониторить систему 78
Хотите подслушать, о чем на самом деле думает ваша система? Присоединитесь к **Нейлу Ботвику** – он роется в файлах системных журналов.



Создавать библиотеки 80
Под занавес данной серии уроков **Ник Вейч** напишет персональные библиотеки для своего легендарного Arduino.



Вникать в Erlang 84
Довольно теоретических рассуждений! **Андрей Ушаков** засучивает рукава и переходит к практическим действиям по многозадачности.



Многостаночничать ... 88
Предоставлять по компьютеру на одного человека – это вчерашний день. **Лео Максвелл** подключает два монитора...

Пропустили номер?

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





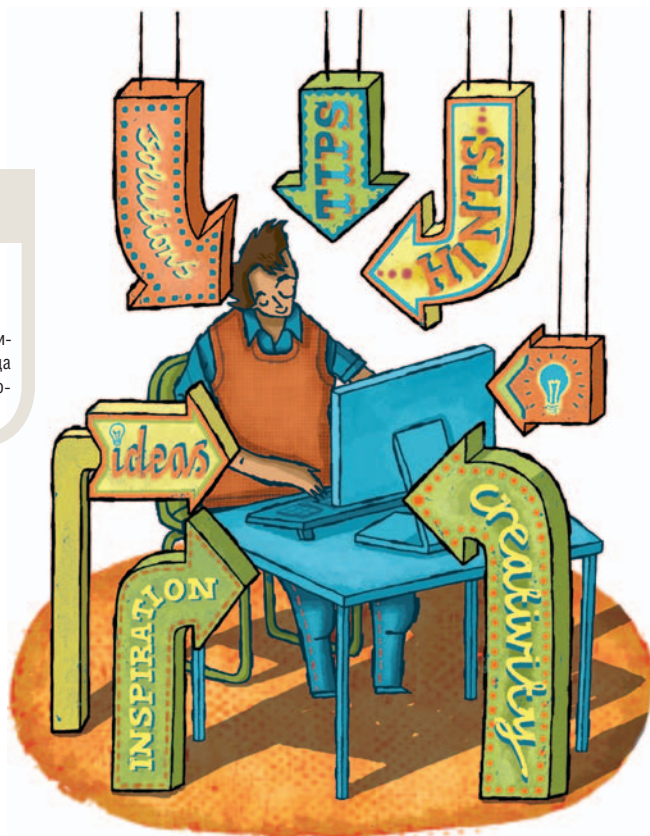
Raspberry Pi:

Дэвид Хейвард возвращает бездарно потраченную юность, эмулируя золотую эру домашних компьютерных игр 1980-х с помощью Pi.



Наш эксперт

Дэвид Хейвард вырос среди тюленей, природных врагов пингвинов, но это никогда не мешало ему любить Linux.



бов «The Perfect Pint», – прим. пер.] и пляж развлечений в Большом Ярмуте – лишь часть из них. Но один обычай стоит плечом к плечу с культовыми памятниками, придающими Британии величие: он породил целое поколение свободных творческих мыслителей. Этот обычай – не что иное, как Великая революция 8-битных домашних компьютерных вычислений. Плоды ее прочно возглавили список игр восьмидесятых, а некоторые из любимейших фигурантов компьютерной истории сегодня украшают собой страницы eBay.

Чтобы воскресить эти ушедшие дни, часто приходится устанавливать эмуляторы в наши громадные компьютеры, что лишает нас части того духа, который олицетворял золотой век домашних компьютеров. Конечно, есть и другая возможность: можно купить один из наших любимых компьютеров и подключить его к старому плазменному телевизору с диагональю 52 дюйма. Однако эти компьютеры уже не юны, и если о них не заботились должным образом, они могут сломаться при подвешивании их на стену.

Но есть альтернатива: сочетание кое-чего современного и кое-чего не слишком современного. Raspberry Pi – возможно, одна из лучших инноваций со времен появления домашних компьютеров, которая возвещает новую золотую эру компьютеров. Этот компьютер размером с банковскую карту находит серьезное применение в творческих проектах: отправляется к границам космоса, превращается в беспроводной приемник из сороковых, становится средством автоматизации дома и используется во многих других проектах, которые доказывают, насколько изобретательными и гибкими являются пользователи и RPi.

Поэтому мы последуем их примеру и посмотрим, что получится, если прикупить кое-что на eBay, взять Raspberry Pi и лист самоклеящегося пластика и, воспользовавшись помощью растущего сообщества пользователей RPi, воссоздать классический домашний компьютер восьмидесятых.

На этом королевском острове [намек на одноименную радиопостановку BBC, – прим. пер.] много прекрасных национальных традиций: рыба и чипсы [любимое блюдо англичан, – прим. пер.], «идеальная пинта» [сеть британских па-

ZX Pi

Наша первая остановка – это милое чудо с резиновой клавиатурой, известное в народе как Спессу. ZX Spectrum 48k и в самом деле стал выдающимся достижением своего времени, мгновенно

превратив доморощенных программистов в коммерческих гигантов сферы разработки ПО, а школьники могли явиться домой после уроков и просидеть за ним до ужина, кропая игры, вошедшие в анналы компьютерной истории.

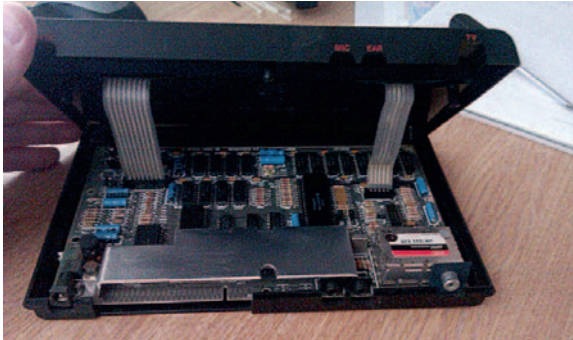
Но довольно ностальгии – от нее сыреют глаза. Побродив по страницам eBay, мы нашли нерабочий ZX Spectrum за плывые 2,99 фунта – в конце концов, разобрать живого Спессу было бы кошунством. Он выглядел не очень хорошо из-за износа, но и большинство из нас, некогда игравших в такие, сегодня тоже не цветут пышным цветом.

Нашей первой задачей было проверить, не устарел ли RPi; мы экспериментировали с ним с момента его появления у нас в почтовом ящике, и поскольку в мире Pi все быстро меняется, мы подумали, что пора обновить его на Wheezy. Ссылку для загрузки системы можно найти на сайте Raspberry Pi, goo.gl/4w4ps, с подробным руководством по ее копированию на SD-карту



➤ Великий ZX Spectrum. Благодарности его Господы!

Ретро-игры



› Внутренности ZX Spectrum, готовые к снятию.

и настройке. Затем мы выполнили команду `sudo apt-get update/upgrade`, и через несколько минут наш RPi обновился и прекрасно заработал.

Затем последовал демонтаж ZX Spectrum, сводящийся к откручиванию пяти винтов на нижней части машины и снятию шлейфа клавиатуры. Материнская плата имела один центральный винт, который после удаления оставил только голый пластмассовый корпус.

RPi существенно меньше настоящей материнской платы Spectrum, и нам пришлось малость потрудиться, чтобы удобно и надежно расположить компоненты RPi в новом корпусе. Мы обнаружили, что порты RPi – RCA Video и Audio прекрасно совпали с портами Spectrum – Mic и Ear, но разъем для подключения SD-карты RPi уперся в стенку пластмассового корпуса. Легкое движение кусачек – и у нас получился самодельный слот для SD-карт, доступный даже при закрытом корпусе; расширив этот слот, мы также смогли запитать устройство от зарядника HTC.

Затем мы подключили HDMI и Ethernet и с помощью старой доброй черной изоленты закрепили RPi на основании корпуса Spectrum, а также закрепили кабели на корпусе, чтобы они не вырывали RPi с ее законного места, если за них дернуть. Тут у нас возникла проблема с клавиатурой. Хотя до нас уже был про-



› Внутри старого Спессу для RPi много места.

ект такого рода – парень по имени Брайан пересобрал ZX Spectrum на BeagleBoard [дешевый одноплатный компьютер, совместная разработка компаний Texas Instruments и Digi-Key, – прим. пер.] и клавиатура у него успешно заработала (см. goo.gl/V5cH3) – наши попытки были не слишком удачны. Короче, мы полностью провалились, запутав шлейф клавиатуры в интерфейсе, который ободрали с обычной клавиатуры USB. Но решили плюнуть на это и подключили обычную клавиатуру и мышь через большой порт ввода/вывода Spectrum.

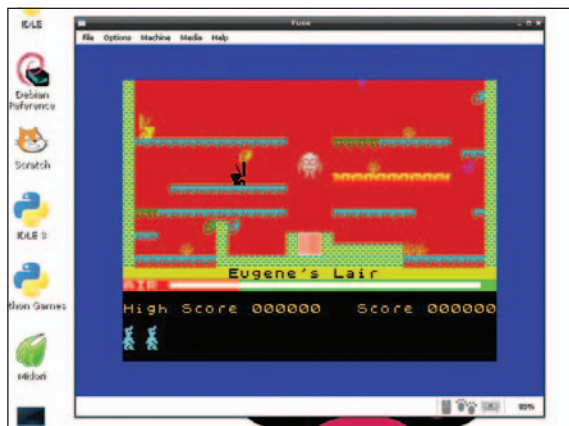
Когда мы надели крышку, ZXRPi выглядел не так уж плохо, хотя кое-что заставило бы Хита Робинсона [W. Heath Robinson, английский иллюстратор XX в., более всего известный рисунками эксцентричных механизмов, – прим. ред.] радостно перевернуться в гробу. Однако, когда мы подключили его к телевизору и элегантно, хотя и осторожно, уложили рядом, все это смотрелось вполне достойно. Нам оставалось только установить приличный эмулятор Spectrum и раздобыть какие-нибудь старые игры.

Установка и запуск эмулятора Spectrum были довольно просты; для этого проекта мы взяли *Fuse Emulator*. Чтобы установить

»



› Наша дань уважения Spectrum – ZXPi.



› Manic Miner, запущенный на ZXPi.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

и запустить его, выполните следующие действия. Зайдите в терминал, наберите

```
sudo apt-get install fuse-emulator-common
```

и нажмите Enter. Нажмите “y” для подтверждения загрузки и установки. По окончании установки вернитесь в командную строку, наберите

```
sudo apt-get install spectrum-roms fuse-emulator-utils
```

и нажмите Enter. Снова вернувшись в командную строку, наберите

```
sudo amixer cset numid=3 2
```

и опять-таки нажмите Enter.

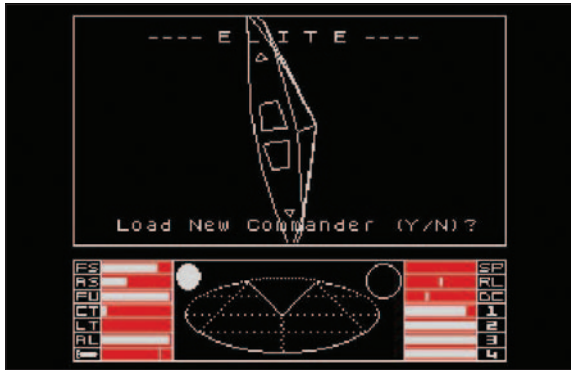
Эта команда включит звук через HDMI, но он довольно нестабилен, так что при необходимости замените «2» на «1», и звук перенаправится на звуковой порт RPi.

Потом выйдите из терминала, нажмите на кнопку Start LXDE (Запуск LXDE) и выберите Games > Fuse Spectrum Emulator (GTK+ Version) [Игры > Эмулятор Fuse Spectrum (версия для GTK+)]. Увеличьте размер окна, потянув за уголок.

Теперь загрузите какую-нибудь игру Spectrum, например, с сайта World of Spectrum (goo.gl/trJvd); затем в меню Fuse выберите Media > Tape > Open [Носитель > Лента > Открыть], после этого в окне Spectrum наберите “J” для загрузки игры, затем “Ctrl+PP” (что соответствует команде Load “”).

Игра должна загрузиться, а в звуковом выходе, к которому можно подключить стереонаушники, должен появиться звук. После этого останется только вернуться к заброшенному 30 лет назад *Manic Miner* и пройти его до конца.

➤ *Elite*, лучшая игра всех времен? Конечно, наш ZXРi — отличный дом престарелых для этого древнего шедевра.



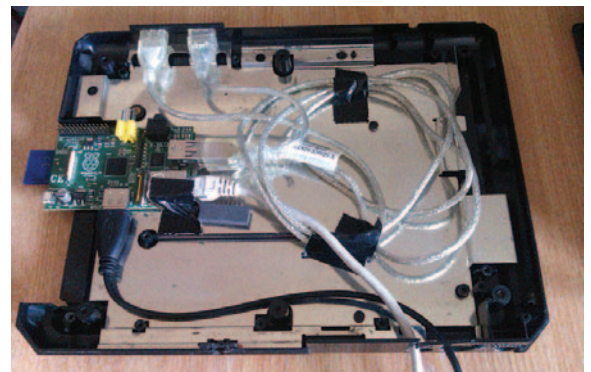
MegaPi

Хотя нам не удалось на 100 % воссоздать функциональность Spectrum, идея ретро-корпуса для Raspberry Pi нам очень приглянулась, и мы призадумались о других старых компьютерах, которые удовлетворили бы наши капризы.

Наше внимание привлекла старая знакомая — Sega Mega Drive, аккуратная, стильная даже по сегодняшним стандартам коробочка; из нее получился бы идеальный ретро-корпус для RPi. Но сначала нужно подготовить RPi для новой жизни внутри лучшей из консолей всех времен, и мы вознамерились играть на ней не только в игры для Sega Mega Drive, но и во все многообразие игр для «железа» той эпохи.

Для этого мы установили прекрасный проект RetroPie с сайта retroRockBlog — достаточно было следовать инструкциям с goo.gl/1sspF. В нашем случае мы решили скомпилировать самые свежие версии программ, скриптов и ядер, выбрав второй пункт главного меню способов установки — Source-based (custom) [На основе исходников (пользовательская)]. Однако, как отмечается на сайте, компиляция всего необходимого занимает несколько

➤ Все прекрасно уместилось, но с кабелями появились бы проблемы.



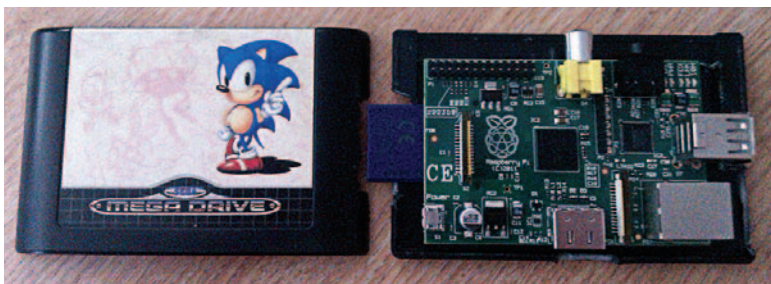
➤ RPi в Mega Drive: много свободного места и повторно использованные USB-порты в передней части консоли.

часов, и все это время ресурсы RPi будут заняты до упора, так что часа три ничего другого на нем делать не получится; но терпите — результат стоит ожидания. После окончания установки RPi получает заслуженный перерыв на отдых в виде перезагрузки. Затем достаточно лишь выбрать игру (мы предполагаем, что вы следуете рекомендациям по соблюдению авторских прав, приведенным во врезке внизу слева) и запустить ее в терминале следующим образом:

```
retroarch -L /home/pi/RetroPie/emulatorcores/Genesis-Plus-GX/libreto.so /home/pi/RetroPie/roms/megadrive/Sonic.md
```

Это мы загрузили игру *Sonic the Hedgehog*, прошивка которой помещалась в каталоге **roms** на “megadrive”. Вам, разумеется, нужно в этой команде изменить пути к эмулятору и к игре.

Купить нерабочую Mega Drive оказалось довольно просто, и она стоила всего несколько фунтов с доставкой, но вместо того, чтобы вытаскивать из нее внутренности, мы придумали кое-что похитрее: не получится ли разместить Raspberry Pi внутри картриджа Sega Mega Drive? Наш экземпляр Mega Drive выглядел немного изношенным, так как с момента своего рождения в 1992 году он-таки получил свою долю экшна, и мы подумали, что с Raspberry Pi в картридже мы лишь добавим еще несколько очков в пользу его ретро-стиля.



Пара слов о легальности

Легальность эмуляции всегда вызывала вопросы, но по сути, если вам принадлежит консоль и игра, то почему, собственно, вам нельзя создать копию соответствующ-

щих прошивок для эмуляции? Во всяком случае, по нашему мнению. Если вы не уверены, лучше выяснить эти вопросы заранее — так, от греха.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.



› Добавьте картридж с играми, и получится фантастический ретро-корпус для Raspberry Pi.

Разумеется, в качестве игры мы выбрали *Sonic*, и так как у нас был старый картридж *Sonic the Hedgehog*, мы подумали, что стоит разместить RPi внутри него. RPi и в самом деле умещается внутри картриджа Mega Drive, но остается очень мало места для кабелей, даже если удалить пластмассовые перегородки для размещения SD-карты. Поэтому вместо того, чтобы поместить RPi внутри грубо искромсанного картриджа, из которого торчал бы пучок проводов, мы решили установить RPi в корпусе Mega Drive, воспользовавшись тем, что там есть, для частичного сохранения функциональности оригинальной консоли.

Мы разместили RPi в основании корпуса Mega Drive; SD-карта легко доступна через боковой порт расширения, который можно прикрыть пластмассовой крышкой, а два кабеля USB от RPi выходят в два передних порта контроллера игровой консоли. Кабели HDMI и Ethernet были пропущены через порт питания и ТВ-выход с задней стороны корпуса, а питающий кабель проведен через боковой проем.

Тщательно закрепив все компоненты Raspberry Pi, чтобы они не выскользнули из корпуса, мы начали просовывать необходимые кабели в соответствующие порты. Мы надежно закрепили крышку Mega Drive, чтобы крепежные винты не касались кабе-



› Знаменитый синий ежик на Raspberry Pi.

лей или самой RPi, уложили консоль в ее новое жилище под телевизором и подали на восстановленную MegaPi питание. Кнопки управления питанием и громкостью на передней панели консоли, естественно, не работали, но они внесли свой вклад в ретро-стиль консоли девяностых с привкусом современности.

Наш MegaPi прекрасно заработал, и мы устроили себе вечеринку классических игр Mega Drive с эмулятором *Genesis* в RetroPie.

Другие ретро-идеи

Вот несколько идей, над которыми вы сможете подумать и начать собственный ретро-проект RPi:

› **C64Pi** Превратите RPi в старый Commodore 64, после чего загрузите его в Commodore OS.

› **MasterPi** Попробуйте сделать с Sega Master System то же, что и с Mega Drive.

› **AtariPi** Раздобудьте старую систему Atari и продолжите волшебство.

› **Spectrum+2** Помните Spectrum+2 со встроенным приводом магнитной ленты? Думаем, RPi туда поместился бы...

› **KeyboardPi** Найдите старую клавиатуру Dip и посмотрите, можно ли встроить в нее RPi.

WinPi

С точки зрения журнала о Linux, наш следующий проект граничит с неприличием. Нам удалось запустить в QEMU на RPi... DOS 6.22 и Windows 3.1! Вместо создания образа в QEMU мы воспользовались готовым образом *VirtualBox*, сконвертировав его в «сырой» файл IMG следующей командой:

```
vboxmanage clonehd "image.vdi" "image.img" --format RAW
```



› DOS 6.22 и Windows 3.1 на RPi. Увы, после этого он перестал работать. Не иначе, заговор, скажете вы?

Замените **image.vdi** и **image.img** на название своего образа, затем преобразуйте «сырой» образ в образ **qcow** QEMU командой

```
qemu-img convert -f raw image.img -O qcow2 image.qcow
```

Результаты оказались впечатляющими, и наблюдение запуска Windows 3.1 под звуки «та-да-та-да» доставило нам извращенное удовольствие. Стоит ли говорить, что все запустилось очень быстро, но на RPi маловато оперативной памяти, и через несколько минут работы ОС от Microsoft все зависло, а QEMU перестал загружать образ, пока мы не удалили и повторно не преобразовали его. Как знать, возможно, команда Эбона добавила в Windows аварийный механизм выключения специально на этот случай? **LXF**

Что дальше

Работать над ретро-корпусом для RPi было очень весело, пусть попытка и была любительской. Мы знаем, что вам, читатель, более чем хватит талантов, чтобы закатать рукава и взять в руки паяльник и пассатижи; как насчет того, чтобы продемонстрировать свои навыки с изящным RPi? Ведь вы смогли бы придумать, как использовать

всякие оригинальные компоненты ретро-компьютера или консоли в проектах с RPi?

Какие бы проекты с Raspberry Pi вы ни выбрали, почему бы не написать о них нам? Добавьте пару картинок и краткое описание, чтобы другие могли последовать вашему примеру. В любом случае, получайте удовольствие!

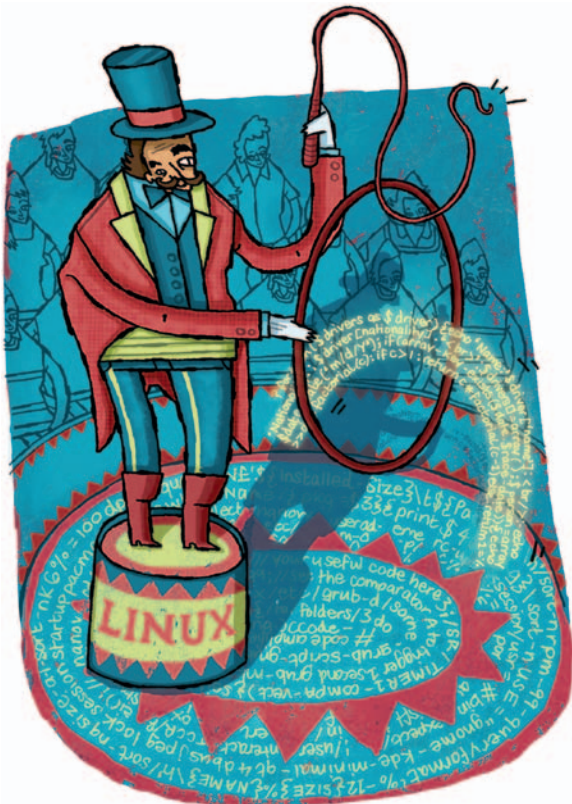
PacketFence: NAC

Шашанк Шарма показывает, как защитить сеть и бодро отражать атаки с помощью *PacketFence*, мощной системы контроля сетевого доступа.



Наш эксперт

Шашанк Шарма более четырех лет пишет о свободном ПО для различных изданий, включая Linux.com. Он соавтор книги «Fedora для начинающих».



печатить достаточной защиты. Система контроля сетевого доступа (Network Access Control – NAC) – это программа, которая может автоматически реагировать на нарушение политики использования вашей сети.

NAC обычно поддерживает регистрацию в сети всех устройств, обнаруживает нарушение политик использования, немедленно по электронной почте о каждом нарушении, изолирует нарушившие политику устройство или компьютер и при необходимости восстанавливает их, после чего они снова получают доступ к сети. *PacketFence* – система NAC на Perl, которая может делать все вышеперечисленное и многое другое через простой web-интерфейс.

С правильно настроенной NAC можно отслеживать активность всех сетевых устройств по MAC-адресам или полностью запрещать определенные действия, такие как сервисы P2P. Сочетание четко определенной политики и внешних утилит с *PacketFence* защитит сеть от шпионского и вредоносного ПО любых типов и допустит в сеть только авторизованных пользователей и рабочие станции, чтобы предотвратить любые атаки.

Зависимости и PacketFence

Поскольку в большинстве дистрибутивов программы находятся в репозиториях, мы уже давно не слышим о том, что было непременной составляющей почти любой установки в конце девяностых – об аде зависимостей.

Если вы любопытны, но еще не испытывали прелестей погоны за зависимостями ради завершения установки, *PacketFence* определенно впечатлит вас надолго. Его нет в репозиториях большинства популярных дистрибутивов, но на сайте проекта можно найти пакеты **.rpm** и **.deb**, а также архив с исходниками. *PacketFence* проще всего установить в RHEL или CentOS, благодаря тому, что они позволяют подключить множество различных репозиторий, где можно будет добыть массу зависимостей и таким образом сэкономить ваши усилия. Но если у вас Ubuntu Server, Slackware или любой другой дистрибутив, приготовьтесь к тому, что зависимости придется искать вручную. Впрочем, не давайте разговорам о зависимостях себя запугать – это проще, чем кажется.

Так как в разных дистрибутивах приложения и библиотеки собираются в пакеты по-разному, мы не можем предоставить вам один универсальный список пакетов, требуемых для установки *PacketFence*. Кроме основного *PacketFence*, у вас должны быть установлены *Apache*, *MySQL*, *DHCP*-сервер, *DNS*-сервер и *Snort*. Они есть в репозиториях большинства дистрибутивов и, возможно, уже установлены и настроены в вашей системе.

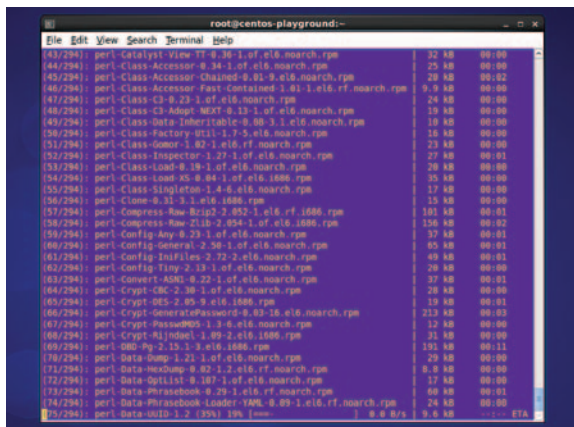
В RHEL и CentOS можно установить и активировать репозитории Repoforge, EPEL и OpenFusion. В руководстве по администрированию [Administration Guide] в формате PDF на сайте проекта описаны инструкции по установке этих репозиторий.

Мы проверяем *PacketFence* в CentOS, и установка начнется с обзаведения репозиториями:

```
# rpm -ivh http://packages.sw.be/rpmsforge-release/rpmsforgerelase-0.5.2-2.el6.rf.i686.rpm
# rpm -ivh http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-7.noarch.rpm
```

По сравнению с сетями десятилетней давности, объединявшими только рабочие станции (компьютеры и ноутбуки), современные сети напоминают научно-фантастические фильмы. Смартфоны, планшеты, роутеры, принтеры, подключаемые компьютеры и всяческие другие устройства и гаджеты, которые поддерживают подключение к сети, соперничают друг с другом за сетевое пространство. При постоянном росте сетей брандмауэры в одиночку не способны обес-

➤ Не старайтесь вчитываться в список почти из 300 пакетов, устанавливаемых как зависимости. Его состав зависит от вашей системы и дистрибутива.



ДЛЯ ВСЕГО

```
# rpm -ivh http://www.openfusion.com.au/mrepo/centos6-x86_64/RPMS.of/openfusion-release-0.5-1.of.el6.noarch.rpm
```

Эти команды, запущенные от имени суперпользователя-root, установят и активизируют три репозитория, и вы все их найдете в каталоге `/etc/yum.repos.d/`. Затем мы загрузим пакет **PacketFence 3.5 rpm** для CentOS с сайта и установим его командой `su -c "yum install packetfence-3.5.0-1.el6.noarch.rpm"`

После установки на экране появится сообщение о том, что нужно открыть в браузере адрес http://@ip_packetfence:3000/configurator для настройки *PacketFence*. Не бойтесь этого странного адреса, просто наберите в адресной строке <http://your-ip-address:3000/configurator>, и откроется мастер настройки.

Подготовка системы

Прежде чем перейти к настройке, нужно сделать кое-что еще.

Прежде всего убедитесь, что у вас есть рабочая установка *MySQL*. По умолчанию у пользователя `root` для *MySQL* нет пароля, и если вы его не задавали, сделайте это сейчас, командой `mysqladmin -u root password "typepassword"`

Это важно, и пароль следует запомнить, так как он понадобится при установке *PacketFence*. Теперь запустите сервер *MySQL*, если он еще не запущен, следующей командой:

```
service mysqld start
```

Одно из лучших свойств *PacketFence* в том, что с его помощью можно блокировать определенные типы сетевой активности, например, трафик P2P. Чтобы реализовать это, понадобится система обнаружения вторжений, и здесь нам поможет *Snort*. *PacketFence* реализует через *Snort* заданные политики использования сети. Обнаружив трафик P2P, *Snort* оповещает *PacketFence*, а тот формирует сообщение о нарушении для клиента.

Snort устанавливается из репозитория командой

```
yum install snort
```

но это не вся сказка. По умолчанию в *Snort* не задано никаких правил, и вам придется создать собственные, описывающие, что такое вторжение, а это обычно дело долгое. Но есть альтернатива: загрузить список готовых правил с сайта *Snort*. Для этого сперва зарегистрируйтесь на сайте *Snort*. Затем выявите свою версию *Snort* командой `snort --version`. Теперь откройте <http://snort.org/snort-rules/#registered> и загрузите файл `rules` для соответствующей версии *Snort*. Например, если у вас установлена *Snort 2.9.1.2*,



загрузите файл `snortrules-snapshot-2912.tar.gz`. Распакуйте архив командой

```
tar xzvf snortrules-snapshot-2912.tar.gz
```

Она создаст четыре каталога: `rules/`, `so_rules/`, `etc/` и `preproc_rules/`. Файлы настройки *Snort* находятся в каталоге `/etc/snort`, а правила – в каталоге `/etc/snort/rules/`. Перейдите в созданный каталог `rules/` и выполните команду

```
cp * /etc/snort/rules
```

чтобы скопировать правила в каталог с правилами.

Все почти готово. Создайте каталог `/usr/local/lib/snort_dynamicrules`, если его еще нет. Затем перейдите в каталог `so_rules` и скопируйте оттуда все файлы в каталог `/usr/local/lib/snort_dynamicrules`.

Все! *Snort* – одна из самых сложных программ в Linux с точки зрения настройки, но на сайте есть масса документации, включая руководства для дистрибутивов CentOS, Debian, Fedora, OpenSUSE, Ubuntu и многих других. Это популярная программа с открытым исходным кодом, и в Интернете можно найти немало обсуждений различных вариантов конфигурации и способов решения проблем, поэтому если документация, пользовательские группы и рассылки вам не помогли, обратитесь к Google.

» Чтобы суметь правильно настроить *PacketFence*, вы должны представлять себе конфигурацию своей сети.



IP-адрес локального компьютера можно определить командой `ifconfig`.

»

Настраиваемые и ненастраиваемые сетевые устройства

Типичная сеть состоит из нескольких узлов, маршрутов и беспроводной или проводной среды, которая соединяет их друг с другом. Узлами можно считать компьютеры в сети – как ноутбуки, так и настольные. В более сложных сетях наряду с вышеперечисленным есть коммутаторы. Функция коммутатора – передавать данные от одного устройства на другое, которому они предназначены. Именно благодаря коммутаторам узлы сети, то есть объединенные

в сеть компьютеры, могут непосредственно взаимодействовать друг с другом.

Коммутаторы бывают двух типов: настраиваемые и ненастраиваемые. Конфигурацию последних нельзя изменить, они просто вставляются в сеть и работают. А вот настраиваемые позволяют пользователю изменять режим своей работы либо с командной строки через `telnet`, либо через мастер настройки с `web-интерфейсом`.

Наряду с руководством по администрированию на сайте *PacketFence* также есть руководство по настройке сетевых устройств [Network Device Configuration Guide] со списком всех устройств, поддерживаемых *PacketFence*, и указаниями по их настройке. Загрузить его можно из раздела Documentation сайта проекта. Однако это неполный список, и некоторые устройства, в нем отсутствующие, прекрасно ладят с *PacketFence*.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Настройка PacketFence

Мастер настройки по адресу <http://youripaddress:3000/configuration> проведет вас по шести этапам настройки, затрагивающим различные аспекты PacketFence. Первый этап настройки – выбор типа системы [Enforcement]. Здесь задается тип системы – VLAN или Inline – и выбор зависит от оборудования в сети. Для ненастраиваемых устройств, таких как простые маршрутизаторы и коммутаторы, не поддерживающие тип VLAN, нужно выбрать тип Inline. В этом случае PacketFence становится интернет-шлюзом, и устройства не могут выходить в Интернет иначе чем через него.

Тип VLAN применяется для приписки устройств к различным виртуальным сетям на основе некоторых критериев. Пусть, скажем, в сети есть три виртуальных сети [VLANs]: обычная, изолированная и регистрационная. Все зарегистрированные устройства без нарушений принадлежат обычной виртуальной сети; новые принадлежат регистрационной сети, а устройство, нарушившее политику использования сети, попадает в изолированную сеть, отсекаемую от обычной сети. Таким образом предотвращается распространение вируса или вредоносного ПО, и атака захлебывается. Статус устройства не является постоянным: его можно изменить на странице настройки.

Создаем VLANs

На втором этапе, Сети [Networks], мы определяем сетевые интерфейсы и создаем виртуальные локальные сети. В окне вы видите все сетевые интерфейсы с IP-адресами, маской подсети, статусом, типом и т.д. Для создания системы Inline определите один сетевой интерфейс как Inline, а другой – как Management, щелкнув по выпадающему списку Type [Тип] и выбрав из вариантов None, Inline и Management. Также надо указать адрес DNS-сервера и шлюз по умолчанию (если вы не уверены, про шлюз можно узнать командой `route -n`). Закончив, нажмите кнопку Continue [Продолжить] в левом нижнем углу для перехода к следующему шагу.

Настройки базы данных PacketFence задаются на шаге 3, так и названном: Настройка базы данных [Database Configuration]. Если сервер MySQL не запущен, запустите его командой

```
service mysqld start
```

Здесь нужно указать имя базы данных PacketFence и создать учетную запись администратора для этой базы данных. Указав имя базы данных, нажмите на кнопку Создать базу данных и таблицы [Create Database and Tables], а указав имя пользователя и пароль, нажмите на кнопку Создать пользователя [Create User].

На этапе 4 нужно указать ряд других параметров (имя хоста, серверы DHCP и др.), а на этапе 5 вас попросят создать учетную запись администратора PacketFence. По окончании первичной настройки только он сможет зайти в интерфейс администрирования.

Наконец, на этапе 6 PacketFence попытается запустить все сервисы, воспользовавшись настройками сети и другими настройками,

Скорая помощь

Преимущество установки пакетов `.rpm` с `util` в том, что она автоматически разрешит зависимости с помощью настроенных репозиториях.

Скорая помощь

На этапе выбора типа системы можно одновременно выбрать Inline и VLAN, чтобы создать гибридную систему.



➤ Если вам нужна помощь, нажмите кнопку «Справка [Help]» в левом нижнем углу интерфейса администратора.

указанными на предыдущих этапах. Если все хорошо, вы увидите поздравление с успешным завершением ('Congratulations'); в противном случае появятся сообщения о возможных проблемах в конфигурации, и их нужно решить до дальнейшей работы с PacketFence.

До версии 3.5, отличавшейся графическим интерфейсом администрирования, для изменения настроек PacketFence приходилось возиться с файлами настройки. Все файлы настройки PacketFence находятся в каталоге `/usr/local/pf/conf/`, и если вы хотите лучше понять, как работает программа, не пожалейте немного времени на их изучение.

По завершении работы мастера настройки в финальном сообщении появится ссылка на интерфейс администратора PacketFence. При необходимости его можно открыть в любой момент по ссылке <http://ipaddress:1443/>. На самом деле, при попытке открыть адрес мастера настройки (<http://ipaddress:3000/configuration>), вас перенаправят в интерфейс администратора. Входите в него от имени администратора, созданного на этапе 5.

После входа в систему откроется страница «Состояние системы [Status]» с полезной информацией о сети, такой как общее количество узлов, использование жесткого диска и памяти, загрузка процессора на сервере и список последних нарушений и последних зарегистрированных устройств. На этой странице также можно сгенерировать массу различных графиков и отчетов по нарушениям, зарегистрированным и незарегистрированным узлам, использованию канала и т.д.

Параметры настройки

Как и во всех крупных проектах с обширными списками возможностей, в интерфейсе администратора PacketFence мы сталкиваемся с важнейшей проблемой: как предоставить пользователю удобный доступ к различным параметрам, не заставляя его копаться в структуре с большим уровнем вложенности?

Для этого все параметры распределены по нескольким разделам во вкладках в верхней части страницы «Администрирование [Administration]»: например, «Лицо [Person]», «Узел [Node]», «Администрирование [Administration]», «Настройка [Configuration]» и др.

PacketFence идентифицирует все сетевые устройства, поэтому всех их, будь то смартфоны, настольные компьютеры, ноутбуки и т.д., можно найти на странице «Узел > Просмотреть [Node > View]». Зайдя на эту страницу в первый раз, вы заметите, что



➤ На каждой странице мастера настройки имеется полезная информация о происходящем на данном шаге.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

все устройства помечены как незарегистрированные. *PacketFence* также покажет, какая операционная система работает на каждом устройстве (Windows, Linux, Android и т.д.). Для более подробной информации о каждом устройстве, например, времени его обнаружения, пользовательском агенте, количестве нарушений политики сети и др., нажмите на кнопку «Отобразить информацию [Show Info]» в левом верхнем углу прямо под верхней панелью.

Обратите внимание, что для каждого устройства на странице «Узел» указан только MAC-адрес устройства, а не его IP-адрес. Это может усложнить определение устройства. MAC-адрес устройства определяется командой `ifconfig` или `ipconfig` в Linux и Windows соответственно. Чтобы узнать MAC-адрес телефона с Android, зайдите в Настройки > Беспроводные сети > Настройки Wi-Fi > Расширенные настройки [Settings > Wireless and network > Wi-Fi settings > Advanced].

Найдя устройство на странице «Узел», наведите мышью на MAC-адрес, не щелкая по нему, и вы увидите три кнопки слева от MAC-адреса. Это кнопки Редактировать запись [Edit This Record], Добавить нарушение [Add Violation] и Удалить запись [Delete This Record]. Нажмите на кнопку Edit [Редактировать], и в открывшемся окне PF::Узел::Редактировать [PF::Node::Edit] введите название в поле Идентификатор [Identifier]. По умолчанию, у каждого устройства есть уникальный идентификатор; их номера начинаются с 1. Номер можно заменить названием, например, Samsung Galaxy S III, что упрощает идентификацию. Для сохранения введенных данных нажмите кнопку Редактировать узел [Edit Node] внизу.

Управление гостями

Хотя *PacketFence* может видеть все устройства и отслеживать их сетевую активность, эти устройства все равно нужно регистрировать. Для этого используется портал подключения устройств [Captive Portal], который, если он включен, перенаправит пользователей сети на специальную страницу регистрации, где они смогут указать свое имя и фамилию и адрес электронной почты, на который будет отправлен код активации. С помощью этого кода устройство можно аутентифицировать.

По умолчанию пользователи могут регистрироваться в *PacketFence* самостоятельно. Это возможно в трех режимах: Адрес электронной почты [Email], SMS и Спонсор [Sponsor], но можно полностью отключить регистрацию или ограничить ее одним или двумя режимами. Чтобы изменить настройки самостоятельной регистрации, выберите Настройка > Самостоятельная регистрация гостей [Configuration > Guests_self_registration].

Гостями можно управлять и из интерфейса администратора, выбрав пункт меню Лицо > Управление гостями [Person > Manage Guests]. При заполнении информации о госте также можно задать дату появления и период, в течение которого устройству разрешается пользоваться сетью. Затем на электронный адрес гостя будет отправлен код доступа, с которым он сможет аутентифицировать устройства. Если задать период действия, скажем, в 3 дня, то по истечении этого срока регистрация устройства будет автоматически снята, и устройство утратит доступ к сети.

Продвинутые возможности

После нашего рассказа вы могли подумать, что на этом возможности программы почти исчерпываются, но это вовсе не так. На самом деле, у *PacketFence* слишком много впечатляющих возможностей, чтобы описать их все подробно.

Достаточно сказать, что если вы знаете, что такое NAC, и представляете несколько конкретных ситуаций ее использования, есть вероятность, что со всем этим справится *PacketFence*. Кроме периода гостевого доступа и помещения устройств в различные виртуальные сети для повышения безопасности, есть и другие возможности для защиты сети. Ширится распространение смартфонов и прочих «умных» устройств, и большинство предприятий

и организаций вынуждены поддерживать политику «Принесите собственные устройства [Bring Your Own Devices – BYOD]» (сотрудник приносит свои мобильные устройства на рабочее место и с их помощью работает с внутренними ресурсами компании, такими как электронная почта, файловые серверы и базы данных, а также со своими персональными приложениями и данными, – прим. пер.). В *PacketFence* для этого используется возможность управления гостями [Guest Management], в которой, наряду с регистрацией по электронной почте или через SMS, также можно задать пароль дня, с помощью которого новые устройства аутентифицируются для получения доступа к сети.

Заодно с большим количеством нарушений в списке Настройка > Нарушения [Configuration > Violations], *PacketFence* поддерживает и еще несколько. Это Пользовательский агент [User-Agent], MAC-адрес [MAC Address] и Отпечаток DHCP [DHCP Fingerprint]. Так как каждый из этих параметров уникален для заданной операционной системы или устройства, по ним можно эффективно блокировать все эти устройства. Например, можно заблокировать доступ в сеть всем игровым консолям, вроде Sony PlayStation и пр. Аналогично можно запретить доступ к различным устройствам с помощью пользовательского агента, который представляет собой встроенный браузер по умолчанию.

Если в сети уже используется какой-то протокол аутентификации, это не обязательно проблема для *PacketFence*, потому что он поддерживает массу различных протоколов, таких как OpenLDAP, FreeRadius, Active Directory, Novell eDirectory и др. Преимущество такого подхода в том, что пользователи сети смогут аутентифицироваться в *PacketFence* по существующим логинам и паролям, и им не придется запоминать новые.

ZEN PacketFence

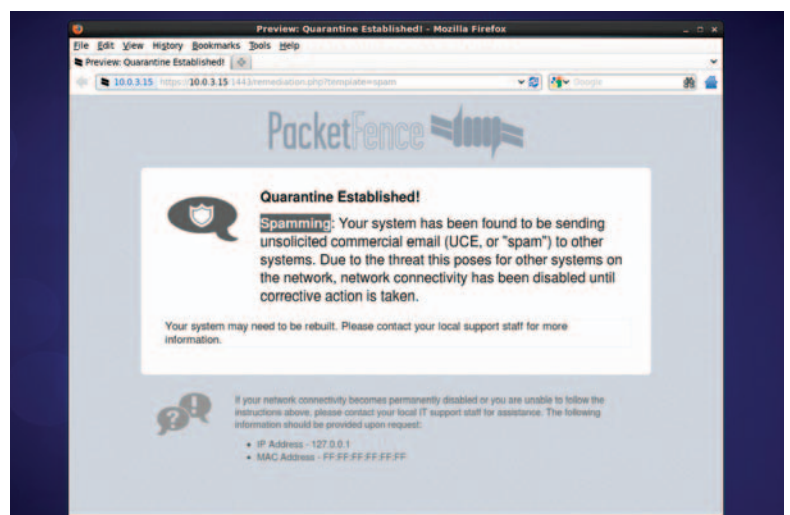
Если вам не очень хочется проходить через довольно сложный процесс установки *PacketFence* и еще более сложный процесс ее настройки, вам повезло! Zero Effort NAC, или ZEN-редакция *PacketFence*, представляет собой полностью установленную и настроенную версию *PacketFence*. Попробовать *PacketFence* или побродить по его лабиринтам можно и с помощью загрузочной флэшки: загрузите образ размером 500 МБ и запишите его на флэшку объемом 4 Гб. Этот образ загрузится в среду на базе Debian. Также можно загрузить образ *VMWare* на базе CentOS 6.3 двух различных версий для *VMWare ESX* или *VMWare Player*.

Объем этой статьи ограничен и не вмещает описание всех отличных возможностей такого обширного проекта, как *PacketFence*. При правильной настройке эта удивительно надежная программа будет сообщать вам буквально о каждом байте, передаваемом по сети. **LXF**

Скорая помощь

Процесс определения IP- или MAC-адреса может различаться в зависимости от операционной системы и версии, в данном случае, Android.

➤ На «утешительной [Remediation]» странице можно проинформировать пользователей о том, почему у них больше нет доступа к сети.





Журналы: Следим

Есть такое место, где Linux прячет свои самые глубокие и темные секреты. Нейл Ботвик бестрепетно заглядывает туда.



Наш эксперт

У Нейла Ботвика по компьютеру в каждой комнате, но он ничем не скажет вам, где находится центральный сервер – по соображениям безопасности.

Вашей системе есть каталог, куда вы, возможно, никогда не заходили; но он жизненно необходим и очень полезен, если что-то разладится. Это `/var/log`; в нем есть файлы журналов (они же – лог-файлы) почти всех действий, происходящих в системе. Большую часть операций журналирования выполняет системный логгер. Это фоновый демон, управляющий записью в лог-файлы – что позволяет не писать в каждой программе свой код для ведения журналов: все они вызывают одни и те же системные функции. В Linux несколько таких логгеров; они делают почти одно и то же. Мы займемся `syslog-ng`, одним из самых популярных. Почти все сказанное будет верно для любого из них.

Зачем нужен системный логгер? Причину мы уже назвали: чтоб не дублировать код. Это также означает, что логгер может быть гораздо более гибким и мощным. Обычно он пишет все сообщения в один файл в каталоге `/var/log` (в случае с `syslog-ng` – файл `messages`). Но это не единственный вариант: сообщения можно отправлять в разные файлы по разным критериям, и даже перенаправлять другому логгеру на другой компьютер.

Обычная строка из лог-файла выглядит примерно так:

```
Aug 16 16:54:51 hactar smartd[3191]: Device: /dev/sdc [SAT], SMART Usage Attribute: 194 Temperature_Celsius changed from 101 to 100
```

Каждая запись в лог-файле – это одна строка, начинающаяся с даты и времени; затем следуют имя хоста и команда. В данном примере это команда `smartd` – демон, отслеживающий состояние жесткого диска, с идентификатором процесса 3191 (идентификатор указывается не для всех процессов, но для большинства). После имени процесса и идентификатора идет двоеточие, а за ним – само сообщение. Единогласного формата сообщений нет: это то, что программа отправляет логгеру. В этом примере `smartd` сообщает об изменении температуры одного из жестких дисков. Зная, что делает системный логгер, вы с его помощью сможете отслеживать происходящее в вашей системе. Одна из нередких ситуаций – посмотреть, что идет не так при попытке запуска программы или использования устройства. Команда `tail` показывает десять

последних строк файла, но добавьте параметр `--follow`, и она будет показывать новые строки файла; таким образом, команда

```
tail --follow /var/log/messages
```

будет выводить новые сообщения по мере их записи в лог-файл. Системный лог-файл часто не имеет доступа на чтение для всех, и может потребоваться выполнить эту команду от имени суперпользователя-`root` или с префиксом `sudo`, иначе ничего не увидишь. Для просмотра, как это работает, запустите команду и вставьте флэшку. Вы увидите системные сообщения о том, что обнаружен USB-диск, идентифицирован как устройство хранения данных и прочитана его таблица разделов. Имя процесса в данном случае – `kernel`, так как горячее подключение устройств обрабатывается напрямую ядром. Так как в фоне выполняются и другие процессы, в вывод команды могут попасть сообщения от любого из них. Отфильтровать их можно с помощью `grep`. Команда

```
tail --follow /var/log/messages | grep kernel:
```

исключит все сообщения, в которых нет слова `kernel`: – завершающее: необходимо, чтобы выделить только сообщения от ядра.

Создаем фильтры

Помещать все сообщения в один файл – не всегда лучший вариант. Некоторые программы генерируют массу сообщений, которые лучше отделить от прочих. Для этого задаются правила фильтров в файле `/etc/syslog-ng/syslog-ng.conf`. В нем источники, места назначения и фильтры представлены объектами, а связывают их записи в лог-файле. В конфигурации по умолчанию стандартный источник определяется так:

```
source src {
    unix-stream("/dev/log" max-connections(256));
    internal();
    file("/proc/kmsg");
};
```

Здесь создан объект источника `src`, принимающий записи журнала от разных источников. Затем определим места назначения:

```
destination messages { file("/var/log/messages"); };
destination console_all { file("/dev/tty12"); };
```

Первая строка выводит записи в указанный файл, вторая – в консоль. В этом можно убедиться, нажав `Ctrl+Alt+F12` или перейдя в стандартную консоль входа в систему 1 (`Ctrl+Alt+F1` с рабочего стола) и затем нажав `Alt`+стрелка влево. Источник и место назначения готовы; теперь определим формат записей, отправляемых с первого на второе. В конфигурации по умолчанию стандартный источник отправляется в оба места назначения.

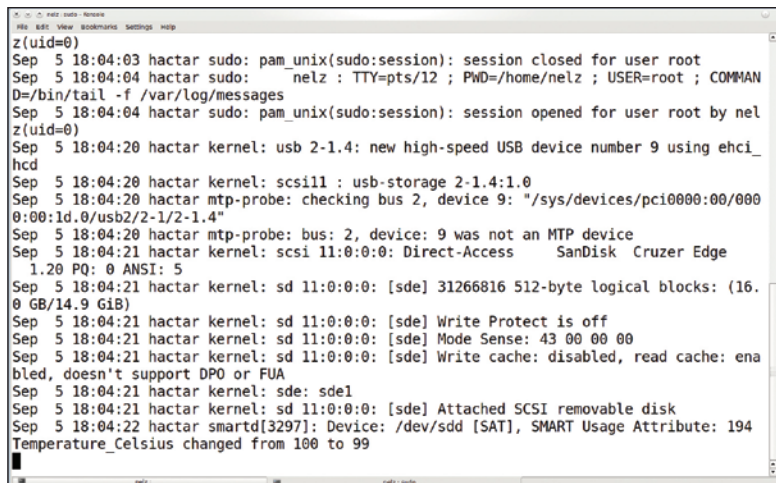
```
log { source(src); destination(messages); };
log { source(src); destination(console_all); };
```

Здесь можно добавить новые источники или места назначения и фильтры, определяющие, куда отправить. Вот пример, где вывод почтового сервера посылается в отдельный файл `mail.log`:

```
destination d_mail { file("/var/log/mail.log"); };
filter f_mail { facility(mail); };
filter f_notmail { not facility(mail); };
```

Строки с источником нет – мы все еще пользуемся стандартным источником. В качестве места назначения задается отдельный

➤ **Просмотр системного журнала командой `tail -f` позволит наблюдать за работой компьютера в реальном времени.**



за системой

лог-файл для почты, а два фильтра с помощью функции `facility` определяют, какие сообщения приходят от почтового сервера. Есть несколько стандартных групп – например, `mail`, `auth` или `news` (в документации есть полный список), и при отправке сообщения процесс указывает одну из групп. Преимущество такого подхода в том, что не нужно знать имя процесса: сообщения будут отфильтрованы для любой почтовой программы. Второй фильтр работает как отрицание и выбирает все сообщения не от почты. Имена объектов источника (`source`), фильтра (`filter`) и места назначения (`destination`) обычно предваряются префиксами `s_`, `f_` и `d_`, чтобы было сразу понятно, какой объект следует за префиксом. Теперь заменим первые две строки на следующие:

```
log { source(src); filter(f_notmail); destination(messages); };
log { source(src); filter(f_mail); destination(d_mail); };
```

Теперь между источником местом назначения появился фильтр, и в лог-файл попадут только должные сообщения. Не создай мы фильтра `notmail`, оставив исходную строку, сообщения попали бы не только в лог-файл почты, но и в лог-файл по умолчанию. Но сообщения по-прежнему отправляются в консоль, так как этого мы не меняли. При любых изменениях в конфигурационном файле нужно сообщить `syslog-ng` о том, что нужно перечитать его, отправив ей сигнал `SIGHUP` таким образом:

```
sudo killall -HUP syslog-ng
```

Держим это под контролем

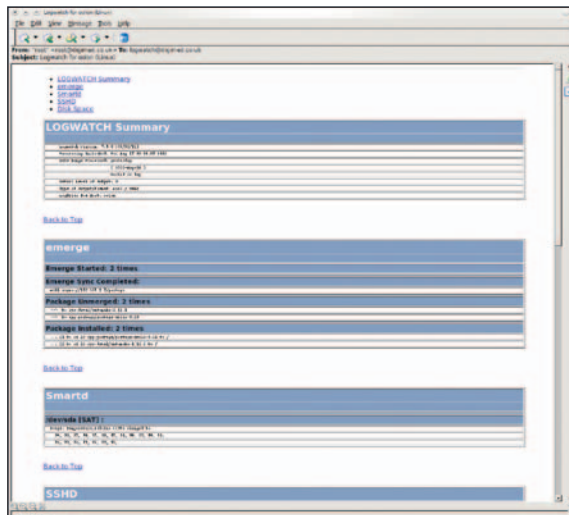
Если ваш корневой раздел (`/`) начал переполняться, проверьте содержимое `/var`. Вышедший из-под контроля процесс может писать в свой лог-файл огромный объем данных, быстро съедая ценное место на диске. Это одна из причин, по которым многие предпочитают размещать `/var` на отдельном разделе. Переполнение раздела `/` способно вызвать нехорошие последствия для системы.

Со временем содержимое каталога `/var/log` будет отнимать все больше места на диске. Чтобы решить эту проблему, установим `logrotate`. Она запускает `cron`, обычно раз в день, и архивирует лог-файлы в соответствии с правилами в `/etc/logrotate.conf` и `/etc/logrotate.d`. `/etc/logrotate.conf` – общий конфигурационный файл, а в `/etc/logrotate.d` находятся отдельные наборы правил, применяемые к конкретным пакетам или лог-файлам. В общем конфигурационном файле находятся настройки вроде

```
weekly
rotate 4
create
compress
```

Эти настройки означают, что по умолчанию лог-файлы архивируются еженедельно, хранятся до четырех последних копий, при архивировании текущего создается новый пустой файл, и что файлы сжимаются. На практике это означает, что основной лог-файл `messages` упаковывается в `messages.1.gz` и создается новый пустой файл `messages`. При следующей архивации `messages.1.gz` переименовывается в `messages.2.gz` и создается новый `messages.1.gz`. Четвертый файл удаляется, чтобы расчистить дорожку остальным. К параметрам можно добавить `dateext`, и вместо цифр 1, 2 и 3 к именам файлов будет добавляться дата. Файл для `syslog-ng` в `/etc/logrotate.d` будет выглядеть примерно так:

```
/var/log/messages {
postrotate
```



► Программы типа `Logwatch` умеют формировать ежедневный отчет о состоянии системы и предупреждать вас о любой подозрительной активности.

```
/etc/init.d/syslog-ng reload > /dev/null 2>&1 || true
endscript
}
```

Здесь указывается архивируемый файл (его имя может задаваться шаблоном) и правила. В данном случае единственное правило, особенное для этого файла – команда `postrotate`, выполняемая после завершения архивации, чтобы `syslog-ng` перечитал конфигурационный файл. Это также снимает блокировку со старого лог-файла и открывает новый. Здесь можно указать любые параметры из глобального файла чтобы перезагрузить их. Например, можно хранить более длинную историю для конкретного лог-файла, включив параметр:

```
rotate 8
```

Если вас беспокоит непомерное увеличение размера лог-файлов, когда пошедшая вразнос программа будет писать туда слишком много сообщений, добавьте параметр

```
maxsize 10M
```

для принудительной архивации при достижении размера файла в 10 МБ, даже если время архивации еще не пришло. Обратите внимание, что это произойдет только при запуске задания `cron`, поэтому файл может успеть стать намного больше. Однако лог-файлы хорошо сжимаются, и сжатие разбухшего файла высвобождает много свободного места.

Просматривать лог-файлы вручную не обязательно: есть программы, которые проанализируют их и сформируют отчет. Такие программы как `logwatch` (www.logwatch.org) и `logcheck` (<http://packages.debian.org/sid/logcheck>) просканируют системные лог-файлы и отправят вам отчет с предупреждениями о любых аномалиях. Есть и программы, которые строят диаграммы для лог-файлов конкретных приложений – например, диаграммы трафика через `Apache`.

Пока мы говорили только о чтении лог-файлов, но командой `logger` в системный лог можно записать данные и из оболочки (или скрипта), примерно так:

```
logger -i -t myscrip "Ваш скрипт рухнул"
```

Параметр `-t` помечает запись, параметр `-i` добавляет в журнал идентификатор процесса, а за ними следует сообщение. **LXF**

Arduino: Пишем

Последняя статья! Написав, как создать собственную библиотеку Arduino и еще немного порезвиться со светодиодами, **Ник Вейч** зачехляет паяльник.



Наш эксперт

Когда **LXF** только появился, его держали на плаву исключительно скрипты *Bash* от **Ника Вейча**. Потом их заменили «люди», и это, по мнению Ника, стало шагом назад...

В прошлый раз мы говорили о мультиплексировании методом Чарли для управления несколькими светодиодами. Сейчас мы кратко рассмотрим, как вынести все проблемы наружу, воспользовавшись специальной микросхемой для реализации логики. Программируемые микросхемы могут сильно озадачить, когда вы сидите и читаете руководство за руководством, чтобы в голове отложились используемые контакты и протоколы. К счастью, в этом случае у нас есть удобная сторонняя библиотека, которая значительно упрощает дело.

Но до того... не написать ли нам несколько своих библиотек? В мире Arduino это обычно делается, чтобы просто и безопасно использовать ту или иную микросхему с собственной логикой – хотя на практике библиотека может делать все что угодно: сложные вычисления, преобразования, процедуры обработки прерываний – что угодно. И вот вам краткое руководство по написанию собственной библиотеки, которая станет частью среды Arduino.

Основы библиотек

Как вы, наверное, знаете из предыдущих уроков, исходный код Arduino написан на C++, и в нем много вспомогательных функций из библиотек Arduino. Чтобы написать собственную библиотеку, нам придется немного углубиться в темные пещеры языка C++.

Для библиотеки на C++ нужны как минимум два файла: сам код библиотеки (в файле с расширением **.cpp**, так как это код на C++) и заголовочный файл для этого кода (с расширением **.h**). Имена файлов должны быть одинаковыми, так компилятор сопоставит их друг с другом.

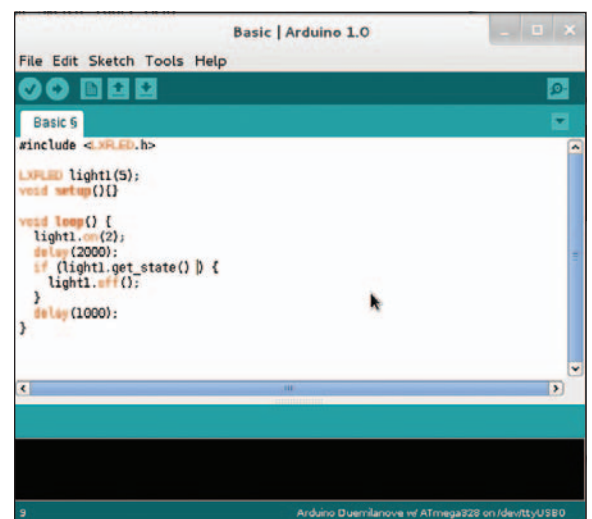
Заголовочный файл – нечто вроде оглавления основного кода. Он содержит «прототипы» функций (короткая форма со списком параметров и возвращаемым значением), помогающие компилятору связать код, откуда происходит обращение к библиотеке, с самими функциями. Сюда также стоит заглянуть, если к библиотеке нет документации, ибо здесь-то и означено, что делает библиотека.

Основной код – это обычный код, как то, что вы пишете в программе для Arduino... почти. Различие в том, что в библиотеке все нужно заключить в определение класса. Мы уже пользовались классами, и даже если вы не слишком знакомы с ними, беспокоиться особо не о чем. Класс – это просто сложная переменная. Вместо одного значения в ней может быть столько значений, сколько вам надо. Кроме того, у нее могут быть собственные функции (по отношению к классу обычно называемые «методами»), выполняющие определенные действия с переменными класса. Гораздо проще пояснить все на примере. Так как на этот раз мы снова возьмемся со светодиодами, можно создать класс LED («светодиод»; кстати, одноименный класс существует и является частью не основного дистрибутива, а проекта Arduino. Это не он – это просто пример создания собственного класса).

Итак, у нас есть класс LED; какие у него должны быть свойства? Подумайте, о каких аспектах (т.е. свойствах) объекта (в этом случае, представляющего реальное устройство) вам нужно знать. Также прикиньте, какие свойства по сути не являются параметрами реального устройства, но помогут вам управлять им.

Свойства объекта LED могут быть такими:

» Номер контакта (т.е. номер физического контакта, к которому он подключен)



» Указав верные ключевые слова, можно активировать подсветку в редакторе Arduino.

» **Месяц назад** Меньше выводов, больше светодиодов – это чарлиплексирование.

библиотеки



» На форумах Arduino полно отличных идей, проектов и ссылок на сторонние библиотеки, которые делают все что угодно.

- » Яркость (если он подключен к контакту с ШИМ)
- » Состояние (можно ли сохранить в нем состояние светодиода или прочесть его, чтобы сказать, включен он или нет)

Одно из этих свойств особенное – это номер контакта. Без него нельзя подключить светодиод или управлять им. Поэтому номер контакта следует сделать частью «конструктора» экземпляров класса. Это означает только то, что при создании объекта LED нам обязательно придется указать номер контакта. Можно было бы создать «виртуальный» светодиод, не подключенный ни к одному контакту, но тогда пришлось бы писать более сложные методы, проверяющие, задан ли номер контакта, поэтому проще задавать номер контакта при создании объекта в обязательном порядке. Тогда наш код в файле **LXFLED.cpp** будет начинаться так:

```

/* LXF LED library */
#include "Arduino.h"
#include "LXFLED.h"
LXFLED::LXFLED(int pin)
{
    pinMode(pin_number, OUTPUT);
    this->_pin_number = pin;
    this->_state = false;
}
    
```

В верхней части файла обычно (да так и удобнее) размещается комментарий с информацией о том, что делает код, а также с информацией обо всех копрайтах и лицензиях и, возможно, с координатами автора.

Далее мы подключаем несколько файлов точно так же, как вы подключали файлы в своих программах для Arduino. В данном случае мы хотим подключить стандартную библиотеку Arduino (скорее всего, нашей библиотеке понадобятся некоторые функции Arduino) и, что чуть более необычно, заголовочный файл для нашего кода.

Затем следует объявление самого класса. Оно всегда начинается с конструкции **<Name>::<Name>**, за которой следует либо пара

пустых скобок, либо список параметров с типами, которые нужно передать классу при создании экземпляра объекта. Мы решили, что нам нужен номер контакта, поэтому привели его здесь. Переменная, здесь указанная, никак не касается того, что находится вне библиотеки, так что не беспокойтесь о конфликтах с переменными в основном коде. В терминах программирования, «пространство имен» ограничивается файлами библиотеки. В фигурных скобках указываются действия, которые должны произойти при создании объекта; мы устанавливаем режим работы соответствующего контакта в OUTPUT, чтобы управлять светодиодом.

Здесь также нужно задать значения всех переменных класса. Конструкция **this->** – стандартная конструкция C++, которая означает, что следующая за ней переменная является членом определяемого класса. Итак, создаваемый экземпляр класса получает свойства «номер контакта» (**_pin_number**) и «состояние» (**_state**). С символа подчеркивания принято начинать имена переменных, которые являются приватными для этого класса. Это означает, что их значения можно изменять только в коде самой библиотеки, и если вы захотите изменить состояние светодиода (или номер контакта), придется написать соответствующие функции (методы!), которые должны быть частью класса. Расширим наш код, добавив несколько простых методов:

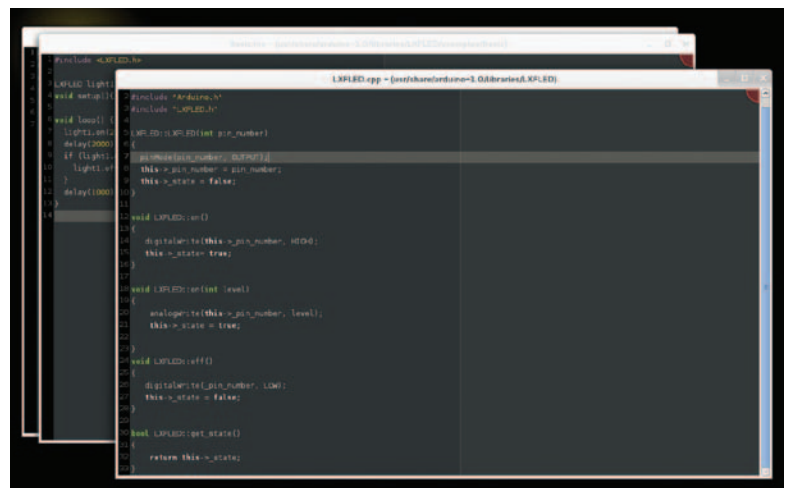
```

void LXFLED::on()
{
    digitalWrite(_pin_number, HIGH);
    this->_state = true;
}

void LXFLED::off()
{
    digitalWrite(_pin_number, LOW);
    this->_state = false;
}
    
```

Эти два метода просто включают или выключают светодиод и обновляют его состояние. Теперь напишем другой метод, который будет возвращать состояние светодиода:

» Для разработки своей библиотеки лучше пользоваться текстовым редактором (желательно тем, что понимает C++), а не редактором Arduino. На рисунке – *Scribes*.



» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

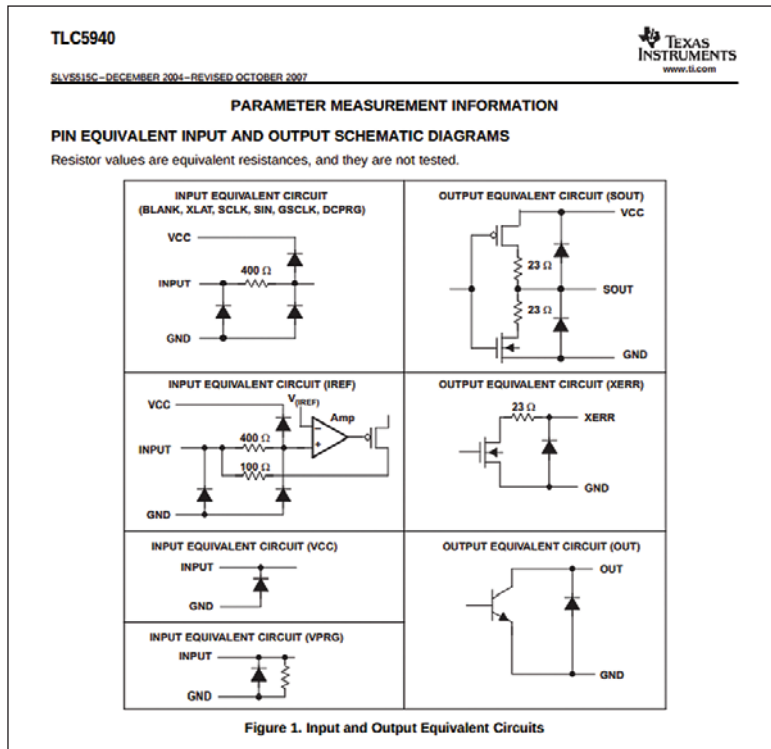


Figure 1. Input and Output Equivalent Circuits

➤ Паспорт TLC5490 довольно подробен, и это очень удобная микросхема для управления светодиодами или другими активными компонентами – например, небольшими электродвигателями.

```
bool LXFLED::get_state()
{
    return this->_state;
}
```

Названию метода здесь предшествует тип возвращаемого значения; в данном случае это **bool** (булевское, т.е. «истина (true)» или «ложь (false)»). Этот метод просто возвращает значение свойства `_state` объекта.

Прежде чем перейти к заголовочному файлу, нужно сделать еще кое-что. Мы сказали, что с помощью этой библиотеки также можно управлять и светодиодами с ШИМ. Тогда при включении нужно каким-то образом задавать их яркость. На уровне Arduino яркость устанавливается легко – включением контакта методом `analogWrite()`, но как работать с этим значением в библиотеке? Для него можно написать отдельный метод – скажем, `PWM_on()` или что-то вроде этого – но в C++ есть отличный трюк, которым мы можем воспользоваться, и это «перегружение [overloading]».

Снова определим метод `on()`, но на сей раз с параметром:

```
void LXFLED::on(int level)
{
    analogWrite(this->_pin_number, level);
    this->_state = true;
}
```

Это совершенно другой метод, который задает значение ШИМ, как мы и хотели, но название метода точно такое же. Он работает благодаря тому, что компилятору C++ хватает ума понять, что если мы вызываем метод с одним аргументом типа `int`, то нужно выполнить именно этот код. Если аргумента нет, выполняется предыдущий метод. Можно определить сколько угодно методов с одинаковыми именами при условии, что у них у всех будут разные списки параметров (примечание: здесь имеются в виду типы параметров, т.е. нельзя создать три метода, в каждом из которых будет по одному аргументу `int` с разными именами).

Теперь, когда основной код библиотеки готов, нужно создать соответствующий ему заголовочный файл. У нас он будет таким:

```
#ifndef LXFLED_h
#define LXFLED_h
class LXFLED{
public:
    LXFLED(int pin);
    void on();
    void on(int level);
    void off();
    bool get_state();
private:
    int _pin_number;
    bool _state;
};
#endif
```

Конструкция `#ifndef...` в начале – еще одно стандартное соглашение C++. В сложных проектах может оказаться несколько частей кода, которые пытаются импортировать одну и ту же библиотеку. Если это произойдет, последствия могут быть неприятными, поэтому в качестве меры предосторожности в заголовочном файле объявляется переменная с именем этого файла. Затем мы проверяем ее в начале – если она уже существует, это означает, что библиотека уже была подключена в какой-то другой части кода и ее можно пропустить.

Единственное, что остается сделать здесь – написать «заглушки» класса в основном коде, т.е. перечислить все переменные класса и все методы с их аргументами. Для перегружаемого метода `on()` нужно привести все варианты.

Чтобы избежать ошибок, лучше взять копию основного файла `.cpp` и просто удалить из него код, оставив заголовки функций. Так вы обеспечите, чтобы аргументы в заголовочном файле и в коде совпадали.

Ключевые слова

Другой полезный файл, который можно найти в каждой библиотеке, называется `keywords.txt`. Вы будете смеяться, но в нем задаются ключевые слова для кода библиотеки. Есть два типа ключевых слов – переменные и имена объекта имеют тип **KEYWORD1**, а методы и функции – тип **KEYWORD2**.

Нужно просто перечислить все это в текстовом файле с таким именем. Сначала идет слово, затем табуляция (не пробелы!), затем – тип ключевого слова. Например, наш файл с ключевыми словами для библиотеки LED будет выглядеть так:

```
# Datatypes
LXFLED KEYWORD1
# methods
on KEYWORD2
off KEYWORD2
get_staus KEYWORD2
```

Включать файл `keywords.txt` в содержимое библиотеки необязательно, но лучше не полениться и сделать это – благодаря ему подсветка синтаксиса в редакторе кода Arduino будет работать правильно.

Приведем пример

Последний этап создания библиотеки – пример, иллюстрирующий ее работу. Его стоит правильно отформатировать и снабдить множеством полезных комментариев – если другие не смогут понять, как пользоваться вашей библиотекой, они подумают, что та не особо хороша, и почти наверняка не угостят вас пивком, столкнувшись с вами где-нибудь на “Maker Faire” [мероприятие, проводимое американским журналом *Make* – семинары и презентации научных проектов, – прим. пер.] или еще где-нибудь.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Создать пример просто. Для начала нужно создать каталог **examples** (буквы названия – обязательно в нижнем регистре). Туда можно поместить все примеры. Если вы сохраняете их из текстового редактора, помните, что каждый пример должен находиться в одноименном каталоге, а файл примера должен иметь расширение **.ino**, это новое расширение для файлов «эскизов» Arduino. Создавайте столько примеров, сколько хотите, но позаботьтесь о понятных именах для них.

Среда Arduino при запуске проверяет наличие примеров в файлах библиотек и добавляет их в меню, чтобы пользователи смогли найти и попробовать ваш код. Их можно найти в меню File > Examples > [имя вашей библиотеки] > [имя вашего примера].

Здесь мы проигнорировали собственный совет и не добавили ни одного комментария, и простой пример работы с нашей библиотекой выглядит так:

```
#include <LXFLED.h>
LXFLED light1(5);
void setup(){}
void loop() {
  light1.on(2);
  delay(2000);
  if (light1.get_state()) {
    light1.off();
  }
  delay(1000);
}
```

Обратите внимание на отличия. Конструкция **#include** в начале нам уже знакома, но затем мы определяем переменную **light1**, объект класса **LXFLED**. Созданная таким образом переменная наследует все свойства и методы класса, который мы объявили ранее. При инициализации мы планировали еще и задать номер контакта, который будет выходом. Теперь мы так и делаем (код конструктора класса вызывается при создании нового экземпляра класса, т.е. при объявлении переменной **light1** этого типа). Среди прочего это означает, что нам ничего не нужно делать в методе **setup()** для данной переменной, так как все сделает библиотека.

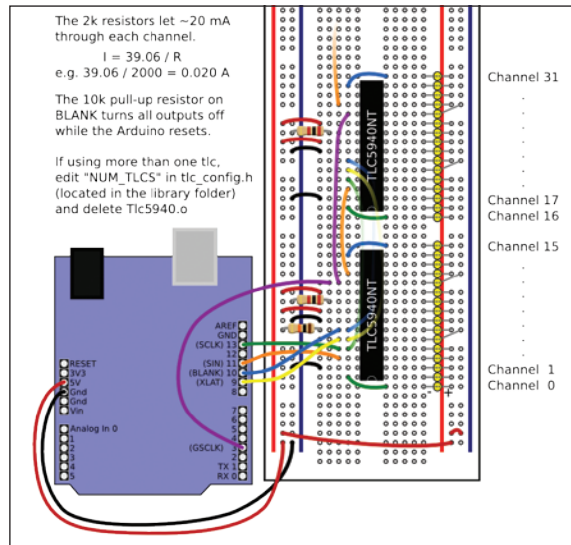
Зажигаем свет, гасим свет

В основном цикле вызываются методы класса **LXFLED** прямо из созданной нами переменной – скажем, чтобы включить светодиод, мы вызываем **light1.on()**. Если бы мы подключили его к контакту с ШИМ, мы вызвали бы **light1.on(xxx)**, чтобы включить его с заданной яркостью. По правде говоря, в нашем недоделанном коде этим методом можно пользоваться даже для контактов без ШИМ (кто-то должен это поправить).

Далее, обратите внимание на то, где мы проверяем состояние светодиода. Можно было бы вызвать **light.off()**, но он не иллюстрирует наше замечательное умение проверять это состояние. Так как переменная состояния задается внутри методов включения и выключения и недоступна нашей программе, вы можете быть уверены в том, что она адекватно отражает состояние. Здесь мы проверяем булево значение в операторе **if**, чтобы проверить, что светодиод включен, прежде чем пытаться его выключить.

В том, чтобы делать это именно так, нет ничего особенного, но приятно увидеть, что оно работает. Метод **light1.get_state** возвращает булево значение, которое затем анализируется оператором **if**. Да, это не бог весть что – но оно работает и иллюстрирует все ограниченные возможности нашей скудной библиотеки.

В комплекте поставки Arduino можно найти массу примеров реальных библиотек, но еще больше библиотек на все случаи жизни создали пользователи Arduino. Поиск на форумах на сайте Arduino или на других сайтах об Arduino, которые вы обычно посе-



Этот пример схемы подключения из библиотеки Алекса Леоне (Alex Leone) иллюстрирует управление 16-ю или большим количеством с TLC5490.

щаете, можно озолотиться. Если библиотека загружается в виде файла ZIP, ее можно распаковать в каталог **arduino/libraries**, и после перезапуска редактора Arduino она появится в нем.

Большинство библиотек на самом деле представляют собой своего рода драйверы для распространенных компонентов, таких как память, устройства хранения данных, контроллеры и т.д. Особенно полезен драйвер для микросхемы TLC5940. Как мы узнали из прошлой статьи о мультиплексировании, большими дисплеями лучше всего управлять с помощью микросхемы драйвера. На рынке их множество, но TLC5490 обладает лучшей функциональностью и достаточно проста, чтобы использовать ее с Arduino.

У нее 16 выходных линий, каждая из которых может работать в режиме ШИМ со значением от 0 до 4096, что восьмикратно превышает разрешение контактов ШИМ в Arduino. В ней также есть собственная схема ограничения тока, и каждый светодиод получает именно такой ток, какой ему необходим – нужно просто установить эталонное значение, подключив подходящий резистор к одному из контактов. Единственная мелкая сложность с TLC5490 – это иногда чрезмерно сложный протокол установки значений отдельных каналов (прежде всего из-за дополнительных возможностей, таких как сообщения об ошибках, которые могут возникнуть из-за неисправных светодиодов).

Так или иначе, большая часть этих сложностей скрыта от мира Arduino, потому что кто-то написал библиотеку для этой микросхемы. Вам придется написать много кода, но схемы для управления дисплеями большего размера можно соединить в гирлянду (хотя, конечно, ее максимальная длина для немерцающего дисплея будет ограничена величиной задержки распространения сигнала).

Библиотека для TLC5490 есть на нашем DVD, и мы также приводим принципиальную схему ее типичного подключения. **LXF**

На прощание

Это последняя статья из серии про Arduino. Серия была долгой и успешной, и из ваших комментариев и писем мы знаем, что она вам понравилась (ну, или, по крайней мере, вы ее прочли). Конечно, мы рассказали далеко не обо всем, но чем больше вы погружаетесь в детали конкретного приложения, тем меньше оно значит для более широкой аудитории. Обязательно зайдите на очень активные и информативные форумы Arduino по ссылке <http://arduino.cc/forum>.

» **Через месяц** Отныне вы в свободном полете. Удачи вам и пылкой фантазии!

Erlang: Практика

Суха теория, мой друг... Андрей Ушаков предлагает приложить руки к практическим решениям.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Последние несколько статей мы занимались тем, что изучали, что такое многозадачные и распределенные системы и какие сложности нас подстерегают на пути создания таких систем. Помимо этого, мы изучали, какие в языке Erlang есть средства для решения подобных задач. Но на одной теории далеко не уехать: без практического опыта не создать более-менее сложную, многозадачную, и, тем более, распределенную систему. Именно поэтому мы открываем цикл статей, посвященных практикуму по созданию многозадачных и распределенных систем средствами языка Erlang.

Нашей задачей в данном практикуме будет написание многозадачных и, с какого-то момента, распределенных версий таких широко распространенных функций, как `map` и `reduce`. Решать данную задачу мы будем постепенно: начнем с обычных, не многозадачных реализаций данных функций и закончим реализацией, работающей в распределенной среде, т.е. на нескольких узлах. Возникает закономерный вопрос: а почему именно эти функции выбраны для нашего практикума? Как мы увидим ниже, реализация этих функций в простейшем, не многозадачном случае очень ясна. Поэтому при реализации разнообразных многозадачных вариантов этих функций мы практически не будем отвлекаться на детали, не относящиеся к многозадачности.

Прежде чем начать работу с примерами, немного поговорим о том, что представляют собой функции `map` и `reduce` и какие ограничения мы накладываем на наши реализации этих функций.

Функция `map` вычисляет результат операции отображения, которая каждому элементу `a` из исходного множества `A` ставит в соответствие элемент `fun(a)` из результирующего множества для заданной функции отображения `fun`. Понятно, что в качестве исходного множества элементов можно взять любой набор элементов, как упорядоченный, так и неупорядоченный, но мы в нашем примере в качестве такого набора всегда будем применять только список.

Функция `reduce` вычисляет результат (некоторое значение) операции свертки (или агрегирования) для заданного множества, функции свертки и начального значения. Примером такой операции является операция нахождения суммы множества чисел. Как и в случае функции `map`, в качестве исходного набора элементов мы будем использовать списки. Результат выполнения операции свертки может быть разным и зависит от того, в каком порядке мы берем элементы из множества, поскольку операция свертки бывает некоммутативной. Действительно, если в качестве исходного множества мы возьмем список матриц, а в качестве функции свертки – операцию умножения этих матриц, результат будет зависеть от порядка обхода списка: слева направо или справа налево. Именно по этой причине модуль `lists` содержит две функции для операции свертки: `lists:foldl/3` и `lists:foldr/3`. Мы во всех наших примерах при реализации операции свертки будем обходить список слева направо (порядок обхода списка справа налево тривиально реализуется по аналогии).

Есть и еще один момент, который связан с операцией свертки: это ассоциативность данной операции. Ассоциативность операции определяет, зависит ли результат операции от того, в каком порядке мы вычисляем результат этой операции, т.е. расставляя скобки. Так, например, операция сложения чисел является ассоциативной: это означает, что значение выражения `(1+2)+3` равно значению выражения `1+(2+3)`. А операция вычитания чисел ассоциативной не является: значение выражения `(1-2)-3` равно `-4`, а значение выражения `1-(2-3)` равно `2`. И опять же для простоты мы полагаем, что имеем дело с ассоциативной операцией свертки (почему нам важна ассоциативность, мы увидим ниже).

Скажем пару слов и об организации наших примеров. Все экспортируемые функции, которые относятся к операции отображения, располагаются в модуле `parallel_map` (и, соответственно, в файле `parallel_map.erl`). Все экспортируемые функции, которые относятся к операции свертки, располагаются в модуле `parallel_reduce` (и, соответственно, в файле `parallel_reduce.erl`).

Помимо этих двух модулей, мы будем определять и использовать дополнительные модули по мере необходимости. Одним из таких модулей, который будет использоваться

«Без практического опыта не создать многозадачную систему.»

практически везде, является модуль `parallel_common` (расположенный в файле `parallel_common.erl`). Как мы увидим, этот модуль содержит общие для наших примеров функции.

А начнем мы наши примеры с обычных, не многозадачных версий функций `map` и `reduce`. Для реализации обычной версии функции `map` мы воспользуемся техникой конструирования списков [List comprehension]:

```
usual_map(_Fun, []) -> [];  
usual_map(Fun, SourceList) -> [Fun(Element) || Element  
<- SourceList].
```

Как видно, реализация этой функции достаточно тривиальна (эту функцию можно было бы реализовать и рекурсивным образом, но реализация при этом стала бы несколько больше). Обычную версию функции `reduce` таким образом реализовать не получится – для этого нам потребуется рекурсивно работать с исходным списком:

```
usual_reduce(_Fun, [], InitValue) -> InitValue;  
usual_reduce(Fun, [H | Rest], InitValue) ->  
NewAgg = Fun(H, InitValue),  
usual_reduce(Fun, Rest, NewAgg).
```

И опять же, реализация этой функции достаточно тривиальна.

Давайте проверим, что наши функции работают правильно. Для этого откомпилируем соответствующие модули и для вызовов функций, приведенных ниже, проверим, что результат их вызовов соответствует приведенным результатам. Вызов `parallel_map:usual_map(fun(X) -> X*2 end, [1, 2, 3, 4])` вернет список `[2, 4, 6, 8]`. Вызов `parallel_map:usual_map(fun(Str) -> "-" ++ Str ++ "-" end, ["aa", "bb", "cc"])` вернет список `["-aa-", "-bb-", "-cc-"]`. Те же самые результаты мы получим и при использовании функции `lists:map/2` из модуля `lists`. Результатом вызова `parallel_reduce:usual_`

МНОГОЗАДАЧНОСТИ

`reduce(fun(Item, Agg) -> Item + Agg end, [1, 2, 3, 4], 1)` будет число 11. Результатом вызова `parallel_reduce:usual_reduce(fun(Item, Agg) -> Agg ++ Item end, ["aa", "bb"], "+")` будет строка `++aabb`. Такие же результаты мы получим и при использовании функции `lists:foldl/3` из модуля `lists`.

Реализовав обычные версии функций `map` и `reduce`, создадим их многозадачные версии. Мы начнем с простейшего случая для функции `map`: когда для отображения каждого элемента (т.е. для вычисления результирующего элемента `fun(a)` для каждого элемента `a` из исходного списка) из исходного множества мы используем отдельную задачу. В многозадачной версии функции `map` у нас будет один главный процесс, который создает дочерние рабочие процессы, раздает им задания и собирает результаты их работы, и несколько рабочих процессов (в нашем случае количество рабочих процессов равно количеству элементов в списке). Главный процесс (в котором мы инициируем выполнение нашей функции `map`) должен сделать следующее: создать задания для рабочих процессов, создать необходимое количество рабочих процессов, раздать всем этим процессам задание, получить результаты от всех рабочих процессов и объединить эти результаты в результирующем списке. Рабочие процессы в данной версии функции `map` являются «одноразовыми»: они получают задание, выполняют его, возвращают его и заканчивают свою работу. Создание задания для рабочих процессов заключается в преобразовании исходного списка в список пар (кортежей из двух элементов), состоящих из порядкового номера элемента и самого элемента. Мы задаем порядковые номера элементов, начиная с 0; почему мы так делаем и зачем вообще нужны порядковые номера элементов, будет ясно чуть ниже. Для создания заданий для рабочих процессов мы создаем пару вспомогательных функций `simple_prepare_data/2` (интерфейсная функция) и `simple_prepare_data/3` (функция, решающая данную задачу) в модуле `parallel_map`:

```
simple_prepare_data([]) -> [];
simple_prepare_data(SourceList) -> simple_prepare_data(0,
SourceList, []).
simple_prepare_data(Index, [Element], PreparedData) ->
lists:reverse([Index, Element] ++ PreparedData);
simple_prepare_data(Index, [Element|Rest], PreparedData) ->
simple_prepare_data(Index + 1, Rest, [Index, Element] ++
PreparedData).
```

Принцип работы этой функции основан на рекурсии (а точнее, функции `simple_prepare_data/3`); при этом сама функция достаточно тривиальна, и детально разговаривать про нее мы не будем. Создание рабочих процессов и раздача им заданий можно объединить в нашем случае, т.к. у нас каждый процесс служит для выполнения только одного задания: применения функции отображения к одному из элементов исходного множества.

А теперь давайте поговорим о том, зачем нам нужно связывать с каждым из элементов его порядковый номер и почему мы нумерацию элементов начинаем с 0. Обычная, не многозадачная версия функции `map` работает следующим образом: последовательно обходит все элементы исходного списка, для каждого элемента вычисляет значение функции отображения от этого эле-

Что такое список в языке Erlang

Если вы работали с такими языками программирования, как C#, Java, Python, где списки являются контейнерами с произвольным доступом (по индексу) к их элементам, то вы можете ожидать от списков в языке Erlang такого поведения, которого здесь у списков нет. Список в Erlang – это структура данных для хранения элементов (которые предполагается обрабатывать одинаковым образом), дающая доступ к головному элементу (или к нескольким головным элементам) и остатку. Это осуществляется посредством операции соответствия шаблону [pattern-matching] `[Head | Tail]` (или `[Head1, ..., HeadN | Tail]`, если нам нужен доступ к нескольким головным элементам сразу), где `Head` – головной эле-

мент, `Tail` – остаток списка без головного элемента. Понятно, что работа со списком предполагает использование рекурсивного подхода (хвостовой рекурсии), когда в некоторой функции у списка выделяются головной элемент и остаток, головной элемент обрабатывается, и происходит вызов функции уже для остатка списка (и так, пока весь список не будет обработан). Список в языке Erlang по поведению подобен реализациям интерфейса `Iterable` в языке C#, `Iterable` в языке Java, итераторам в C++ и т.д. Если же в языке Erlang необходима структура данных с произвольным доступом к элементам (по индексу), следует использовать массивы, определенные в модуле `array`.

мента, и полученное значение добавляется к результирующему списку. Очень важно понять, что все это происходит последовательно! В случае же многозадачной версии, мы не можем гарантировать, что все задачи пришлют нам свои результаты работы в правильном порядке, даже если (как в нашем случае) мы их создавали (и запускали на выполнение) в правильном порядке. Решение этой проблемы достаточно простое и очевидное: необходимо с каждым элементом передавать его порядковый номер. Если функция рабочего процесса написана так, что она вместе с результатом работы возвращает и этот исходный порядковый номер, то мы сможем сохранять получаемые результаты по его порядковому номеру в некоторое хранилище. В момент готовности всех результатов работы их можно будет извлечь из хранилища и поместить в подходящую структуру данных. Хранилищами, которые позволяют сохранить некоторое значение по его порядковому номеру, являются массивы (определенные в модуле `array`). Так как нумерация элементов в массиве начинается с 0, то именно по этой причине мы также начинаем нумеровать элементы из исходного списка с 0. Ну, а подходящей структурой данных, как мы уже говорили, является список.

Функциональность (или средство), которая приостанавливает выполнение одной задачи, пока не будут получены все необходимые результаты от других задач, и собирает эти результаты, называется барьером. Данная функциональность у нас будет общей для нескольких вариантов реализаций функций `map` и `reduce`, поэтому ее реализация находится в паре функций `collect_result/2` (интерфейсная функция) и `collect_result/3` (функция, решающая данную задачу) модуля `parallel_common`:

```
collect_result(ResultStorage, TotalCount) -> collect_
result(ResultStorage, TotalCount, 0).
collect_result(ResultStorage, TotalCount, TotalCount) ->
ResultStorage;
```

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Сообщения об окончании жизни процесса

Если два процесса являются связанными и один из них – супервизор, то этот процесс-супервизор получит сообщение вида `{EXIT, From, Reason}`, когда второй процесс закончит свою работу. Здесь **From** – идентификатор процесса, закончившего работу, **Reason** – причина, по которой процесс закончил работу. Если второй процесс закончит свою работу

естественным образом, т.е. выполнив свою рабочую функцию, то причиной будет атом **normal**; в противном случае **Reason** будет содержать информацию о произошедшей ошибочной ситуации. Если же два процесса являются связанными, и ни один из них не является супервизором, то в случае естественного завершения одного из процессов

другой об этом никак не узнает, а в случае завершения одного из процессов из-за ошибки второй процесс также будет завершен. Естественно, что все изложенное выше справедливо и для случая, когда связанных процессов несколько (при этом процессы-супервизоры будут получать сообщения об окончании жизни, а обычные процессы – нет).

```
collect_result(ResultStorage, TotalCount, ProcessedCount) ->
receive
  {EXIT, _From, normal} -> collect_result(ResultStorage,
    TotalCount, ProcessedCount);
  {EXIT, _From, Reason} -> error({internal_error, Reason});
  {result, Index, DestElement} -> UpdatedResultStorage
    = array:set(Index, DestElement, ResultStorage), collect_
    result(UpdatedResultStorage, TotalCount, ProcessedCount + 1);
  _Other -> collect_result(ResultStorage, TotalCount,
    ProcessedCount)
end.
```

Для сбора результатов работы рабочих процессов наша барьерная функция должна уметь взаимодействовать с рабочими процессами, а если точнее – принимать от них сообщения с результатами работы. В этих сообщениях нам необходимо знать индекс исходного элемента и результирующий объект; идентификатор рабочего процесса нам не нужен, т.к. используемые рабочие процессы являются «одноразовыми». Поэтому мы ожидаем от рабочих процессов сообщения вида `{result, Index, DestElement}`, где **Index** – индекс исходного элемента, **DestElement** – результирующий элемент. Помимо этого сообщения, мы также обрабатываем сообщения об изменении состояния рабочих процессов: завершился ли рабочий процесс обычным образом или из-за ошибки. В первом случае мы ничего не делаем, во втором – завершаем главный процесс с ошибкой. И, наконец, мы обрабатываем все остальные сообщения; т.к. они не имеют для нас смысла, мы их просто извлекаем из очереди сообщений главного процесса и ничего не делаем.

Теперь можно перейти непосредственно к телу рабочей функции главного процесса. Это функция `simple_pmap/2`, определенная в модуле `parallel_map`. Эта функция является и точкой входа в реализуемый нами вариант, т.е. экспортируемой:

```
simple_pmap(_Fun, []) -> [];
simple_pmap(Fun, SourceList) ->
  process_flag(trap_exit, true),
  MasterPid = self(),
  ElementCount = length(SourceList),
  PreparedData = simple_prepare_data(SourceList),
  lists:foreach(fun({Index, Element}) -> spawn_link(fun() ->
    simple_worker(Fun, Element, Index, MasterPid) end) end,
    PreparedData),
  EmptyStorage = array:new({size, ElementCount}, {fixed, true},
    {default, none}),
  FullStorage = parallel_common:collect_result(EmptyStorage,
    ElementCount),
  process_flag(trap_exit, false),
  array:to_list(FullStorage).
```

Как уже говорилось выше, в рабочей функции главного процесса мы создаем задания для рабочих процессов, создаем необходимое количество рабочих процессов и раздаем им задания,

получаем результаты работы всех рабочих процессов и объединяем эти результаты в итоговом списке. Помимо этого, перед созданием рабочих процессов мы делаем главный процесс супервизором, а после сбора результатов работы рабочих процессов (после барьера) мы делаем главный процесс обычным процессом. Главный процесс обычно делаем супервизором для того, чтобы отслеживать завершение вспомогательных процессов и, при необходимости, реагировать на это (например, если вспомогательный процесс завершился из-за ошибки, то перезапустить его). В нашем случае мы для процессов, завершившихся с ошибкой, генерируем ошибку более высокого уровня (но содержащую исходную ошибку в качестве дополнительной информации). Так делают, чтобы абстрагироваться от деталей реализации, но, тем не менее, позволяя эти детали получить при необходимости.

Нам осталось рассмотреть, что делают рабочие процессы для выполнения своего задания. Функция `simple_worker/4` из модуля `parallel_map` является рабочей функцией таких процессов:

```
simple_worker(Fun, SourceElement, Index, MasterPid) ->
  DestElement = Fun(SourceElement),
  MasterPid ! {result, Index, DestElement}.
```

Так как рабочие процессы у нас «одноразовые», то их рабочая функция имеет очень простой вид: выполнить задание и послать результат выполнения задания обратно главному процессу. Для нашей задачи, задание рабочего процесса – это просто вычислить значение функции отображения для заданного исходного элемента.

Давайте проверим, что созданный нами многозадачный вариант функции `map` работает правильно. Для этого компилируем соответствующие модули и запускаем консоль среды выполнения языка Erlang. В ней набираем `parallel_map:simple_pmap(fun(Item) -> 3*Item end, [1, 2, 5, 8])` и получаем в результате список, состоящий из утроенных элементов исходного списка `[3, 6, 15, 24]`. Аналогичным образом набираем `parallel_map:simple_pmap(fun(Item) -> lists:reverse(Item) end, [{"str13", "str666"}])` и получаем список, состоящий из обратных строк `["31rts", «666rts»]`. Таким образом, мы можем сделать вывод, что данный вариант многозадачной реализации функции `map` работает правильно.

С простой многозадачной версией функции `map` мы закончили. Теперь возникает вопрос: можем ли мы, руководствуясь теми же принципами, написать столь же простую многозадачную версию функции `reduce`? Для ответа на этот вопрос давайте рассмотрим принципиальное отличие операции отображения (метод `map`) от операции свертки (метод `reduce`). В операции отображения (`map`) мы обрабатываем все элементы независимо друг от друга; именно поэтому мы можем обрабатывать их все параллельно. В операции свертки (`reduce`) мы вычисляем одно значение по всем элементам; это означает, что мы не можем взять какой-либо элемент и работать с ним независимо от остальных элементов. Рассмотрим, например, как мы вычисляем сумму элементов в списке. У нас есть начальное значение суммы (обычно это `0`);

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

мы берем значение первого элемента и складываем его с начальным значением, потом полученный результат складываем со значением второго элемента и т.д. Видно, что мы не можем взять и одновременно обработать первый и второй элементы.

Может показаться на первый взгляд, что мы не сможем создать многозадачную версию функции `reduce`, но не стоит отчаиваться: нам поможет такое свойство операции, как ассоциативность. Как уже говорилось выше, ассоциативность операции определяет, зависит ли ее результат от того, в каком порядке мы его вычисляем. Другими словами, если операция ассоциативна, то мы можем расставить скобки (выделить подмножества элементов) так, как нам хочется, вычислить результат операции согласно расставленным скобкам, и получим в качестве результата всегда одно и то же значение. Так, например, сумма чисел $1+2+3+4+5+6$ равна как сумме $(1+2)+(3+4)+(5+6)$, так и сумме $(1+2)+(3+4+5+6)$. Следует также сказать, что когда мы выделяем подгруппы для операции свертки, мы должны также задать начальное значение для операции свертки в подгруппе, или «ноль». Для нахождения суммы чисел этот «ноль» является числом `0`, для нахождения произведения чисел – `1`, для конкатенации строк – `""` (или `[]`), для нахождения произведения матриц – единичная матрица, и т.д. Понимание этого факта важно

по той причине, что такие функции свертки, как `lists:foldl/3` и `lists:foldr/3`, позволяют задать начальное значение для всей операции свертки, которое может отличаться от «ноля». Итак,

видно, что для создания многозадачной версии функции `reduce` нам необходимо сделать следующее: разбить исходный список на порции, порции обработать параллельно, после чего результаты параллельной обработки свернуть в итоговое значение.

Давайте еще поговорим о том, что делать, если операция свертки не ассоциативна. Возьмем, например, следующую разность: $1-2-3-4$, значение которой равно `-8`. Если мы сгруппируем элементы так, как мы это делали для суммы, то получим совсем другое значение. Так, например, группировка $(1-2)-(3-4)$ дает значение `0`. Но с точки зрения арифметики это неправильно, а правилен один из следующих вариантов: $(1-2)-(3+4)$ или $(1-2)+(-3-4)$. Давайте перепишем эти варианты следующим образом: $1-(2)-(3+4)$ и $1+(-2)+(-3-4)$. Становится ясно, что для выполнения операции свертки по группам (как для вычитания чисел, так и в общем случае) нам необходимы две операции свертки элементов: исходная и некоторая дополнительная. При этом мы либо применяем исходную операцию свертки для свертки внутри групп, а дополнительную для свертки результатов свертки для групп, либо наоборот. Также видно, что первый элемент в операции свертки не входит ни в какую группу. Как итог: написание операции свертки с группами для неассоциативной операции свертки приводит к дополнительным сложностям, никак не связанным с многозадачностью. Чтобы избежать этих сложностей, реализовывать многозадачный вариант функции `reduce` для таких операций свертки мы не будем, как уже говорилось выше.

Итак, мы пришли к такому понятию, как разбиение данных на порции. Мы обычно разбиваем исходный набор элементов на порции тогда, когда есть возможность обработать данные параллельно, но обработка одного элемента данных невыгодна. В случае операции свертки (функции `reduce`), мы можем разбить список исходных данных на группы из одного элемента и обработать их, после чего полученные результаты свернуть в итоговый результат. Очевидно, что это невыгодно и бессмысленно. В случае операции отображения (функции `map`), если функция отображения простая (например, удвоение аргумента), то параллельная обработка всех элементов также невыгодна. И в этом случае гораздо выгоднее обрабатывать параллельно порции данных.

В связи с этим возникает вполне естественный вопрос: а как выбирать размер таких порций данных? Хотя мы и можем теоретически прикинуть размеры порций, обычно их размеры ищут экспериментальным путем на примерах типичных данных. Мы об этом еще поговорим в нашем практикуме.

Пора двигаться дальше. Но прежде чем браться за реализацию параллельной версии функции `reduce`, следует создать несколько полезных функций для разбиения данных на порции. Нас интересуют две задачи: вычисление количества порций данных по размеру одной порции (и размеру исходных данных) и собственно разбиение исходных данных на порции. Первая задача реализована в функции `calc_portion_count/2` модуля `parallel_common`, которая весьма тривиальна:

```
calc_portion_count(TotalSize, PortionSize) when TotalSize rem
PortionSize == 0 ->
    TotalSize div PortionSize;
calc_portion_count(TotalSize, PortionSize) when TotalSize rem
PortionSize /= 0 ->
    (TotalSize div PortionSize) + 1.
```

Эта функция учитывает тот факт, что если размеры исходных данных и порции не кратны, то у нас появляется остаток (размер которого меньше размера порции), который также необходимо учитывать. Разбиение исходного списка данных на порции реализовано в функциях `prepare_data/2` (интерфейсная функция) и `prepare_data/3` (функция, решающая данную задачу) модуля `parallel_common`:

```
prepare_data(_PortionSize, []) -> [];
prepare_data(PortionSize, SourceList) -> prepare_data(0,
PortionSize, SourceList, []).
prepare_data(Index, PortionSize, SourceList, PreparedData)
when length(SourceList) =< PortionSize ->
    lists:reverse([Index, SourceList] ++ PreparedData);
prepare_data(Index, PortionSize, SourceList, PreparedData) ->
    {Portion, Rest} = lists:split(PortionSize, SourceList),
    prepare_data(Index + 1, PortionSize, Rest, [Index, Portion] ++
PreparedData).
```

Так же, как и в ситуации с простой многозадачной реализацией функции `map`, нам придется собирать результаты со всех рабочих процессов. Это означает, что нам точно так же необходимо связывать с порциями данных индексы, начинающиеся с `0` (только это уже будут индексы порций). Именно это и делают функции `prepare_data/2` и `prepare_data/3` – они создают список из пар (кортежей, состоящих из двух элементов: индекс порции и собственно порция), который мы используем в дальнейшем. Следует сказать, что все дальнейшие варианты многозадачных реализаций функций `map` и `reduce` будут использовать порции в качестве единицы работы для рабочих процессов.

На этом мы, пожалуй, сегодня остановимся: к сожалению, место в журнале для статьи ограничено. Давайте подведем промежуточный итог: мы увидели, что даже в самом простейшем случае (на примере многозадачной версии функции `map`) многозадачная версия больше и сложнее соответствующей не многозадачной версии функции. Мы увидели, что не всегда возможно подойти к задаче распараллеливания процесса вычисления в лоб: мы не всегда можем распараллелить обработку каждого элемента исходных данных. И мы начали рассматривать случай, когда параллельно у нас обрабатываются не единичные элементы, а порции. В следующем номере мы продолжим наш практикум и, в том числе, закончим создание многозадачных версий функций `map` и `reduce`, которые параллельно обрабатывают порции исходных данных. **LXF**

«Не всегда возможно подойти к распараллеливанию в лоб.»



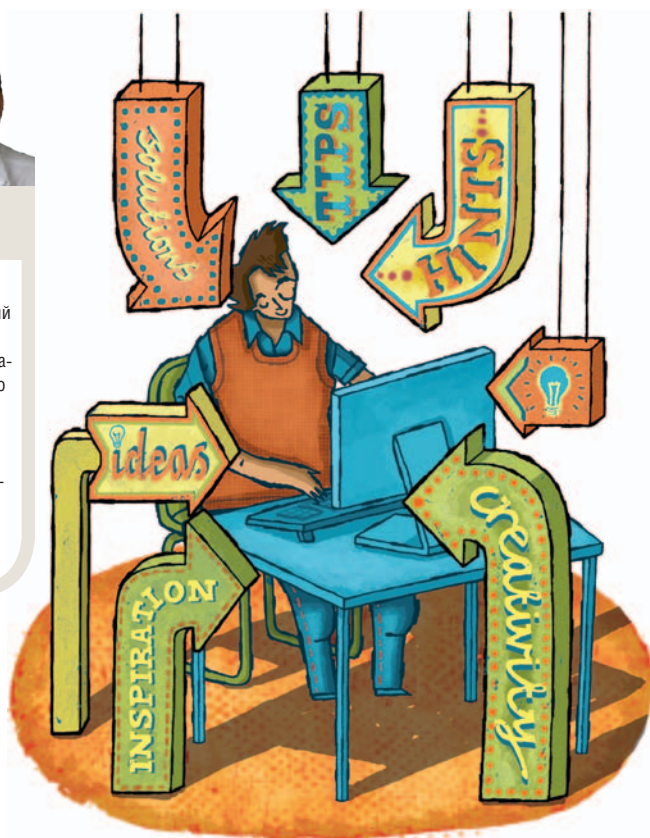
МНОГОМЕСТНОСТЬ:

Лео Максвелл показывает, как подключить к одному компьютеру несколько мониторов, мышей и клавиатур и осчастливить несколько пользователей.



Наш эксперт

Директор IT-компании и системный администратор **Лео Максвелл** оказывал аппаратную и программную поддержку пользователям своей и других сетей более 15 лет. Неудивительно, что он почти облысел.



Ваши дети переругались друг с другом из-за компьютера в гостиной? Счет за электричество заставляет потрясти головой и проверить баланс на банковском счете? Вам помогут многоместные системы. Существуют системы, в которых используется несколько мониторов, а в многоместных системах – несколько клавиатур, мышей, колонок и одновременно работающих пользователей.

Причины выбора таких систем – экономия пространства, денег или хотя бы электроэнергии. В некоторых странах, в том числе в ЮАР и Бразилии, в школах и общественных центрах установлены тысячи таких систем, чтобы доступ к компьютеру и Интернету получало максимальное количество учеников и обычных граждан. Такую систему можно развернуть даже на довольно слабом компьютере.

Для многих задач, таких как просмотр web-страниц, редактирование текста или чтение электронной почты, современный компьютер излишне производителен. Большинство настольных компьютеров, особенно многоядерных, большую часть времени простаивают. Если два человека пользуются одним компьютером, то его энергопотребление будет лишь немного превышать энергопотребление компьютера с одним пользователем и будет существенно ниже энергопотребления двух отдельных компьютеров.

Домашняя простоя двухпользовательская система может быть полезной, если нужен еще один дополнительный компьютер для ребенка, делающего домашнее задание, или для зашедших на время гостей.

Для нашего урока мы использовали компьютер с Ubuntu 12.04 с процессором AMD64×2 4200+, 4 Гб оперативной памяти, встроенной видеокартой Nvidia и внешней видеокартой Nvidia PCIe. Не слишком производительный по современным стандартам, этот компьютер вполне соответствовал поставленной задаче. Мы сознательно взяли более старое «железо», чтобы прояснить все вопросы с разделением задач. С целью основательной проверки мы попросили нескольких добровольцев поработать в обеих системах одновременно.

И мы были приятно удивлены: никаких проблем с производительностью! Два пользователя прекрасно бродили по Интернету, смотрели видео на YouTube, играли в простые игры и выполняли повседневные задачи, такие как обработка текста, без видимых задержек. В этой статье пользователей два, но их число ограничено только количеством видеокарт, которые можно установить в основном компьютере, так как количество доступных USB-портов всегда можно увеличить с помощью хабов.

На современной материнской плате обычно есть один слот PCIe и один или несколько слотов PCI, но есть и материнские платы, которые поддерживают до пяти видеокарт PCIe. Также существуют видеокарты USB.

В идеальном случае многоядерный процессор или процессор с гипертренингом предоставит каждому пользователю по одному ядру или процессу, а дополнительная память пригодится для трехмерных игр или обработки видео. Одного сетевого подключения в 100 Мбит более чем достаточно для нескольких интернет-пользователей.

Подготовка системы

Если что-то пойдет не так, система вполне может оказаться нерабочей. Если вы пользуетесь своим основным компьютером, обязательно создайте резервную копию данных.

Для начала вам понадобятся как минимум две учетных записи пользователя с отдельными домашними каталогами. Это важно, так как два «экземпляра» одного и того же пользователя будут обращаться к одному и тому же каталогу **home**, что может привести к нестабильности системы и потере данных.

Права администратора должны быть только у одного пользователя, главным образом для того, чтобы избежать некорректного выключения системы и бесконтрольных изменений файлов.

Тщательный выбор устройств поможет сделать работу системы более ровной. Настроить два монитора в Linux нетрудно. Если обоих подключить при установке, есть вероятность, что они будут обнаружены и настроены, но если нет, перед их использованием нужно убедиться, что они будут работать каждый в отдельном сеансе X.

Далее мы опишем этапы создания параллельного сеанса X в многопользовательской системе путем изменения файлов настройки вручную.

2 ГОЛОВЫ ЛУЧШЕ

Несколько отдельных сеансов

По сути, мы параллельно запустим два отдельных X-сервера. Нужно изменить файл `/etc/X11/xorg.conf`, создав две отдельных схемы, с монитором, мышью и клавиатурой каждая. Сервер X11 поддерживает несколько сессий, но базовый X-сервер ограничен одной сессией на видеокарту.

На нашем уроке мы пользуемся видеокартами Nvidia и их проприетарными драйверами. Учитывая изобилие возможных конфигураций, мы можем лишь научить вас настройке системы для конкретного «железа». Однако использованные здесь методы можно применить к самым разнообразным ситуациям.

Неплохо будет создать резервную копию `/etc/X11/xorg.conf`, чтобы в случае сбоя войти в систему с командной строки, по SSH с другого компьютера или с Live CD, и восстановить его, чтобы восстановить систему. Настроив два сеанса, нужно параллельно запустить видеотерминалы с помощью менеджера дисплея.

Первоначально мы настроили только одну схему, закомментировав половину разделов, чтобы убедиться, что у нас два рабочих сеанса.

Проблема современных настольных дистрибутивов – степень их автоматизации. Вам нужно сделать так, чтобы HAL и Udev не добавляли устройства автоматически, иначе все устройства будут работать на всех мониторах, что, мягко говоря, раздражает.

Идентификация устройств

Видеокарты можно определить, набрав команду `lspci | grep VGA`. Проприетарные драйверы Nvidia позволяют настраивать мониторы с помощью утилиты `nvidia-settings`. Она также создаст базовую версию `xorg.conf` с информацией о видеокарте и мониторе. Затем эту версию можно изменить. Драйверы AMD также должны работать.

Для клавиатуры и мыши мы воспользуемся драйвером `evdev`, который использует такие события, как `xxx-event-kbd` и `xxx-event-mouse`. Они находятся в каталоге `/dev/input` – это файлы вроде `mice`, `mouse0`, `mouse` и т. д. Они привязаны к драйверу мыши и не подходят для нашей задачи, так как могут меняться или быть захвачены событиями клавиатуры или мыши.

В `/dev/input` также есть пронумерованные события, связанные с событиями в подкаталогах `by-id` и `by-path`. Так как это динамические ссылки, устройства, на которые они ссылаются, могут измениться, если они отключаются или если перед загрузкой добавляется другое устройство.

Чтобы еще больше запутать ситуацию, иногда возможны множественные события, у клавиатуры могут быть события мыши, а у мыши – события клавиатуры. Разумеется, у каждого устройства должен быть уникальный идентификатор.

Следующие команды:

```
ls -R /dev/input
ls -R /dev/input | grep mouse
ls -R /dev/input | grep kbd
```

выведут список всех событий. Записи в `/dev/input/by-id/` удобны, если используются различные модели, так как они остаются постоянными.



» Один компьютер, два сеанса. Здесь у нас два монитора, две беспроводные клавиатуры и мыши и два набора USB-колонок.

Если мыши и клавиатуры одинаковые, то некоторые файлы в `by-id` могут использоваться совместно, и нужно использовать файлы в каталоге `by-path` (они изменятся, если подключить устройство к другому порту USB). Пример вывода приведен ниже, корректные значения в нем выделены жирным. Первое – из каталога `/dev/input/by-id`, а второе – из каталога `/dev/input/by-path`. Нам нужны записи, которые заканчиваются на `-event-mouse` или `-event-kbd`.

```
$ ls -R /dev/input | grep mouse
mouse1
usb-Logitech_USB_Receiver-event-mouse
usb-Logitech_USB_Receiver-if01-event-mouse
```

»

«Железо» для многоместных систем

Нужно подчеркнуть, что чем проще и стандартнее «железо», тем проще заставить систему работать. Приведенные ниже спецификации – конфигурация по абсолютному минимуму, но вы удивитесь, как хорошо будут работать дешевые системы.

» **Процессор** Как минимум P4 2,4 ГГц.

В идеальном случае – многоядерный процессор, но это не обязательно.

» **Оперативная память** 512 МБ. 1 ГБ – для двух нетребовательных пользователей уже много, но чем больше, тем лучше. Рекомендуется добавлять не менее 256 МБ на пользователя.

» **Дисковое пространство** Минимум – 20 ГБ для двух пользователей, но чем больше, тем лучше.

» **Видеокарты** Не менее 16–32 МБ видеопамяти. По одной видеокарте на пользователя, желательно одинаковые карты одного производителя с одними и теми же драйверами. Вообще говоря, это не обязательно, но так гораздо проще устранять неисправности. Сгодятся даже карты PCI.

» **Мониторы** Желательно одинаковые или поддерживающие одно и то же разрешение.

» **USB-хабы** С целью получения лучших результатов для каждого сеанса нужно

использовать отдельный хаб. Рекомендуются хабы с внешним питанием, так как некоторые устройства потребляют довольно много энергии.

» **Клавиатуры и мыши** Любые мыши или клавиатуры с интерфейсом USB, в том числе и беспроводные. Хотя последние помогают уменьшить количество проводов, будьте осторожны, так как беспроводные клавиатуры/мыши могут влиять друг на друга.

» **Звук** Подойдут звуковые карты или колонки с интерфейсом USB со встроенной звуковой картой. Если включить их в хаб с клавиатурой и мышью, все будет чисто. Единственное, о чем стоит беспокоиться – шум. Если эта конфигурация будет постоянной, вам понадобятся либо длинные кабели, либо наушники для одного или нескольких пользователей.

» **USB-брелки, камеры и т. д.** По умолчанию устройства USB и Bluetooth будут доступны только первому пользователю, вошедшему в систему. Если это нежелательно, измените конфигурацию `policykit`.

» **Принтеры, сканеры и т. д.** Должны подключаться в порты на материнской плате или по сети и быть доступны всем сеансам.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

► **LibreOffice Writer** в стерео! Даже печать работает отлично.



```
usb-Logitech_USB_Receiver-if01-mouse
usb-Logitech_USB_Receiver-mouse
pci-0000:00:10.0-usb-0:2:1.1-event-mouse
pci-0000:00:10.0-usb-0:2:1.1-mouse
```

Идентифицировав устройства, добавьте их в файл `/etc/X11/xorg.conf`. В следующем разделе мы разберем пример этого файла. Информацию об устройствах, разумеется, нужно изменить в соответствии с вашей.

Первый раздел – **ServerFlags**, в нем находятся настройки по умолчанию для всех серверов. Наиболее важны строки, где отключается автоматическое добавление устройств.

```
# X configuration file
```

```
Section "ServerFlags"
```

```
Option "Xinerama" "off"
```

```
Option "AutoAddDevices" "off"
```

```
Option "AllowEmptyInput" "on"
```

```
Option "AllowMouseOpenFail" "on"
```

```
Option "DisableModinDev" "on"
```

```
Option "DRM" "on"
```

```
EndSection
```

Для каждого сервера вам понадобится раздел **ServerLayout** с информацией о мониторе и устройствах ввода. Опция **SingleCard** ограничивает resets одним устройством:

```
Section "ServerLayout"
```

```
Identifier "Layout0"
```

```
Screen 0 "Screen0" 0 0
```

```
InputDevice "Keyboard0"
```

```
InputDevice "Mouse0"
```

```
Option "SingleCard" "on"
```

```
EndSection
```

```
Section "ServerLayout"
```

```
Identifier "Layout1"
```

```
Screen 0 "Screen1" 0 0
```

```
InputDevice "Keyboard1"
```

```
InputDevice "Mouse1"
```

```
Option "SingleCard" "on"
```

```
EndSection
```

Теперь создайте раздел для каждого устройства ввода. Обычно это мыши и клавиатуры, но возможны и другие устройства, скажем, графические планшеты или сенсорные экраны. Мы использовали два разных набора беспроводных клавиатур/мышей. Опция **GrabDevice** запрещает другим драйверам доступ к устройству. Раскладку клавиатуры лучше задать в системных настройках Ubuntu, так как они хранятся отдельно для каждого пользователя.

```
Section "InputDevice"
```

```
Identifier "Mouse0"
```

```
Driver "evdev"
```

```
Option "Device" "/dev/input/by-id/
```

```
usb-Logitech_USB_Receiver-event-mouse"
```

```
Option "GrabDevice" "on"
```

```
EndSection
```

```
Section "InputDevice"
```

```
Identifier "Mouse1"
```

```
Driver "evdev"
```

```
Option "Device" "/dev/input/by-id/
```

```
usb-MOSART_Semi._Wireless_Keyboard___Mouse-event-mouse"
```

```
Option "GrabDevice" "on"
```

```
EndSection
```

```
Section "InputDevice"
```

```
Identifier "Keyboard0"
```

```
Driver "evdev"
```

```
Option "Device" "/dev/input/by-id/
```

```
usb-Logitech_USB_Receiver-event-kbd"
```

```
Option "Xkb_Rules" "evdev"
```

```
Option "Xkb_Model" "evdev"
```

```
Option "GrabDevice" "on"
```

```
EndSection
```

```
Section "InputDevice"
```

```
Identifier "Keyboard1"
```

```
Driver "evdev"
```

```
Option "Device" "/dev/input/by-id/
```

```
usb-MOSART_Semi._Wireless_Keyboard___Mouse-event-kbd"
```

```
Option "Xkb_Rules" "evdev"
```

```
Option "Xkb_Model" "evdev"
```

```
Option "GrabDevice" "on"
```

```
EndSection
```

Наконец, укажите мониторы, устройства (видеокарты) и экраны, к которым они подключаются. Мы пользовались двумя 19-дюймовыми мониторами с разрешением 1440×900 пикселей – опция **DPMS** активирует режим пониженного энергопотребления **Energy Star**:

```
Section "Monitor"
```

```
Identifier "Monitor0"
```

```
VendorName "Unknown"
```

```
ModelName "Bit 3 Vw19B2"
```

```
HorizSync 31.0 - 83.0
```

```
VertRefresh 56.0 - 76.0
```

```
Option "DPMS"
```

```
EndSection
```

```
Section "Monitor"
```

```
Identifier "Monitor1"
```

```
VendorName "Unknown"
```

```
ModelName "Acer AL1916W"
```

```
HorizSync 31.0 - 84.0
```

```
VertRefresh 56.0 - 76.0
```

```
Option "DPMS"
```

```
EndSection
```

Оба наших устройства – видеокарты Nvidia, одна на материнской плате, а другая в слоте PCIe.

Про файл `xorg.conf`

Файл `xorg.conf` (расположен в каталоге `/etc/X11`) запугал немало опытных линуксоидов. Он анализируется при запуске X-сервера, а его синтаксис темен.

Сегодня в большинстве дистрибутивов X-сервер настраивается автоматически, и файла `xorg.conf` может не быть. Хотя это может приводить к трудно решаемым проблемам, настройка вручную сегодня почти не употребляется. Обстоятельства вроде наших, когда надо сделать нечто необычное – это исключение.

Формат может показаться немного сложным, но лишь из-за его гибкости. Многие примеры из тех, которые можно найти в сети, имеют явные признаки сборки методом копирования и вставки. Люди часто копируют и вставляют фраг-

менты, забывая убрать ненужное. Скажем, в разделе **ServerFlags** задаются настройки по умолчанию, но их можно перегрузить в любом другом разделе. Параметры и значения не чувствительны к регистру, порядок разделов не важен, а булевские параметры можно задавать как 0 и 1, "on" и "off", "true" и "false" или "yes" и "no". Есть и почти бесконечный набор опций, и не все они хорошо задокументированы. Мелкая опечатка – и все может забастовать, но, по крайней мере, у нас есть хорошие логи.

В наших примерах я постарался быть последовательным и исключил максимально возможное количество «лишних» строк, но если что-то будет работать не так, первым делом загляните в каталог `/var/log`.

Другие способы

Как всегда в мире Linux, многоместную систему можно создать разными способами. Можно воспользоваться конфигурацией с двумя мониторами в диспетчере виртуальных машин, таком как *VMware* или *VirtualBox*, и запустить два сеанса в виртуальной машине. Видеокарты с двумя выходами можно запустить на вложенных X-серверах на Xephyr, а настроит такую систему скрипт на Python, версию которого для Ubuntu можно загрузить с <http://code.google.com/p/multiseat-wizard-bicefalo>. Он зависит

от пакета *xserver-xephyr* и работает только с открытыми драйверами по умолчанию. Оба этих варианта более требовательны к ресурсам по сравнению с нашим. Существует коммерческая версия многопользовательской системы под Linux от канадской компании Userful. Она поддерживает до 10 пользователей на компьютер через параллельные сеансы X-сервера и широко используется в Бразилии и других странах. Домашнюю лицензию для двух пользователей можно получить бесплатно, зарегистрировавшись на www.userful.com. Доступны версии системы для нескольких дистрибутивов Linux, но они довольно старые. Последняя поддерживаемая версия Ubuntu на данный момент – 10.04 LTS.

Fedora 17, видимо, поддерживает Zero-clients через *systemd*. Существуют хабы со встроенными видеоадаптерами и звуковыми картами, подключаемые к компьютеру по Ethernet или USB. Производители модулей включают HP, Acer, Viewsonic и Plugable.

Если у вас есть проблемы с идентификацией устройств, отключайте и заново подключайте их по одному и наблюдайте за выводом команд *lsusb* и *dmesg*.

```
Section "Device"
Identifier "Device0"
Driver "nvidia"
VendorName "NVIDIA Corporation"
BoardName "GeForce 8100 / nForce 720a"
BusID "PCI:2:0:0"
EndSection
Section "Device"
Identifier "Device1"
Driver "nvidia"
VendorName "NVIDIA Corporation"
BoardName "GeForce 8400 GS"
BusID "PCI:3:0:0"
EndSection
```

Экран – это пара видеокарта–монитор.

```
Section "Screen"
Identifier "Screen0"
Device "Device0"
Monitor "Monitor0"
DefaultDepth 24
Option "metamodes" "nvidia-auto-select +0+0;1440x900_75 +0+0"
SubSection "Display"
Depth 24
EndSubSection
EndSection
Section "Screen"
Identifier "Screen1"
Device "Device1"
Monitor "Monitor1"
DefaultDepth 24
Option "metamodes" "nvidia-auto-select +0+0;1440x900_75 +0+0"
SubSection "Display"
Depth 24
EndSubSection
EndSection
```

Когда у вас будет рабочий файл *xorg.conf* (т.е. тот, с которым операционная система загружается без ошибок, при этом будут работать оба монитора, обе мыши и обе клавиатуры), настройте менеджер дисплея. Он позволит пользователям заходить в различные сеансы видеотерминала и распределять между ними доступные устройства.

Да будет LightDM

Для этой статьи мы выбрали менеджер дисплея Ubuntu по умолчанию, LightDM, потому что он поддерживает несколько сессий с различными мониторами и синтаксис его файла настройки довольно прост. Он также поддерживает как KDE, так и Gnome. KDM (менеджер дисплея KDE) тоже поддерживает несколько сессий, но GDM (менеджер дисплея Gnome) – нет. В файле */etc/lightdm/lightdm.conf* по умолчанию текста немного, но скоро мы это из-

меним! Нам нужно создать записи для каждого пользовательского места, указав команды X-сервера и соответствующие схемы.

```
[LightDM]
seats=Seat:0 Seat:1
[SeatDefaults]
user-session=ubuntu
greeter-session=unity-greeter
[Seat:0]
xserver-layout=Layout0
xserver-command=/usr/bin/X :0 -layout Layout0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
[Seat:1]
xserver-layout=Layout1
xserver-command=/usr/bin/X :1 -layout Layout1 -auth /var/run/lightdm/root/:1 -nolisten tcp vt8 -novtswitch -sharevts
```

Опция *novtswitch* запрещает переключение между сессиями, а *sharevts* позволяет держать открытыми несколько видеотерминалов. Для первого сеанса включать ее не стоит – можно повредить монитор. Если вам нужен автоматический вход в систему, добавьте следующие строки (подставив свое имя пользователя) в каждый раздел:

```
autologin-user=username
autologin-timeout=0
```

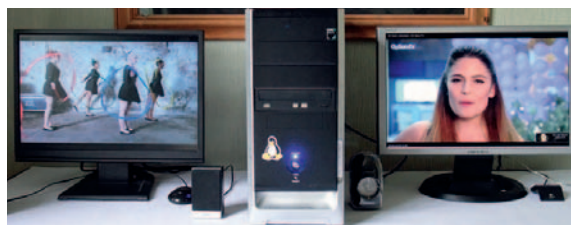
Это также решает проблему с не отображавшимся полем пароля у некоторых пользователей.

Звук

Звук проще всего настроить для каждого пользователя отдельно, с несколькими звуковыми картами. Для начала добавьте обоим пользователям в группу *audio*.

После этого каждый пользователь сможет войти в систему и задать настройки звука для своего устройства. При использовании *PulseAudio* настройки автоматически сохраняются в их файле по умолчанию в */home/username/pulse*. Конечно, если в следующий раз пользователи поменяются местами, звук придется перенастраивать.

Настольный Linux сейчас находится в состоянии активных перемен. Большая автоматизация снижает гибкость, а множественные сеансы могут также повлиять на безопасность. Этот подход может показаться несколько нудным, но зато работает в большинстве дистрибутивов. **LSB**



Скорая помощь

Если у вас есть проблемы с идентификацией устройств, отключайте и заново подключайте их по одному и наблюдайте за выводом команд *lsusb* и *dmesg*.

► Octopus против GuruTV. На обоих мониторах YouTube запущен в полноэкранный режим, и никакого подторжания. Все конфликты между пользователями свелись к музыкальным вкусам.

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru, и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Создание загрузочных дисков аварийного восстановления системы
- 2 Копирование файлов на сервер
- 3 Настройки VOIP-телефонии
- 4 Настройки мультимедийных загрузочных систем
- 5 Основы DHCP

1 Спасательный DVD

В мои первый опыт общения с Linux был Slackware 2.2 на 80 установочных дисках с Fidonet для почты. Теперь все не так, и у меня openSUSE 12. Помогая друзьям и домашним в решении проблем с Windows, я устал от того, что систему каждый раз приходится переустанавливать с нуля — потому что они не могут найти диски восстановления системы или вообще не создавали их.

Я хотел бы приводить соответствующий компьютер в рабочее и актуальное состояние и затем создать набор загрузочных CD/DVD в Linux для восстановления компьютера.

Грэм Поузи (Graham Pawsey)

Это возможно, но придется пожонглировать дисками, так как вам понадобятся несколько DVD для хранения ряда текущих версий системы. Другой вариант — создать диск аварийного восстановления системы, а резервные копии хранить на внешнем жестком диске или даже на объемной флешке, так как сейчас они дешевле.

Это также означает, что не придется хранить по несколько дисков для каждого компьютера — достаточно стандартного загрузочного диска и диска с резервными копиями для всех компьютеров.

Одно из лучших средств для решения этой задачи — Clonezilla, специально предназначенный для нее Live CD. Clonezilla может создавать резервные копии на жестком диске, USB-дисках, DVD-дисках или образы ISO для последующей записи на оптические диски или на сетевой сервер. Загрузить его можно с сайта <http://clonezilla.org/downloads.php>, но версий дистрибутива существует несколько.

Для текущего стабильного релиза их не менее шести. Версия amd64 предназначена для 64-битных систем (как AMD, так и Intel), i686 — для современных 32-битных платформ, а i486 работает почти на всех платформах. Версию выбирайте в соответствии со своей архитектурой, а резервные копии они создают одинаковые.

Каждую из этих версий можно загрузить в виде ISO-образа для записи на CD и в виде архива ZIP для копирования на флешку. Второй вариант чуть сложнее, но на сайте Clonezilla есть подробные инструкции.

После загрузки Clonezilla и выбора языка можно запустить Clonezilla [Start Clonezilla] и пройти по этапам создания резервной копии, или открыть командную строку [Enter shell] и сделать все вручную. Выберите первый вариант и укажите расположение резервных копий — это может быть локальный каталог или один из типов сетевого сервера. Выбор режима начинающего [Beginner] или эксперта [Expert] обычно очевиден — если у вас есть сомнения, выбирайте первый. Затем нужно выбрать, стоит ли создавать резервную копию всего диска или отдельных разделов. Для диска аварийного восстановления создайте копию всего диска, потому что она содержит таблицу разделов.

После выбора диска, резервную копию которого нужно создать, установите опцию проверки образа на восстановление. Это увеличивает время его создания, но образ, который невозможно восстановить, полностью бесполезен.

На создание резервной копии может потребоваться некоторое время.

Другие опции включают разбиение копии на фрагменты по размеру CD или создание самодостаточного диска восстановления. Если запустить этот процесс снова, вы увидите дополнительные опции, так как теперь у нас есть резервная копия для восстановления.

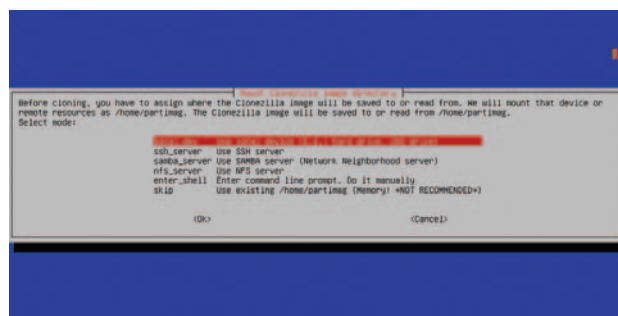
2 Ну что, готово?

В у меня есть файл, который копируется на сервер (с ftp или scp). По окончании копирования нужно перенести его еще кое-куда. Есть ли изящный/простой способ узнать, что файл скопирован полностью, прежде чем копировать его в другое место? Можно выполнить следующие команды:

```
MD5sum $file; sleep 2s; MD5sum $file
```

и сравнить контрольные суммы, чтобы определить, что файл еще копируется, но этот способ выглядит довольно сложным.

Сделать это можно разными способами, в зависимости от обстоятельств. Если копирование было запущено на локаль-



Создайте загрузочный диск аварийного восстановления Linux или Windows (или обеих систем) с помощью Clonezilla.

ном компьютере, можно просто подождать, пока процесс копирования завершится, даже если он выполняется в фоновом режиме. Если команда копирования была запущена в той же оболочке, то команда

```
wait PID
```

завершится, когда завершится процесс с заданным идентификатором. Если копирование по SSH/SCP было запущено удаленно и вы знаете имя хоста или IP-адрес отправителя, можно поискать в системном лог-файле окончание сеанса.

```
tail -f -n 0 /var/log/messages | grep --quiet --max-count 0 'Received disconnect from IPADDRESS'
```

Команда `tail` следит за системным лог-файлом, выводя все новые записи. Команда `grep` ищет сообщение об отключении. Параметр `--max-count` задает количество вхождений. Оно установлено в ноль, так как после обнаружения данного количества вхождений `grep` завершается, и эта команда завершится по завершении сеанса SSH. Учтите, что если отключения не будет обнаружено, то команда не завершится никогда, поэтому нужно добавить несколько проверок. Те же действия можно сделать для лог-файла FTP. Другой вариант — определить, блокируется ли файл каким-то процессом, с помощью команды `lsof` или `fuser`. Если файл заблокирован, значит, он еще копируется.

Это не только экономит ресурсы по сравнению с `md5sums`, но и позволит избежать вероятных ошибок при изменении файла, контрольная сумма которого вычисляется. Команда

```
fuser --silent /path/to/file
```

вернет код 0, если файл используется, и 1, если нет, поэтому команды

```
while fuser --silent /path/to/file; do  
sleep 1m
```

```
будут выполняться в цикле до тех пор, пока копирование не завершится. Было ли оно завершено успешно или было прервано, предстоит определить вам. Для более общих действий, например,
```


для обработки каждого файла, загружаемого в каталог, вам пригодится *incron* (<http://incron.aiken.cz>). Он работает так же, как и *cron*, но срабатывает не по времени, а по оповещениям доступа к файлам. Поместите примерно такую строку в файл `/etc/incron.d`:

```
/path/to/watch IN_CLOSE_WRITE /command/to/run $#
```

Эта команда отслеживает каталог, указанный в начале строки, на события закрытия файлов, открытых на запись – именно то, что вам нужно. При обнаружении таких событий выполняется указанная команда, причем `$#` заменяется именем файла. В качестве команды можно указать скрипт, который будет проверять, корректен ли файл, и затем выполнять любые необходимые действия.

3 Осложнения в сети

В Это должно быть так просто. Для установки VOIP-телефона наберите адрес в браузере *Firefox*. IP-адрес моего Cisco SPA 301—192.168.1.4, но открыв его, я получил сообщение «Cannot access the device [Устройство недоступно]». Я попробовал установить беспроводную точку доступа D-Link (DWL-2100AP) с адресом 192.168.0.50 и получил такое же сообщение. Пока, насколько мне известно, точки не защищены паролями.

В руководстве по D-Link написано, что перед началом установки нужно назначить компьютеру статический IP-адрес, но я не знаю, должен ли это делать я или провайдер (и если я, то как). И какой статический адрес назначать? Тот же, что и у устанавливаемого устройства, текущий IP-адрес компьютера или вообще любой IP-адрес, который взбрдет в голову?

Я довольно хорошо знаком с IT, но уже этой ужасной истории хватит, чтобы превратить меня в луддита. Можете ли вы вернуть мне веру в чудеса технического прогресса?

Морис Жорж [Maurice George]

О Это не имеет отношения к провайдеру. Адреса вашего компьютера, телефона и всех прочих устройств, в том числе и маршрутизатора – это адреса в локальной сети. Провайдер назначает вам один IP-адрес, внешний адрес маршрутизатора (узнать его можно на странице статуса маршрутизатора в его web-интерфейсе или зайдя на специальный сайт, например, www.whatismyip.com). Он маршрутизирует трафик между локальной сетью (LAN) и Интернетом в целом, который иногда называют WAN (Wide Area Network – глобальная сеть).

Указанные вами адреса 192.168 – это локальные адреса из диапазона, зарезервированного для частных сетей, но они принадлежат разным подсетям. В диапазоне адресов вида 192.168.x.y первые три числа задают подсеть, поэтому все ваши адреса, в том числе адрес маршрутизатора, должны начинаться с 192.168.1 или 192.168.0, но только с одного из них. Заглянув в любую программу настройки сети, например, в раздел Network (Сеть) системных настроек Ubuntu, вы найдете свой IP-адрес и маску подсети.

Какие адреса относятся к локальной подсети, определяется маской подсети. Обычно это 255.255.255.0. Это означает, что первые три числа каждого локального адреса должны быть одними и теми же. Для задания статического адреса нажмите на Options [Параметры], перейдите на вкладку IPv4 Settings [Настройки IPv4], выберите

Manual [Вручную] и задайте адрес, маску подсети и шлюз. Маска и шлюз должны быть такими же, какие были заданы DHCP. Адрес должен быть другим, но находиться в той же подсети, что и шлюз. Серверы DHCP выделяют адреса из определенного диапазона подсети, например, от 192.168.1.100 до 192.168.1.200. Статические адреса нужно брать вне этого диапазона.

Заглянув в web-интерфейс своего маршрутизатора (если вы не делали этого прежде, то достаточно набрать в адресной строке тот же адрес, что был задан в сетевых настройках роутера), вы найдете там диапазон адресов, выделяемых DHCP. Выберите какой-нибудь адрес вне этого диапазона. Затем назначьте устройству другой адрес из того же блока адресов вне DHCP. Тогда у каждого из устройств будет уникальный адрес, и все они будут находиться в одной подсети и смогут взаимодействовать друг с другом.

4 Много-много Linux'ов

В Я новичок в мультизагрузочных системах, и мне нужен совет, как найти лучшую конфигурацию для своей системы (систем). Должен ли /boot находиться на первом основном разделе? Это не одно и то же, что и MBR? Должен ли быть у всех устанавливаемых дистрибутивов общий раздел /boot? Если да, то как построить его так, чтобы следующий дистрибутив не перезаписал его? Ведь если он перезапишется, я смогу загрузить только последний установленный дистрибутив.

Можно ли сделать так, чтобы все дистрибутивы безопасно работали с общим разделом подкачки? (Я знаю о том, что перевод одной системы в спящий режим и затем загрузка другой может



Коротко про...

X-проброс

X, основа всех графических интерфейсов Linux – это сетевой протокол. Хотя ваш монитор обычно связан с X-сервером на локальном компьютере, это не обязательно. Возможно, вы подключались к удаленным компьютерам с помощью VNC или Nx, но их недостаток в том, что они полностью воспроизводят рабочий стол другого компьютера. В свою очередь, для этого на удаленном компьютере должен быть рабочий стол. Если удаленный компьютер является автономным сервером, то рабочий стол – напрасная трата системных ресурсов. Вместо того чтобы запускать рабочий стол целиком, можно запустить только ту программу, которая вам нужна, по SSH (это означает, что подключение будет безопасным даже в публичных сетях). В своем любимом терминале наберите:

```
ssh -X user@remote.computer someprogram
```

Если *someprogram* – графическое приложение, то на вашем рабочем столе должно от-

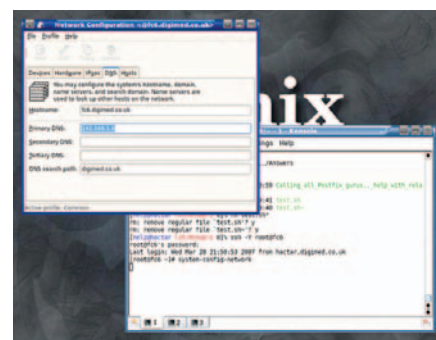
крыться его окно. И здесь есть несколько моментов, о которых нужно помнить.

Во-первых, у вас должен быть доступ к компьютеру по SSH с паролем или ключом. На сервере SSH на другом компьютере должно быть включено перенаправление X (X-проброс). Обычно оно включено по умолчанию, но на всякий случай проверьте, что в файле `/etc/ssh/sshd_config` есть строка

```
X11Forwarding yes
```

У ssh имеются и другие полезные параметры. Вместо `-X` можно использовать `-Y`. Он работает только с доверенным перенаправлением X11 (Trusted X11 Forwarding), зато заметно быстрее. Еще один параметр, прилично увеличивающий скорость при низкоскоростном подключении – `-C`; он включает сжатие всех данных, хотя для по-настоящему медленных подключений X не годится, здесь лучше будет выбрать *TightVNC* или *Nx*.

Интересно отметить, что в MacOS для отображения видеоданных также используется X, поэтому таким способом вы сможете отображать окно с программой Mac на компьютере с Linux и наоборот.



➤ Запуск программы настройки сети Fedora на рабочем столе другого компьютера.

привести к ошибкам.) С общим разделом /home проблемы могут возникнуть, насколько я понимаю, с разными настройками одних и тех же приложений (или различных версий приложения) в разных дистрибутивах. Сколько места на диске нужно выделить каждому дистрибутиву? Можно ли в одном из дистрибутивов запустить другой, установленный на жестком диске, в виртуальной машине *VirtualBox*?

Я понимаю, что можно было просто установить один «основной» дистрибутив и затем установить другие в виртуальные машины. Но я хотел бы разместить их виртуальные диски на отдельных разделах, чтобы не заполнять свой раздел /home большими файлами.

С форумов

Главная загрузочная запись (MBR) – небольшая, размером 512 байт, область на жестком диске, которая содержит таблицу разделов и загрузчик. Это не то же самое, что /boot, который представляет собой просто каталог в файловой системе. Его можно сделать общим, но это не рекомендуется, потому что вам придется заботиться об именах ядер. Раздел подкачки можно сделать общим, если он не используется для спящего режима. Я бы выбрал схему с общим разделом подкачки, отдельным корневым разделом для каждого дистрибутива и общим домашним разделом с отдельными каталогами для каждого дистрибутива на нем.

Ни один из дистрибутивов не обязан находиться на первичном разделе. *Grub* (загрузчик, используемый в большинстве дистрибутивов) может загрузиться и с логического раздела, и в системе Linux может не быть ни одного первичного раздела. Если у вас общий раздел /home, то на разделах для дистрибутивов нужно оставить место только для файлов операционных систем – по 10–12 ГБ на дистрибутив достаточно.

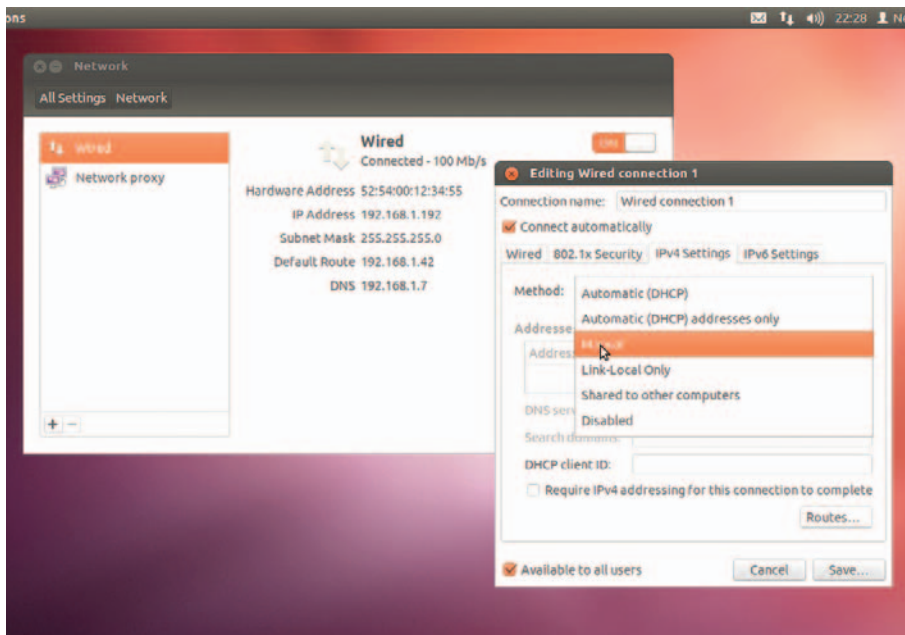
Хотя общий раздел /home – хорошая идея, для своего пользователя в каждой системе следует создать отдельные домашние каталоги. Можете пользоваться одним и тем же именем пользователя – домашний каталог не обязательно должен быть вида /home/user, это лишь общепринятое соглашение.

5 Неувязки DHCP

У меня есть приставка *MythTV* с сетевым именем *civo* и статическим IP-адресом 192.168.1.10. Иногда у нее возникает желание подключиться к моей сети по Wi-Fi, и она получает адрес по DHCP (например, 192.168.1.100). После этого при пинговании *civo* я получаю адрес 192.168.1.100 – это не то, чего бы мне хотелось. В web-интерфейсе *OpenWrt* я задал *civo* как имя устройства для IP-адреса 192.168.1.10.

Я думаю, что в *OpenWrt* для DNS используется *Dnsmasq*, так что, наверное, это ее проблема или «особенность». Можно ли отключить эту возможность или изменить порядок запроса адресов, чтобы использовались имена хостов, заданные первыми?

bobthebob1234, с форумов



Хотя в настройках сети по умолчанию обычно используется DHCP, при необходимости все настройки можно задать вручную.

Как и большинство DHCP-серверов, *Dnsmasq* определяет адрес, который следует назначить сетевой карте, по ее MAC-адресу. Тогда, если вы отключитесь и подключитесь снова, то обычно получите один и тот же адрес. У вас две сетевые карты, одна проводная и одна беспроводная, с разными MAC-адресами. Даже если воспользоваться DHCP для обеих, их адреса все равно будут разными, но задать проводному интерфейсу статический адрес, то сервер DHCP даже не узнает о нем, поэтому он и не замечает статического адреса.

Исправляется это разными способами. Простейший – отключить беспроводной адаптер, если он не нужен, и тогда подключение всегда будет осуществляться по проводному интерфейсу. Если его можно отключить аппаратно или в BIOS, сделайте это. Если нет, можно добавить его модуль в «черный список». В терминале выполните команду:

```
lspci -k
Для просмотра используемого интерфейсом драйвера может потребоваться снабдить ее префиксом sudo. Затем добавьте строку:
blacklist modulename
```

в файл в каталоге /etc/modprobe.d. Дайте ему понятное имя, например *blacklist-wireless.conf*. Файл должен иметь расширение *.conf*.

Второй вариант – сделать так, чтобы у беспроводного интерфейса тоже был статический адрес, и задать ему тот же адрес, что и у проводного. Это сработает, если вы не будете подключать оба одновременно. Наконец, можно включить DHCP для проводного интерфейса и настроить *Dnsmasq* на назначение всем интерфейсам одного и того же адреса. Для настройки интерфейса в *Dnsmasq* используется параметр *dhcp-host*; например, строка:

```
dhcp-host=00:22:68:66:B2:79,192.168.1.16,oolon,
infinite
```

задает компьютеру, имеющему MAC-адрес 00:22:68:66:B2:79, IP-адрес 192.168.1.16, имя хоста *oolon* и бесконечное время аренды (это клиент *MythTV* в моей сети). Указание второго MAC-адреса через запятую означает, что данное правило нужно применять к запросам с любого из них, сбрасывая время аренды для другого MAC-адреса. Как и с двумя статическими адресами, вы должны позаботиться о том, чтобы оба интерфейса не попытались подключиться в одно и то же время.

Если отключить беспроводной интерфейс нельзя, воспользуйтесь последним вариантом: несколькими MAC-адресами в конфигурации *Dnsmasq*. Сеть с DHCP проще в обслуживании, даже если сервер выделяет фиксированные адреса, поскольку все настройки сети задаются в одном месте.

В многопользовательских системах у меня всегда прекрасно работала схема /home/user-distro. В порядке альтернативы, можно добавить название дистрибутива к имени пользователя. Домашний каталог можно задать во время установки или изменить позже. Убедитесь, что во всех дистрибутивах у вашего пользователя один и тот же идентификатор, в большинстве из них они начинаются с 1000 у первого пользователя – так вы получите доступ к файлам в домашнем каталоге другого дистрибутива. Основная причина, по которой нельзя сделать домашний каталог общим для всех дистрибутивов – потенциальная возможность путаницы с файлами настройки, однако каталоги с общими данными, такими как документы, фотографии и электронная почта, можно создать в одном дистрибутиве и привязать к домашним каталогам других с помощью символических ссылок.

Что касается меню загрузки, выберите один основной дистрибутив и предоставьте ему отвечать за загрузку. Каждый раз после установки

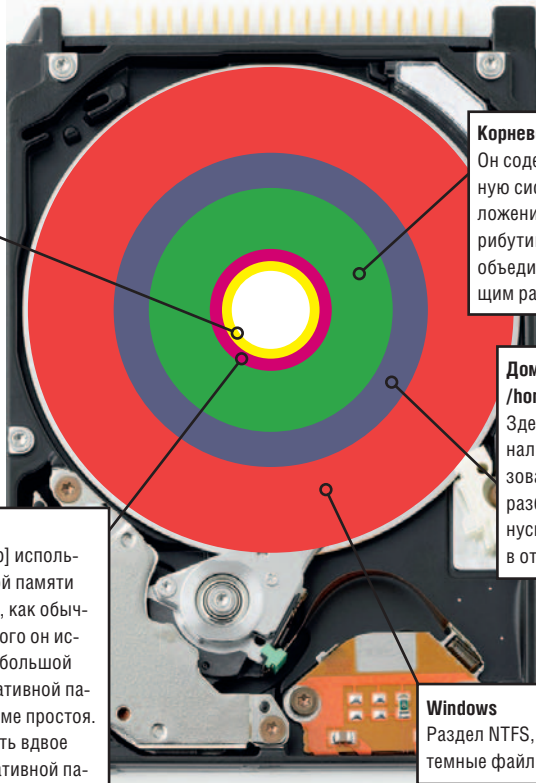
Помогите нам помочь вам

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
```



MBR — Главная загрузочная запись
Это не раздел, а небольшой фрагмент диска, содержащий первую часть загрузочного кода и таблицу разделов, в которой указаны операционные системы, имеющиеся на диске.

Корневой раздел Linux (/)
Он содержит операционную систему и файлы приложений, т.е. сам дистрибутив Linux. Нередко объединяется со следующим разделом.

Домашний раздел /home
Здесь хранятся персональные файлы пользователей. Далее мы разберем плюсы и минусы выделения /home в отдельный раздел.

Раздел подкачки
Раздел подкачки [swap] используется для виртуальной памяти и не содержит файлов, как обычный раздел. Вместо этого он используется ядром как большой медленный блок оперативной памяти для задач в режиме простоя. Его размер должен быть вдвое больше размера оперативной памяти, но не более 4 Гб, за исключением крайних случаев.

Windows
Раздел NTFS, содержащий системные файлы ОС Windows.

нового перезагрузитесь в главный дистрибутив и выполните команду:

```
sudo grub-mkconfig -o /boot/grub/grub.cfg
```

для обновления меню загрузчика.

Запустить виртуальную машину с настоящим разделом возможно, но делить ее с настоящей

установкой сложно, потому что в двух системах разные «устройства» – обычно это под силу только экспертам.

Если вас беспокоит, что виртуальные диски отнимают место на диске, создайте для них отдельный раздел. **LXF**



Часто задаваемые вопросы

DNCP

› Что такое DNCP?

Протокол динамической настройки узла [Dynamic Host Configuration Protocol].

› Ну да, так расшифровывается аббревиатура, но ЧТО это?

При подключении компьютера к сети ему нужно сообщить определенную информацию, такую как его IP-адрес, адреса серверов DNS, а также куда перенаправлять запросы компьютерам, которые ему не известны.

Вместо того, чтобы делать все это вручную (так называемая статическая адресация), можно воспользоваться DNCP и получить эту информацию с сервера. Это упрощает работу с большими сетями и перемещение компьютера из одной сети в другую, что часто происходит с ноутбуками.

› DNS? А это еще что?

Сервис доменных имен [Domain Name Service], который преобразует доменное имя, например, www.linuxformat.co.uk, в сетевой адрес, такой как 212.113.202.71.

› И как он работает?

Клиент (ваш компьютер) отправляет в сеть широковещательный «запрос обнаружения». Он принимается всеми компьютерами, но отвечает на него только серверы DNCP. Каждый сервер отправляет обратно «предложение». Если предложений несколько, клиент выбирает одно из них и запрашивает необходимые данные: IP-адрес компьютера, один или несколько серверов DNS и адрес шлюза.

› Шлюз? Это которые делают компьютеры?

[Имеется в виду американская компания Gateway Inc., – прим. пер.] Да, делают, но это другой шлюз. Шлюз по умолчанию – это адрес, на кото-

рый отправляется весь трафик для компьютеров вне вашей сети, например, для интернет-серверов. Затем шлюз перенаправляет вам трафик и ответы на запросы.

› Где взять сервер DNCP?

Обычно отдельный пакет для него не нужен. В большинстве роутеров, в том числе широкополосных и/или беспроводных модемах/маршрутизаторах есть сервер DNCP, который выделяет адреса для локальной сети.

Провайдеры кабельных модемов также используют DNCP для предоставления вам адреса при включении компьютера и модема.

Если вы пожелаете запустить собственный сервер DNCP для своей сети, то в дистрибутиве почти наверняка есть и сервер, и клиент (последний часто установлен по умолчанию). Поищите по строке «DNCP» в менеджере пакетов. Или загрузите dnsmasq с www.thekelleys.org.uk/dnsmasq.



› Настройка DNCP в веб-интерфейсе маршрутизатора.

› Я что-то слышал о сроке аренды в DNCP. Что это?

При отключении соединения серверу DNCP направляется запрос «освобождения», и сервер знает, что выданный вам адрес уже не используется и можно воспользоваться им повторно. При некорректном отключении сервер не сможет повторно воспользоваться вашим адресом. Время аренды – срок действия адреса. Клиент DNCP знает, что должен отправить запрос на обновление адреса в течение этого периода.



LXF Hot Picks



Ник Вейч

Ника однажды укусил радиоактивный менеджер скачиваний, и с тех пор у него болезненная страсть к экспериментам с установкой программ. И дурное пищеварение.

GLMixer » Knotter » Parted Magic » Emergent » Bacula » Powertop » VirtualBox » Ultimate Stunts » O A.D » Profanity » Din

Программа для микширования видео

GLMixer

Сайт <http://code.google.com/p/glmixer>

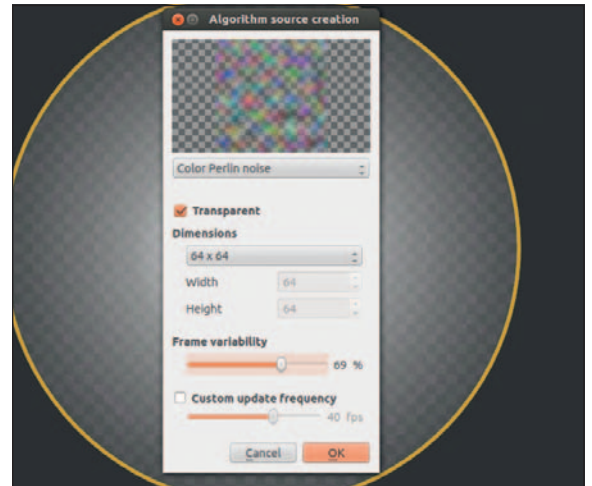
Возможно, вы думаете, что все программы для видео стоят на одной линии, являясь нелинейными редакторами (линии... нелинейные... ладно, забудем). Но нередко вам незачем заранее с высокой точностью редактировать видеоматериал, а просто нужно средство микширования видео в реальном времени.

Сам я отказываюсь использовать термин «виджей [VJ]» иначе, чем в саркастическом смысле, но эта программа считает видео скорее исполнительским искусством, чем способом надоедать друзьям. А может быть, и тем, и другим.

Главный экран должен навести вас на мысль о том, что здесь все по-другому.

Видеопотоки загружаются из массива источников с помощью кнопок вверху, а затем помещаются в круглый дисплей. То, куда вы их поместите, определит их коэффициент непрозрачности в окончательной версии на выходе. Конечно, у вас может быть более одного видео, и их можно соединять с помощью эффектов. А еще их можно перетаскивать поодиночке или группами для добавления эффектов.

«Это — удовольствие, особенно если вы принадлегли на Vimto.»



» Алгоритмы predeterminedены, если только вы не решите намикшировать собственный. Попробуйте perlin noise — это нечто!

Среди источников — стандартные видеофайлы, видеоустройства, алгоритмы (т.е. вы можете запрограммировать эффект) и, самое интригующее, источники памяти с распределенным доступом, с помощью которых программа может заглянуть в видеопамять другого работающего приложения. Но это чистая теория: как мы ни пытались, нам не удалось найти ни одной совместимой программы в списке, за которой можно было бы пошпионить.

Несомненно, при наличии навыков и практики вы, вероятно, сумеете обрушить на зазевавшихся зрителей весьма впечатляющее видео, но стоит также помнить и то, что это просто большое удовольствие, особенно если вы принадлегли на Vimto.

Перетаскивать клипы по основной сфере можно не только ради собственного удовольствия, но и потому, что это делается очень мягко и без всяких усилий. Ряд разнообразных режимов курсора позволяет изменять движение, используя, например, латентность (когда вы перетаскиваете треки и смотрите, как за ними следуют клипы) или эффекты нагрузки (когда создается видимость некоей инерции).

Работа с ним доставляет такое удовольствие, что вы даже можете забыть сохранить продукт для потомков.

Исследуем интерфейс GLMixer

Выберите источник

Выберите источник из предварительно записанных клипов, съемок, алгоритмов или совместно используемой памяти.

Вкладка Layer

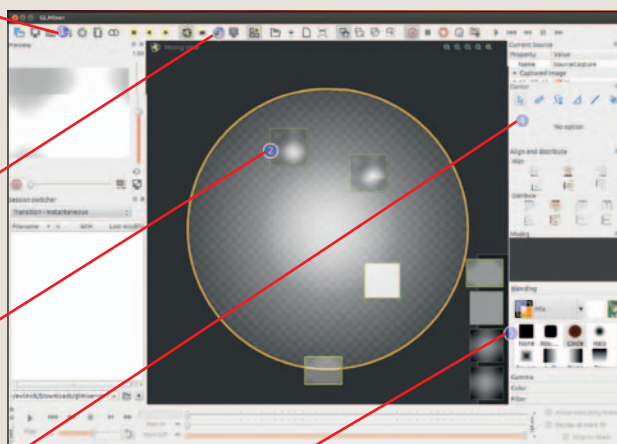
Вкладка Layer позволяет делать более точную настройку клипов и конечного результата.

Непрозрачность

Перетаскивайте клипы по сфере, чтобы настроить непрозрачность.

Управление курсором

Выберите управление курсором для более мягкого и плавного движения.



Подробная информация об источнике

Здесь отображаются детальные сведения о выбранном источнике.

Дизайнер узлов

Knotter

Версия 0.6 Сайт <http://bit.ly/SRnmnQ>

Вы же слышали о кельтах? Если они не убивали по 50 человек направо и налево, подумывая, что это все же маловато, ничто не доставляло им большего удовольствия тихими вечерами у огонька, чем возможность расслабиться и отвязаться, завязывая всякую всячину. Иными словами, любили они вязать разнообразные узлы.

Другие люди делали узлы, просто чтобы прикрепить что-нибудь, но кельтам обязательно нужно было возвести подобную взаимосвязь до уровня искусства. А может, им было больше не к чему приложить руки.

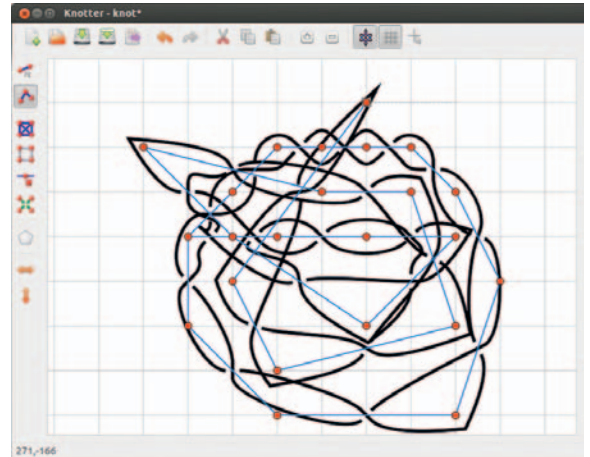
Так вот, *Knotter* – это инструмент, который поможет вам пройти по запутанным тропам искусства плетения узлов. Потратив буквально несколько минут и приложив некоторые усилия, вы сможете изобразить чудесный узел, просто шедевр. Вероятно. Даже если до этого момента все ваши навыки работы с многомерными переплетениями ограничивались

попытками распутать эти треклятые кабели, образовавшие колтун в ящике.

Knotter использует стандартный метод протягивания «веревки» между узлами. Помимо некоторого подбора стиля отдельных сегментов узла, настраивать здесь особо нечего – просто присоединяйте узелки к основному узлу с помощью инструментов на левой панели, пока не получите желаемый результат.

Инструмент «многогранник» позволит вам нарисовать замкнутые формы, но и их тоже можно будет присоединить к остальной части вашего изображения простым соединением или, в некоторых случаях, слиянием узлов. Закончив работу, можете экспортировать результат в файл SVG.

«Завершив работу, можете экспортировать результат в файл SVG.»



► При всей той помощи, которую предлагает *Knotter*, крайне сложно создать нечто уродливое, но нам это удалось...

Knotter не использует каких-то сложных библиотек, и у него нет витиеватых зависимостей, поэтому, хотя на данный момент его нет в репозиториях каждого дистрибутива, известного человечеству, компиляция из предоставленного исходника – дело пары минут.

Разбиение диска на разделы

Parted Magic

Версия 2012-9-12 Сайт <http://partedmagic.com>

Когда дело доходит до разделения и перетасовки небольших участков магнитных полей, которые хранят все ваши данные (если, конечно, вы не перешли на SSD), немного найдется инструментов, которые используются столь же часто, как *Parted*.

Редактирование разделов, особенно в системе, уже содержащей множество живых данных – не та задача, к которой можно отнестись легкомысленно, поэтому будет обнадеживающим узнать, что почти с момента появления Linux существует программа, способная целиком и полностью взять на себя эту работу, а вам остается просто нажимать на нужные кнопки.

Но иногда случаются неприятности, и вы остаетесь с системой, которая не грузится, и все ваши бесценные базы данных с анекдотами и архивы электронной почты, простирающиеся до позднего мезозоя, остаются запертыми в ней.

Parted Magic – это программа восстановления диска, созданная на инструменте почтенного возраста, но тем не менее превосходном для разбивки диска на разделы. При наличии полного понимания обычной структуры диска и поддержки типов разделов, от обычных форматов ext до более экзотических разновидностей JFS и Btrfs, вряд ли вы сможете удивить *Parted Magic* чем-то ему неизвестным.

Но *Parted Magic* используется не только в крайних случаях. Это отличный способ обезопасить себя от всякого рода цифровых катаклизмов, используя предоставляемые инструменты для создания резервных копий основных разделов (весьма хитро, когда вы работаете с ними в режиме

«Лучшее в этой программе – то, что ее не надо компилировать.»



► Он служит не только в случае неприятностей. ОК, по большей части он нужен именно для этого. Но какой он симпатичный!

live) и клонирования разделов или даже целых дисков. Почувствовали себя уверенно и в безопасности? Не стоит, пока вы не сделали всех резервных копий! Пожалуй, лучшее в этой программе – то, что ее даже не надо компилировать. Предоставленный образ будет работать на архитектуре i586 и выше (для RISC-машин придется компилировать собственный образ), и вам просто остается прожечь его на диск.

Можете запускать программу из рабочей системы Linux, но многие операции намного проще, если загрузиться непосредственно с диска.

Искусственный интеллект

Emergent

Версия 6.0.1 Сайт <http://grey.colorado.edu/emergent/>

Иногда случается откусить больше, чем можно проглотить. Но как узнать, сколько вы сможете проглотить, если не попробовать? Попытка рассказать о пакете моделирования нейронной деятельности на половине журнальной страницы, даже и напечатанной на бумаге такого отменного качества, как у *Linux Format*, явно выглядит чересчур амбициозной.

Поэтому уж поверьте нам на слово, что если вас интересуют искусственный интеллект и обучение машин, экспертные системы и моделирование поведенческих моделей, то вам обязательно надо познакомиться с этой программой.

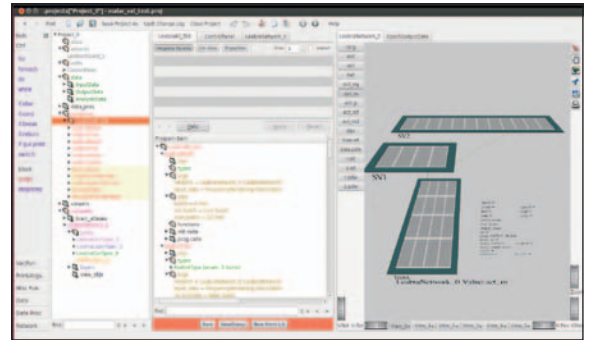
Emergent – прямой потомок работы, проводимой в Стенфордской лаборатории PDP, так что на ее создание ушли буквально десятилетия исследований.

Хотя нейронные сети очень полезны и как инструмент обучения (и, взглянем правде в глаза, как весьма занимательная игрушка), и как серьезный инструмент по-

знания происходящего вокруг нас (возьмите финансовые рынки, системы приливов, глобальное потепление, да, наконец, кто выигрывает в X-Factor в этом году...), перекрыть все части спектра одна система сумеет лишь в очень редких случаях.

С помощью относительно дружелюбного и снисходительного GUI довольно легко следовать руководствам и узнать, как работает *Emergent*. Если вам нужно нечто более сложное, чем простая система, основанная на правилах, стоит заняться изучением языка CSS (своего рода интерпретированная версия C++), и всяких прочих хитростей (таких, как полный физический симулятор).

На самом деле нет никакой возможности рассказать здесь обо всех из мно-



➤ **Всего лишь тремя с половиной строчками кода мы завершили симулятор Майка...**

жества функций *Emergent*, но примите наш добрый совет: читайте документацию! Вас, возможно, утомят пространные комментарии по каждому пункту меню, которыми набито руководство для начинающих [Getting started], но перейдите в раздел Build your first network [Создание собственной сети], и вам станет намного интереснее, поверьте.

Emergent – серьезный инструмент, и в него включено множество отличных примеров; но помимо этого, работа с ним – истинное удовольствие. Да и компиляция из исходника не представляет особой сложности.

«Легко следовать руководствам и узнать, как работает Emergent.»

Резервное копирование

Bacula

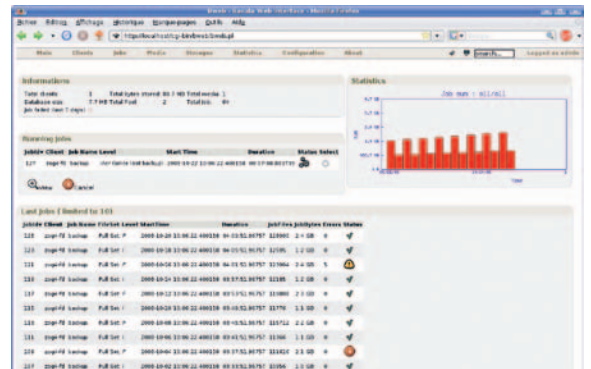
Версия 5.2.11 Сайт www.bacula.org

Бла-бла-бла... резервные копии... бла-бла-бла. Да, слушать про обязательность резервного копирования всем уже до смерти надоело. Да, все мы знаем, что это надо делать постоянно, но оно ж отнимает время. А все наши важные документы и так в безопасности в облаке, правда? Примерно такие мысли бродят в головах тех везунчиков, которые не знают, что такое полный отказ системы и полная потеря данных. Ну, может, и не полная, но все же утрата значительной части полезных данных.

Есть масса инструментов для резервного копирования, от простых до абсурдно сложных, и вам нет никаких оправданий, если вы до сих пор не смогли найти подходящего. Для многих *Bacula* покажется запредельным. Впервые он появился в те времена, когда резервные копии делались на кассету, а маленькие роботы таскали туда и сюда магнитофонные ленты. В наше время вы, возможно, решите сделать

копию на внешний носитель или на удаленный сервер в сети, или даже на один из этих пошлых сервисов облачного хранения. Все это хорошо и мило, но если ваша схема становится чуть сложнее, чем просто «скопировать этот файл туда-то», то *Bacula*, быть может, решение для вас.

Хотя этот пакет программ и разработан для предприятия (здесь есть компонент для управления; для хранения; для контроля за всем; для слежения за тем, чтобы все работало), *Bacula* будет в равной мере хорош при управлении вашей собственной небольшой флотилией технологий. Ручное создание резервных копий на паре ноутбуков, нескольких настольных ПК и, возможно, на файловом сервере может



➤ **О, если ты в своих скриптах уверен, когда теряют все скрипты вокруг... [аллюзия на стихотворение Р. Киплинга «Если», которое начинается строками: О если ты покоен, не растерян/ Когда теряют головы вокруг... — прим. пер.]**

доставить немало проблем. Изучение настройки и управления процессом резервного копирования *Bacula* (и крайне важным процессом восстановления – ваши данные не могут считаться в безопасности, пока вы им этого не обеспечите и не будете знать, как их можно восстановить) тоже задача не из легких, но, справившись с ней, вы, по крайней мере, сможете ощутить удовлетворение от мысли, что ничто, кроме термоядерного взрыва или прямого попадания метеорита, не лишит вас ваших драгоценных данных.

«Слушать про обязательность резервного копирования надоело.»

Утилита мониторинга

Powertop

Версия 2.1 Сайт <https://01.org/powertop>

Ну, ну, спокойнее. Я не собираюсь снова пускаться в разглаживание по поводу всяких топовых топ'ов, хотя складывается впечатление, что в последнее время на свет появилось множество их первоклассных версий. Как вы, возможно, помните по выпускам темного прошлого, *Powertop* – это тот, которого интересуется интенсивность занятости вашего CPU.

Изначально проект появился в Intel, и стал истинным благом как для пользователей, так и для разработчиков. Эффективно подсчитывая, сколько раз приложения выводят CPU из его естественного состояния ожидания, когда же что-нибудь произойдет, вы можете получить надежный прогноз того, насколько хватит батареи в вашем ноутбуке при работе определенного процесса.

Для пользователей это означает, что они наконец-то поймут, почему обещанных девяти часов работы батареи с трудом хватает на средней продолжительности поездки

на электричке, или она вдруг отключается, не дав вам досмотреть третью серию «Пип-шоу» [*Peep Show* – британский телесериал, – прим. пер.]. Для разработчиков это означает, что надо всерьез отладить производительность своего кода, поставившись, чтобы не его выкидывали в первую очередь, когда *Powertop* сочтет его чересчур прожорливым.

На самом базовом уровне *Powertop* может сообщить вам о состоянии вашего CPU. Не в том смысле, что он замахрился по краям, а в смысле уровня мощности, на котором он находится (или в режиме C).

А ниже будет располагаться список приложений, которые и заставляют его прервать сон. Намного более подробная

Usage	Events/s	Category	Description
188.2%			
9.8 ms/s	63.5	Timer	Audio codec hC002: Realtek
26.6 ms/s	53.6	Process	tick sched timer
22.3 ms/s	55.1	Process	/usr/bin/perl /usr/bin/shutter
6.9 ms/s	45.1	Timer	/usr/bin/x11-common /usr/bin/lightdm/root:/0 -mo
13.8 ms/s	35.8	Process	lxtime_wakeup
16.6 ms/s	31.9	Process	/usr/lib/firefox/firefox
7.5 ms/s	31.4	Process	compiz
8.9 ms/s	17.6	Process	/usr/sbin/libvirtltd -d
2.3 ms/s	19.1	Process	/opt/google/chrome/chrome
4.2 ms/s	16.4	Process	/usr/bin/python /usr/share/virt-manager/virt-man
3.7 ms/s	16.4	Process	/usr/lib/firefox/plugin-container /usr/lib/Flash
1.8 ms/s	13.5	Process	/usr/lib/ban/ban/daemon
1.3 ms/s	10.2	Process	/opt/google/chrome/chrome --type-renderer --lang
2.5 ms/s	9.5	Process	python
12.3 ms/s	4.9	Process	perl
6.7 ms/s	8.7	Process	/usr/bin/gtk-window-decorator
2.1 ms/s	5.5	Process	/bin/dbus-daemon --fork --print-pid 5 --print-a
13.6 ms/s	5.9	klmork	console_callback
355.8 ms/s	4.8	Process	/usr/lib/gnome-settings-daemon/gnome-settings-da
134.8 ms/s	4.5	Timer	lxtime_timer
1.2 ms/s	3.9	Interrupt	(4) tasklet(softirq)

» Не вините X.org! Хоть он и лидирует в этом списке постоянно, обычно в сжирании ресурсов виноват кто-то другой.

информация имеется на сайте проекта. *Powertop* – удобный инструмент, и он совершенно необходим, если вы твердо решили выжать из вашего ноутбука каждый эрг мощности.

В данном релизе улучшена поддержка обретающих все большую популярность процессоров ARM, и теперь *Powertop* способен предоставить вам более подробную информацию о производительности видеочипа; но на настоящий момент – только при условии, что таковым является встроенная модель Intel.

«Powertop может сообщить вам о состоянии вашего CPU.»

Виртуальная машина

VirtualBox

Версия 4.1.22 Сайт <https://www.virtualbox.org>

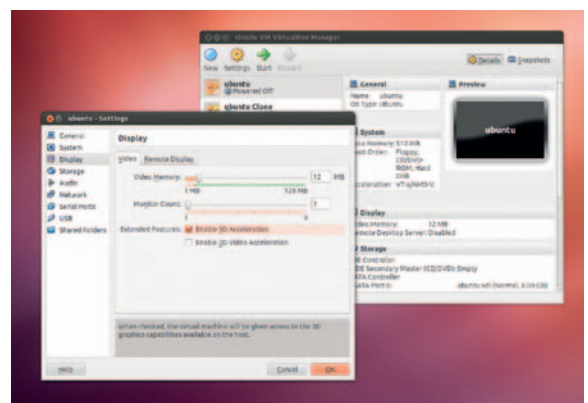
VirtualBox когда-то был серьезным соперником на арене менеджеров виртуальных машин (VM). Когда его приобрела Sun, он стал еще интереснее, поскольку обзавелся ранее отсутствовавшими функциями, в том числе очень симпатичным и простым в использовании GUI. Когда Sun была приобретена Oracle... они ведь были не самыми активными защитниками интересов открытого ПО, верно? Но *VirtualBox*, похоже, эти перемены не затронули, и он по-прежнему процветал.

Главное в этом релизе – лучшее управление несколькими экземплярами VM. Это вовсе не так дико, как может показаться на первый взгляд: несколько VM часто используются для тестирования, так что воспроизведение нескольких разных ОС или одной и той же ОС с разным набором программ – главная задача хорошего менеджера VM. Сгруппировав виртуальные машины, намного проще

активировать их партиями или внести важные изменения во всю систему.

Есть две разновидности *VirtualBox*. В Open Source edition, которую мы включили, отсутствуют некоторые графические функции; их нет в свободной версии в связи с лицензионными требованиями (бу-у – страшно?). Но они практически одинаковы. Особенно если вы просто хотите опробовать новый дистрибутив, не замусоривая свою систему, или создать тестовый кластер из полудюжины серверов (у каждого свое хобби).

Используя мастера настройки, трудно допустить ошибку: просто отвечайте на вопросы, и ваши устройства будут активированы. Она требует некоторых



» *VirtualBox*: процветание вопреки Oracle?

специфичных модулей ядра, которые обычно компилируются просто на лету при установке, но все же намного лучше установить ее с помощью вашего обычного менеджера пакетов, если это возможно, или, по крайней мере, убедиться, что активирована Dynamic Kernel Module Support (dkms).

Есть и другие способы установки и управления виртуальных машин в Linux; полностью открытые системы *Qemu/KVM* тоже весьма популярны, но удобство простого GUI означает, что *VirtualBox* еще поработает.

«Главное – лучшее управление несколькими экземплярами VM.»

HotGames Развлекательные приложения

Игра с автомобильными трюками

Ultimate Stunts

Версия 0.7.7.1 Сайт www.ultimatestunts.nl

Я думаю, что слово «игра» в названии несколько лишнее. Конечно, самым потрясающим трюком будет нечто вроде свободного падения с высоты 30 000 футов таким образом, чтобы единственным, кто мог бы помешать вашему падению и выживанию, был бумажный единорог в стиле оригами. На самом деле это стоило назвать Потрясающие Трюки На Автомобиле, Пытающиеся Игрючи Опровергнуть Основные Законы Физики.

Как бы то ни было, эта игра скорее гоночная, чем какая-либо другая. Трюки же, по большей части, случайность. Вы будете мчаться по трассе, пытаться получить, скажем, хорошее время круга, а затем обнаружите, что дорожное покрытие вдруг стало таким, будто его настелили рабочие, скакавшие на резиновом шаре. Или вы будете ехать по отличной прямой трассе и вдруг заметите

странную высоченную штуковину прямо перед собой – и вот, не успев даже толком этого осознать, вы уже мчитесь по самому большому кругу HotWheels из всех, когда-либо существовавших.

Уровни, конечно, можно настроить под себя, если вам не жаль потратить время и силы на их редактирование, но вы, возможно, обнаружите, что насколько бы безумную трассу вы ни придумали, она уже существует где-то среди множества примеров (именно поэтому архив игры столь много весит – все эти образы уже добавлены туда).

Здесь есть несколько действительно интересных достопримечательностей,

«Уровни можно настроить под себя, их редактированием.»



► Следите за дорогой. Куда бы она ни вела...

но они могут быть опасны. Пару раз интересно выглядящая декорация привела к незапланированному отклонению от оптимального маршрута и, возможно, влекла за собой некоторые поломки.

Ultimate Stunts основана на старой DOS-игре, и будем честны: при сравнении их играбельности оригинал, возможно, окажется лучше. Но всё равно игра доставляет немало удовольствия, предоставляя возможность погонять в невероятных условиях по причудливым трассам в поисках по-настоящему интересного укрытия.

Стратегия

O A.D.

Версия Alpha 11 Сайт www.wildfiregames.com/Oad

Мы уже обозревали эту игру, давненько, когда всё это сопровождалось чудесной графикой и сказочным основным постулатом «собирай, съедай, убивай». Что ж, сейчас всё изменилось! Графика стала намного чудеснее!

На самом деле, изменений накопилось множество, и на эту игру стоит взглянуть заново. Она по-прежнему в основном является игрой с управлением ресурсами и стратегией – представьте себе нечто вроде *Age of Empires*, но с улучшенной графикой. И анимацией. И музыкой. Есть несколько потенциальных «племен», за которые можно сыграть, и множество прекрасных местностей, где можно провести время. Они дышат атмосферой, и вы можете приблизить их и наблюдать за тщательно выполненной анимацией, к примеру, курей, которые кудахчут

на ходу, отыскивая съестное, пока кто-то не отрубит им головы и не сделает съестное, наоборот, из них.

Как и во всех подобных играх, идея заключается в том, чтобы накопить ресурсов, построить специальные здания и экстра-юниты – или собрать еще больше ресурсов, чтобы построить ещё больше зданий, или пойти на поиски противника, чтобы задать ему жару. Хотя Искусственный Интеллект улучшен по сравнению с предыдущей версией, для сражений не требуется особой стратегии – это просто состязание, кто соберёт больше юнитов; но от этого игра

«Здесь есть несколько племен и множество прекрасных мест.»



► В древности защита животных сводилась к превращению кур в пищевые ресурсы. Изменилось немного.

не становится менее увлекательной. Даже сейчас, когда она находится в альфа-стадии. То есть одиночный вариант весьма играбелен. Также на сайте имеется множество информации о том, как играть и как внести свой вклад в будущее развитие игры. Всего лишь ради ознакомления с игрой скачивание добрых 300 МБ данных может вас не прельстить. Но если вы любите стратегии и располагаете достаточной полосой пропускания, то почему бы и не попробовать?

Инструмент для чата

Profanity

Версия 0.1.8 Сайт <https://github.com/boothj5/profanity>

Если вы регулярно читаете наш журнал, то, вероятно, знаете один из золотых принципов Башен LXF: если чего-то нельзя сделать в командной строке, значит, это незачем и делать. Ничто не вносит больше радости в наш быт, чем новый способ сделать в оболочке *Bash* нечто, ранее непременно требовавшее возни с запуском X Windows – по этому, особенно при его замечательном названии, *Profanity* был просто обречен на появление здесь.

Profanity – это клиент Jabber, или, точнее, клиент XMPP. А значит, его можно использовать для соединения с вашими учетными записями в Google Chat, или с любым количеством других сервисов XMPP, и всю отвлекать народ от трудов праведных. В работе он похож на сессию IRC – вы печатаете, они печатают, все это появляется в окне Shell. Но минималистская манера отнюдь не означает отказа от всех удобств. Например, *Profanity*

по-прежнему привязан к системе уведомлений рабочего стола, и если кто-то решит начать с вами разговор, вам немедленно сообщат. Также вы получите сообщение «печатает сообщение», и будете знать, что на том конце кто-то есть. Командам предшествует префикс */*, как в IRC, хоть их и не так много. Но можно вывести список всех своих контактов, заходить на разные серверы и начинать разговор.

Конечно, революции в общении *Profanity* не совершит, но зато избавит вас от возни с окнами.

```
prof@profan:~$ ./profan
23:34 - prof@profan: but the plans were on display
23:34 - me: on display?
23:35 - me: I eventually had to go down to the cellar to find them
23:35 - prof@profan: thats the disolay department
23:35 - me: with a torch
23:35 - prof@profan: why will the lights had probably gone
23:36 - me: so had the stairs
23:36 - prof@profan: but look, you found the notice didn't you?
23:36 - me: yes
23:36 - me: yes i did
23:36 - me: it was on display in the bottom of a locked filing cabinet
23:37 - me: stuck in a disolay lavatory
23:37 - me: with a sign on the door saying "beware of the leopard"
23:37 - prof@profan: is offline, "Disconnected: connection closed"
23:37 - prof@profan
```

► Болтайте с друзьями при помощи Shell* (*пожалуйста, предоставьте собственных друзей).

Шумелка

Din

Версия 4.0 Сайт <http://dinisnoise.org>

Мы уже не раз повествовали о *Din* в HotPicks, но виноваты ли мы, что он постоянно улучшается? Может быть, не настолько, чтобы дорасти до нового главного релиза, но для нас он достаточно хорош, чтобы снова побеспокоить вас рассказом о нем.

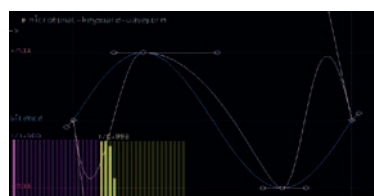
Если раньше вы его не встречали, сообщаем, что *Din* (*Din is Noise*) делает именно то, что декларировал. Вы можете создать грандиозный шум, приятность которого будет зависеть от вашей чуткости и врожденного музыкального таланта. Главное отличие *Din* от обычных программ для создания шума в том, что он не заботится о мелодичности.

Ноты, звуки, огибающие – все это изображается кривыми Безье. То есть вы не столько воспроизводите музыку, сколько ее рисуете. Формы, соотносящиеся на экране со звуками, кажутся за пределами восприятия некоторых людей, но если вы способны «услышать» визуализированную звуковую волну, то у вас не будет

с этим проблем. Либо можно просто продолжать валять дурака, пока не добьетесь желаемого звучания.

Главная проблема, возникающая при запуске *Din*, заключается в том, что он использует Jack. Не сдавайтесь, оно того стоит. Если вам кажется, что ваши силы уже на исходе, загляните на сайт, там вы найдете live CD версию программы.

Много работы в этом новом релизе ушло на изменение т.н. «дронов». Под дроном здесь подразумевается не беспилотный самолет, управляемый войсками для шпионажа, а длительно звучащая нота, которая может изменяться и модулировать на фоне, но присутствует постоянно.



► Мы не будем заходить чересчур далеко и утверждать, что это музыка, но это забавно.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

► Liferea 1.9.5

Важные отладки в системе безопасности и исправленные ошибки для популярного агрегатора новостей рабочего стола.

<http://liferea.sourceforge.net>

► Lilypond 2.17.2

Система нотной записи музыки теперь поддерживает синодальную нотацию.

О радость!

<http://lilypond.org>

► Codebrowser 4.2

Интересная альтернатива *gedit/kate* для программистов.

<http://tbleiz.net/code-browser>

► LittleCMS 2.4

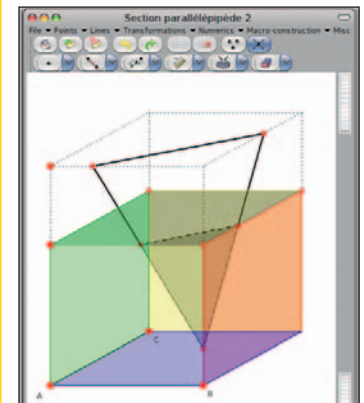
Улучшенное распознавание и трансформация точек черного для этого графического инструмента.

www.littlecms.com

► Dr Geo 12.10

Изучение начертательной геометрии должно приносить удовольствие! Или, как минимум, радовать глаз.

www.drgeo.eu



► А если мы рассечем куб этой плоскостью, то он развалится...

► Gfeedline 1.7

Теперь с множественной поддержкой учетных записей Twitter, пока его еще не запретили.

<http://code.google.com/p/gfeedline>

► Tor 0.2.2.39

Не полагайтесь на судьбу, когда речь идет о конфиденциальности – обновитесь уже сегодня.

<https://www.torproject.org>

► Samba 3.6.8

Важные обновления для ровной работы сети с совместным доступом.

www.samba.org

На диске

Пробуйте новую операционную систему уже сегодня!

Не пропустите
BSMBench —
сравнительный
тест НРС —
на DVD этого
месяца!



Некоторые месяцы заваливают нас в Башнях *Linux Format* огромным количеством релизов, зато в другие месяцы приходится выбирать всего из нескольких новых дистрибутивов.

Но в любом случае нам важно знать, что вы, наши читатели, хотите видеть на диске. Конечно, мы не можем обещать включить все. Некоторое ПО имеет технические ограничения, и нам всегда

трудно принять решение, на что именно следует отвести ограниченное пространство диска.

Тем не менее, чем больше вы рассказываете нам, что вы хотите видеть на диске, тем лучше мы сможем составлять его, чтобы он соответствовал вашим ожиданиям. Присылайте мне свои идеи на ben.everard@futurenet.com

Силен как прежде

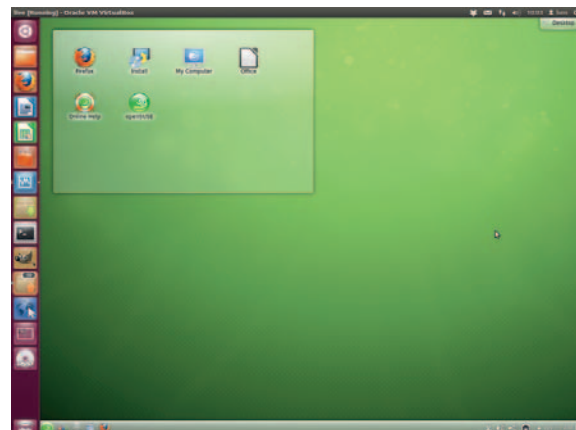
OpenSUSE

20 лет назад родился Software und System Entwicklung GmbH (S.u.S. E.), и стал оригинальным коммерческим дистрибутивом Linux. В те времена было далеко не очевидно, что маленькое ядро в его сердцевине превратится в программу, которая изменит мир. Но так и произошло. По мере изменения Linux менялся и S.u.S. E. В его названии исчезли точки, и он разделился на коммерческую и свободную ветви.

В этом году релиз OpenSUSE запоздал. Однако слухи о смерти OpenSUSE были сильно преувеличены, и 5 сентября он появился. Новейшая версия (12.2) со-

ответствует высоким стандартам, которых мы привыкли ожидать от, наверное, популярнейшей в мире KDE-версии Linux.

Здесь нет радикальных перемен — по крайней мере, ничего такого, что не было бы очевидно постоянным пользователям. Включены самые новые программы, и разработчики утверждают, что новая версия быстрее и более стабильна. Однако внутренние изменения более очевидны (горячее всего предвсказываемые — *Grub 2* и *BTRFS*). *BTRFS* (B-Tree File System) — файловая система нового поколения, и через пару лет она должна бы появиться во всех дистрибутивах. В ней есть ряд пре-



» OpenSUSE: хамелеон отказывается менять цвет!

имущества перед семейством ext, но они в большей степени будут заметны на серверах, чем на настольных системах.

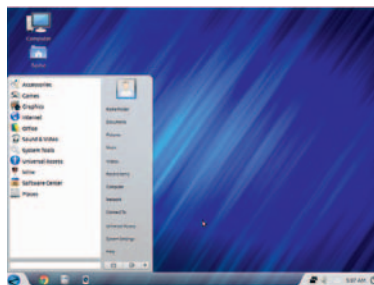
Linux для пользователей Windows

Zorin 6

Версия 6 отметила собой трехлетнюю годовщину этого дистрибутива. Его цель благородна: вербовать пользователей Linux в рядах поклонников Windows, а не пытаться увести их из других дистрибутивов. Однако делается это отнюдь не из голого альтруизма; компания-производитель нацелена на получение прибыли.

Мы включили на диск версию Free (она же — Core). Есть также платные версии — Multimedia, Gaming, Business и Ultimate (7–18 евро, в зависимости от версии и опций загрузки). И если опытные пользователи могут усомниться в том, что предлагаемые опции оправдывают эту цену, новички в Linux скорее всего оценят предлагаемую поддержку. Чтобы облегчить миграцию на Linux, Zorin включает при-

ложение *Look Changer*, которое позволяет пользователю выбрать, на какую систему, отличную от Linux, будет похож их рабочий стол. Это позволит новообращенным ощутить себя в знакомой системе, с какой бы ОС они ни совершили переход.



» Снимим ничего не подозревающих пользователей с Windows в Linux, на Zorin 6.

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, для получения содействия.

Новый дистрибутив с большими амбициями

SolusOS

Mint стремительно взлетел к успеху в мире дистрибутивов. И уж если один из его разработчиков создает новый дистрибутив, на это стоит обратить внимание. SolusOS – продукт Айки Догерти [Ikey Doherty], разработавшего Linux Mint Debian Edition, и неудивительно, что SolusOS также построен на базе Debian. Как и в Mint, большая часть процесса разработки была отведена отладке ин-

терфейса пользователя. Среда рабочего стола отличается простотой, быстротой и отзывчивостью, подкрепленными глубиной репозитория Debian. Дистрибутив предназначен скорее для домашних пользователей, чем для программистов и разработчиков. Единственное, что удивляет – это отсутствие игр среди установленных программ. Включено *Play on Linux*, чтобы помочь вам запустить любимые игрушки из Windows.



► Мы включили версию Legacy, которая должна работать на любом оборудовании x86 (почти).



Проскользни под радаром

Liberté

Конфиденциальность ныне – горячая тема. Правительство Великобритании усиленно продвигает свою т.н. Snoopers' Charter [Шпионская Хартия – законопроект, предлагающий отслеживать любую онлайн-деятельность, – прим. пер], и во многих странах ситуация немногим лучше: конфиденциальность более нельзя считать само собой разумеющейся. Мы уже писали о Tails, и Liberté – еще один дистрибутив из того же ряда. Оба используют

Tor, способ анонимного доступа в Интернет, чтобы действовать обеспечению вашей конфиденциальности. Но это не идеально; не забудьте ознакомиться с ограничениями протокола на www.torproject.org

Liberté идет в несколько необычной форме: как ZIP-файл. Для создания загрузаемого устройства USB распакуйте его и скопируйте файл, затем откройте терминал, перейдите в `<path to usb stick>/liberte` и запустите `sudo ./setup.sh auto`.

► Если, подобно британской королевской семье, вы озабочены сохранением своей конфиденциальности, используйте Liberté, чтобы обезопаситься в режиме онлайн.

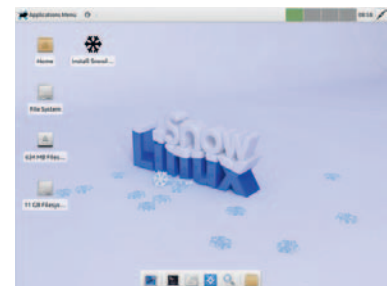
Пусть падает снег

Snow Linux XFCE

Сайт проекта гласит: «Linux должен быть не сложнее в использовании, чем Microsoft Windows». Snow создан на этом принципе. [Let it snow – популярная песня, – прим. пер.]

Он существует в нескольких разновидностях; мы включили версию на базе Xfce в виде ISO-образа – вам надо записать ее на диск или ознакомиться с ней в виртуальной машине. При первом запуске Xfce предоставит вам довольно типичный рабочий стол. Од-

нако среда рабочего стола – это не просто базовый интерфейс. Это все части, сочетание которых обеспечивает работу пользователя, и важнейшая из них – файловый менеджер; в Xfce это Thunar. Он менее популярен, чем Nautilus (Gnome), и не столь мощный, как Dolphin (KDE), но эффективен, прост и отлично вписывается в философию Xfce. Другие специфичные для Xfce программы – Ristretto (для просмотра изображений) и Squeeze (менеджер архивов). LXF



► Популярность Xfce недавно резко выросла. Познакомьтесь с ним через Snow Linux – узнаете, почему.

На диске

На нашем диске есть не только дистрибутивы. Мы также включили коды из журнала, все hotpicks, некоторые из недавних подкастов и подборку сетевых книг по Linux. Эта последняя категория стала появляться на нашем диске с прошлого выпуска. Здесь вы найдете руководства, которые помогут вам начать работу в Linux или улучшить ваши навыки работы в командной строке. Здесь есть и классика Linux: The Cathedral And The Bazaar [Собор и Базар] и Linux Kernel In A Nutshell [О ядре Linux в нескольких словах]. А еще – Руководство администратора Debian и словарь Linux. Хотите использовать

дистрибутивы на компьютере без DVD-привода? Мы подготовили удобный скрипт, чтобы перенести содержимое DVD на устройство USB. Вам понадобится устройство USB размером более 4,6 ГБ, поскольку это размер диска. В терминале введите `sudo /media/LXFDVD163/dvd2usb.sh` или, альтернативно, `sudo bash /media/LXFDVD163/dvd2usb.sh`

Если первое не работает, поскольку ваша система не дает оптическим дискам прав разрешений на исполнение, видимо, потребуется изменить путь, в зависимости от того, куда подмонтирован DVD.

Альтернатива – скачать ISO [английской версии, – прим. ред.] с www.linuxformat.com/archives, записать его на DVD или перенести на USB-брелок по dd.

Не все дистрибутивы загружаются с устройства USB. У каждого из них свой метод поиска расположения основной файловой системы squashfs для монтирования. Некоторые проверяют только оптические носители, и не смогут найти ее, если она хранится на флэшке. Однако ситуация, похоже, улучшается, и все больше дистрибутивов поддерживают загрузку с устройств USB, поскольку этот способ знакомства с ними становится все популярнее.

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 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:



LXF161
Сентябрь 2012

250 руб.

- » **Обновляйтесь!** Разумный выбор нового оборудования
- » **Строим сайт** Системы управления контентом
- » **OpenStack** Linux облачных вычислений
- » **Moodle** Жаждете поделиться знаниями? Открывайте курсы!

LXFDVD: ROSA Marathon 2012, Linux Mint 13 и еще 2 дистрибутива, Phoronix Test Suite, DigiKam, Packet Fense, видеоролик по работе с микрокомпьютером MK802 и прочее...

Печатная версия: shop.linuxformat.ru/lxf_161/
PDF-версия: shop.linuxformat.ru/elxf_161/



LXF162
Октябрь 2012

250 руб.

- » **Матч гигантов** Какой дистрибутив лучше?
- » **За нашу свободу** Лицензии на открытое ПО
- » **SecureBoot** Загрузка-то безопасна, но с установкой ясно не все
- » **Скажи пароль** А вот "password" в ответ говорить не надо!

LXFDVD: Arch, CentOS, Fedora, Mageia и еще 11 дистрибутивов, 9 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_162/
PDF-версия: shop.linuxformat.ru/elxf_162/



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/
PDF-версия: shop.linuxformat.ru/elxf_163/

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 или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на 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



Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- » **имя_программы-1.0.1.tgz** – тот же файл, что и выше этажом по списку; “tgz” – это сокращение от “tar.gz”;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: Preview файла в Unity, интеграция с web-службами и многое другое

» **Firefox 3.5.0** » **Gnome 3.6.1** и **Unity 6.8.0** » **LibreOffice 3.5.2** и **Firefox 16.0.1** » **Nautilus 3.4.2** и **Python 3 Desktop** и **Server 32- и 64-разрядные сборки**



ubuntu 12.10

А ТАКЖЕ: Vassila, GLMixer, PartedMagic и многое другое

И ЕЩЕ: **Liberte** » **Snow Linux** » **Solus OS** » **Zorin OS 6 Live**

Live CD с рабочим столом KDE

opensuse 12.2



Декабрь 2012
LXF DVD 164

LINUX
FORMAT

Декабрь 2012
LXF DVD 164

LINUX
FORMAT

Сторона 1

ДИСТРИБУТИВЫ

Liberte 2012.3 Дистрибутив на базе Gentoo для безопасного доступа в Интернет.
32-разрядная сборка (ISO-образ)
Snow Linux 3 Дистрибутив на базе Debian с рабочим столом Xfce. 32-разрядная сборка (ISO-образ)
Solus OS Облегченный дистрибутив на базе Debian. 32-разрядная сборка (загрузка с LXF DVD)
opensUSE 12.2 KDE Live. 32-разрядная сборка (загрузка с LXF DVD)
Zenit OS 6 Live Дистрибутив с интерфейсом, максимально приближенным к Windows. 32-разрядная сборка (загрузка с LXF DVD)

Сторона 2

ДИСТРИБУТИВЫ

Ubuntu 12.10 Установочные DVD (ISO-образы) для рабочих станций
Server 32- и 64-разрядные сборки для серверов
Desktop и Server amd64-нац 64-разрядные сборки с поддержкой компьютеров Apple
ПРОГРАММЫ
razorqt-0.4.1 Исходный код графической оболочки Razor-Qt

ДОКУМЕНТАЦИЯ: 9 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Bash Scripting Подробное руководство по программированию на Bash
Bourne Shell Scripting Начальное руководство по программированию на Bash
Cathedral Bazaar Классический текст Эрика Раймонда (Eric S Raymond) «Сбор и базар»

The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
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 – Фредом Краас-Хартманом (Fred Kroah-Hartman)
System Administrators' Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

NOTICKS

OAD 0.0.11 Игра, пошаговая стратегия в стиле Age of Empires
Bacula 5.212 Одна из наиболее популярных свободных программ для резервного копирования по сети

Din 4.0 Генератор звуковых волн произвольной формы
Emergent 6.04 Программа для моделирования работы нейронных сетей, разработанная в Университете Колорадо
GL Mixer Программа для микширования видеопотока из различных источников
Knitter 0.6 Программа для построения орнаментов в стиле «кельтских узлов»
PartedMagic 2012-09-12 Специализированный Live CD для управления разделами дисков
PowerTop 2.1 Средство диагностики проблем, касающихся энергопотребления
Profanity 0.18 Клиент Jabber/XMPP, написанный на curses
UltimateStunts 0.771 Игра, имитатор автомобильных гонок
VirtualBox 4.1.22 Популярнейшая кросс-платформенная среда виртуализации

Пожадуйста, перестаньте использовать одним диском. Диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Писать ваши пожелания и предложения по электронной почте: info@lxf.com
ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@lxf.com

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 резервного копирования с нашего сайта. Мы благодарим вас за то, что вы выбрали издание журнала.
 Этот диск изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР 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

Аппаратно-программный комплекс

DR.WEB OFFICE SHIELD

Комплексное решение задачи антивирусной и антиспам-защиты для малых и средних предприятий



88 679 руб.

DR.WEB OFFICE SHIELD TWISTER
Рассчитан на 250 ПК



65 240 руб.

DR.WEB OFFICE SHIELD NEO
Рассчитан на 50 ПК

Dr.WEB®

Linux center
www.linuxcenter.ru

WWW.LINUXCENTER.RU/SHOP/ANTIVIR/DR_WEB/

LINUX FORMAT

Главное в мире Linux

Как разместить рекламу в разделе Classifieds?

1/4 полоса (210 × 297 мм)	165 200 руб.
1/2 полосы горизонтально (197 × 144 мм)	88 500 руб.
1/2 полосы вертикально (102 × 278)	88 500 руб.
1/4 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/

allbuntu.ru



**сообщество
ПОЛЬЗОВАТЕЛЕЙ
UBUNTU**

Unix Education Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: + 7 (812) 611-1575







**RHCSA/RHCE
Certification**

Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/

Школа LXF

Спонсор рубрики
Mandriva.ru
разработчик
дистрибутива
EduMandriva
www.mandriva.ru

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

KompoZer Основные возможности и внешний вид

Михаил Выграненко занялся Созиданиями, с философским введением и лирическими отступлениями. Созидание первое из четырех...



Наш
эксперт

Михаил Выграненко – идеолог свободы в мыслях и порядка в делах на ниве просвещения, каким бы парадоксальным ни казалось это сочетание. В миру – учитель-методист.

Введение

– *KompoZer*? Осколок древнего сайтостроя? Какой смысл использовать эту программу в наши дни? Кому и зачем она нужна? – предвижу, что так, скептически усмехнувшись, выскажется поверхностно-продвинутый обозреватель Интернета, мечтающий о карьере фрилансера. Что ж, эти философские вопросы имеют место быть. Действительно, приложение историческое, последняя версия открыта года три назад. Правда и то, что нынче не в моде создание статичных информационных html-страниц, и программе, предназначенной для их конструирования, в современных портално-облачных реалиях жить не просто. Но всякому философскому вопросу должен найтись философский ответ, пусть даже и субъективный. Здесь и сейчас я тот, кто сделает это для программы *KompoZer*.

Я утверждаю, что ей самое место в школе, в качестве учебного средства, и поясню почему.

По действующим федеральным учебным программам 2004 г. (<http://www.websib.ru/noos/informatika/progr.htm>) в базовом курсе информатики средней ступени практического сайтостроения нет вообще, а в базовом курсе старшей школы на него отведено примерно 10 уроков. Понятно, что разгуляться в дебрях web-конструирования учителю нет возможности. Вряд ли положение кардинально улучшится со вводом учебных программ по новым стандартам. Ситуацию могут поддержать дополнительно выделенные руководством учебные часы из регионального и школьного компонентов или элективные курсы, но это уж где какой местный расклад интересов. Я исхожу из того, что есть гарантированно у всех.

Итак, что же нужно учителю, использующему школьную платформу Linux в учебном процессе, от программного приложения для таких занятий?

- » Архив или пакет с максимальной простотой установки
- » Свободная версия win-дистрибутива (желательно)
- » Сочетание визуального конструирования с работой в коде
- » Понятный и удобный инструментарий
- » Русскоязычный интерфейс (желательно)

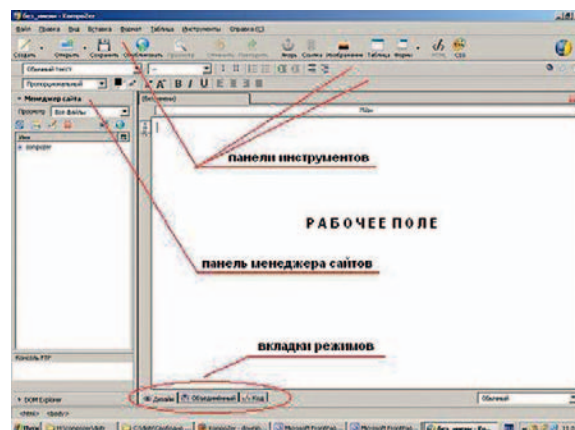
И хотя в школьных Linux-сборках имеются средства web-разработки, получается, что *KompoZer* – единственная программа, удовлетворяющая этим условиям, позволяющая вести занятия

практически с нуля. Ведь *KDevelop* и *Quanta Plus*, которыми оснащен, например, Альт Линукс 5.0 Школьный Мастер (ПСПО 5, <http://www.altlinux.ru/products/5th-platform/school-box/>) – это только редакторы кода, работа в них в принципе не отличается от создания html-страниц в «блокноте». Понятие web-дизайна при этом практически теряет смысл, а ведь видеть картинку сразу – очень важно для школьника и вообще новичка. А *KompoZer* – объектно-визуальный редактор сайтостроения (WYSIWYG-редактор [What You See Is What You Get, «что видишь, то и получишь», см. страницу Википедии]), свободный, открытый и кроссплатформенный. И очень дружелюбный по отношению к пользователю.

Так что в нем не нужно искать недостатки в сравнении с проприетарными лидерами такого рода от Microsoft (*SharePoint Designer* в составе *Office 2007* и *2010*, *Expression Web* в составе пакета *Expression Studio*,

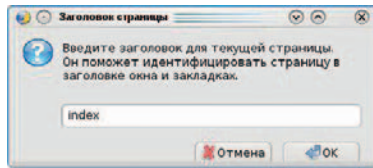
FrontPage в составе пакета *Office 2003*) и Adobe (*Dreamweaver*, ранее принадлежавший Macromedia). Он является замечательной стартовой площадкой для дальнейшего любительского и профессионального сайтостроения как с использованием названных почтенных программ, так и динамических систем управления контентом (CMS-системах, см. в Википедии), и online-конструкторов

«KompoZer – редактор, очень дружелюбный к пользователю.»



» Рис. 1. Интерфейс окна программы KompoZer.

► Рис. 2. Диалоговое окно, появляющееся при создании новой страницы.



(см., например, конструктор ВебСтолица.ру или соответствующий сервис Google), в которых так или иначе сохраняется преемственность базовых инструментов и навыков.

Сайт программы – <http://kompozer.net/>, где можно получить все необходимые дистрибутивы и пакеты. Я буду использовать в предлагаемом пособии по начальному сайтостроению последнюю, развивающуюся [development], русскоязычную версию 0.8b3 (2010-02-28) на платформе ПСПО 5.

Для установки lin-дистрибутива нужно:

- 1 Записать его архив в Домашнюю папку пользователя.
- 2 Вызвать для архива контекстное меню и выбрать пункт «Распаковать в эту папку».
- 3 Открыть появившуюся в Домашней папке директорию **Kompozer** и запустить файл **kompozer**.
- 4 При желании можно создать ярлык для программы на Рабочем Столе.

Считаю, что это тот случай, когда начать надо сразу с практики. Теория – по ходу дела, в зависимости от технологических перерывов, праздного любопытства и Википедии. Для выполнения Созиданий (в простонародье – практических заданий, упражнений) требуется исправное интернет-соединение и учебные файлы в папке **pict** (см. **LXF DVD**).

Вот и все для начала. Но нет, чего-то не хватает. Чего? Мне кажется, лирического отступления. Поэтические строки способствуют интересу и стимулируют мышление, настраивают на проникновенность и возвышенность. Например, гекзаметры. Они высококультурны, как все эллинское, надежны и проверены временем – как раз под стать программе *Kompozer*. Да, именно гекзаметры.

*Так приготовься вникать, о пылливый и вдумчивый юзер,
Чтобы в беседе с собой смог бы достойно предстать,
Не говоря про людей, и вершин мастерства сайтостроя
Светлых хотя бы достичь, если уж не покорить.*

Установим KompoZer

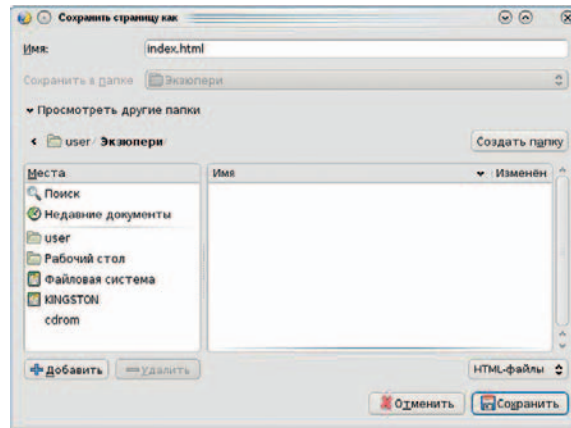
Перед тем, как начать созидать, проясним следующее.

KompoZer позволяет:

- » работать в трех режимах: режиме дизайна (объектно-визуального конструирования), режиме кода и объединенном режиме, а также включать предварительный просмотр;
- » открывать несколько страниц и вести обработку слоев на странице;
- » управлять файлами и папками сайта (менеджмент сайта);
- » публиковать сайт (обновления) в Интернет и контролировать этот процесс в консоли протокола FTP;
- » выполнять форматирование символов, абзацев и страниц, создавать таблицы и формы;
- » открывать текущую страницу во внешнем текстовом редакторе в виде html-кода;
- » вести CSS-редактирование (редактирование стиля);
- » выполнять вставку объектов (изображения, таблицы, ссылки, символы и др.).

Внешне (рис. 1) программа близка к *Microsoft SharePoint Designer* со всем привычными составляющими окна. Элементы легко находятся по подписям и всплывающим подсказкам.

В рабочем поле каждая страница выглядит в виде вкладки. Имеются измерительные линейки, полосы прокрутки, вкладки режимов и раскрывающийся список просмотра. В левой части ок-



► Рис. 3. Сохранение созданной страницы.

на программы располагается менеджер сайта и его инструменты. Вверху – как обычно: меню и панели инструментов.

Созидание первое: Простая начальная страница сайта

Замер компьютерный мир: каждый пиксель застыл на экране.

Клавиатура молчит – что она может без рук?

Даже затих как-то кулер во чреве системного блока...

Этот магический мир действий осмысленных ждет.

Мы создадим сайт, посвященный отважному летчику и замечательно французскому писателю Антуану де Сент-Экзюпери.

Предварительно следует:

- » Приготовить (взять с **LXF DVD**, проверить в папке **pict**) учебные файлы **pict.jpg**, **plume.png**.
- » Убедиться в исправности интернет-соединения.

Теперь можно браться за дело.

1 Запустим программу с помощью меню системы или ярлыка на Рабочем Столе.

2 Новая страница открывается автоматически, и ее следует сразу сохранить. При этом в диалоговом окне нужно набрать имя начальной страницы сайта **index** (это универсальное правило для сайтостроения) – см. рис. 2.

Следует иметь в виду, что файлы сайта должны иметь в именах только строчные латинские буквы и цифры без пробелов и знаков препинания.

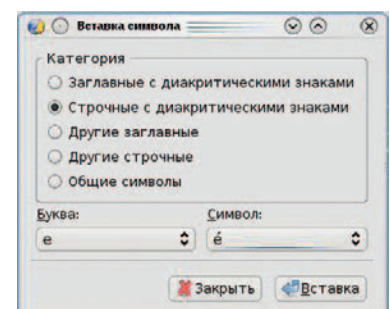
- 3 Нажатие на кнопку ОК приведет в диалоговое окно (рис. 3), запрашивающее имя папки для сохранения. Ее можно создать прямо сейчас (или заблаговременно) с помощью соответствующей кнопки диалогового окна. Наше имя папки – **Экзюпери**. Стандартный тип файла html для страницы сайта задается автоматически.
- 4 Наберем в первых строках страницы текст с именем нашего героя, по-русски и по-французски, и датами его жизни:

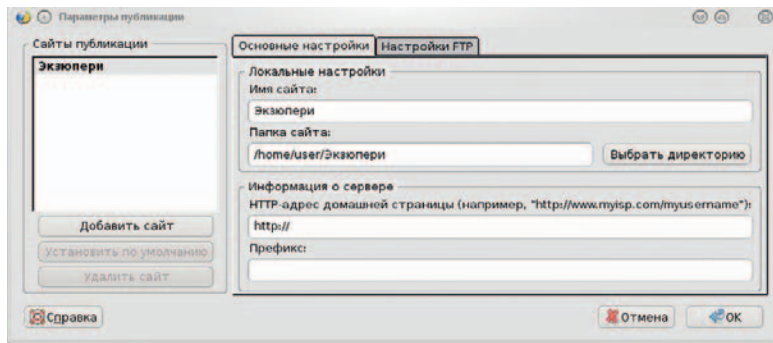
Антуан Мари Жан-Батист Роже де Сент-Экзюпери
Antoine Marie Jean-Baptiste Roger de Saint-Exupéry
(1900-1944)

Текст можно копировать из внешних источников (документ, Интернет), но обязательно сперва поместите (сохраните) его в простой текстовый файл и переносите на страницу уже оттуда. Промежуточное размещение освободит нас от форматирования источника, которое портит код создаваемой страницы и нередко приводит к нежелательным последствиям.

Разместим текст по центру и применим к нему инструменты форматирования

► Рис. 4. Вставка символа с диакритическим знаком.





► Рис. 5. Сохранение папки сайта.

по своему усмотрению (вид шрифта, размер, начертание и цвет символов). Обратили внимание на пунктирную красную рамку? Она обозначает блок единого форматирования и никак не ограничивает свободы внесения изменений.

Единственное, что требует пояснения – это символ **é**: на клавиатуре он отсутствует. Для его вставки в меню программы выбираем пункты Вставка/Буквы и символы... и в диалоговом окне задаем положение:

Не забудем закрыть после вставки символа диалоговое окно.

5 На очереди – вставка изображения. Все картинки (элементы оформления), размещаемые на страницах сайта, должны находиться в его структуре в отдельной папке (или папках). Поэтому обратимся к Менеджеру сайта, располагающемуся в левой части окна программы, и нажмем его кнопку Редактировать сайты.

6 Выберем папку **Экзюпери**, созданную в п. 3, и нажмем ОК.

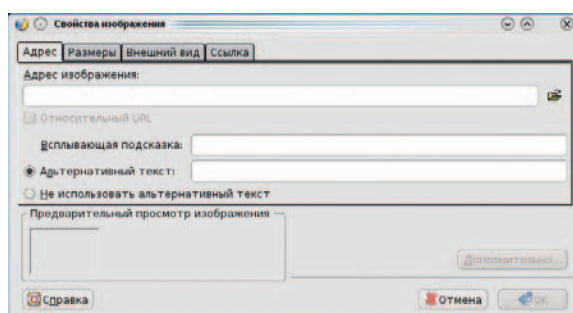
7 Убедимся в появлении имени папки в поле иерархической схемы сайта и выберем ее. Затем нажмем кнопку Новая папка Менеджера сайта и зададим имя **pict**. Важно запомнить: *все имена папок и файлов в структуре сайта определяются строчными буквами латинского алфавита.*

8 Раскроем структуру сайта, нажав на + слева от имени, и убедимся в появлении папки **pict**.

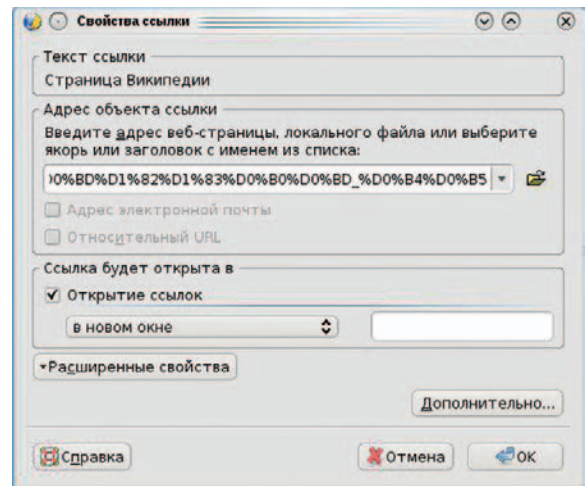
9 Для помещения в папку **pict** учебных файлов **pict.jpg** и **plume.png** свернем окно *KompoZer* и воспользуемся любыми доступными для этого средствами. Затем вернемся в окно программы и нажмем кнопку Обновить Менеджера сайта. Убедимся в появлении в схеме сайта файлов изображений.

10 Установим теперь текстовый курсор в конце текста на странице и нажмем клавишу Enter, определив место для вставки. С помощью приема перетаскивания курсором мыши разместим файл из схемы сайта **pict.jpg** (портрет нашего героя) в заданном месте. Убедимся в появлении изображения на странице. Это самый простой и удобный способ вставки картинки. Однако если воспользоваться пунктами меню Вставка/Изображение..., мы получаем возможность использовать важные настройки – добавление всплывающей подсказки, альтернативного текста, гиперссылки; изменение размеров, положения на странице для выбранного изображения.

11 Вызовем для изображения на странице контекстное меню и выберем пункт Свойства изображения... В диалоговом окне



► Рис. 6. Настройка параметров вставленного изображения.



► Рис. 7. Задание свойств ссылки.

выберем вкладку Внешний вид и установим текстовый курсор в поле ввода Сплошная рамка:. Введем значение 1 и нажмем ОК. Убедимся в появлении заданного элемента оформления для изображения.

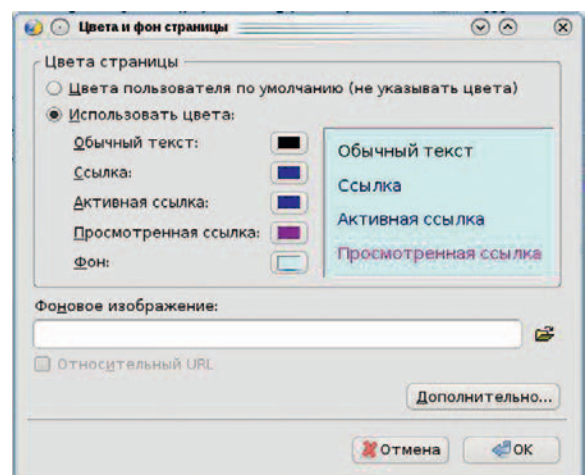
12 Мы хотим, чтобы посетители нашего сайта при желании могли обратиться к справочной информации и узнать о нашем герое поподробнее. Тут как нельзя более пригодится Википедия – вставим на нее гиперссылку. Для этого опять придется «отложить в сторону» (свернуть) окно программы и воспользоваться интернет-обозревателем (браузером). С помощью любого поисковика получим результат запроса на слово «Экзюпери» и откроем соответствующий ресурс Википедии.

13 Скопируем адрес ресурса в оперативную память (буфер обмена) и вернемся в окно *KompoZer*. На следующей строке после изображения, сохраняя размещение по центру, наберем текст **Страница Википедии**. Выберем его и вызовем во встроенном меню пункт Создать ссылку...

14 В окне диалога вставим адрес из памяти в соответствующем поле. Установим флажок Открытие ссылок и положение списка выбора в новом окне. Нажмем ОК и убедимся в появлении гиперссылки.

15 Вы можете самостоятельно определить и добавить еще 2–4 ссылки на ресурсы по теме сайта. Затем на новой строке вставим разделяющую черту с помощью пунктов меню Вставка/Разделитель.

16 Сохраним текущее состояние страницы.



► Рис. 8. Раскрасим нашу страницу.

17 Пропустим пару строк и перейдем на третью, расположив текстовый курсор по левому краю. Три раза нажмем кнопку Панели форматирования Увеличить отступ (сдвиг вправо).

18 Поместим на странице названия самых известных книг Экзюпери. Наберем следующее:

Ночной полет (1931)

Планета людей (1938)

Маленький принц (1943)

Строки с названиями можно пометить символом писательской деятельности – гусиным пером (**plume.png**).

19 Следующий шаг – фон страницы и установка цветов. Выполним пункты меню Формат/Цвета и фон страницы...

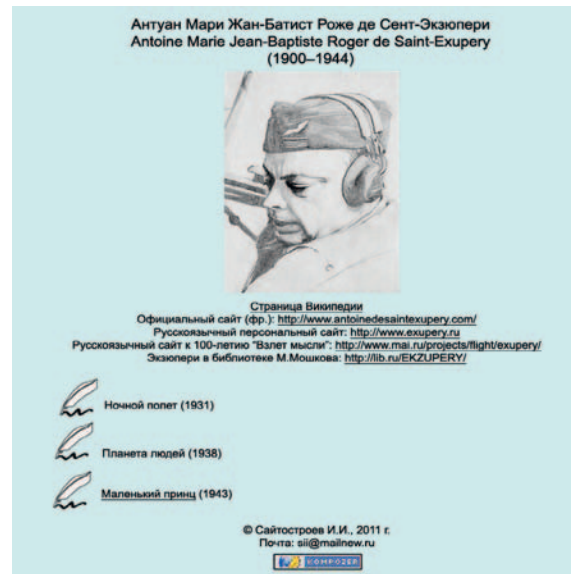
20 В диалоговом окне установим переключатель в положение Использовать цвета. Обратим внимание, что можно изменить стандартное цветовое оформление шрифтов по своему усмотрению, но выберем только кнопку Фон. Зададим цвет страницы, руководствуясь при этом правилами дизайна и здравым смыслом. Также заметим, что в качестве фона можно использовать внешнее изображение. Нажмем ОК.

21 Не забудем упомянуть и себя. В конце страницы по центру создадим «служебный колонтитул» в соответствии с правовыми нормами и правилами «нетикета» (см. Википедию) примерно такого вида:

Знак © получим с помощью пунктов меню Вставка/Буквы и символы..., при положении переключателя Общие символы в соответствующем окне диалога. Фамилию и почтовый адрес, конечно, напишем свои.

Баннер (картинку-ссылку) программы вставим с помощью пунктов меню Вставка/Украшения/Кнопка «Создано в KompoZer». При этом вставка произойдет через несколько секунд (в зависимости от соединения). Можно оснастить баннер всплывающей подсказкой Создано в KompoZer с помощью пункта встроенного меню Свойства изображения и ссылки...

22 В результате первого нашего Созидания должно получиться примерно то, что показано на рис. 10. **LXF**



► Рис. 10. Вот что у нас получилось!



Академическая программа для учебных заведений

Mandriva Linux

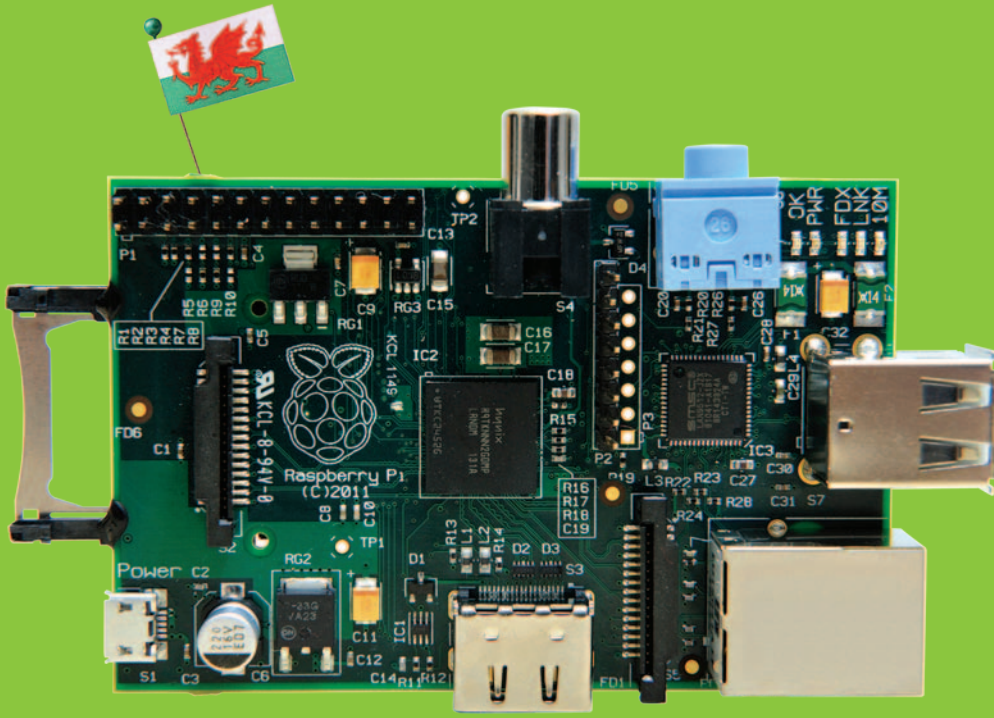
Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

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
 Тел.: (812) 309-06-86, (499) 271-49-55
 info@mandriva.ru

По этой программе учебное заведение получает:

- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку



В январском номере

Проекты Raspberry Pi

Развлеките малышей, восхитите своих пассий и обманите ожидания учителей благодаря нашему празднику творческого легкомыслия.

Патенты – это просто

Патенты внушают трепет своей сложностью, как ядерная физика или система отбора в колледжи США. Но это ненадолго!

Построено на Django

Вы этого хотели, ну и получайте: совершенствуем web-программирование в рабочей среде на базе Python.

Bitcoin

Активируйте свою форму количественного упрощения и повысьте наши акции банка RBS с помощью сервера Bitcoin.

Содержание будущих выпусков может меняться – вдруг мы увязнем в недрах Швейцарии.

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

Редактор диска

Кирилл Степанов

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Ланинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 7362

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] graham_morrison@futurenet.com

Заместитель редактора Эндрю Грегори [Andrew Gregory]

agregory@futurenet.com

Редактор диска Бен Эверард [Ben Everard] ben_everard@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] efrain_hernandez-mendoza@futurenet.com

Выпускающий редактор Гэри Уокер [Gary Walker] gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Бен Эверард [Ben Everard], Дэвид Хейвард [David Hayward], Джульетта Кемп [Juliet Kemp], Лео Максвелл [Leo Maxwell], Грэм Моррисон [Graham Morrison], Адам Оксфорд [Adam Oxford], Джонатан Робертс [Jonathan Roberts], Майк Сондерс [Mike Saunders], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Ник Вейч [Nick Veitch], Евгений Балдин, Михаил Выграненко, Артем Зорин, Андрей Ушаков, Алексей Федорчук, Роман Ярыженко

Художественный ассистент Кейт Мак-Доннелл [Kate McDonnell]

Иллюстрации Шейн Коллиндж [Shane Collinge], Ely Walton Illustrations,

iStockPhoto, Саймон Миддлвек [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

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

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензируются Future Publishing Ltd (Future plc group company). Все права защищены. Никакая часть этого журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает исключительное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на 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 повреждения или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux – зарегистрированный товарный знак Линуса Торвальдса [Linus Torvalds].

«GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

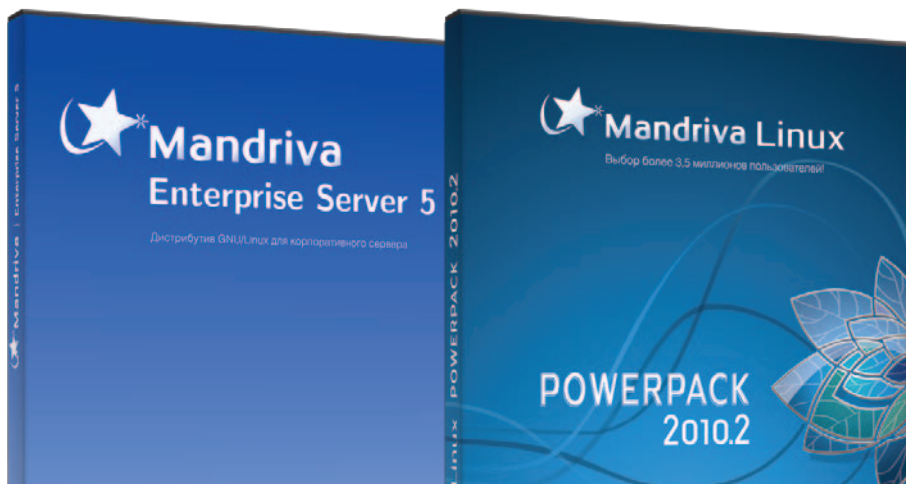
За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO



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.
Корпоративные продукты Mandriva	
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
Продукты Mandriva для образования	
Комплект программного обеспечения Mandriva Linux и EduMandriva для школ	3 500
Сертифицированные ФСТЭК продукты Mandriva	
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 ИСПОЛЬЗУЮТ:
 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



РАЗЫСКИВАЕТСЯ!

Надёжный и доступный по цене выделенный сервер шлет владельца пользователя!



*Предлагаю:
полный root-доступ, отличную производительность,
максимальную доступность и высокую эффективность.*

ru.hetzner.com

ru.hetzner.com

HETZNER DEDICATED ROOT SERVER EX 4

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 1900 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER EX 5

- Intel®Core™ i7-920 Quadcore с поддержкой технологии Hyper-Threading
- 24 ГБ DDR3 RAM
- 2 x 750 ГБ SATA 3 Гбит/с (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 0 рублей

2500
рублей в месяц



GreenIT
Best Practice Award
2011

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!



RU.HETZNER.COM

*Трафик предоставляется бесплатно. При превышении порога 10000 ГБ/месяц скорость соединения будет ограничена до 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТВ.