

# LINUX

ЧИТАЮТ ВСЕ ПОЛЬЗОВАТЕЛИ LINUX

## FORMAT

**БЕЛЕНДОРФ**  
Хакер-музыкант из Apache – о том, куда движется Web с.34



Будь то рисование или издательское дело, открытые инструменты – это будущее профессионального дизайна с.38

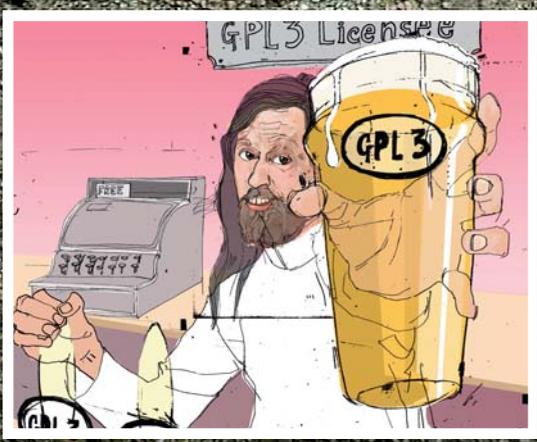
№6 (80)  
ИЮНЬ 2006

## ИЗУЧАЕМ LINUX

- Шифруем файловую систему с.80
- Изучаем регулярные выражения с.74
- Пишем макросы для OpenOffice.org с.70

- GIMP ●
- INKSCAPE ●
- SCRIBUS ●
- XARA 3D ●

**GPL 3** Столлмен, Моглен и Торвальдс – лицом к лицу с.52



*«Если, в один прекрасный день мы проснемся и скажем: «Ого, да Novell – это Microsoft от Linux», тут-то движение Linux и закончится.»* Грег Манкузи-Унгаро с.50



# LINUX FORMAT

## К вашим услугам

Проникнувшись духом Libre Graphics, мы решили спросить у команды LXF: **Если бы вы могли делать любой интересный вам журнал в Scribus, какой журнал бы вы выбрали?**



**Пол Хадсон**  
*Linux Format.* Я так долго работал над ним, что забыл, как выглядит все остальное.



**Грэм Моррисон**  
*Linux Format.* Ведь сегодня Scribus могут использовать даже сторонники Mac.



**Эфраин Эрнандес-Мендоса**  
*Linux Format.* В Playboy слишком мало пингинов.



**Майк Сондерс**  
*Linux Format.* Просто для того, чтобы всем пришлось изучить новую программу. Ха!



**Ребекка Смолли**  
*Enid Blyton Magazine,* закрытый в 1959 году. Но ему придется стать LXF.



**Эндрю Грегори**  
*Linux Format,* потому что вы, ребята, лучшие!



**Дэвид Картрайт**  
*Linux Format.* В нем будут статьи вроде: «Переходим с InDesign на Scribus» и «Почему мне надоел InDesign».



**Энди Ченнел**  
*Linux Format.* Мое начинание может убедить разработчиков добавить настраиваемое поле для даты, которое я жду с версии 0.0.0.4.



**Ричард Коббетт**  
*Linux Format.* Потому что во всех остальных журналах дизайнеров приходится отгонять от их Mac'ов хворостинкой.



**Марк Бейн**  
*Linux Format.* Лучшим авторам — лучшую программу.

# Libre graphics



## О единстве



Нам часто приходится слышать — «Linux фрагментирован», «GNOME ведет священную войну против KDE» и так далее. Утверждается, что подобное положение дел есть артефакт свободы: каждый делает так, как ему нравится, а глобального контролирующего органа нет — отсюда неразбериха. Но так ли это?

Тем, кто изучал основы экономической теории, знакомо понятие саморегуляции, представленное там в форме «невидимой руки рынка». В результате свободной конкуренции — за кошелек ли покупателя или место на компьютере пользователя, неудачные решения отмирают, а удачные — совершенствуются. В этом выгодное отличие открытого ПО от закрытого, подчиненного одной организации, которая может нарушить этот естественный баланс в угоду собственным соображениям.

Оглянитесь вокруг — технологии «единства Linux» уже здесь: это D-BUS, HAL, Project Portland и другие, развивающиеся под «зонтиком» проекта Freedesktop.org. Примечательно, что хотя все они несут на себе едва уловимый отпечаток Gnome, их можно использовать практически из любого инструментария. Не так давно было завершено портирование kdelibs4 на D-BUS, а это значит, что одним универсальным решением в мире открытого ПО стало больше. Сравнительно молодые приложения вроде NetworkManager изначально разрабатываются с прицелом на этот единый набор стандартов. В выигрыше же оказываются простые пользователи: предпочитаете ли вы Fedora Core 5 (GNOME) или OpenSUSE 10.1 (KDE), NetworkManager всегда будет доступен для вас, причем в таком виде, к которому вы привыкли.



С уважением,  
**СИНИЦЫН ВАЛЕНТИН**  
главный редактор LinuxFormat в России

### КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации:

[letters@linuxformat.ru](mailto:letters@linuxformat.ru)

Подписка и предыдущие номера:

[subscribe@linuxformat.ru](mailto:subscribe@linuxformat.ru)

Техническая поддержка:

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

Проблемы с дисками:

[disks@linuxformat.ru](mailto:disks@linuxformat.ru)

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

Website: [www.linuxformat.ru](http://www.linuxformat.ru)

Адрес редакции: Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54  
Телефон редакции: (812) 717-00-37  
Дополнительная информация на стр.120

### МИССИЯ ЖУРНАЛА

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО



# Содержание

**LINUX**  
FORMAT

Добро пожаловать в *LinuxFormat* – ваш гид в мире Linux!

**LXF6(80) ИЮНЬ 2006**

»» ЧИТАЙТЕ В ЭТОМ ВЫПУСКЕ



## FREE, LIBRE GRAPHICS 38

Эти революционеры сбросят одряхлевшие коммерческие приложения с парохода современности

### 26 HotPicks

Наконец-то! Открытый Flash, который работает

### 48 iFolder за 10 минут

Резервирование и общий доступ – проще пареной репы

### 52 GPL 3: вкус свободы

Ключевые моменты черновой версии

### 84 СКОРОСТНОЙ СТАРТ

Советы по ускорению загрузки вашей системы



### 52 Бармен свободы

МЫСЛИ  
ВСЛУХ

«Вы должны почувствовать разницу между **тяжелым металлом** и классическим хором.»



12 Грег Моррисон об индикаторе настроения в Amarok

«Это было ужасно! **Просто ужасно**. Но мы должны были узнать, что это такое – перевести **большую организацию** на Linux.»



50 Грег Манкузи-Унгаро из Novell

## 12 Создай настроение



### 4 Новости

Linspire на свободе

### 10 KOffice 1.5

Ужасно хорош или просто ужасен?

### 12 Amarok 1.4

Большой и толстый

### 13 Gaim 2.0

Summer of Code прошло не зря!

### 14 Seapine Surround SCM

Управление версиями за деньги

### 15 Damn Small Linux

Легкий и быстрый – как МиГ-29

### 16 Moneydance 2006

Считайте ваши денежки

### 18 Distrowatch

Мир LiveCD сошел с ума?

### 20 Roundup: LiveCD/DVD

Так оно и есть! Вот 7 самых адекватных

### 26 HotPicks

Лучшие новинки этого месяца

### 32 Что такое... Xgl?

Рабочий стол будущего или...

### 34 Брайан Белендорф

Анегл Apache

### 38 Libre Graphics

И у них тоже есть слои!

### 48 iFolder за 10 минут

Резервирование – это просто

### 50 Грег Манкузи-Унгаро

Почему Novell – это не Microsoft

### 52 GPL 3

За или против?

### 56 Конкурс

IBM WebSphere Community Edition Contest 2006

### 58 PDF и Flash

Не JPEG'ом единым жив линуксоид

### 62 It-Robot

Сайт как конструктор

### 66 Учебник: первые шаги

Сводим видео и звук

### 70 Учебник: OOo Basic

Собственный макрос для Writer

НОВАЯ СЕРИЯ

### 74 Hardcore Linux

Долгий путь к регулярным выражениям

### 78 Учебник: PHP

Как это будет... по-польски?

### 80 Учебник: шифрование

Защитите свои данные

### 84 Учебник: загрузка

Грузитесь быстрее – работайте дольше

### 88 Программирование для Qt/KDE

Красивый интерфейс – одной мышью

### 92 Программирование для Linux

Файлы – это наше все

НОВАЯ СЕРИЯ

### 96 Уроки Python

Истинное лицо ваших скриптов

### 100 Учебник: MetaPost

Заключительная серия

### 106 Вопрос? Ответ!

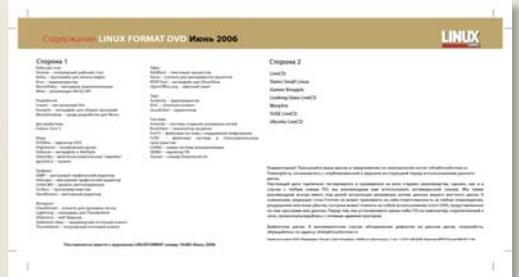
Мы вам поможем

### 112 Диск Linux Format

Почти трехсторонний DVD!

# На диске

Целый DVD интересных программ **112**



## »» ДВУСТОРОННИЙ DVD!

**FEDORA CORE 5** Используйте Mono, SELinux и Xep или же приукрасьте рабочий стол с помощью AIGLX. Классический дистрибутив – почему бы и не попробовать?

**ЖИВЫЕ ДИСТРИБУТИВЫ** Шесть вариантов – и никакой инсталляции

SUSE, Ubuntu, Morphix, DSL и другие

**AMANDA** Держите данные в безопасности с помощью этой масштабируемой системы резервного копирования

**LIGHTNING** Календарь для Mozilla Thunderbird

**HIGHMOON** Как Worms, но в космосе!

Вырежьте свою обложку для DVD на стр. 117-118



Перед использованием диска ознакомьтесь с инструкцией на стр. 117



34 Брайан Белендорф



26 HotPicks



# ГЛАВНЫЕ НОВОСТИ

- Linspire на свободе • Picasa для Linux • Devicescape открывает Wi-Fi стек • Новая лицензия Java
- «|p|» значит «рука помощи»

## Linspire на свободе

Закрытый коммерческий дистрибутив становится ближе к сообществу

**Едва ли вам удастся найти** пользователя Linux (да, пожалуй, и пользователя вообще), который ничего не слышал о *Linspire*, в «девичестве» - *Lindows*. Труднее этого может быть только найти пользователя, который реально работал с данным дистрибутивом. Однако, все может измениться и уже в самом ближайшем времени.

Идея *Lindows* первоначально зародилась в голове создателя портала MP3.com Майкла Робертсона (Michael Robertson). В то время предполагалось, что *Lindows* будет закрытой коммерческой ОС (даже слово «дистрибутив» в отношении нее как-то не употреблялось), которая будет практически

полностью совместима с Windows. На первых порах компания активно инвестировала в проект Wine, но потом внесла коррективы в свою стратегию, решив сосредоточиться не на двоичной совместимости, а на разработке достойных аналогов популярным Windows-приложениям. Итогом этого стало появление на свет кросс-платформенной среды web-разработки Nvu ([www.nvu.com](http://www.nvu.com)), менеджера фотографий LPhoto и мультимедиа-проигрывателя LSongs и других программ, распространявшихся уже по открытым лицензиям. Более радикальные сдвиги в позиционировании *Linspire* на рынке произошли с уходом Робертсона с поста президента компании. Его преемник,

Кевин Кармони (Kevin Carmony), быстро подружился с проектом Ubuntu (это было несложно, так как оба дистрибутива преследуют одинаковые цели и являются потомками Debian), а затем предпринял и более решительный шаг: анонсировал свободную версию *Linspire* - *Freespire*.

Завсегда датаи новостных лент могут заметить, что это название уже мелькало в них - и будут правы. Термин «*Freespire*» впервые появился на страницах DistroWatch.com летом 2005 года, но тогда речь шла о любительском проекте, который вскоре был вынужден сменить название по требованию *Linspire*. В отличие от своего более раннего предшественника, сегодня *Freespire* - это абсолютно официальная инициатива, имеющая собственный сайт ([www.freespire.org](http://www.freespire.org)) и поддерживаемая *Linspire, Inc.* Более того, модель разработки *Linspire* перестала быть полностью закрытой: будущие версии дистрибутива будут выпускаться по принципам, сходным с OpenSUSE/SUSE Linux.

Что же такое *Freespire*? Согласно описанию на домашней странице, это, в первую очередь «управляемый сообществом, основанный на Debian дистрибутив Linux, который всегда будет свободным для использования, распространения и модификации». *Freespire* будет состоять только из открытых компонентов, однако, желающие смогут легально установить мультимедиа-кодеки, средства работы с DVD, проприетарные драйвера, шрифты и тому подобное из репозитория *CNR Warehouse*. Клиент *CNR Warehouse* является открытым, однако, доступ к хранилищу (часто называемому главным достоинством коммерческого *Linspire*) будет бесплатным, хотя и не дорогим: 20 долларов в год. Не торопитесь рвать журнал и кричать: «Нас опять обманули!» *Freespire* действительно является community-дистрибутивом, поэтому вы можете заработать купон для бесплатного доступа, выполняя различные общественно-

полезные задачи. Присылайте патчи, разрабатывайте документацию или делайте переводы (кстати, хорошая многоязыковая поддержка является одним из приоритетов *Freespire*, интересующихся отсылаем к проекту *IRMA* - [irma.linspire.com](http://irma.linspire.com)) - и ваши усилия будут вознаграждены. Впрочем, и сторонникам идеи «если кто-то не дает мне доступ к чему-то за просто так, этот кто-то ограничивает мою свободу» нет повода обижаться: *Freespire* (как и Debian) может использовать apt-get, но (естественно) только для установки свободных приложений. Это является существенным отличием от *Linspire*, который тесно связан с CNR и не сможет (официально!) использовать apt-get как минимум до версии 6.0.

Теперь, вероятно вы задаете себе вопрос: «А когда же можно будет попробовать эту новинку в деле?» К счастью, ждать осталось не так уж долго: первая бета-версия ожидается 1 августа 2006 года, а финальный релиз - уже 1 сентября. Как поясняют разработчики, *Freespire 1.0* базируется на хорошо протестированном коде *Linspire 5.0*, поэтому длинная обкатка перед выпуском не обязательна. Следует также иметь в виду, что *Freespire 1.0* станет формальной презентацией платформы, выпуск которой обусловлен скорее стремлением развести процессы разработки *Linspire* и *Freespire*. Планируемая к выходу в четвертом квартале этого года *Freespire 1.1* будет куда более интересным продуктом. В ее состав войдет поддержка нескольких языков (будет ли среди них русский, пока не ясно), клиент CNR, ядро 2.6.17, KDE 3.5 и GCC4. Дистрибутив теперь открыт и кто знает - быть может, мы даже разместим *Freespire 1.1* на одном из дисков Linux Format!

Принять участие в развитии *Freespire* можно уже сейчас. Просто зарегистрируйтесь на сайте [www.freespire.org](http://www.freespire.org) и откройте страничку Community.



# Picasa для Linux

Google портировал в Linux одно из своих настольных приложений

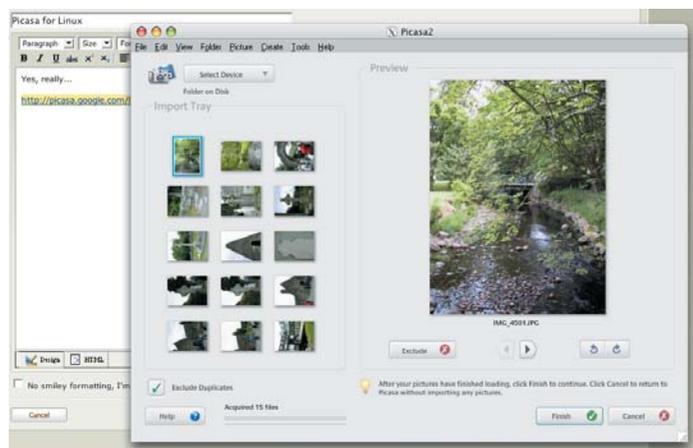
**Лаборатории Google Labs** объявили о выпуске бета-версии менеджера фотографий *Picasa* ([picasa.google.com](http://picasa.google.com)) для Linux. Как и *Picasa* для Windows, Linux-порт доступен только в двоичной форме, однако, должен работать на большинстве дистрибутивов. На сайте *Picasa* доступны пакеты в формате RPM/DEB, а также самораспаковывающийся инсталлятор.

Следует отметить, что *Picasa* не стала Linux-приложением в полном смысле этого слова. Перенос программы в открытую среду был осуществлен компанией *CodeWeavers* с использованием технологий Wine. В ходе портирования *Picasa* было создано более 100 патчей к «не-эмулятору», которые были переданы сообществу, поэтому, даже если вы никогда не занимались цифровой фотографией (и *Picasa* как

таковой вам не интересен), то все равно окажется в выигрыше. Помимо Wine, *Picasa* for Linux использует Gecko – движок Mozilla.

Как уже говорилось выше, текущий выпуск *Picasa* является бета-вер-

сией. Среди отсутствующих функций можно отметить запись компакт-дисков, многоязыковую поддержку и экспорт в TiVo. *Picasa* также не будет корректно работать на системах с включенным SELinux.



© 2006 DIGITAL INSPIRATION - TECHNOLOGY, A LA CARTE

## КОРОТКОЙ СТРОКОЙ

- Вышли обновления для двух популярных систем семейства BSD: **FreeBSD 6.1/5.5** (а также PC-BSD 1.1) и **OpenBSD 3.9**.
- Открытый векторный редактор **Xara** обновился до версии 0.5. Теперь поддерживается запись файлов и экспорт в растровые форматы.
- Увидели свет **XFree86 4.6.0** и **X.Org 7.1**.
- Самый старый из существующих ныне текстовых редакторов – **VI[M]** – разменял очередной номер версии. Встречайте **VIM 7!**
- Компания Intel подарила проекту Apache Harmony (открытая реализация Java) собственную версию библиотеки **Swing/AWT**.
- Компания Xandros, производитель одноименного настольного дистрибутива, представила серверный вариант – **Xandros Server**.

# Devicescape открывает Wi-Fi стек

День, когда беспроводные сетевые карты будут «просто работать» в Linux, не за горами

**Компания Devicescape**, выпускающая «конструкторы», из которых сторонние производители затем собирают свои беспроводные сетевые адаптеры, опубликовала исходные тексты своего Wi-Fi стека «*Advanced Datapath*» по лицензии GPL. Компания выразила надежду, что этот шаг позволит работать с Wi-Fi устройствами, не дожидаясь, пока поставщики оборудования выпустят Linux-драйвера. Кроме этого, стек *Advanced Datapath* обладает улучшенной поддержкой мультимедиа-поточков (голоса и видео), так что его открытие упростит создание новых устройств наподобие Wi-Fi телефонов.

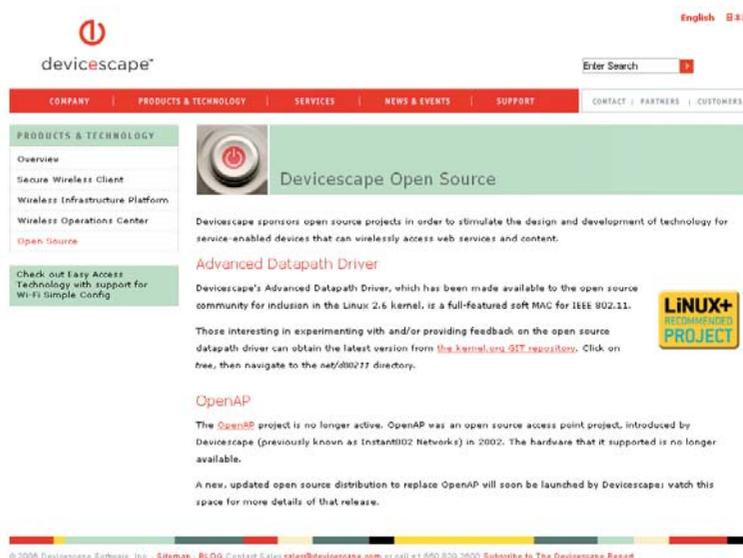
Будучи включенным в ядро (такое предложение уже прозвучало и нашло весьма положительный отклик), стек *Devicescape* решит ряд проблем, в настоящее время осложняющих создание родных Wi-Fi драйверов для Linux. В состав решения входит программный контроллер среды (MAC), *hostapd* (режим точки доступа), реали-

зации WEP, WPA, WME и QoS. Как сказал Гленн Финлхбо (Glenn Flinchbaugh), менеджер *Devicescape* по маркетингу: «[Наш драйвер] делает задачу разработки других драйверов гораздо проще – вам нужно всего лишь написать

тонкий слой совместимости с нашим API, и в вашем распоряжении окажутся самые современные беспроводные чипсеты».

Примечательно, что стек *Advanced Datapath* был написан, по существу,

двумя людьми: Симоном Барбером (Simon Barber), ведущим научным сотрудником *Devicescape* и Йони Малиненом (Jouni Malinen), автором открытого *wpa\_supplicant*.



## ЧТО БЫЛО

★ Санкт-Петербургская Конференция Интернет и Реклама - Всероссийский Интернет-МАРАФОН  
31 мая 2006 года,  
г. Санкт-Петербург

31 мая 2006 года в отеле Ольгино (20 минут езды от г. Санкт-Петербурга) в рамках Санкт-Петербургской Конференции «Интернет и Реклама» прошел Всероссийский Интернет-МАРАФОН. Были подробно рассмотрены вопросы сборки и настройки web-серверов на базе ОС Linux, обеспечения информационной безопасности и стабильной работы сайта под высокой нагрузкой. Кроме того, в рамках конференции специалисты компании R-Style провели эксклюзивную презентацию нового (пятого) поколения серверов ProLiant от HP, официально не объявленных на рынке. Для посещения презентации даже требовалось подписать соглашение о неразглашении информации.

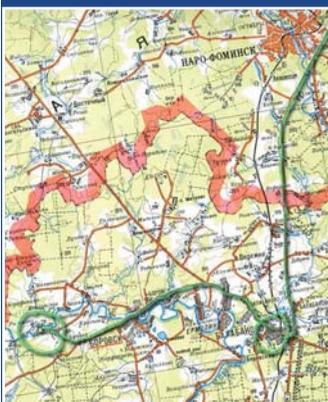
[www.spkir.ru](http://www.spkir.ru)

## ЧТО БУДЕТ

★ LinuxFest-8  
28-30 июля 2006 года, Калужская область

Ежегодный фестиваль LinuxFest в этом году будет проходить в рамках «Недели Linux На Протве» с 28-ого по 30-е июля 2006-ого года в Боровском районе Калужской области на берегу живописной реки Протвы. Количество участников фестиваля не ограничено, места под солнцем хватит всем. Обращаем ваше внимание, что организаторы фестиваля предоставят для его участников только место для установки палатки, интересных собеседников и запас дров. Все остальное (продукты питания, туалетные принадлежности и наборы для развлечений) необходимо брать с собой. Оргвзнос составляет 200 рублей с участника.

<http://lnr.ru/index.php?module=linuxfest>



# И снова о Java

**Новая лицензия позволит включать среду времени выполнения и комплект разработчика в свободные дистрибутивы**



### Компания Sun Microsystems

изменила условия распространения Java Platform, Standard Edition (Java SE) 5 таким образом, чтобы ее можно было включать в свободные дистрибутивы Linux. Ранее это было невозможным и, хотя некоторые дистрибутивы (как правило, разрабатываемые энтузиастами) закрывали глаза на нарушение лицензионного соглашения, более крупные проекты были вынуждены выносить Java в различные «дополнительные» репозитории или включать ее только в состав коробочных версий.

Соглашение, названное «*Operating System Distributor's License for Java*» (или «*Distro License for Java*» - DLJ) явным образом разрешает дистрибьюторам распространять *Java SE 5.0*.

Java Development Kit (JDK) и Java Runtime Environment (JRE) в виде установочных пакетов для соответствующих операционных систем. Поставщики вправе выбирать способ упаковки, установки Java

и обеспечивать поддержку технологии в рамках своего дистрибутива.

Свое намерение распространять JDK уже выразили Ubuntu, Gentoo и Debian, а также дистрибутивы OpenSolaris: NexentaOS, Schillix и BeleniX. «Мы действительно рады видеть, как Sun завязывает все более и более тесные взаимоотношения с движением свободного ПО, начиная с открытия Solaris и заканчивая новой лицензией Java, совместимой с дистрибутивами GNU/Linux,-

сказал Энтони Таунс (Anthony Towns), лидер проекта Debian.

Впрочем, не все настроены столь оптимистично. Брайан Стивенс (Brian Stevens), технический директор Red Hat, считает, что Sun следовало бы просто открыть виртуальную машину Java, чтобы разработчики могли улучшить функции реального времени. «Они же пытаются ограничиться минимально возможными уступками», -сказал г-н Стивенс, имея в виду DLJ.



# Пишите с пользой

**Вновь созданная служба электронной почты собирает деньги на благотворительность и помощь открытым проектам**



### Запущена в эксплуатацию

новая служба электронной почты, которая собирается жертвовать заработанные деньги на благотворительность и поддержку открытых проектов. *Ippimail* - детище сторонника свободного ПО Симона Мартина (Simon Martin), который давно искал способ принять участие в

движении Open Source. «Все это началось как попытка не-программиста сделать что-то полезное для свободного ПО», -говорит он. «Именно поэтому мы используем открытое ПО и оказываем финансовую поддержку сообществу».

*Ippimail* - это почтовый сервис, напоминающий *Gmail*. Используя стандартную

баннерную рекламу, компания надеется привлечь средства для благотворительных организаций, в том числе, Save The Children, Breast Cancer Care и The Born Free Foundation. Однако, утверждает Мартин, к инициативе может присоединиться любая официально зарегистрированная организация. Пользователи сами могут выбрать, кому будут перечислены заработанные ими деньги. Он также отметил, что некоторые «крупные компании» побуждают своих сотрудников использовать эту службу. Благотворительным фондам перечисляется 45 процентов прибыли сервиса, еще 10 жертвуются различным открытым проектам.

Мартин надеется, что поддержка сообщества поможет *Ippimail* успешно конкурировать на рынке, лидерами которого является Google, Microsoft и Yahoo. «Если мы хотим создать по-настоящему хороший сервис, который сможет потягаться с лучшими из лучших, в этом должны принять участие все. Мы возлагаем большие надежды на разработчиков open-source».

[www.ippimail.com](http://www.ippimail.com) LXF



'Ippi' в переводе с японского означает «рука помощи», пояснил Майк Сондерсан.

# Интернет-марафон: подробности

**В Ольгино были проведены эксклюзивные презентации и актуальные мастер-классы**

30 и 31 мая в Санкт-Петербурге состоялась конференция «Интернет и Реклама» – первое в Северо-Западном регионе подобное мероприятие – в рамках которой был проведен Интернет-марафон.

Организаторы конференции, которыми выступили Региональный общественный центр Интернет-технологий (РОЦИТ), SEO TEAM, AdMedia, Мезон. Ру, TRINET и Ленвендо, старались привлечь внимание компаний округа к бизнес-возможностям Интернета и продемонстрировать решения актуальных задач на практике.

В пресс-конференции, состоявшейся в день открытия в ИА Росбалт, приняли участие представители интернет-индустрии, госструктур и средств массовой информации.

Одним из самых активных участников Интернет-марафона стала компа-

ния R-Style, которая провела эксклюзивную презентацию нового поколения серверов HP ProLiant G5, официально не представленных на рынке.

Также специалистами компании был проведен целый ряд мастер-классов, посвященных разработанным компанией решениям на базе Интернет-технологий:

- Корпоративные порталы на базе IBM WebSphere и Oracle application Server 10g.
- Средства коллективной работы на базе Oracle Collaboration Suite – интегрированный набор приложений работы с почтой, файлами, голосовой почтой и факсимильными сообщениями.
- Система документооборота на базе Oracle Collaboration Suite (RS-Doc).
- Система дистанционного обучения на базе IBM, Oracle.

• Новые решения по удаленному администрированию ИТ-инфраструктуры на базе HP Inside, iLO, Rapid.

Также в рамках конференции специалисты R-Style представили новые

услуги хостинга департамента информационных решений на серверах HP и продуктах Oracle Application Server 10g и Oracle Collaboration Suite.



Передача символической эстафеты Интернет-Марафона

## Делать больше с меньшими затратами.

Если Вам нужны мощные серверы, которые позволят делать больше работы и консолидировать вычислительные ресурсы, выбирайте серверы ASNORD® на базе двухъядерных процессоров Intel® Xeon®.



Серверы ASNORD® - это работа 24 часа в сутки, высокие уровни реагирования и управляемости, 3 года стандартной гарантии, различные виды расширенной поддержки.

Компания INVICTUS специализируется в области решения задач по созданию систем обработки, хранения и защиты данных на базе кластеров, сетей хранения данных SAN, внешних RAID-массивов, NAS-серверов, ленточных накопителей. Подробнее о наших предложениях Вы можете узнать на нашем сайте по адресу [www.invictus.ru](http://www.invictus.ru). Ждем Ваши запросы по адресу [invictus@invictus.ru](mailto:invictus@invictus.ru) и по телефонам...

**(812) 449-8888 (495) 647-0080**

**INVICTUS®**  
OPEN STORAGE TECHNOLOGIES

Обозначения Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Intel, Intel Core, Intel logo, Intel Inside, Intel Inside logo, Intel SpeedStep, Intel Viiv, Intel Xeon, Itanium, Itanium Inside, Pentium и Pentium Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.

# SUSE на офисное время

Novell приближается к Windows несколькими путями. Майк Сондерс анализирует ее стратегию рабочего окружения Linux...



Некоторое время назад мы говорили об анонсе SUSE Linux Enterprise Desktop (SLED), наследника коммерческого продукта Novell – Novell Linux Desktop. Но это не просто анонс продукта: это показатель сдвига курса Novell как фирмы, бросающей вызов Red Hat Enterprise Linux и Microsoft. Благодаря приобретению Ximian и SUSE, Novell обладает интеллектуальным потенциалом для создания настольного дистрибутива, способного затмить все остальные. Но сможет ли она привлечь на свою сторону основную компьютерную прессу, невзирая на недовольство некоторых членов сообщества открытого кода тем, что часть ее разработок остается закрытой?

На демонстрации для прессы в Париже известный специалист и руководитель разработки SLED Нат Фридман проиллюстрировал блестящие возможности, намеченные для окончательного выпуска SLED 10 этим летом. Используя *Beagle*, *F-Spot* и 3D-эффекты рабочего стола *Xgl*, Фридман просто заворожил многих журналистов, никогда не видевших эти технологии в

действии. Сверхновым этот функционал назвать нельзя, и у пользователей Linux со стажем челюсти не отвалятся, но приятно видеть их хорошо увязанными вместе в дистрибутиве уровня предприятия.

Одной из полезных для офиса особенностей является добавление Deskbar, реорганизованного меню кнопки «Пуск» в Gnome: оно включает группы приложений и сетевой статус. Присвокупите к этому удаление верхней панели меню Gnome в SLED – и получите рабочий стол, во многих отношениях поразительно похожий на Windows XP. (Более подробно – см. ниже беседу с Фридманом). Понятно, что Novell хочет сделать SLED максимально доступным для перебежчиков с Windows, но как хотелось бы видеть нечто уникальное, кричащее: «Я – новое слово в GUI от Linux!»

## Настольный ренессанс

Novell называет свою переориентацию на конечного пользователя настольных компьютеров «возрождением». Фирма считает, что теперь обладает продуктом для



Готовый офис? Упрощенное главное меню SLED, а на заднем плане – стремительно улучшающийся поисковый инструмент Beagle.

стандартного офисного пользователя, тогда как ранее, с Novell Linux Desktop, целью были устройства с конкретными функциями (например, торговые точки) и технические рабочие станции, где рядом всегда должен быть специалист. На данном этапе мы не уверены, что переход с NLD на SLED так уж необходим – большинство «новых» возможностей уже давно в деле, и во многих отношениях это скорее обычное обновление дистрибутива, хотя и отшлифованного до блеска.

Но пусть это и не революционный скачок, все равно SLED 10 включает ценнейшие функции для привлечения внимания рынка офисных пользователей. Одна из наиболее заметных – поддержка VBA-макросов Excel в *OpenOffice.org*, облегчающая процесс перехода пользователей Windows

(и их файлов!) на Linux. SLED – скорее NLD+1, чем совершенно новый дистрибутив, но уровень внимания, уделенный GUI, заслуживает большой похвалы.

И что же из этого следует? Главным конкурентом SLED на Linux-арене является Red Hat Enterprise Workstation, и несомненно, Novell лидирует, имея смачные улучшения рабочего стола типа *Xgl*. Но у Red Hat большой кредит доверия, так что Novell должна всех убедить, что SLED – это прочное и надежное, а не только красивое решение для корпоративных компьютеров. Учитывая, что Novell продолжает выдавать хорошо скомпонованные выпуски SUSE, а также достойным одобрением способом управляет проектом OpenSUSE, мы можем сказать, что фирма отлично вооружена для грядущих битв за рынок. **LXF**

«NOVELL ЛИДИРУЕТ, С ЕЕ СМАЧНЫМИ УЛУЧШЕНИЯМИ РАБОЧЕГО СТОЛА ТИПА XGL.»

## НАТ – НАМ

Мы поймали Ната Фридмана (Nat Friedman) в Париже и задали ему наши вопросы касательно рабочего стола...

**LXF:** Gnome в SUSE Linux Enterprise Desktop стал больше походить на Windows, исчезла верхняя панель меню в стиле Mac...

**НФ:** В 2000 году мы в моей фирме [Ximian] разработали расширение Gnome, добавляющее панель к верхнему краю, как в Macintosh.

Но в наших прошлогодних тестах на удобство использования мы обнаружили, что такая панель на самом деле – плохая идея. Она приводила в замешательство и пользователей Windows, которые используют только одну нижнюю панель, и пользователей Macintosh,

привыкших к кардинально отличающейся верхней панели. Таким образом, мы сделали вещь, которая неудобна всем, и, просмотрев сотни часов видео [заснятых во время тестов], решили: «Нам этого не надо!»

**LXF:** Как вы думаете, наличие двух конкурирующих рабочих столов (KDE и Gnome) влияет на прогресс Linux, если вообще влияет?

**НФ:** Пока, вероятно, нет, но это может произойти в будущем.

Независимым поставщиком ПО (ISV) становится трудно писать приложения для Linux; они хотят писать просто для Linux, а не для KDE

или Gnome. В конце прошлого года у нас была встреча на OSDL...

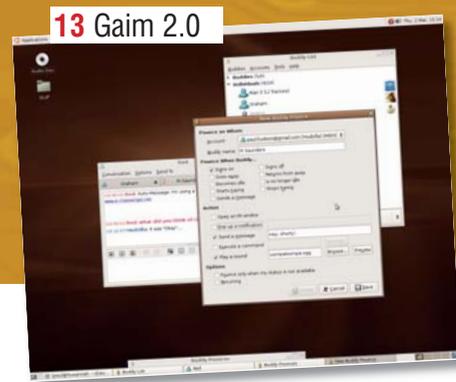
**LXF:** Встреча по общим стандартам проекта Portland?

**НФ:** Да, [там] было шестеро или семеро ISV. Одной из их претензий была: «Нам нужен один общий вариант выполнения задач – один способ размещения иконки на рабочем столе или поиска принтера. А не так, что для Gnome один способ, а для KDE – другой».

В этом и цель проекта Portland: разработка стандартного интерфейса для выполнения стандартных функций.

# Обзоры >>>

Новинки программного и аппаратного обеспечения с точки зрения наших экспертов.



13 Gaim 2.0

## XUBUNTU: В БЛАГОРОДНОМ СЕМЕЙСТВЕ ПРИБЫЛО



Алексей Федорчук  
«сначала читает – потом ставит»

До сего дня Ubuntu распространялся в двух вариантах – собственно Ubuntu с Gnome в качестве рабочего стола, и Kubuntu, в котором его роль выполняет KDE. Что отсекало от него тех, кто не питает симпатий ни к одной из этих сред. И

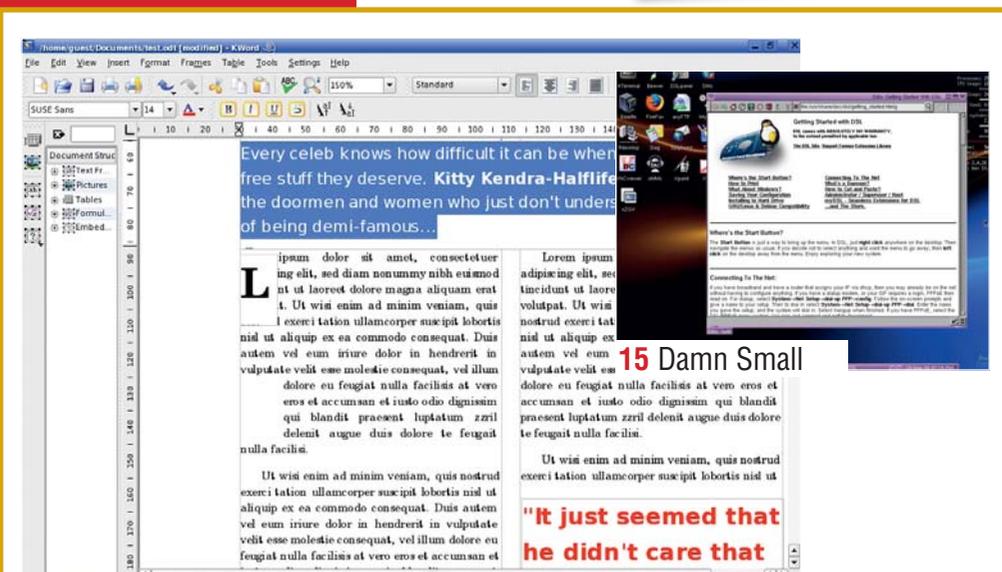
вот – выход еще одного варианта, Xubuntu (<http://xubuntu.org>), приобщающего пользователя к африканскому гуманизму посредством XFce. Последний представлен здесь в бета-версии 4.4, существенно отличной от всех предыдущих.

А так Xubuntu – это самый обычный Dapper, укомплектованный штатными приложениями XFce (файловым менеджером Thunar, текстовым редактором mousepad, и т.д.). Сторонних приложений – также минимум: в качестве браузера – Mozilla Firefox, почтовую службу отправляет Mozilla Thunderbird, обработкой изображений занят Gimp. Что же до канторских обязанностей – они возлагаются на AbiWord, ни малейшего OpenOffice.org мы тут не увидим.

Вполне достаточно, чтобы сразу начать работать, а для восполнения недостающего – весь репозиторий Ubuntu. То есть политика разработчиков – в том, чтобы не навязывать пользователю своих решений, предоставляя ему полную свободу выбора. Особенно в области мультимедийных приложений, в наибольшей степени подверженных влиянию вкусов и моды.

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

## ЧТО НОВЕньКОГО?



10 KOffice 1.5

### 10 KOFFICE 1.5

Надоело ждать, пока загрузится OpenOffice.org? За то же самое время KSpread успевает запуститься, не прочесть таблицу Excel и рухнуть!

### 12 AMAROK 1.4

Музыкальный проигрыватель KDE теперь имеет индикатор настроения и другие функции. До сего момента вы даже не подозревали, что они вам нужны!

### 13 GAIM 2.0

Мгновенный обмен сообщениями должен быть простым, гибким и мгновенным. Gaim фокусируется на этих трех функциях – и выигрывает

15 Damn Small



### 14 SEAPINE SURROUND SCM 4.1

Вести контроль версий можно с помощью отличного свободного программного обеспечения, а можно и за деньги, но без излишних удобств...

### 15 DAMN SMALL LINUX 2.3

Не так страшен черт, как его малюю... Малюсенький, чертовски маленький Линукс, который неплохо иметь у себя на CD-визитке!

### 16 MONEYDANCE 2006

Если у вас есть финансы – управление ими стоит поручить этой программе. Сколько стоит? Примерно \$30 – ну так ведь у вас есть финансы!

## НАШ ВЕРДИКТ: ПОЯСНЕНИЕ

Все продукты оцениваются по 10-балльной шкале. 4 обычных параметра оценки: возможности, производительность, простота использования и соотношение «цена/качество», но для свободного ПО последний параметр может быть заменен на оценку документации. Независимо от набора категорий, мы всегда вычисляем общий рейтинг, подводящий итог нашим высказываниям.



Продукты, выделяющиеся из основной массы, получают престижную награду *Linux Format Top Stuff Award*. Выбираются самые-самые – и только лишь высокая оценка здесь еще ни о чем не говорит.

Для тестирования серверов используется наша собственная разработка, LFXBench 2004, состоящая из 4 основных подсистем: Multi CPU, Single CPU, RAM и Hard Disk. Оценки усредняются и дают общий рейтинг.

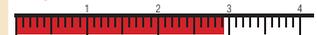
1 очко означает, что тестовый компьютер равен нашему эталону: Pentium 4 1,8ГГц, 512 Мб оперативной памяти и жесткий диск IDE. 2 очка означает, что он вдвое быстрее.

Все тесты выполняются под управлением Red Hat Enterprise Linux 3 AS на соответствующей платформе (x86, AMD64, Itanium). Код компилируется при помощи GCC, если не указано иное.

### РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

MULTI CPU	6.07
SINGLE CPU	3
RAM	2.17
HARD DISK	0.46

### ОБЩИЙ РЕЙТИНГ 2.93



ОФИСНЫЙ ПАКЕТ

# KOffice 1.5

На рынке офисных пакетов в 2006 году ожидается много инноваций. Энди Ченнел думает, что *KOffice* вряд ли займет там достойное место.

## САМОЕ ГЛАВНОЕ

Интегрированный офисный пакет для KDE. Аналоги: *OpenOffice.org*, *Gnome Office (AbiWord, Gnumeric)*, *ThinkFree Office*.

- **РАЗРАБОТЧИК:** Команда разработчиков KOffice
- **САЙТ:** [www.koffice.org](http://www.koffice.org)
- **ЦЕНА:** Бесплатно по лицензии GPL



Хороший офисный пакет в силах помочь становлению операционной системы. Перенеситесь в 1992 год, когда Microsoft выпустила *MS Office 3.1*. Всего за сутки было продано свыше трех миллионов копий, и благодаря этому свершился переход от MS-DOS к Windows.

Вернемся в наши дни – и увидим, что планируется похожий сдвиг. Microsoft решила заново изобрести весь пользовательский интерфейс в грядущем обновлении *MS Office* (шаг, чреватый переходом миллионов пользователей на другие ОС); формат пакета *OpenOffice.org* OpenDocument принят на вооружение другими приложениями; Google взяла под крыло онлайн-текстовый процессор *Writely*; а ThinkFree выпустила в свободное плавание Java-версию своего пакета. Конкуренция становится жестче, мы решили взглянуть на *KOffice 1.5*, офисный пакет с большой историей, на которому еще

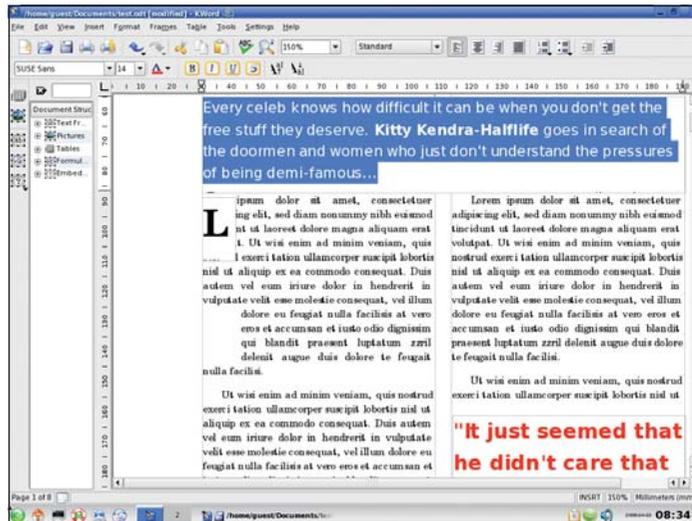
много предстоит пройти. В новую версию вошло множество исправлений, а также наведен блеск. Достигла ли она полного расцвета?

*KOffice 1.5* доступен с сайта проекта ([www.KOffice.org](http://www.KOffice.org)) в виде исходных текстов и двоичных файлов. В пакет входит 11 приложений: восемь основных и три вспомогательных, для построения графиков, формул и докладов. Пакет занимает одну треть размера *OpenOffice.org*, и тот факт, что он основан на едином наборе библиотек, позволяет легко разделять функции между приложениями. Это хорошая возможность для просмотра (Konqueror может отображать содержимое файлов *KWord*), также облегчает встраивание документов одного приложения в другое в пределах офисного пакета.

А еще это означает, что приложения (по большей части) имеют единый пользовательский интерфейс, и что текстовый процессор имеет доступ ко всем функциям таблиц (*KSpread*) и программы создания иллюстраций (*Karbon14*).

## Курс по ухабам

К сожалению, сильная сторона *KOffice* – интеграция – одновременно и его слабость: ошибки одного приложения появляются в других приложениях. Ошибки версии *1.5* действительно странные (например, загадочное отсутствие переноса строки). Вдобавок любая экономия времени,



*KWord* имеет и средства для обработки документов, и прерастанные ошибки!

полученная благодаря интеграции, может сойти на нет из-за необходимости перезапуска пакета после каждого сбоя, и для создания даже простых документов иногда требуются достаточно нетривиальные обходы программных ошибок.

Оболочка, включающая возможность доступа ко всем частям пакета, предлагает выбор шаблонов документа, например, расчет ссуды или служебные записки. Она реализована неплохо, но было бы полезнее, если бы она убиралась подальше, когда приложение или документ уже выбраны. Все приложения следовало бы объединить в общем меню, например 'Office Apps', а не рассовывать их под разные заголовки, вроде Graphics или Office. Это симптом большой проблемы *KOffice* (помимо сбоев): мелкие ухищрения в ущерб основным задачам.

Появились некоторые долгожданные возможности, например, режимы вставки/замещения. Важный пункт – принятие формата OpenDocument: дело-то хорошее, но разработчики приняли этот формат, не позаботившись отладить фильтры, чтобы обработка выполнялась корректно. В тестовом документе из *OpenOffice.org*, например, текстовые врезки [text box] оказались не на своих местах и потеряли заголовки, а параграфы форматировались неправильно. Не самый удачный старт.

## Большой оригинал

Для многих пользователей текстовый процессор, *KWord*, является ключевым приложением. *KWord* содержит много хороших инструментов, демонстрирующих, что разработчики подумали о выборе вариантов для пользователя. Беда в том, что многие из них совершенно излишни. Процессор явно спроектирован программистами-теоретиками.

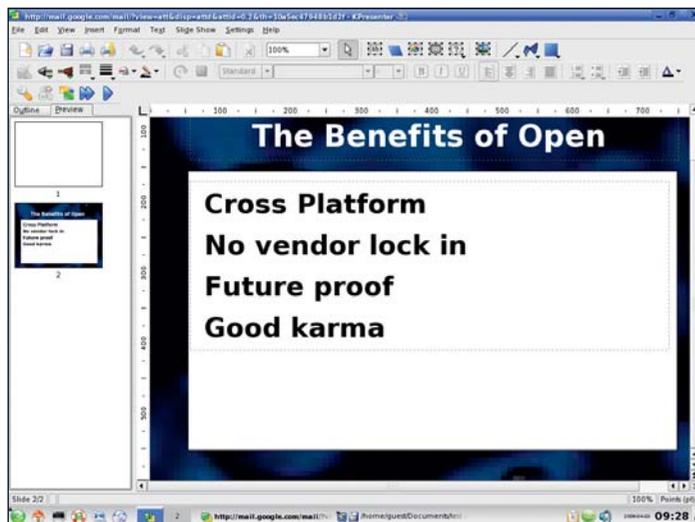
Попураживая *KWord* на множестве документов, мы пришли к выводу, что его свое-

обычность скорее раздражает, чем умиляет, и к тому же программа подвержена случайным сбоям – даже более частым, чем в версии 1.4. Нам всегда нравилась идея Редактора Выражений, который группирует всякие обращения, подписи и инструкции в меню Вставка; однако вряд ли эффективно проделывать мышью путь Вставка > Выражение > Личные > Привет! вместо написания Привет! в начале письма. Когда речь идет о номерах страниц или колонтитулах, это можно понять – но зачем же столько мучиться ради обращений, которые проще напечатать самому!

На нас произвела было впечатление идея иметь доступ ко всем инструментам рисования, используя объект *Karbon14*, однако *KWord* даже не имеет собственных инструментов для рисования линий. Если вам понадобилась жирная черта поперек страницы, придется или создавать объект и рисовать линию с помощью K14 (не слишком ли роскошно – вызывать векторный редактор, чтобы нарисовать черточку?) или добавлять таблицу и делать видимой одну из горизонтальных границ. Операция, которая не должна быть сложнее, чем протаскивание мыши от одного края страницы до другого, вызывает массу проблем, а уж если говорить о создании документов в коллективе с пользователями других пакетов, то горизонтальные линии, экспортированные из Word или *OpenOffice.org 2*, в *KOffice* не редактируются. Более того, экспорт документов со встроенными объектами ненадежен даже в родных форматах.

## KSpread слабоват

*KSpread* продвинулся вперед по сравнению с версией 1.4, но проигрывает и *OOo Calc*, и *Gnumeric* по возможностям и стабильности. В этой версии присутствуют заметные улучшения, например, возмож-



Где же источник? *KPresenter* нахально путает документы *OOo*.



**ПОЛ СЧИТАЕТ...**

«*KOffice* лучше всего сочетать с *OpenOffice.org*. Используйте *Kexi* и *Kivio*, а от всего остального держитесь подальше. Чем раньше *KWord* переписут с чистого листа, тем лучше!»

Разработчикам *KOffice* предстоит трудный выбор. Какие задачи в офисном пакете, например, должна выполнять *Krita*? Это хорошее многообещающее приложение, но большинство офисных работников (или домашних пользователей) обошлось бы коррекцией цвета, простыми эффектами, масштабированием и понятным инструментом создания текста. Вместо этого приложение пытается соперничать с *Gimp* или *Photoshop*, и чувствуется, что в офисном пакете ему не место. Новые возможности, безусловно, в первую очередь бросаются в глаза в программном обеспечении, но не следует забывать основное: 80% пользователей задействуют всего 20% задач. Загадочные опции типа вышеупомянутого Редактора Выражений не просто отнимают время разработчиков, но и усложняют интерфейс, не внося в пакет особого вклада.

В целом, *KOffice* проигрывает в сравнении с *OpenOffice.org* и *AbiWord/Gnumeric*, а его качество несравнимо хуже остальных интегрированных KDE-приложений (правда, документация написана очень понятно и снабжена множеством скриншотов).

Мы надеемся, что следующий основной выпуск *KOffice* будет радикально упрощен, как по количеству приложений, так и по количеству доступных возможностей. Слова «маленький, легкий, быстрый и стабильный» должны звучать как лозунг... **LXF**

**СОСТАВ KOFFICE**

- **KWord** Текстовый процессор с большими возможностями.
- **KSpread** Табличный процессор.
- **Kexi** Среда разработки для создания баз данных.
- **Kivio** Приложение для составления блок-схем.
- **Karbon14** Создание иллюстраций с приличной поддержкой SVG.
- **Krita** Небольшой аналог Photoshop.
- **KPlato** Новая система управления проектами.
- **KChart** Инструмент для построения графиков и диаграмм.
- **KFormula** Редактор формул.
- **Kugar** Генератор бизнес-отчетов.

ность проработать в программе больше 25 минут; добавлены также функции, перекрывающие большинство финансовых расчетов. Открытие таблиц с заполнением свыше 1000 ячеек вызвало трудности – как часто бывает и в *OpenOffice.org* – но с небольшими документами все было в порядке. Как и в версии 1.4, импортирование файлов из *Excel* не проходит гладко.

*KPresenter* имеет несколько хороших шаблонов, но опять-таки страдает нестабильностью, особенно при попытках сменить тему презентации или использовать гиперссылки для открытия внешних сайтов. Как и *KWord*, это приложение неохотно импортирует документы. Например, простая презентация в ODP (созданная в *OpenOffice Impress*) потеряла информацию о прозрачности (что можно простить), а также маркеры, выделяющие пункты в списках (чего простить нельзя). Кроме того, программа не смогла корректно обработать предустановленные переходы (смены кадра) и интерполировать изображения как *OpenOffice.org* – в *Impress* они выглядели гладкими, а в *KPresenter* зазубренными.

Графическое приложение *Krita* также страдает частыми сбоями, особенно заметными при работе с корректировочными слоями. [Они хранят информацию о преобразованиях, применяемых к параметрам оригинала, таким как оттенок, насыщенность, контрастность, в то время как сами данные оригинала остаются неизменными. Это позволяет свободно экспериментировать с изображением, не опасаясь его испортить – прим.ред]. Увы, результатов увидеть нам не удалось, так как приложение норовит рухнуть через несколько секунд после попытки добавить такой слой.

Теперь о приятном: как это ни фантастично, *Krita* способна использовать цвета СМЮК – 8- и 16-битовый режимы; хороший задел на будущее. Цвет на основе значений СМЮК выбирать пока нельзя, но реализация довольно устойчивая и неплохая – преобразование 6-мегапиксельной фотографии из RGB в 16 бит СМЮК заняло 8 секунд (на Athlon 2500+ 512 МБ ОЗУ), против 4 секунд в *Photoshop*. Одно это делает *Krita* достойным партнером другим системам подготовки документов, например, *Scribus*.

*Kexi*, молодое приложение для работы с базами данных, тоже развивается хорошо. Этот первый .0 релиз стал стабильнее, улучшил поддержку изображений в формах и, если использовать дополнительный модуль, более надежен при работе с базами *Access*. Программа хорошо работает с небольшими проектами, а при работе с большими наборами данных (то есть содержащими тысячи, а не сотни записей) способна подключаться к *SQLite*, *PostgreSQL* или *MySQL*.

Между приложениями существует некоторая несогласованность. Например, в текстовом процессоре, на главной панели, рядом с установкой масштаба, находится иконка в виде лупы. По ее изображению и местоположению можно принять ее за инструмент для ручного увеличения – и в *Krita* очень похожая иконка для этого и предусмотрена – но, оказывается, это команда Поиск. Вдобавок в пользовательский интерфейс прокралось несколько грамматических ошибок – например, одно окно в *Krita* обозначено как 'restauration tool'.

**Тест по документам**

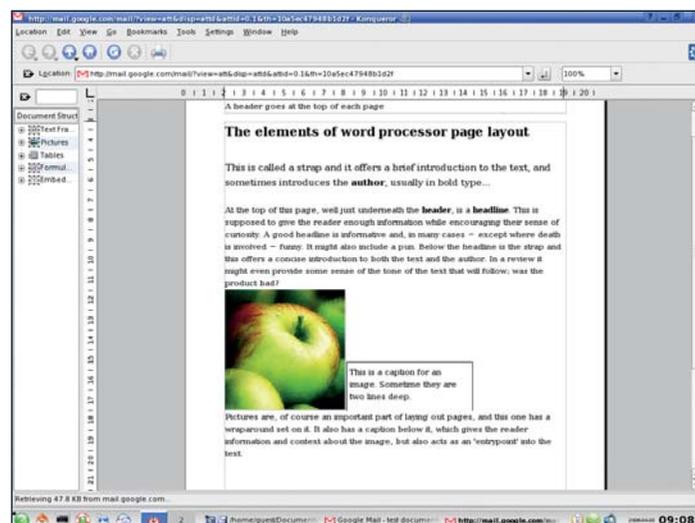
Офисное приложение обязано уживаться с программами-коллегами. Мы протестировали возможность *KOffice* работать с другими форматами. С помощью

Расширенного поиска Google мы нашли 100 документов (различных поддерживаемых форматов) и сравнили их с интерпретацией *KOffice*. Оказалось, что с текстовыми документами *KWord* справляется, но при легком усложнении документа (вставка графики или комментария) результаты будут воспроизведены не полностью, а чаще всего приложение рухнет, причем так происходит и с ODT-документами, созданными в *OpenOffice.org*. Большое разочарование, поскольку именно этот формат разработчики приняли в качестве родного.

*KSpread* показал себя лучше, так как открыл большую часть тестовых документов; однако еще остаются проблемы с открытием больших таблиц, а также таблиц, содержащих макросы и изображения. Этого нельзя сказать о *KPresenter*, который остается самым слабым звеном среди ключевых приложений. Импортирование PPT-файлов обычно ведет к краху программы, а когда документ все-таки удается открыть, вид его далек от исходного из-за утраты встроенных картинок. Файлы из *OpenOffice.org Presenter* импортировались немного лучше, но теряли жизненно важные части презентации. Лучшее, естественно, выглядели графические приложения: они работают с кроссплатформенными форматами файлов, такими как SVG и JPEG/PNG.

**Приложить усилия**

Если проект предполагает создание жизнеспособной альтернативы офисного пакета для среднего пользователя, то поработать есть над чем. В пакете имеются хорошие куски – например, mail merge, и даже отличные – способы использования переменных (все это в *KWord*). Релиз украшают *KSpread* и *Kexi*. Однако налицо некоторые проблемы, включая частые падения программы и хрошащий импорт/экспорт документов.



OpenDocument – родной формат пакета, но импорт ODT-файлов не идеален.

**ВЕРДИКТ LINUX FORMAT**

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	3/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	5/10
ДОКУМЕНТАЦИЯ	7/10

Наряду с инновационными идеями – проблемы стабильности и назначения приложений.

**РЕЙТИНГ 4/10**

●●●●○○○○○○○○

АУДИОПЛЕЙЕР

# Amarok 1.4

Грэм Моррисон удивлен шумихой, вызванной обычным обновлением классики.

## САМОЕ ГЛАВНОЕ

Воспроизводит музыку, загружает нужные сведения и принимает Интернет-радио. Аналоги: Juk или XMMS.

- **РАЗРАБОТЧИК:** Amarok Development Squad
- **САЙТ:** <http://amarok.kde.org>
- **ЦЕНА:** Бесплатно по лицензии GPL



Когда *Amarok* появился, в июле 2003, все его свойства можно было пересчитать по пальцам ваших рук. Теперь же при таком способе подсчета вам придется собирать гостей, чтобы охватить хотя бы малую часть списка.

*Amarok* был создан на замену старым аудиоплеерам, входившим в состав KDE. Название одолжили у альбома Майка Олдфилда [Mike Oldfield], а интерфейс сконструировали из двух окон: окна плеера (в стиле достопочтенного XMMS) и плей-листа. Это был умный, удобный в использо-

достижение, если припомнить, сколь жалок встроенный звуковой драйвер KDE. Среди движков, GStreamer – самый яркий: в своё время он несомненно был судьбоносным для звука в Linux. В новейшей версии *Amarok*, GStreamer 0.10 обзавёлся ещё и дополнительным эквалайзером.

## Бесшовное воспроизведение

Другой движок – Xine, верный соратник Linux – теперь способен на бесшовное воспроизведение. Удивительно, что программа для проигрывания одной музыкальной дорожки сразу после другой, без паузы, разрабатывалась так долго; но наконец она есть, и это здорово. Ещё два движка появились благодаря таинственно звучащим аббревиатурам, NMM и MAS (два конкурирующих стандарта прослушивания аудио по сети). Пользователи SUSE обнаружат, что заработал Helix, движок *RealPlayer 10*, а новейшая версия *Amarok* умеет делать плавный переход (постепенные затухание

## «НАША 650-МГЦ МАШИНА НАМЕРТВО ЗАВИСЛА...»

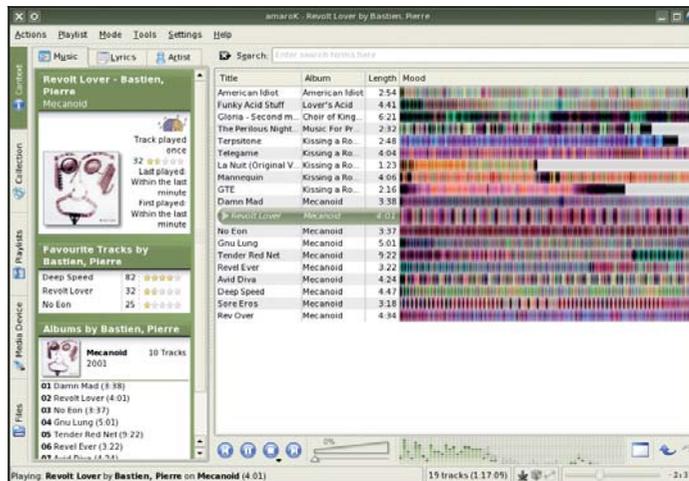
вании плей-лист, который и стал основным козырем *Amarok*. Добавление и удаление файлов не отвлекало от прослушивания музыки, дорожки можно было запускать, приостанавливать и переключаться между ними в одном и том же окне... у пользователей KDE *Amarok* сразу стал фаворитом.

«Перемотав» два года, *Amarok* превратился в один из самых тяжеловесных музыкальных плееров для всех платформ – вызывающим в равной мере похвалы и изумление. Ядро программы умеет воспроизводить аудио с помощью нескольких разных движков; это замечательное

и нарастание звука) между звуковыми дорожками.

Глупо было бы перечислять все подряд – скажем только о наиболее эффективных нововведениях, а главное из них – превосходный контекстный браузер.

О тех, кто сотрясает ваши барабанные перепонки, браузер сообщит вам даже больше, чем вы хотите знать. Окно с тремя вкладками покажет вам: обложку альбома, время последнего прослушивания дорожки, слова песни и информацию об исполнителе, найденную в Wikipedia. Слова песен разыскиваются на специальных сайтах с помощью скриптов; правда, поиск может быть как успешным, так и не очень. К счастью, в *Amarok* есть встроенный редактор для вво-



У каждой дорожки – свой «индикатор настроения», но что значит эти цвета? Классический хор выглядит суматошное пивной дискотеки «Tuborg»...

да слов вручную, текст можно сохранить как метаданные в самом аудиофайле.

За контекстным браузером следует ещё одна понравившаяся нам особенность *Amarok*: это тесная связь с Last FM, Интернет-радиосайтом, прежде известным как Audioscrobbler. Вы можете счесть *Amarok* ябедой, но после создания вами учётной записи на Last FM он сообщит о ваших музыкальных пристрастиях на сервер сайта, где на основе ваших плей-листов для вас будет создан музыкальный профиль. Сервер сравнивает его с другими профилями и выдвигает предположения о том, каких еще исполнителей вы хотели бы услышать, а заодно снабжает вас контактами с людьми похожих вкусов. *Amarok* может использовать полученные данные для пополнения своего контекстного браузера сведениями о ваших любимых исполнителях.

*Amarok* не был бы собой без визуальных украшений. Некоторые скачущие, ползущие и вертящиеся видеоэффекты (в основном заимствованные из *XMMS*) выглядят очень неплохо. Зато нововведённый «индикатор настроения» (mood bar) достоин премии за бесполезность. Это попытка описать характер аудиофайла полоской в плей-листе, цве-

та которой зависит от изменения частоты звука во времени. Идея состоит в том, что вы сможете отличить «тяжёлый металл» от солидного «4'33"» Джона Кейджа [John Cage]. Мы – не смогли.

В *Amarok*, как в капле воды, отразились различия между KDE и Gnome. Он набит функциями, которые большинству людей не понадобятся никогда, поэтому совершенно непригоден для «пожилых» компьютеров – последняя версия намертво завесила нашу 650-МГц тестовую машину, а это уже перебор. **LXF**

## ВЕРДИКТ LINUX FORMAT

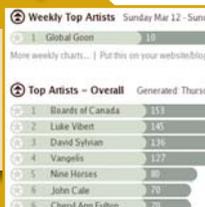
ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	5/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	6/10
ДОКУМЕНТАЦИЯ	7/10

*Amarok* до того велик, что требует отдельного Linux-дистрибутива, и всё-таки он нам нравится.

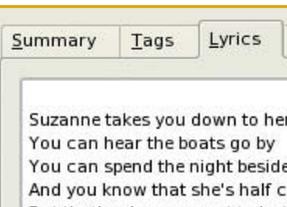
**РЕЙТИНГ** **7/10**

## В ДВУХ СЛОВАХ

Создайте свой топ-парад музыкантов благодаря Last.FM.



Спишите для себя тексты песен. Эй, кто на караоке?



*Amarok* поддерживает iPod и много стандартных устройств.



КЛИЕНТ МГНОВЕННОГО ОБМЕНА СООБЩЕНИЯМИ

# Gaim 2.0

Благодаря упорной работе 15 победителей Summer of Code, появлению новых возможностей и дизайна, Gaim достиг настоящей *Kompetentности* даже в сравнении с конкурентами. Рассказывает Пол Хадсон.

## САМОЕ ГЛАВНОЕ

Клиент мгновенного обмена сообщениями, поддерживает множество различных сетей. См. также: Kopete и Skype.

- **РАЗРАБОТЧИК:** Команда разработчиков Gaim
- **САЙТ:** <http://gaim.sourceforge.net>
- **ЦЕНА:** Бесплатно по лицензии GPL

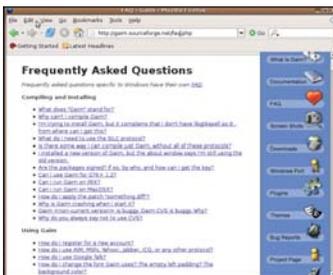


Проект Google Summer of Code (SoC) принес сообществу Open Source миллионы долларов, а в выигрыше оказался *Gaim*: программа быстрого обмена сообщениями на основе *GTK*. Он поддерживает больше сетей, чем самый навороченный мобильный телефон. А до «Лета за Кодом» процесс его разработки находился почти в замороженном состоянии, особенно по сравнению со стремительным развитием его ближайшего конкурента, *Kopete* для KDE.

Пятнадцать студентов и руководителей работали над 15-ю новыми и нужными возможностями, вошедшими почти в полном составе в версию 2.0. Было сделано много других улучшений, и этот релиз *Gaim* достоин восхищения. Правда, не чрезмерного: не реализована нормальная поддержка голоса и видео.

## Новые возможности

Последний крупный релиз *Gaim* был 18 месяцев назад. За это время команда разработчиков попыталась внедрить долгожданные изменения. Во время бета-тестирования опции включались либо нет по следующему принципу: «Удалим ее и посмотрим, будут ли жалобы». Опцию восстанавливали, если жалоб хватало, поэтому несмотря на большой скачок *Gaim 2.0*



Документация к *Gaim* все еще имеется только в сети и только в виде длинного списка часто задаваемых вопросов. Доколе?

## ГОЛОС И ВИДЕО

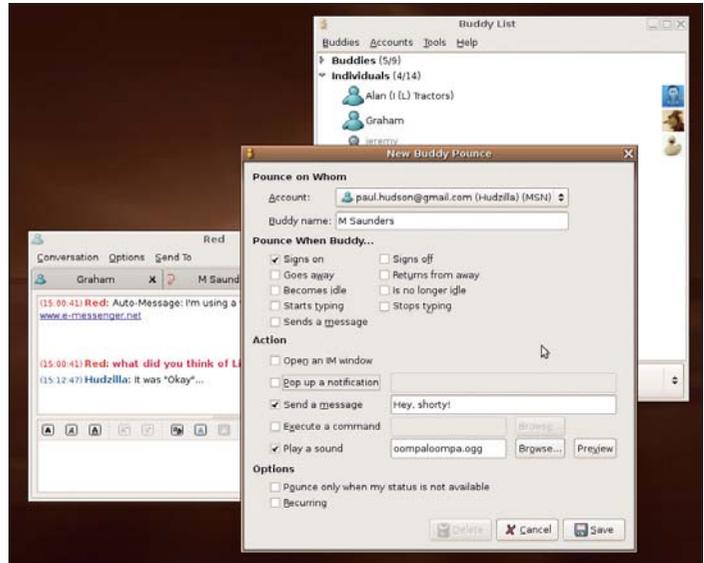
Если хотите всех обогнать, сконфигурируйте *Gaim* с помощью ключа `--enable-vv`, и получите поддержку голоса и видео. Мы находим ее крайне слабой, поэтому используйте последнюю CVS-сборку для увеличения шансов успешно поболтать!

вперед в плане функциональности, диалог Preferences (Свойства) не ошеломляет количеством возможностей.

Так что же изменилось на самом деле? Начнем с главного: поддержка Bonjour (ранее известная как Rendezvous), интеграция с D-BUS, поддержка Mono и Perl, доска объявлений, IRC через SSL, поддержка UPnP (Universal Plug and Play) и другие. Фактически, сделано чересчур много, поэтому неудивительно, что некоторые функции работают ненадежно. Например, *Yahoo Doodle* хорошо работает между пользователями *Gaim* и *Yahoo* взаимодействия добиться не получается.

Поддержка UPnP работает отлично: мы даже не заметили, что она есть – и это правильно. Для тех, кто не в курсе, UPnP – протокол Microsoft, позволяющий вашему компьютеру общаться с роутером и автоматически устанавливать перенаправление портов (port forwarding) на основе запросов. Это важный пункт для быстрого и простого обмена файлами, когда требуется открывать произвольные порты в системе. Без UPnP вам пришлось бы превращать свой брандмауэр в швейцарский сыр или вообще посылать файлы электронной почтой.

Поддержка Bonjour – это хорошо (он посылает широковещательные запросы с целью обнаружить всех использующих *iChat* от Apple или другой Bonjour-совместимый клиент и завязать с ними беседу), но нам особенно полюбилась поддержка D-BUS. D-BUS – будущее межпроцессного взаимодействия в Linux. Она уже интегрирована в Beagle, CUPS, HAL, udev и другие проекты. Предыдущие выпуски *Gaim* использовали индивидуальную систему протоколов, никем другим, похоже, не употребляемую, а поддержка D-BUS в новой версии означает, что *Gaim* может взаимодействовать с любой другой D-BUS-совместимой программой. Для начала, новый скрипт *Gaim-remote* предлагает несколько



Выскакивать на экран – хороший способ позлить друзей и семью.

базовых опций, доступных из командной строки – отправка сообщения, добавление собеседников и так далее.

## Gaim, подстройка

Если новые возможности не убедили вас выбрать *Gaim*, то обратите внимание на искусный интерфейс программы. Диалог Preferences (Свойства) тщательно пересмотрен, улучшена организация просмотра журналов, новый диалог Plugins позволяет активировать расширения без перезапуска программы, а Buddy Pounce уведомляет вас, когда появляется особо ценный товарищ. Кроме визуальных изменений, произошли изменения в звуках. Раньше ужасные звуки *Gaim 1.5* заставляли людей выключать колонки, а теперь они добавляют приятности в финальный релиз.

Судя по возможностям, *Kopete* выигрывает у *Gaim*, причем отрыв растет: последние CVS-сборки *Kopete* уже имеют в своем составе работающую поддержку Jingle (см. *LXF78*, Что такое... Jingle). Но если сравнивать эти программы по стилю и удобству использования, то победит *Gaim*, имеющий изящный GUI, ясные иконки и превосходную поддержку обмена сообщениями из скриптов. Обмен мгновенными сообщениями – одно из новых направлений в Linux, и за наше внимание борются два действительно стоящих проекта, а в результате мы будем избалованы выбором. **LXF**

## ПОДДЕРЖКА СЕТЕЙ

*Gaim* поддерживает следующие протоколы:

- AIM
- Gadu-Gadu
- GroupWise Messenger
- ICQ
- IRC
- Jabber
- MSN
- SILC
- Yahoo
- Zephyr

## ВЕРДИКТ LINUX FORMAT

ВОЗМОЖНОСТИ	7/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	10/10
ДОКУМЕНТАЦИЯ	7/10

Клиент превзошел *Gaim 1.5* по всем параметрам и на равных спорит с *Kopete*.

## РЕЙТИНГ 8/10



СИСТЕМА УПРАВЛЕНИЯ ВЕРСИЯМИ

# Seapine Surround SCM 4.1

Кажется, кто-то забыл сделать check-in нововведений в этом выпуске, подозревает Пол Хадсон.

**САМОЕ ГЛАВНОЕ**

Проприетарная система управления версиями с приятным GUI. Аналоги: *Subversion*, *BitKeeper*, *Bazaar-NG*.

- **РАЗРАБОТЧИК:** Seapine
- **САЙТ:** www.seapine.co.uk
- **ЦЕНА:** 395 фунтов стерлингов за именную или 995 за плавающую лицензию

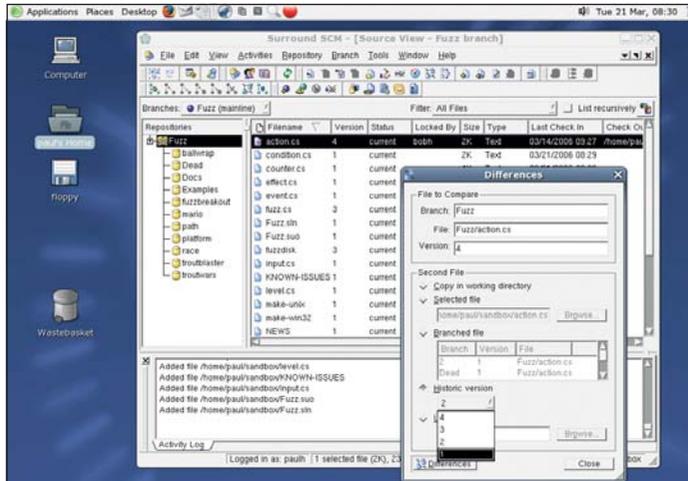


SourceForge недавно перешла от CVS к *Subversion* для управления своим репозитарием исходных текстов. Этот шаг сделал *Subversion* номером один среди систем управления версиями. Конечно, есть другие системы: *BitKeeper* сохранил популярность даже после фиаско с ядром Linux; новичок *Bazaar-NG* уже задействован в нескольких проектах; и конечно же, проект Git, детище команды разработчиков ядра, достиг зрелого состояния.

Однако рынок систем контроля версий не ограничен одними открытыми программами, о которых мы слышим каждый день. Мы следим за развитием проприетарной системы контроля версий *Seapine Surround SCM* с ее первого выпуска: многообещающее приложение, которое было бы превосходным, если бы разработчики добавили в него пару ключевых возможностей, расцвело в... опять-таки многообещающее приложение, которому не хватает все тех же возможностей.

**Большая разница**

Последний раз мы обсуждали *Surround SCM* версии 3.0 в *LXF61*. С тех пор было



GUI Surround умеет делать мышью сравнения, слияния и ветвления, но нам это скоро надоело, и мы вернулись в свою IDE.

добавлено несколько интересных возможностей, включая поддержку WebDAV (чтение и запись репозитория через веб-сервер), поддержка Unicode и усиленная интеграция с различными IDE. Особенно хороша поддержка WebDAV, так как

мось устанавливать *Surround*-клиент.

Поддержка IDE пригодится вам лишь в случае, если ваша среда поддерживается программой. В список поддерживаемых IDE входят *Visual Studio*, *Dreamweaver*, *WebSphere*, *CodeWarrior*, *IntelliJ IDEA*,

ций. Например, если вы хотите зафиксировать несколько файлов, причем изменения произошли только в малой их части, то *Surround SCM* будет неэффективно использовать ваш канал: перекинет все файлы на сервер, а потом заявит, что в большинстве файлов изменений нет. Реализации примитивнее и придумать нельзя, поэтому вы вправе удивляться, чего ради надо приплачивать к 395 фунтам за честь его пользоваться.

Кстати о цене. Да, программа стоит 395 фунтов. Прибавьте к этому НДС, а также обязательный единовременный платеж в виде 20% от цены за дальнейшие обновления и поддержку, и все это только за именную лицензию (то есть выданную конкретному разработчику). Если хотите плавающую лицензию, то готовы 995 фунтов + НДС + единовременный платеж. Для сравнения, пользователям Windows две полных копии *Microsoft Visual Studio 2005* обойдутся дешевле, чем одна копия *Seapine Surround* с именной лицензией – причем у MS не самые низкие цены.

*Surround SCM* имеет приятный пользовательский интерфейс, но ничего более. Однако новизна GUI меркнет, как только вы осознаете, что никому не нужно отдельное приложение для задач, которые и без того выполняются в вашей IDE: *Eclipse*, например, уже имеет первоклассную систему *Subversion*. Во времена, когда царил CVS, *Surround* явно тянулась в лидеры. Но с приходом *Subversion* остался единственный выбор, и это не *Seapine Surround*. **LXF**

«АТОМАРНОСТЬ ОПЕРАЦИЙ ОБЕСПЕЧИВАЕТСЯ ЛИШЬ НА УРОВНЕ ОТДЕЛЬНЫХ ФАЙЛОВ.»

Windows, Mac OS X и Linux могут читать и осуществлять запись на серверы WebDAV, что во многих случаях устраняет необходи-

мость устанавливать *Surround*-клиент. Поддержка IDE пригодится вам лишь в случае, если ваша среда поддерживается программой. В список поддерживаемых IDE входят *Visual Studio*, *Dreamweaver*, *WebSphere*, *CodeWarrior*, *IntelliJ IDEA*,

**ЗАЧЕМ НУЖЕН SUBVERSION?**

*Subversion* использует особую систему транзакций: файлы, которые выгружаются в репозиторий, недоступны другим пользователям до тех пор, пока не будут зафиксированы изменения в последнем из них. Например, мы фиксируем три файла, а в это время кто-то другой пытается получить доступ к одному из них — тогда *Subversion* отправит ему старую версию файла без внесенных нами изменений. Таким образом обеспечивается целостность проекта, независимо от числа разработчиков и их месторасположения.



Пользователь слева выгружает три файла в Surround, а пользователь справа загружает их до подтверждения изменений – чревато разрушением проекта.

**Провал Seapine**

Новые возможности неплохи, но как насчет старых проблем? Для начала, *Surround SCM* все еще поддерживает атомарность выгрузки в репозиторий отдельных файлов, но не задания в целом, так что кто-то другой вполне может загрузить некорректную версию ПО, пока вы фиксируете свои изменения. (см. врезку *Зачем нужен Subversion?*)

Программа продолжает игнорировать потребность в работе без подключения к сети: постоянное общение с сервером идет даже при выполнении простейших опера-

**ВЕРДИКТ LINUX FORMAT**

ВОЗМОЖНОСТИ	5/10
СКОРОСТЬ	4/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	6/10
ОПРАВДАННОСТЬ ЦЕНЫ	1/10

То, что оно работает под Linux, еще не повод его использовать.

**РЕЙТИНГ** **4/10**

●●●●○○○○○○○○

ДИСТРИБУТИВ LINUX

# Damn Small Linux 2.3



Хотите реанимировать ПК из девяностых? Майк Сондерс изучает дистрибутив объемом всего 50 мегабайт...

**САМОЕ ГЛАВНОЕ**

Наилегчайший дистрибутив, призванный вдохнуть жизнь в старые машины. Аналоги: Puppy Linux и Feather Linux.

- **РАЗРАБОТЧИК:** Команда DSL
- **САЙТ:** [www.damnsmalllinux.org](http://www.damnsmalllinux.org)
- **ЦЕНА:** Бесплатно по лицензии GPL



Давно ли гигабайтов было немаленько... кто ж их сожрал? Вот и верь после этого мнению, что

Linux нетребователен к ресурсам. Но хотя аппетиты SUSE или Red Hat на оперативную и прочую память не уступают Windows XP, это далеко не весь мир Linux. Модульное строение Linux и его открытая для совершенствования графическая оболочка позволяют работать буквально на бросовом оборудовании.

Конечно, байтоедов вроде Gnome, KDE или OpenOffice.org на слабых машинах жить не заставишь, но есть ведь и другие способы выполнения их задач. В этом и состоит задача Damn Small Linux: взять наилегчайшие настольные приложения, собрать из них добротное настроенный дистрибутив и удержать системные требования на минимальном уровне. Результат? Полноценная система Linux, довольствующаяся всего 50 МБ и уживающаяся на древностях вроде 486 с 16МБ ОЗУ.

Damn Small Linux задуман отнюдь не как первый порт приписки для новичков в Linux, да и не как суперскоростная замена распространённым настольным дистрибутивам. За ее основу приняли Кнопприх, унаследовав красочные загрузочные сообщения и прочие зрелые черты Debian, сделавшие Кнопприх звездой.

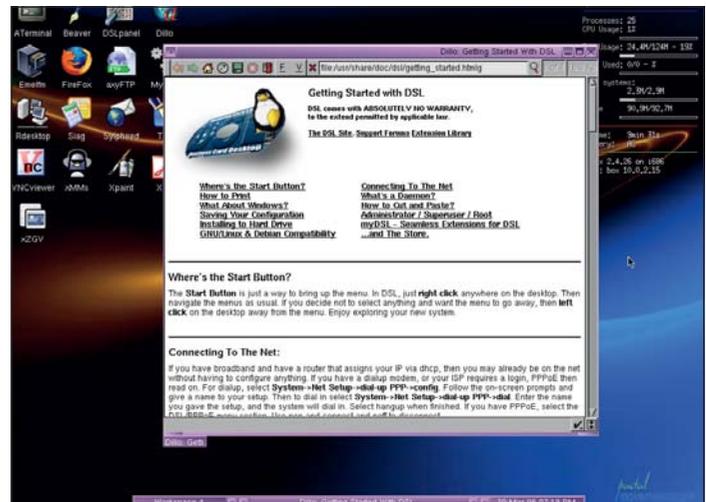
Загружается Damn Small Linux – Чертовски Компактный Linux – Чертовски

Быстро. На нашем Pentium-III, 650МГц, 128МБ ОЗУ, DSL загрузился с Live CD за 85 секунд – существенно быстрее, чем SimplyMepis на той же машине (4 минуты 32 секунды). Затем мы урезали ОЗУ до 32 МБ и установили DSL на винчестер: загружается за 58 секунд, выдающееся достижение для столь медленного железа. Сравните: Ubuntu на машине с 1,5 ГГц и 384МБ ОЗУ для загрузки понадобилось 108 секунд, а DSL с Live CD на той же машине хватило 38.

Чтобы Live-дистрибутив целиком уложился в оперативную память, понадобится 128 МБ, несколько не в русле минимальных системных требований; зато отсюда можно инсталлировать на жесткий диск или USB-накопитель, где достаточно всего 16 МБ. Для комфортной работы большинства прилагаемых программ рекомендованы 32 МБ, но эти цифры по зубам почти всем компьютерам, изготовленным за последнее десятилетие.

## Скромный рабочий стол

Опрятный рабочий стол, открывающийся перед вами после загрузки – результат сочетания быстрейшего оконного менеджера Fluxbox и файлового менеджера Emelfm. Замечателен экран помощи, он появляется после загрузки и объясняет, как настроить сеть и принтер и инсталлировать систему на винчестер, а заодно указывает, где находится кнопка «Пуск» – очень удобно! Контрольная Панель, объединяющая массу графических инструментов настройки, скромна, но функциональна – почти весь дистрибутив можно настроить с помощью диалоговых окон в стиле «point-and-click» («наведи и нажми»), хотя установка на винчестер потребует некоторых навыков обращения с командной строкой.



Рабочий стол Damn Small Linux под управлением Fluxbox – привлекателен, удобен и не падок на системные ресурсы.

Чтобы понять, как разработчики DSL умудрились заставить дистрибутив работать на 16 МБ ОЗУ, достаточно глянуть на использование памяти приложениями по умолчанию. Почтовый клиент Sylpheed обходится 4,3 МБ (KMail необходимо 24 МБ); текстовому процессору Ted нужно 4,1 МБ (Ooo Writer занимает 54 МБ); а Emelfm потребляет лишь 3,1 МБ, не сравнить с аппетитом Nautilus (14 МБ). Siag для электронных таблиц, XMMS для проигрывания музыки и ассортимент прочих столь же скромных программ, всё это на базе ядра 2.4.2 и X.org 6.7.0.

Установив DSL на жёсткий диск, вы можете расширить его за счёт Deb-пакетов из запасников Debian (есть там и специальные расширения MyDSL). Разработчики DSL не планируют разрастание дистрибутива сверх 50 мегабайт, с тем расчётом, чтобы его всегда можно было записать на малый компакт-диск.

Если вы уже имели дело с прежними версиями DSL, то обратите внимание на нововведения: это автоматическая загрузка расширений MyDSL, новая тема, способность полностью загружаться в оперативную память с USB-накопителей и усовершенствованная документация. Гибкость Damn Small Linux заслуживает отдельного описания. Работает ли он с Live CD, USB-накопителя или из эмулятора, малые размеры DSL способствуют мирному сосуществованию с другими ОС. Возможен даже такой вариант: приобретение USB-брелоч-

ка с новейшей версией системы вместе с Qemu для эмулированной загрузки из-под Windows!

За последние пять лет мы повидали бесчисленное множество «мини-дистрибутивов». Подавляющее большинство из них были просто безжалостно урезанными версиями «старших» собратьев, без малейших попыток облегчить обращение с ними или гармонизировать рабочую среду. Разработчики DSL искусно подогнали все части друг к другу без помощи Gnome или KDE, и хотя удобства а ля Mandriva достичь не удалось – не возникает ни впечатления полной беспомощности, ни ощущения разлаженности, как у прочих «мини-соперников» типа Puppy Linux. DSL – полноценный дистрибутив, не раздутый излишествами, как его старшие сородичи. Возьмите систему с нашего диска, и покрывшийся было паутиной ПК образца 1995 года снова улыбнётся вам. **LXF**



Контрольная панель DSL, может, и неказиста, зато функциональна, в отличие от прочих «мини» с их текстовыми настройками.

## ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	5/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	6/10
ДОКУМЕНТАЦИЯ	8/10

Великолепное решение для старых PC. Поразительно, сколько достоинств уместилось в этом крошечном дистрибутиве.

**РЕЙТИНГ 8/10**



ПЕРСОНАЛЬНЫЙ ФИНАНСОВЫЙ УЧЕТ

# Moneydance 2006

Грэм Моррисон теперь учитывает все до последнего пенни с помощью кросс-платформенной программы, являющейся соперником *GnuCash*.

**САМОЕ ГЛАВНОЕ**

GUI-приложение, поможет сбалансировать ваши доходы и расходы. Аналоги: *GnuCash* и *KMyMoney*.

- **РАЗРАБОТЧИК:** Reilly Technologies
- **САЙТ:** www.moneydance.com
- **ЦЕНА:** \$29,99



Управлять своими финансами не менее увлекательно, чем смотреть старые сериалы о семейном наследии. Заставить длинный список цифр понравиться кому-нибудь, кроме бухгалтеров – задача сложная, но авторы *Moneydance* находятся на правильном пути. Если название программы дано ей неспроста, оно вселяет надежду, что баланс можно рассчитать приплясывая.

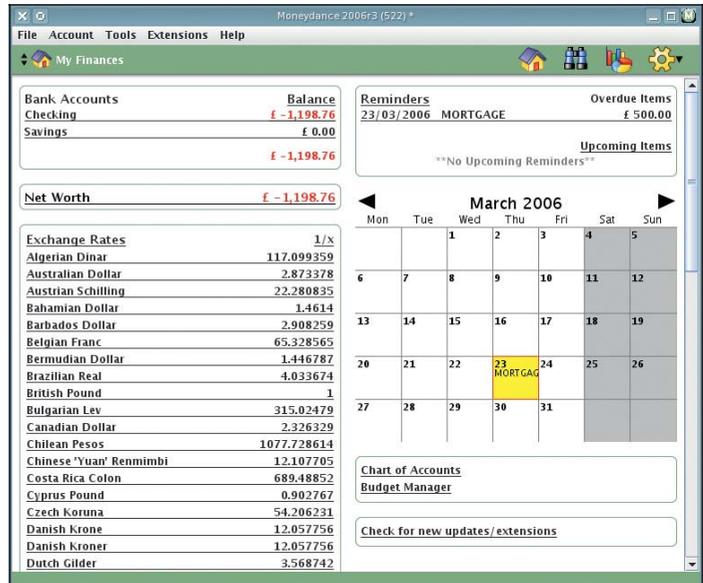
*Moneydance* написана на Java и по умолчанию ставит себя в каталог /opt. Первое, что вы видите при запуске программы – окно My Finances. Оно похоже на *Evolution* и *Kontact* с их большим календарем и напоминаниями. Вверху слева находится итог ваших счетов, а сразу под ним расположился чудной список из 40 валют. Это отнюдь не обновляющаяся через сеть сводка, а постоянная информация; более того, в списке содержатся валюты, уже вышедшие из обращения. Будь *Moneydance*

нашим советником по финансам, мы бы забеспокоились.

**Важные данные**

Чтобы начать пользоваться *Moneydance*, потребуется внести в нее определенные данные. Базы *GnuCash*, самой известной программы управления финансами, вы сможете автоматически импортировать, только преобразовав их через *GnuCash* в формат QIF или OFX. Также существует превосходный сторонний скрипт на Perl – *migrate.pl* – преобразующий базы *GnuCash* в более понятный *Moneydance XML*.

Однако в основном вы будете получать файлы данных о транзакциях у банка и импортировать в программу. Предпочтительным форматом является QIF (Quicken Interchange Format), и хотя он немного устарел, его поддерживают и используют многие организации. Он также будет вашим единственным выбором при первом запуске *Moneydance* – так как QIF содержит достаточно информации для *Moneydance*, чтобы сделать необходимые предположения о типе создаваемого счета (типов в программе 8). Банк является основным счетом для ежедневного использования, существуют также счета для кредитных карт, основного капитала и займов. Нас расстроило отсутствие сче-



Страница My Finances показывает ваш счет в Игре Жизни. Длинный список валют включает ныне покойные бельгийские франки...

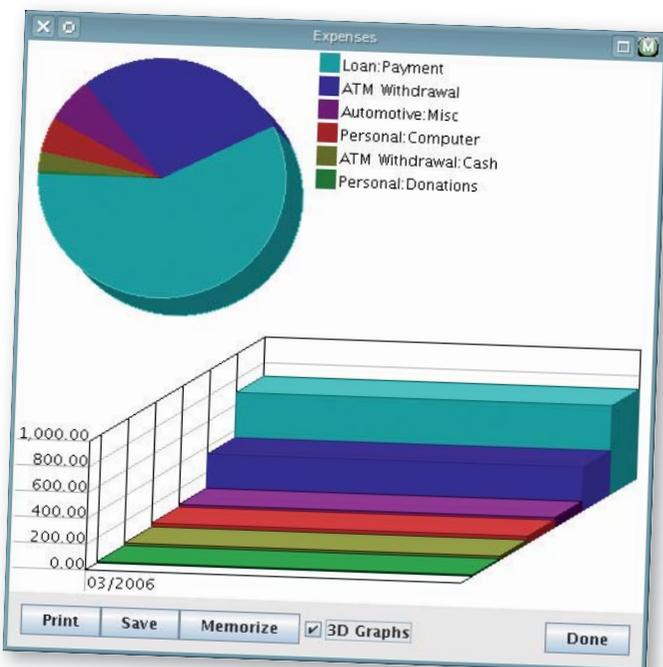
та для Закладной, однако в принципе ее функции перекрывает тип Заем.

Создав счет, вы сможете импортировать данные OFX (Open Financial Exchange). Этот формат менее распространен, чем QIF, но предлагает дополнительные возможности, например, перенаправление потоков данных – то есть можно получать данные прямо из банка. Проблема только в том, что поддерживаемые банки, из числа способных на расчеты в реальном времени онлайн, в основном находятся в США. Мы не обнаружили в списке *Moneydance* ни один банк из Великобритании. Но если вам известно, что ваш банк работает с другими финансовыми приложениями, например, *Quicken* или *Microsoft Money*, то попробуйте настроить его вручную, он имеет шансы заработать и с *Moneydance*.

Ваши транзакции отображаются в нижней панели. Каждой из них надо будет присвоить категорию, а также сопоставить ее с другими используемыми вами счетами. Это тот же метод двойной записи, который использует *GnuCash*, только в *Moneydance* он реализован немного лучше.

Имеется, однако, большое упущение, а именно – отсутствие использования гибкой логики при назначении категорий транзакциям. *GnuCash* пытается отгадать категорию, в зависимости от прошлых выборов для похожих транзакций, это реальная экономия времени. В *Moneydance* вы должны

трудолюбиво назначать категорию каждой транзакции. Этот, а также другие небольшие недочеты, означают, что хотя *Moneydance* имеет более продуманный пользовательский интерфейс и лучшую документацию по сравнению с открытым аналогом, единственная причина, по которой стоит выбрать *Moneydance*, это поддержка вашим банком расчетов в реальном времени. **LXF**



Графиков в Moneydance меньше, чем в GnuCash, но они получше.

**ВЕРДИКТ LINUX FORMAT**

ФУНКЦИОНАЛЬНОСТЬ	5/10
ПРОИЗВОДИТЕЛЬНОСТЬ	6/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ОПРАВДАНОСТЬ ЦЕНЫ	7/10

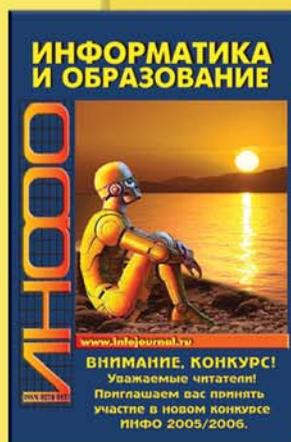
Подходит для онлайнных расчетов (в США), но Moneydance отстает от GnuCash по возможностям и цене. Хотя составляет красивые отчеты.

**РЕЙТИНГ** **6/10**

●●●●●●●●○○○○

# Научно-методический журнал «ИНФОРМАТИКА И ОБРАЗОВАНИЕ»

Учредители — Российская академия образования,  
издательство «Образование и Информатика»



12 номеров в год

## ОСНОВНЫЕ РУБРИКИ ЖУРНАЛА:

Общие вопросы. Информатизация школы.  
Методика. Задачи. ИКТ в образовании.  
Педагогический опыт. ИКТ в предметной области.  
Зарубежный опыт. Информатика в начальной школе

## ПОДПИСНЫЕ ИНДЕКСЫ

в каталоге «Роспечать»:

70423 — для индивидуальных подписчиков;

73176 — для предприятий и организаций;

в объединенном каталоге «Пресса России» — 26097

Приложение к журналу «Информатика и образование»

## «ИНФОРМАТИКА В ШКОЛЕ»



8 тематических  
номеров в год

## ПОДПИСНЫЕ ИНДЕКСЫ

в каталоге «Роспечать»:

81407 — для индивидуальных подписчиков;

81408 — для предприятий и организаций;

в объединенном каталоге «Пресса России» — 45751

## Б И Б Л И О Т Е К А

журнала «Информатика и образование»



6 выпусков в год

**Зубрилин А. А. Сборник задач по непрерывному курсу информатики. Часть 1**

Подписной индекс в каталоге «Роспечать» — 46411

**Зубрилин А. А. Сборник задач по непрерывному курсу информатики. Часть 2**

Подписной индекс в каталоге «Роспечать» — 46412

**Проект на уроках информатики**

Подписной индекс в каталоге «Роспечать» — 46673



# Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.

## СМЕНА СОСТАВА



**Ладислав Боднар**  
основатель,  
начальник и  
сотрудник сайта  
DistroWatch.com

Уход из Mandriva Гаэля Дювала (Gaël Duval) заставил меня задуматься: если вы сегодня начнете делать дистрибутив Linux, каковы шансы, что вы всё ещё будете его лидером или разработчиком, скажем, лет через

10? Судя по истории популярных дистрибутивов Linux, эти шансы невелики. Вспомним историю Debian. Стартовав в 1993 году, Debian очень быстро стал одним из крупнейших проектов совместной разработки программного обеспече-

## «ИЭН МЭРДОК ТАК И НЕ УВИДЕЛ РЕЛИЗА DEBIAN»

ния в Интернете. Однако, его основатель, Иэн Мердок (Ian Murdock), так и не увидел первого релиза. К моменту анонсирования первой стабильной версии проекта в июне 1996-го Мердок уже его покинул.

Похожая история произошла и с Gentoo. Даниел Роббинс, запустивший Gentoo в 1999-м под именем Epoch Linux, уволился в апреле 2004-го по личным обстоятельствам. Затем он немного поработал на Microsoft, после чего присоединился к небольшой фирме в своём родном городе Альбукерке, штат Нью-Мексико.

Коммерческие дистрибутивы — не исключение: основатели Red Hat, SUSE и, не так давно, Mandriva, все либо покинули, либо вынуждены были покинуть свои проекты.

Так есть ли ещё хоть один из известных дистрибутивов, чей основатель не оставил его в поисках более тучных пастбищ? Конечно, Патрик Фолкердинг (Patrick Volkerding) продолжает быть (по большому счету, единственным) разработчиком Slackware Linux, самого старого из уцелевших дистрибутивов. Хотя почтенный Slack за свои годы пережил и взлёты и падения, похоже, его создатель и хранитель сдаваться не собирается.

# ЛУЧШИЙ СТАНОВИТСЯ ЛУЧШЕ

В дистрибутив **Knoppix 5.0** Live добавили Mono, Java и ограниченную поддержку Gnome.

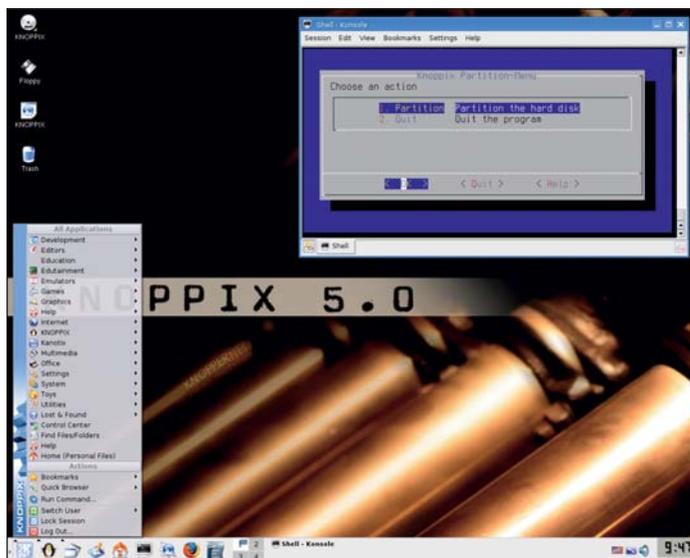


## Разработчики Knoppix 0-дистрибутива, первым предложившего концепцию Live CD около шести лет назад — выпустили крупное обновление, версию 5.0.

В отличие от ранних дней дистрибутива, которые характеризовались частыми, но мелкими релизами, содержащими исправления ошибок и добавляющими функциональность, сегодняшний более спокойный цикл является признаком вызревания Knoppix в профессиональный инструмент, пригодный для широкого круга задач, например, демонстрации возможностей Linux или восстановления файлов.

Возможно, наиболее впечатляющие изменения — это добавление X.org 6.9.0, переход на Udev для задач монтирования устройств хранения данных и обновление многих приложений. Java (и относящиеся к Java пакеты) и Mono также включены в дистрибутив впервые. С новой темой рабочего стола дистрибутив выглядит гораздо профессиональнее, чем предыдущие версии, а внутренние изменения привели к более быстрому процессу загрузки и более отзывчивому десктопу.

Как и Knoppix 4.0, дистрибутив вышел



Улучшенный инсталлятор поможет обновить старый Knoppix.

в двух редакциях: Live CD и Live DVD. Более быстрый и лучше отлаженный Live CD понравится системным администраторам и обычным пользователям; те же, кто любит всегда иметь под рукой большое количество приложений, будут рады Live DVD-версии. В обеих редакциях по умолчанию используется KDE 3.5.1, но на DVD включен и Gnome 2.12, и набор популярных приложений Gnome. В дистрибутив также включены Xfce, IceWM, Fluxbox и несколько других оконных менеджеров, а DVD-версия еще предлагает OpenOffice.org 2.01.

Есть ли что-либо в этом дистрибутиве, что стоит улучшить? К сожалению, не излечены некоторые болезни предыдущего релиза:

бессистемно разбросанные пункты меню KDE по-прежнему его захламляют, затрудняя поиск приложений. Кроме того, Gnome и другие альтернативные рабочие столы не изобавлены поддержкой на уровне KDE; например, удобные пункты меню для запуска специализированных утилит настройки Knoppix только в KDE и предусмотрены.

Но грех жаловаться на совершенно бесплатный продукт, предоставляющий доступ к богатой коллекции программ с открытым кодом на Live CD и DVD. При всех попытках соперников улучшить оригинальную концепцию, включая экспериментальные заплатки на ядро, более интуитивные графические инсталляторы и лучшую поддержку оборудования, неизменная популярность Knoppix служит четким индикатором: Knoppix всё ещё король Live CD для Linux.

[www.knoppix.com](http://www.knoppix.com)



**Клаус Кноппер (Klaus Knopper)** представляет Knoppix 5.0 на выставке CeBIT в Ганновере, Германия.

# Тяжёлая артиллерия

**Fedora 5 • SUSE 10.1 • Ubuntu 6.06** Первые серьёзные релизы 2006-го года – не идеально, но многообещающе.



**Впечатляющие усилия по разработке, предпринятые командами основных дистрибутивов** с целью добавления новых возможностей и дополнительной отладки, привели к задержкам трёх весенних релизов Fedora Core, SUSE и Ubuntu.

Первые впечатления от Fedora Core, чей 5-й релиз был выпущен в марте, оказались смешанными. Необычно долгий процесс разработки, существенные изменения в инсталляторе, новая тема рабочего стола и добавление приложений на основе Mono – всё это привело к чрезвычайно высокому числу закачек.

К сожалению, серьёзная ошибка, затеявшаяся в выпущенный релиз, сделала его непригодным для владельцев машин с закрытыми драйверами видеокарт Nvidia и ATI, и хотя исправление ядра было выпущено достаточно быстро, имидж дистрибутива пострадал. С другой стороны, мно-

гих пользователей порадовали упрощённый инсталлятор, новая система обновлений *Pup* и профессиональный вид темы рабочего стола.

Существенные изменения в SUSE 10.1 привели к задержкам в процессе тестирования; тем не менее второй из основных релизов этого года должен появиться к моменту, когда вы читаете эти строки.

Что касается мира Debian, то перенос нового релиза Ubuntu Dapper Drake на 1 июня был воспринят положительно, как признак того, что качество и надёжность дистрибутива будут на уровне. Основная причина задержки – желание предоставить разработчикам дополнительное время на подготовку локализации для азиатских языков и отладку дистрибутива, обновления безопасности которого будут поддержи-



**Проект Rosetta от Ubuntu помогает пользователям публиковать свои собственные переводы online.**

ваться в течение трёх лет для настольного варианта и пяти для серверного.

<http://fedora.redhat.com>  
[www.opensuse.org](http://www.opensuse.org)  
[www.ubuntu.com](http://www.ubuntu.com)

## ХИТ-ПАРАД ДИСТРИБУТИВОВ

10 самых посещаемых страниц на DistroWatch.com в марте (среднее число визитов в день)

1	Ubuntu Linux	2,996	↔
2	Fedora Core	2,159	↑
3	SUSE Linux	1,999	↓
4	Mandriva Linux	1,203	↔
5	Damn Small Linux	1,024	↑
6	Knoppix	858	↑
7	Debian GNU/Linux	835	↑
8	Mepis Linux	830	↓
9	Gentoo Linux	756	↓
10	Slackware	621	↔

DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве визитов на страницы сайта, посвящённые конкретным дистрибутивам. Хотя эти цифры и не отражают настоящее количество установок, они являются индикатором популярности дистрибутива на данный момент времени.

# Все золото Berry

**Berry Linux 0.68** Новехонький Live CD для фанатов Fedora



**Berry – один из самых дружелюбных пользователей дистрибутивов Live CD.** Хотя он включает элементы Knoppix, например,

модули определения аппаратного обеспечения и инсталлятор на жёсткий диск, базовая часть Berry взята от Fedora Core. Возможно, это лучшая комбинация обоих миров – превосходный набор приложений для рабочего стола и отлаженная система Fedora вкупе с проверенными и протести-

рованными утилитами от разработчиков Knoppix.

Разработанный Юичиро Накада (Yuichiro Nakada), Berry по умолчанию настроен на японский язык (используется модуль Whiz от многоязыкового «наборщика» SCIM). Английский тоже поддерживается, но его нужно явно выбрать из меню при загрузке системы. Дистрибутив загружает приятный рабочий стол KDE с очень удачным выбором сглаженных шрифтов и приличным набором



**Berry Linux включает Berry Control Centre – простую утилиту настройки, с модулями для сети и обновлений.**

приложений для повседневных задач. Сюда входят: *OpenOffice.org 2.0* – для офисных нужд; *Gimp* и *Inkscape* – для редактирования изображений; *BMP*, *MPlayer* и *Xine* – мультимедиа (проигрывание зашифрованных DVD и MP3 файлов поддерживается прямо из коробки); а также *Firefox* и *Thunderbird*

для Интернета. Беспроводные сети также поддерживаются.

Новые версии Berry Linux всегда поставляются с самым последним ядром Linux и рабочим столом KDE и выходят в свет в среднем раз в месяц. **LXF**

<http://berry.sourceforge.jp>

## ОПЕРАТИВНО ЛИ РАБОТАЕТ ГРУППА БЕЗОПАСНОСТИ ВАШЕГО ДИСТРИБУТИВА?

Существенным показателем качества дистрибутива является скорость, с которой разработчики реагируют на обнаруженные в системе уязвимости. Ни администратор, ни пользователь не станут запускать систему, содержащую серьёзные дыры в безопасности, как ни малы шансы успешной удалённой атаки. Система, подсоединённая к Интернету, подобна банковской карте: если её данные похищены, вы ожидаете от банка как можно быстрее реакции. В случае операционной системы разницы быть не должно.

22 марта Sendmail Inc. объявила об обнаружении уязвимости стабильных версий её популярного почтового сервера. *Sendmail*, используемый очень широко (по некоторым оценкам, он доставляет больше половины почтовых сообщений в мире), является критической частью инфраструктуры Интернета. Хотя вы и сами в состоянии залатать код, обычно вы ожидаете, что поставщик вашей ОС предоставит легко устанавливаемое обновление.

Давайте оценим скорость, с которой основные операционные системы отреагировали на эту уязвимость.

Операционная система*	Время, потребовавшееся на исправление уязвимости
Red Hat Enterprise Linux	0 часов 16 минут
Lineox Enterprise Linux	0 часов 56 минут
SUSE Linux	1 час 10 минут
Fedora Core	2 часа 58 минут
Gentoo Linux	3 часа 50 минут
FreeBSD	4 часа 9 минут
Slackware Linux	4 часа 14 минут
CentOS	6 часов 16 минут
Mandriva Linux	8 часов 21 минута
Debian GNU/Linux	17 часов 27 минут

\*Ubuntu не попал в список, поскольку в последнем стабильном релизе Sendmail не является поддерживаемым приложением.

# Сравнение

Каждый месяц мы просматривает тонны ПО, чтобы вам не приходилось заниматься этим самостоятельно.



## ЖИВЫЕ ДИСТРИБУТИВЫ

Ричард Коббет перетряхнул целый мешок LiveCD/DVD, претендующих на ваше внимание.



Вся прелесть «живых» дистрибутивов в их разнообразии. Они могут иметь всевозможные формы и размеры; мы навиделись всяких. LiveCD в основном делятся на три категории.

К первой, да и самой полезной категории относятся дистрибутивы вроде KNOPPIX и Damn Small Linux. Они помогут исправить сбой, поковыряться в настройках, загрузить заплатки – или станут палочкой-выручалочкой, способной спасти и сохранить ваши файлы перед полной переустановкой системы.

Вторая категория повеселее: принадлежащие к ней LiveCD позволяют посмотреть на новый дистрибутив, не проходя мучительного (для многих)

этапа установки. Многие из них могут использоваться и как обычные установочные диски – и пусть комплект софта здесь не столь богат, как у коробочной версии SUSE, но основные системные пакеты и полноценное рабочее окружение на месте. Просто вставьте такой LiveCD в привод, посмотрите на него, выберите опцию установки (обычно предусматривающую даже автоматическое перераспределение дисковой памяти), и готово – новая вежа в развитии подхода «попробуй, прежде чем покупать».

Недостаток «живых» дистрибутивов – необходимость их сочетания с подходящим носителем, например, USB-брелком, потому что файлы, которые вы мусолили в ОЗУ, после

выключения питания исчезнут навеки.

Третья категория – узкоспециализированные LiveCD. Например, чисто технологическая забава в стиле «собери дистрибутив сам» или игровая система для убийства времени и гоблинов по первому требованию души.

Как и многое в мире Open Source, «живые» дистрибутивы легко добыть и протестировать. В этом обзоре мы обращали внимание на их объем, скорость загрузки, совместимость с оборудованием и прилагаемое программное обеспечение. Однако их цели столь различны, что мы в первую очередь смотрели, как дистрибутив справляется с задачей, для которой создан, и насколько эта задача актуальна – вместо сравнения с

### УЧАСТНИКИ СРАВНЕНИЯ

<b>Damn Small Linux</b> .....	<b>22</b>
<b>Games Knoppix</b> .....	<b>24</b>
<b>Knoppix</b> .....	<b>21</b>
<b>LG3D Live CD</b> .....	<b>24</b>
<b>Morphix</b> .....	<b>22</b>
<b>SUSE Live</b> .....	<b>23</b>
<b>Ubuntu Live</b> .....	<b>23</b>

соперниками напрямую. Самое приятное – LiveCD настолько просты, что даже если какой-то из них вызывает одно лишь любопытство, он и то будет не хуже любого инструмента для эксперта.

# Knoppix

Вставьте диск. Загрузитесь. Самое то для дебюта.

• ВЕРСИЯ: 4.0.2 • WEB: www.knoppix.org • Цена: Бесплатно

Knoppix – это не просто LiveCD, а своего рода эталон в мире живых дистрибутивов. Его можно применять как угодно: и как средство «реанимации» системы, и как портативный офис. Knoppix – мастер на все руки, который поможет вам в трудную минуту, и достойный попутчик в командировке (на пару с USB-брелком).

С какой бы целью вы ни использовали Knoppix, без поддержки в трудную минуту не останетесь. Речь идет не только о традиционных web-ресурсах и wiki: про этот дистрибутив выпущено множество книг, от серии «для чайников» до солидных томов издательства O'Reilly.

Как и все LiveCD, KNOPPIX работает только с памятью: никакие файлы на жесткий диск не устанавливаются, ничего не удаляется (хотя, конечно, вы можете управлять файлами самостоятельно). Knoppix затронет на диске только раздел подкачки, если он у вас есть и если ОЗУ вдруг не хватит. Естественно, в этом случае будет соответствующее падение производи-

тельности. Knoppix можно запустить при 64 МБ оперативной памяти, но для нормальной работы необходимо 256 МБ, а если вы хотите работать с *OpenOffice.org* и обрабатывать фотографии в *Gimp*, то рекомендуемый объем памяти увеличится до 512 МБ – 1 Гб.

Knoppix – невероятно емкий дистрибутив, уместивший на диск около 2 Гб программ. Он основан на Debian, а в качестве основного рабочего окружения выбрал KDE. Есть тут и другие оконные менеджеры, например, *Fluxbox* и *IceWM*, но тем, кто любит работать с Gnome, придется взять другой LiveCD – Knoppix, на базе Ubuntu.

## Что душе угодно

Ну и, конечно, в нем есть куча программ, начиная с утилит администратора типа *Ethereal* и заканчивая дизайнерским ПО (*Scribus*) и играми. Большую часть программ, которые устанавливаются по умолчанию во многих дистрибутивах, можно найти и здесь: например, *Firefox* (лидер среди



Установка Knoppix на жесткий диск возможна, но не факт, что проста. Действуйте на свой страх и риск.

браузеров), *OpenOffice.org*, *Gimp*... и т.д. Точный список пакетов постоянно меняется, и Knoppix старается угодить всем, поэтому вы найдете в нем инструмент практически для любой задачи.

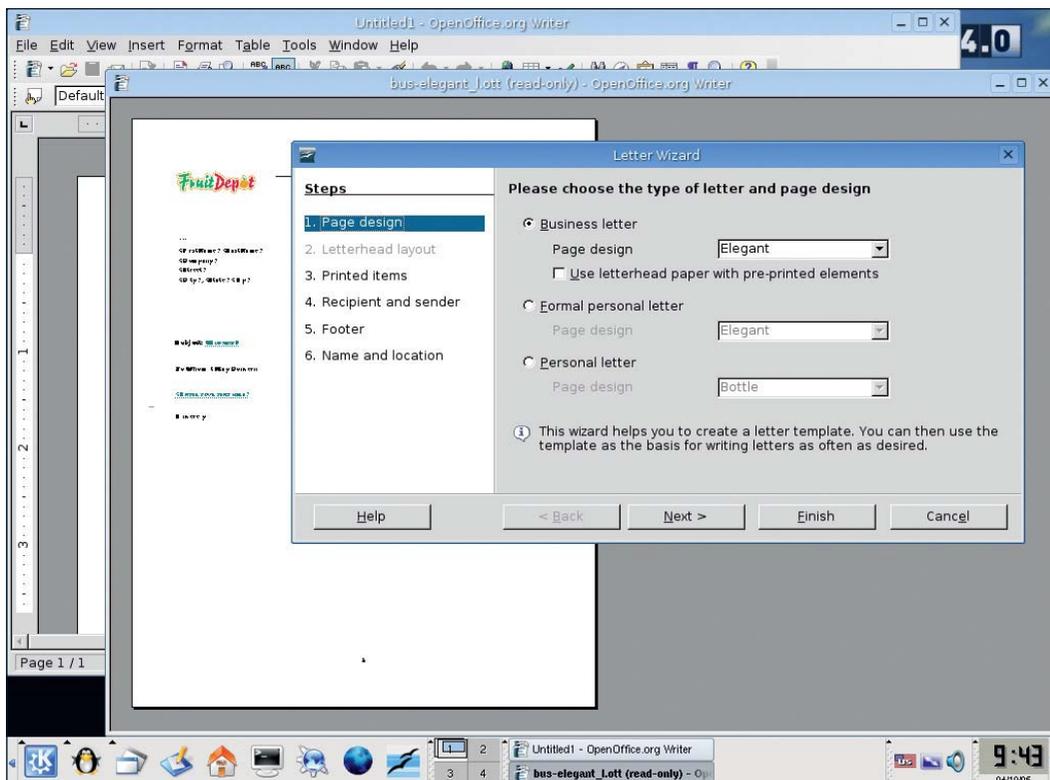
Настройка проходит при загрузке системы, автоматически и почти незаметно. Knoppix особенно хорош в обнаружении оборудования. Отдельного упоминания заслуживают личные настройки пользователя. Здесь поосторожнее: как и в любом LiveCD, вам позволят сохранить любые файлы на

рабочем столе или в другом месте виртуальной файловой системы, но постарайтесь скопировать их на внешние носители, ибо от системных сбояв и отключения питания никто не застрахован.

Лучше всего пользоваться Knoppix вместе с USB-брелком. Меню Knoppix даст вам возможность переместить домашнюю директорию на USB-носитель. С брелком и Knoppix вы можете все свое носить с собой – и систему, и файлы: подключите брелок, вставьте диск – и вот оно, любимое рабочее окружение.

Последним стабильным релизом на данный момент является Knoppix 4.0 (пятая ветвь пока дебютировала лишь в версии для SeBIT). Нестабильный релиз широко распространен в сетях BitTorrent, и содержит такие полезные функции, как прозрачная запись файловой системы NTFS.

Knoppix – отличный дистрибутив и отличная демонстрация технологии LiveCD. Knoppix никоим образом не специализируется на отдельной задаче, он пытается обеспечить все ваши нужды: вызовет «скорую помощь» для рухнувшей системы или позволит в гостях быть как дома. Для какой-то конкретной задачи Knoppix может оказаться не самым удобным, но если что из LiveCD нужно иметь под рукой – это Knoppix.



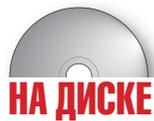
На любом хорошем LiveCD – куча программ. Точный список варьируется от диска к диску.

**ВЕРДИКТ LINUX FORMAT**

МЧС в вашем офисе; славную репутацию Knoppix заслужил по праву.

**РЕЙТИНГ** **8/10**

# Morphix



НА ДИСКЕ

Не нашли идеальный дистрибутив? Создайте свой.

• ВЕРСИЯ: н/д • WEB: [www.morphix.org](http://www.morphix.org) • Цена: Бесплатно

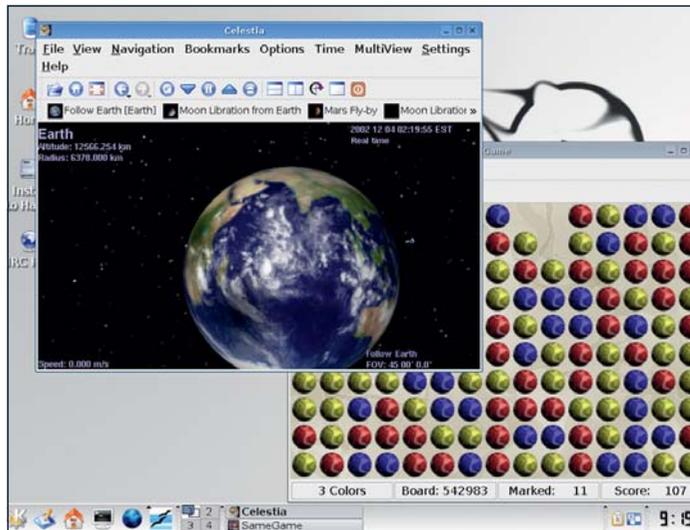
Morphix, основанный на Knoppix (то есть на Debian), использует совершенно иной подход, нежели большинство LiveCD: пользователю предоставляется широчайший спектр настроек, а уж вы коверкайте систему любым удобным для вас образом.

Поэтому Morphix отнюдь не тот дистрибутив, который можно просто скачать и использовать – в большинстве случаев, для специалистов есть и получше. Это скорее выбор продвинутого пользователя, желающего создать свой LiveCD, комбинируя имеющиеся пакеты, а их великое множество: от KDE до демо *Unreal Tournament 2004*. Идея простая: вы получаете 700 мегабайт для заполнения и возможность порулить.

Загвоздка с ним три. Во-первых, последнее время развитие Morphix выглядит не очень активным – разработчики и поклонники у дистрибутива

есть, но ISO-образы на главной web-странице едва ли можно назвать свежими. Во-вторых, хотя собрать свой LiveCD на базе Morphix значительно проще, чем с нуля – берите нужные пакеты да кладите на диск – но и тут сложности встречаются. В-третьих (и это самое печальное), официальный список пакетов на [www.morphix.org/wiki/index.php/ModuleListing](http://www.morphix.org/wiki/index.php/ModuleListing) довольно чухлый.

Все это делает Morphix интересной игрушкой, перескакивать на который, однако, не посоветуешь. Knoppix-подобные дистрибутивы обычно избегают лишней работы, так что если не нашлось LiveCD, заточенного под ваши задачи, то и в Morphix вы вряд ли сможете найти готовые модули с нужными вам программами. Вы можете скачать готовый комбинированный образ, куда более похожий в работе на стандартный LiveCD, однако и в



Не любите KDE? Выкиньте. Хотите игр? Добавьте. В Morphix есть все для создания личного LiveCD.

нем не будет ничего такого, чего нету у других.

Наиболее интересен Morphing-Morphix ISO, построенный как мини-учебник и позволяющий поупражняться в несложных настройках вроде смены фона рабочего стола или автозапуска Firefox, и не ждет, что вы с ходу сбациаете готовый собственный LiveCD.

### ВЕРДИКТ LINUX FORMAT

Мощное решение, если вы любите копаться в настройках, но туристам здесь показывать нечего.

### РЕЙТИНГ

8/10



# Damn Small Linux



НА ДИСКЕ

Компьютер в бумажнике.

• ВЕРСИЯ: 2.6 • WEB: [www.damnsmalllinux.org](http://www.damnsmalllinux.org) • Цена: Бесплатно

Угадать главное преимущество «Чертовски маленького Linux» ничего не стоит – невероятная польза его в том, что он легко умещается даже

в бумажнике: ведь его объем – всего 50 мегабайт, а значит, он прекрасно запишется на мини-диск размером с визитку.



Справа – Firefox, слева – Dillo. Даже в таком крошечном дистрибутиве у вас есть выбор.

Поразительно, сколько всего разработчики впили в этот крошечный объем. Основная идея DSL – использование более легких альтернатив распространенным программам. Конечно, приходится идти на компромиссы: вместо KDE вы получаете Fluxbox, вместо полного OpenOffice.org – облегченный FLWriter; вашу жажду общения удовлетворяет Naim (клиент на базе командной строки) и GPhone.

Но даже если минимализм вам не по вкусу, у DSL есть еще один туз в рукаве: онлайн-сервис MyDSL. Простенькое меню предлагает множество программ, игр и файлов для скачивания прямо из сети, буквально правым щелчком мыши. DSL основан на Debian, и APT с Synaptic в нем, конечно, имеются, но MyDSL для DSL значительно эффективнее.

Все это создает прекрасный баланс между малым размером и функциональностью – OpenOffice.org целиком скачивать было бы мучительно,

зато небольшие приложения мгновенно доступны для работы через службу MyDSL. А еще можно комбинировать базовый диск и USB-брелок, то есть иметь нужные программы при себе.

DSL, как и прочие, устанавливается на жесткий диск, но даже документация советует этого не делать. DSL – прежде всего мобильная система, и вряд ли она подойдет для ежедневного использования на рабочем столе.

DSL напоминает швейцарский армейский нож – худо-бедно управится с любой задачей, а небольшой размер позволяет всегда носить его с собой. Пусть это не самый функциональный или милый LiveCD, обязательно держите его в резерве.

### ВЕРДИКТ LINUX FORMAT

По удобству – особенно если Интернет всегда есть – с этим LiveCD ничто не сравнится.

### РЕЙТИНГ

9/10



# Ubuntu Live

Самый популярный дистрибутив живее всех живых.

• ВЕРСИЯ: 5.10-live • WEB: www.ubuntu.com • Цена: Бесплатно

Несмотря на меньший размер (650 МБ), Ubuntu Live (как, например, и SUSE Live) – обычный дистрибутив общего назначения, трансформированный в LiveCD, а не специальная LiveCD-разработка. Есть несколько причин популярности Ubuntu: прежде всего, это крепкий фундамент Debian, простой в использовании рабочий стол Gnome и грамотный подбор приложений.

Немного разочаровывает тот факт, что Ubuntu Live сразу же не загружает рабочую систему: пользователя сначала проведут по нескольким текстовым меню, в которых он должен будет выбрать язык и разрешение монитора и загрузить нужные драйверы. Но когда вы покончите с этой процедурой, все пойдет гладко.

Привлекательнейшая черта этого дистрибутива – здесь, как и в DSL, можно устанавливать дополнительные программы, причем так же просто, как в DSL, по принципу «для загрузки программы жми здесь», а не «если ты знаком с apt-get или Synaptic, ну тогда вперед».

Главное меню Ubuntu Live делает упор на специализированные средства загрузки: в DSL вы видите наскоро накиданный список, а Ubuntu Live предлагает огромный выбор, причем программы сортированы по категориям и снабжены полным описанием и иконками. Конечно, «установка» в терминологии LiveCD – не совсем верное понятие, поскольку «установлен-

ные» пакеты (и все остальное тоже) испарятся при первой же перезагрузке. Выбор пакетов вполне приличный, все более или менее известные программы здесь налицо. Поэтому Ubuntu Live идеальна и как демонстрационная платформа, и как портативный дистрибутив.

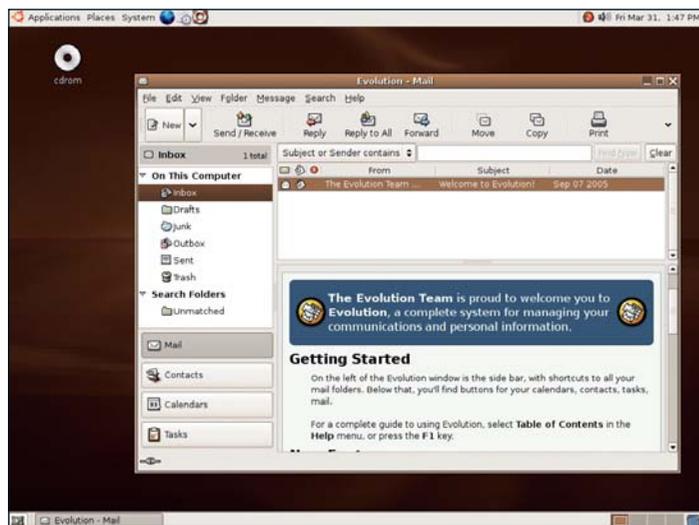
В отличие от SUSE Live, Ubuntu можно установить на жесткий диск... хотя соломки тут не подстелено: познакомиться придется изрядно. Но несмотря на то, что лучше уж скачать настоящий инсталляционный диск Ubuntu, на wiki имеются инструкции по распаковке LiveCD на жесткий диск (и на USB-брелок, для постоянного хранения). Различия между LiveCD и установочным дистрибутивом минимальны.

Ubuntu Live вряд ли займет место в вашем наборе инструментов, это просто «живая» версия Ubuntu, которая служит в первую очередь для знакомства с этим дистрибутивом. Вы можете выступить гидом по Linux, держа боязливых за ограждением – гораздо впечатление гарантировано.

## ВЕРДИКТ LINUX FORMAT

Не будем о грустном при начале установки. С этим LiveCD мы забыли, что находимся не в полноценном Ubuntu – такое бывает нечасто.

**РЕЙТИНГ** **8/10**



Читая почту из LiveCD, убедитесь, что установлена опция Оставить Письма На Сервере, не то ваши сообщения пропадут.

# SUSE Live

Принцип «сначала надкуси»: бесплатная версия.

• ВЕРСИЯ: 10.0 • WEB: www.suse.com • Цена: Бесплатно



Благодаря стильному внешнему виду, SUSE – отличная возможность показать Linux во всей красе.

Вряд ли SUSE нуждается в особом представлении. SUSE Live – «живой» дистрибутив, являющийся свободно скачиваемой демо-версией полной коммерческой версии SUSE (а не свободной OpenSUSE). Если вы хотите показать своим друзьям, что они получат в коробке с SUSE Linux, его-то вам и надо.

Система сразу же радует глаз зелено-голубым экраном с градиентной раскраской, и, не прибегая к допросу на техноязые, загружает менеджер входа в KDE (еще лучше был бы автоход, но и так вам не придется у тестового пользователя LiveCD). Рабочий стол дополнен всплывающими подсказками, а в меню доступна опция поиска нужного приложения, и вы можете сразу попасть... ну, к примеру, в аудиоплеер. Новичкам не придется гадать, что такое *amaroK* – не покер ли? Не совсем, правда, соблюдено единообразие подписей: плеер значится как «Audio Player (amaroK)», а *Gimp* назван просто «Редактор изображений».

Набор программ достаточен, на DVD их влезает много. Вы найдете всех подозреваемых (ядро 2.6.13, Firefox, OpenOffice.org, Evolution) плюс фирменные утилиты SUSE, например, YaST и всевозможные инструменты для записи дисков, проверки почты, форматирования, создания разделов...

больше и желать нечего. SUSE Live – один из немногих «живых» дистрибутивов, содержащих и KDE, и Gnome, не разветвляясь для этого на разные версии (как Ubuntu/Kubuntu).

Установка программ не так проста, как в Ubuntu: скорее всего, вы заблудитесь в утилитах. Но хуже всего то, что вы не сможете установить эту систему себе на жесткий диск, если она вам понравится. Существует отдельная пробная версия SUSE, а LiveDVD предлагается фирмой Novell лишь как демонстрация возможностей системы и средство убедить вас в необходимости покупки коробочной версии. К счастью, никакого давления на вас LiveDVD не оказывает и никакой рекламы не содержит, если не считать крошечной заметки о возможности приобрести контракт на поддержку SUSE.

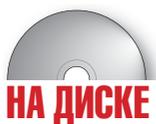
## ВЕРДИКТ LINUX FORMAT

Красива и свободна... но только на DVD, на жесткий диск установить нельзя. И если вас не интересует конкретно SUSE, то и использование ограничено.

**РЕЙТИНГ** **6/10**



# Games Knoppix



Бойцам нужно развлечься!

• **ВЕРСИЯ:** 3.7.0-1 • **WEB:** <http://tinyurl.com/429g2> • **Цена:** Бесплатно

О серьезных вещах – достаточно. Если вам вдруг захотелось поиграть, поможет Games Knoppix. Вы поймете это еще при загрузке, когда увидите, что обычный статичный фон KDE заменен на заснеженный пейзаж с анимированным Санта-Клаусом на лыжах. Все остальное похоже на обычный Knoppix. Те же утилиты, то же оформление, и много таких же программ (вряд ли когда-либо *Tomb Raider* вытеснит *Vim...*).

Конечно же, главная прелесть этого дистрибутива скрывается в подмену «Игры». Есть две версии Games Knoppix, на CD и DVD – не стоит и говорить, что последняя предпочтительней. Здесь есть широкий выбор игр большинства жанров, кое-какие вы, наверное, и не встречали: *FreeCiv*, *Armagetron*, *Chess*, *Freecell...*

Теперь готовьте помидоры: начинается критика.

Забавно, конечно, носить развлекуху в кармане, но Linux просто не имеет достаточной поддержки игр, чтобы включать этот диск в список необходимых. Приятно, что нужно всего-навсего перезагрузиться, чтобы увидеть *NetHack*, и есть где повеселиться. Беда в том, что лишь немногие из игр хоть как-то впечатляют, да и то не чрезмерно. Обычные проблемы с лицензиями означают (заметное) отсутствие коммерческих проектов, вроде *Unreal Tournament* или *America's Army*, хотя и имеется полная копия отличной научно-приключенческой игры *Beneath A Steel Sky*, заодно со снотворной *Flight Of The Amazon Queen* – та и другая любезно предоставлены проектом *ScummVM* (он тоже включен).



Всем встать! *Frozen Bubble* – игра королей. Или во что они там играют – в скачки, поло, *Beneath A Steel Sky*? Все равно игра королей.

По большей части, Games Knoppix предназначен для запуска старых игр и незнакомых новинок без вмешательства в вашу среду, и вы не возьмете его в дорогу ради любимого развлечения. Не надо возиться с установкой – это хорошо, но если игра вам понравится, вы скорее всего установите ее на свою систему, а диск с Games Knoppix отложите на черный день.

## ВЕРДИКТ LINUX FORMAT

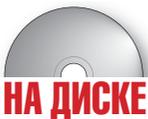
Задумано неплохо, но большинство известных игр не поддерживается.

РЕЙТИНГ

6/10



# LG3D Live CD



Looking Glass – путешествие в Зазеркалье.

• **ВЕРСИЯ:** 2.3 • **WEB:** <https://lg3d-livecd.devjava.net> • **Цена:** Бесплатно

Для получения полного эффекта от этого дистрибутива читайте данную статью, медленно раскачиваясь и поворачивая журнал под разными случайными углами.

Проект Looking Glass – попытка компании Sun сделать трехмерный рабочий стола, сама по себе интересная. Большинство людей не станут скачивать и устанавливать целую систему просто для ознакомления с какой-то технологией, поэтому специализированный «живой» дистрибутив придется весьма кстати.

Кроме собственно демонстрации технологии, ни для чего другого он не пригоден. Здесь мало программ, мало опций – просто последняя версия трехмерного рабочего стола, и вы можете с ней поиграть. Причем запуск этого дистрибутива на наших офисных компьютерах был не осо-

бо успешным; если ваша видеокарта не распознается, вы получите лишь сообщение об ошибке, и – в порядке компенсации – помахивая плавниками, всплывет Slackware.

Однако в качестве демонстрационного LiveCD, LG3D со своей задачей справляется. Looking Glass автоматически запускается сразу после загрузки – или не запускается (подайте 240 МБ ОЗУ, пожалуйста).

По крайней мере, LG столь же пригоден к использованию, как и установленный в системе – увы, на практике постоянное раскачивание и окна, поворачивающиеся то так, то эдак, вызывают скорее подташнивание, чем улыбку на ваши уста. Можно просматривать web в 3D и блуждать среди фотографий на компакт-диске в 3D, но вряд ли вы ради этого поспешите дать пинка KDE или Gnome.



3D-интерфейсы – отличная идея или тупик? Проверьте сами, для этого не надо сметать вашу систему!

Дистрибутивы такого типа много обещают в будущем. Они не просто позволяют запускать новые программы, избегая их настройки (представьте, как вы показываете друзьям «живой» Xgl прямо из-под Windows), но и «ощутить» их в полной мере. В данном случае, ощущается морская болезнь. Но уж так бывает в Зазеркалье. И вообще, на вкус-цвет...

## ВЕРДИКТ LINUX FORMAT

Крайне ограниченная задача, но на большее LiveCD не претендует. И всегда интересно испытать новую технологию.

РЕЙТИНГ

5/10



# LiveCD ВЕРДИКТ

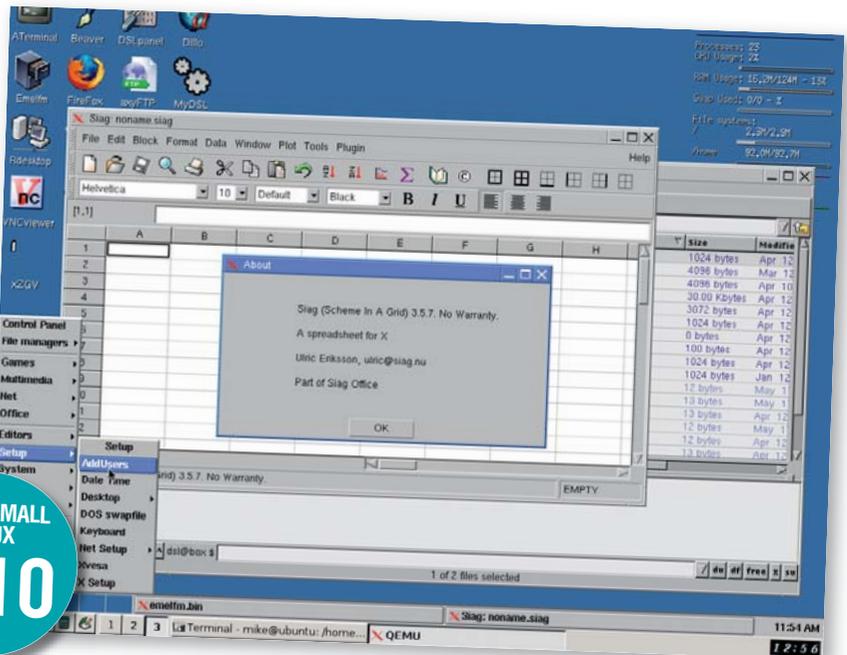
 Все дистрибутивы интересны, но по-нашему, Damn Small Linux – самый полезный из них (если на машине, которая его приютит, есть доступ к Интернету). Крошечный объем позволяет быстро его скачать, записать на диск и, что важнее всего, всегда держать при себе; если каких-то программ не хватает, MyDSL придет на помощь и загрузит их на лету.

Вас может удивить, что мы не рассматривали дистрибутивы для каких-то особых задач, например для аудита безопасности. Но ведь это задача скорее для утилиты, чем для LiveCD. У всех нас есть любимые утилиты и методы работы, и лучшей чертой систем вроде DSL, Knoppix и Morphix является в первую очередь их расширяемость. Если чего-то не хватает, добавить всегда возможно. «Живой» дистрибутив во всей красе – это не просто компьютер на диске, а ваш

собственный компьютер, от темы рабочего стола до набора любимых приложений.

Сейчас, правда, добиться этого нелегко. Будем надеяться, что системы вроде *MorphingMorphix* сумеют значительно упростить создание личного LiveCD. Планка, которую надо взять – это выбор прямо из списка, запись диска, и в путь! Мы уже на пороге, но кто знает, когда настанет это время?

Здесь и сейчас, самые интересные LiveCD – те, что демонстрируют новейшие технологии. Практически каждый LiveCD уже способен распознать ваше оборудование (проблемой пока являются USB и прочие Windows-центристские устройства), настроить звук и сеть через Ethernet; если Project Looking Glass справляется с этим, то



Потрясающий баланс между мощностью и портативностью. DSL уютно устроится у вас в бумажнике.

почему бы не справиться и с играми, и с офисными пакетами?

Чтобы попасть в светлое будущее, LiveCD должны еще много поработать. Если ни один из рассмотренных дистрибутивов вам не подошел – посетите страницу <http://www.frozentech.com/>

[content/livecd.php](http://content/livecd.php), тут представлено свыше трехсот претендентов на вашу CD-матрицу. Что-нибудь вам да понравится. А нет, так просто выкиньте диск на помойку: цена вопроса составит стоимость пустой болванки. **LXF**

## СРАВНИТЕЛЬНАЯ ТАБЛИЦА ВОЗМОЖНОСТЕЙ

	DSL	Games Knoppix	Knoppix	Morphix	SUSE	Ubuntu	LG3D
<b>Формат</b>	Business Card CD or CD	CD/DVD	CD	CD	DVD	CD	CD
<b>Размер</b>	50Mб	700Mб/2.7Гб	700Mб	Различается	1GB	650Mб	240Mб
<b>Возможность установки</b>	Да		Да			Да	
<b>Специализация</b>	Портативный рабочий инструмент	Игры	Портативное рабочее окружение	Создание своего LiveCD	Демо	Демо	Технологическое демо
<b>Среда</b>	Fluxbox	KDE	KDE	Любой	KDE or Gnome	Gnome	Fluxbox/Project Looking Glass
<b>Версия ядра</b>	2.4.26	2.4.27	2.6.12	Любая	2.6.13	2.16.12.6	2.4.29
<b>Возможность ремастеринга</b>	Да		Да	Да			
<b>Основа</b>	Debian	None	Debian	Debian	SUSE	Ubuntu/Debian	Slax

# Hot Picks



Лучшее на планете новинки открытого ПО!

## Майк Сондерс

избегает синдрома статической нагрузки при наборе кода, воздевая руки к небу и завывая.



Здесь мы проводим обзор некоторых из наиболее популярных программ в мире.

Каждый месяц мы прочесываем тысячи проектов с открытым кодом и отбираем для вас самые новые, самые изобретательные и просто отличные. Большая часть программ, попавших в обзор, записана на прилагаемый к журналу диск, но мы также предоставляем и веб-ссылки, чтобы вы могли скачать новейшие версии.

Если у вас есть идеи о том, какие проекты с открытым кодом нам стоит включить в обзор, пишите нам на адрес:

## БЛОГ-СЕРВЕР

# LnBlog

• ВЕРСИЯ 0.6.4 • WEB [www.skepticats.com/lnblog](http://www.skepticats.com/lnblog)

Сейчас в сети уже больше блогов, чем фонарных столбов в Западной Европе. Даже команда LXF вскочила на поезд блоггинга несколько месяцев назад: [www.linuxformat.co.uk/blog](http://www.linuxformat.co.uk/blog). Хотя многие серьёзные сайты и порталы вроде Google предлагают услуги по ведению блогов, многие пользователи предпочитают вести блог на собственном сервере: либо по соображениям безопасности, либо предпочитая

при этом в правой части интерфейса удобно расположен календарь. Другие темы не слишком живописны – мы навидались таких на WordPress и других популярных блог-серверах, но выглядят профессионально. Кроме того, редактирование и создание новых тем не составляет труда, так как их дизайн целиком построен на строгих XHTML и CSS. С правами администратора можно добавлять новых пользова-

работу и снизить нагрузку. Это хорошая реализация модульности, помогающая также укрепить безопасность. На данный момент многие модули влияют на раскладку страницы – например, добавляя боковые панели и окна поиска или META-теги (например, для генерации карты сайта). Мы надеемся, что если LnBlog обзаведётся стойкими последователями, его пользователи создадут дополнительные модули, которые добавят мощную функциональность, не усложняя само приложение.

LnBlog поддерживает пинги TrackBack, RSS, загрузку файлов и почтовые извещения о комментариях. Запись в блог можно добавить двумя способами: используя разметку в стиле BBCode (удобную для новичков) или обычный HTML. Вместе эти возможности покрывают всё, что вы ожидаете от блог-сервера. Производительность и стабильность работы также вызывает уважение. Единственное, что мы можем покритиковать – это отсутствие пользовательской документации: приложено довольно много текста, но он рассчитан на продвинутого пользователя, а хотелось бы увидеть нечто более пригодное для первого знакомства. Впрочем, от версии 0.6.4 всего на свете требовать рано, и LnBlog – прекрасный выбор, если вам нужен законченный блог-сервер без всяких осложнений.

«ЗАПИСЬ В БЛОГ ВОЗМОЖНА С РАЗМЕТКОЙ В СТИЛЕ BB CODE ИЛИ ПРОСТО HTML.»

иметь больший контроль над своим журналом. Создатели сервера LnBlog делают многое для облегчения поддержки сервера, одновременно предлагая модульный дизайн, и хотя сервер ещё в стадии разработки, он уже может использоваться блоггерами на постоянной основе.

## Рай для блоггера

Установка проста: распакуйте zip-архив в каком-нибудь каталоге, доступном для вашего web-сервера, и направьте браузер по адресу этого каталога. Вы увидите на экране начало процедуры установки, позволяющей выбрать базовые настройки блога и убедиться, что права доступа к файлам настроены корректно. Хотя большинство пользователей используют Apache, LnBlog не привязан к какому-то конкретному web-серверу: всё, что ему нужно – это PHP 4.1 и PCRE. Многие блоги требуют для хранения данных MySQL или другой сервер баз данных, а LnBlog хранит все записи в обычных текстовых файлах, так что настраивать базу данных вам не придётся.

Блог-тема по умолчанию – окна и голубые тени, вперемешку. На первый взгляд она больше походит на wiki, чем на блог,

а чтобы подавить нудные спамоподобные комментарии, вы можете создать чёрные списки для слов и IP-адресов.

Большая часть функциональности LnBlog поставляется в виде дополнительных модулей, которые в случае необходимости можно отключить, чтобы упростить

## HOT PICKS В ЭТОМ РАЗДЕЛЕ

BlinkenSisters .....	29
Gnash .....	27
GtkShots .....	30
LnBlog .....	26
Protoeditor .....	27
ReactOS .....	31
Smb4K .....	31
VirgoFTP .....	30
VoR .....	29
Workrave .....	28

## ОБРАТИТЕ ВНИМАНИЕ НА ПОБЕДИТЕЛЯ HOT PICKS

Все, что попадает в раздел HotPicks, заслуживает самого пристального внимания. Однако, каждый месяц мы выбираем всего один самый яркий проект. Побеждает лучший!



Котенок (одна из тем, поставляемых с LnBlog) создаст уют на вашем блоге.

## ПРОИГРЫВАТЕЛЬ FLASH

## Gnash

• ВЕРСИЯ CVS 06-04-2006 • WEB [www.gnu.org/software/gnash](http://www.gnu.org/software/gnash)

**Shockwave Flash и Linux не всегда** хорошо уживаются. Хотя Macromedia активно продвигала поддержку своего формата на Linux, её проигрыватель – всё ещё закрытый продукт, и фанаты свободного программного обеспечения чувствуют себя обиженными. *Gnash* – новый открытый проект, нацелившийся объединить высокое качество воспроизведения *Flash v7* и Linux, опередив усилия сообщества, едва добившегося совместимости с *Flash v5*.

В данный момент *Gnash* доступен только через CVS, официальных его релизов пока не существует. На нашем диске вы



Тестируйте возможности *Gnash* с помощью игр с сайта [b3ta.com](http://b3ta.com).



найдёте срез исходного кода, актуальный на момент выпуска журнала. Его можно собрать обычными командами `./configure`, `make` и `make install` (от имени суперпользователя). *Gnash* требует библиотек OpenGL, SVGLib и SDL, а также файлов для разработки *Firefox*, чтобы собрать соответствующий подключаемый модуль.

Разработчики пока не советуют брать подключаемый модуль *Firefox* для ежедневного Интернет-серфинга. Вместо этого рекомендуется использовать автономный плеер, запускаемый командой `Gnash filename.swf`. Вы увидите окно, где будет проигрываться анимация *Flash*; качество анимации зависит от сложности ролика. Функциональность *Gnash* базируется на *GameSWF*, и, как можно видеть из демонстрационного теста, поддерживает большую часть спецификации *Flash v7*, однако ролики из реальной жизни склонны сбоить.

Мы тестировали *Gnash* на многих *Flash*-роликах в форматах *v5*, *v6* и *v7*. Некоторые

из них проигрывались просто отлично, с полноценным звуком, анимацией и интерактивностью, тогда как у других – особенно игр – пропал звук и имелись проблемы при работе с мышью. Несколько роликов, начавшись очень хорошо, затем обрастали серьёзными визуальными ошибками. Хотя для утилиты, находящейся на столь ранней стадии разработки, когда количество вносимых изменений и исправлений каждый месяц исчисляется сотнями, это вполне приемлемо.

*Gnash* ещё не обкатан, но имеет большой потенциал. Мы считаем его чрезвычайно важным для создания задела полностью открытых Интернет-протоколов и форматов. Миллионы сайтов используют *Flash*, и если разработчики *Gnash* продолжат разработку с той же скоростью, мы сможем наслаждаться широким выбором Интернет-контента (а заодно и раздражающей рекламой...), сохраняя верность открытому ПО.

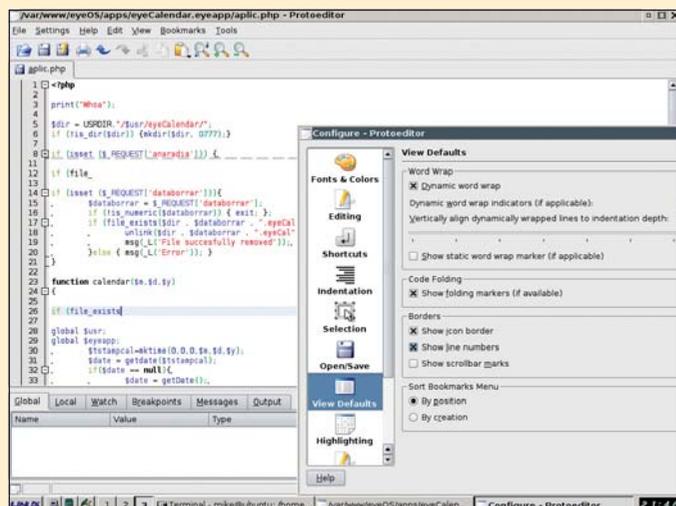
HULA HOOP KITTEN ILLUSTRATION BY SARAH BEAR, WWW.DOMESTICDESIGN.COM

## РЕДАКТОР И ОТЛАДЧИК PHP-КОДА

## Protoeditor

• ВЕРСИЯ 0.8 • WEB <http://protoeditor.sourceforge.net>

**PHP – очень популярный язык для** создавать очень сложные приложения разработки web-сайтов, и неспроста. Его (пример – среда *PHP-Nuke*), и он доступен легко изучить, с его помощью можно практически на всех разновидностях Unix.



Всевозможные настройки всегда к стати в приложениях для хакеров.

*Protoeditor* разработан с целью облегчить процесс редактирования и отладки PHP-скриптов, не привнося миллионы лишних свойств IDE, только сбивающих с толку залетных кодеров. Ни сложных панелей организации проектов, ни диалогов с двадцатью вкладками: просто редактор кода и отладчик.

*Protoeditor* построен на базе KDE, так что для его сборки из исходных текстов понадобятся пакеты разработчика KDE (3.2 или новее), а также *Qt*. Возможности отладки обеспечиваются опциональным пакетом – в данный момент он поддерживает *DBG*, *Xdebug* и *Gubed*, все они записаны на наш диск. Архитектура *Protoeditor* строится вокруг повторного использования компонент KDE, поэтому его редактор кода, предоставленный *Katepart* и используемый также в редакторе *Kate*, поддерживает подсветку синтаксиса для огромного числа языков программирования (аж 119).

Другие полезные возможности – операции над блоками кода и примитивная форма подстановки ключевых слов с помощью дополнительных модулей. По большей части *Protoeditor* функционирует

как обычный редактор для разработчика, но ценность программе придает панель отладчика, расположенная в нижней части экрана. Здесь вы можете отслеживать вывод исполняемой программы, проставлять точки останова и просматривать значения переменных по мере работы программы. По сравнению с такими IDE, как *Eclipse* и *Anjuta*, набор возможностей покажется хакерам ограниченным, но обычные PHP-программисты найдут всё, что им нужно.

Особо следует отметить возможности настройки *Protoeditor*: в присутствии KDE стили, отрегулировать можно почти каждый аспект приложения, от косметических нюансов (шрифты и цвета) до обработки типов файлов и комбинаций клавиш. Широка диапозона настраиваемых опций делает *Protoeditor* отличным выбором для кодеров также и на языках, отличных от PHP. Возможно, в нём нет всех примочек продвинутых IDE, но если вы хотите слепить программку в небольшом и быстром редакторе, который к тому же хорошо ладит с KDE, то почему бы не попробовать *Protoeditor*?

ЗАЩИТА ОТ СТАТИЧЕСКИХ НАГРУЗОК

# Workrave

• ВЕРСИЯ 0.8.2 • WEB [www.workrave.org](http://www.workrave.org)

Если только этот журнал не попал к вам в руки случайно, вы, скорее всего, так или иначе связаны с компьютерами, и знакомы с существованием синдрома мышечных болей, вызванного долгой работой за клавиатурой (RSI – Repetitive Strain Injury, русский термин – СДСН, синдром длительных статических нагрузок). А то и испытывали этот синдром на себе – в зависимости от рода вашей деятельности. Например, программисты и операторы по вводу данных в компьютер весьма подвержены действию

можете заняться физкультурой.

Просто попросить пользователя проделать упражнения несколько неконструктивно, поэтому в *Workrave* включены и сами упражнения, они снижают риск возникновения RSI. Тренировкой руководит кокетливая 3D-девушка, показывающая, как нужно вытянуть руки и расслабить плечи, шею и прочие части тела, подверженные RSI. Великолепно придумано: сделать перерыв в работе с компьютером – хорошо, и лучше в это время не бежать к кофейному автомату

лит вам осуществлять мониторинг работы сотрудника и подстраивать необходимые перерывы, если он начал замечать признаки RSI-синдрома.

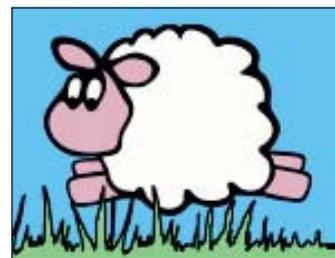
Похожим образом можно назначить глобальные настройки сразу для нескольких машин в сети, чтобы в случае, если пользователь переходит с одного компьютера на другой, перерывы в работе остались теми же. Продуманность этой функциональности и то, как она реализована, достойны всяческих похвал.

## Брось работу. Живо!

Даже при всех своих возможностях, *Workrave* не вышвыривает вас с компьютера, заблокировав его или мешая вам делать свое дело, если вы действительно в цейтноте. Предложения прерваться можно отложить до любого момента, а всю систему временно заглушить, и она подстроится под меняющийся рабочий график, продолжая работать.

К счастью, окна не выскакивают, пока вы отлучились на перерыв; кроме того, с помощью *XScreenSaver* можно заблокировать доступ к вашему рабочему столу, чтобы вы могли спокойно отойти от компьютера.

Другая полезная возможность программы – хорошая многоязыковая поддержка: обеспечено большинство западноевропейских языков плюс китайский и русский. Вдобавок есть удобный просмотр статистики, показывающий количество перерывов, которые вы сделали или проигнорировали. Вы даже можете отслеживать количество



Что, барашек, симптомчики? Талисман *Workrave*.

нажатий на клавиатуре, сделанных, пока работал *Workrave*, и дистанцию, которую прошёл курсор мыши в метрах! Подобные мелкие, но милые черточки разбросаны по всей программе, а на веб-сайте вы найдёте множество благодарностей от пользователей, которые считают, что *Workrave* гораздо душевнее, чем многие платные аналоги.

Другая положительная черта *Workrave* – его универсальность. Настраивается длительность каждого перерыва либо промежутка между перерывами (с точностью до секунд), вид статусного окна, звуковые эффекты и многое другое. Разработчики программы очень серьёзно относятся к своей работе, поэтому программа выглядит отлаженной и законченной, как коммерческое приложение. *Workrave* стоит внимания, если от сидения за компьютером у вас что-то начало побаливать, а уж если вы менеджер или администратор, обслуживающий несколько рабочих станций Linux, то в нашу эпоху осознания важности заботы о здоровье вы просто обязаны попробовать это приложение.

## «ВСЕ МЫ ДОЛЖНЫ ЧАЩЕ ОТДЫХАТЬ ОТ КОМПЬЮТЕРА.»

синдрома, часто затрудняющего их работу, да и обычные пользователи компьютера могут страдать из-за неестественной позы и движений.

Оказывается, нам всем нужно отдыхать от нашего компьютера гораздо чаще – вот бы кто-нибудь напоминал нам об этом, когда мы с бешеной скоростью переделываем пробелы на знаки табуляции или занимаемся подобной столь же захватывающей деятельностью.

Для решения этой проблемы разработчики *Workrave* спроектировали небольшую программу-помощницу: она выскакивает на экран и просит пользователя оторваться ненадолго от работы и сделать упражнения. Это приложение рабочего стола представляет собой нечто большее, чем просто апплет, предлагая широкие возможности по настройке и массу информации об RSI. Программа работает как под Windows, так и под Linux – это плюс для менеджера, заботящегося о здоровье сотрудников, которые работают в смешанной среде. Интерфейс для Linux написан с использованием *GTK 2*.

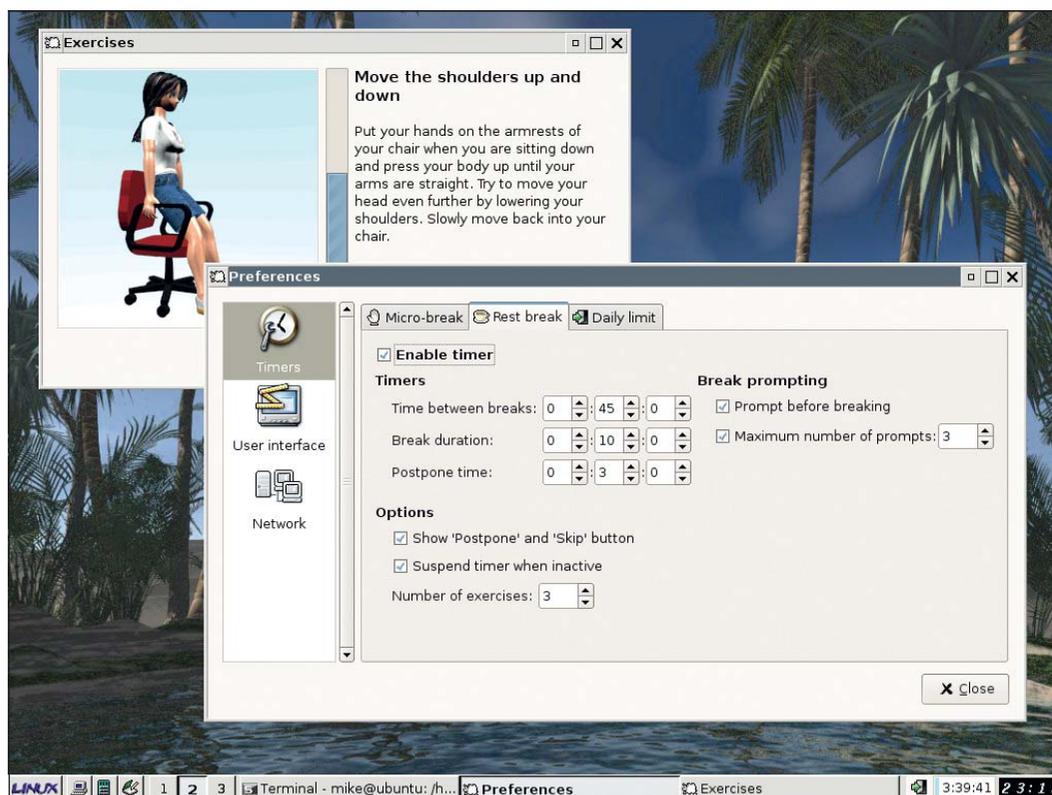
После запуска *Workrave* добавляет небольшую иконку в системный лоток поддерживаемых рабочих столов (а это Gnome, KDE, *Xfce*, *IceWM* и некоторые другие). Если этого сделать не получится, программа отобразит небольшое окно, с таймером обратного отсчета и с возможностью доступа к настройкам по правой кнопке мыши. Окно ненавязчиво: оно просто показывает, сколько времени осталось до перерыва.

В терминологии *Workrave* существуют микро-перерывы, позволяющие на некоторое время отвлечь ваше внимание от компьютера, и перерывы на отдых, когда вы

ту за очередной чашкой, а заняться поправкой здоровья.

## Ваше здоровье

В качестве автономного приложения *Workrave* предлагает более чем достаточный набор возможностей, однако его сетевые функции поднимают программу на новый уровень. Если вы администратор, вы можете настроить *Workrave* на отсылку данных центральному серверу, что позво-



Если вы затрудняетесь в выполнении упражнения, 3D-девушка вам все объяснит.

ИГРА-ПЛАТФОРМЕР

# BlinkenSisters: поиск пропавших пикселей

• ВЕРСИЯ 0.3 • WEB <http://blinkensisters.sourceforge.com>

**Потеря пикселя, может, беда и не великая, однако для агента Кейт (звезды BlinkenSisters) это важное событие: несчастный случай, произошедший в секретной лаборатории высоких энергий BlinkenArea, привёл к рассеянию «самозарядного пиксельного оборудования» по случайным территориям, и она должна собрать все пиксели, пока не произошла катастрофа. Ура! Вот вам платформенная игра, в которой не нужно спасать Foo от ловушек злобного Bar.**

После установки GCC и SDL вы можете распаковать архив и набрать в каталоге **LostPixels\_0.3** команды **make** и **make install** (с правами суперпользователя). После этого наберите **blinkensisters** из командной строки, и игра запустится. От версии 0.3 не стоит ждать обворожительной графики: в игре в основном используются спрайты из 16-битной эпохи, хотя и в элегантно-ностальгическом стиле.

*BlinkenSisters* – платформенная потеха; вы управляете агентом Кейт (с помощью клавиатуры или джойстика), прыгая по платформам, собирая потерянные пиксели и давя врагов, преграждающих вам путь. Игра приправлена элементами головоломки, вроде откидных переключателей для обнаружения спрятанных пикселей. В данный момент в игре восемь уровней – для столь юной версии, отличный коктейль из традиционных элементов платформенных игр с шансами поднапрячь мозги.

Почти как у принцессы Поганки (Princess Toadstool) из *Super Mario Brothers 2*, платье агента Кейт обладает подъёмной силой во время прыжков в воздух, в результате чего вы вынуждены прилагать усилия, пытаясь попасть на место приземления. Над рудиментарными спрайтами переднего плана явно не перетрудились, зато сюрреалистический фон вконец отлично подобранными радостными мело-



Агент Кейт выбрала причёску в стиле Тэтчер.

диями платформ формируют неотразимую обстановку для игры.

Нам в самом деле понравилась игра *BlinkenSisters*, даже на столь ранней стадии разработки. Смотреть особенно не на что,

но для подвигов место есть. Bravo команде разработчиков!

ИГРА В СТИЛЕ ACTION

## VoR

• ВЕРСИЯ 0.4 • WEB <http://jasonwoof.org/vor>

**Возьмите классику всех времён – Asteroids, уберите стрельбу, и что вы получите? Нечто тоскливое? Только не в случае Variations on Rockdodger (VoR).** Это переделка раннего... угадали, *Rockdodger'a*, и хотя игровой процесс не так уж сложен, сама игра неожиданно забавная. Для

ее компиляции понадобится SDL: просто запустите **make** и затем **./vor** в каталоге, появившемся после распаковки игры.

*VoR* предлагает вам управлять безымянной металлической сферой, блуждающей в космосе. Что ж, бывали мы там. Со всех сторон экрана появляются астероиды,

постепенно оказывающиеся прямо по курсу вашего корабля – самое время проявить молниеносную реакцию. Клавишами курсора вы можете включать боковую тягу двигателей, чтобы изменить траекторию полета и избежать удара. Если ваш корабль все же задет астеро-

идом, посреди экрана взрывается эффектный фейерверк пикселей.

Ловко перебирая пальцами, вы избежите столкновений; а можно вместо этого отпихивать астероиды, все время следя, не мчитесь ли вы прямоком к очередному скоплению. Описывать игру – всё равно что анализировать перемещения футболиста, приводящие к голу: базовая стратегия у вас есть, но в игре вы о ней особо не задумываетесь, просто идете на адреналине. Выиграть нелегко, но проиграв, вы не чувствуете себя обманутым.

Косметических ухищрений немало – например, тысячи пикселей, возникающих при столкновении; однако нам бы очень хотелось слышать какую-нибудь злорадную фоновую музыку, усиливающую ощущение напряженности. *VoR* не из тех игр, за которые вы готовы выложить 40 фунтов, и все равно это игра скоростная, непростая и захватывающая. Но

бойтесь синдрома RSI!



Растёрли в пыль! Наш корабль (слева) тюкнул астероид (тот, что сверху).

АВТОМАТИЗАЦИЯ СОЗДАНИЯ ЭКРАННЫХ СНИМКОВ

# GtkShots

• ВЕРСИЯ 0.1 • WEB [www.flagar.com/software/gtkshots](http://www.flagar.com/software/gtkshots)

**GtkShots** – небольшая утилита для автоматизации процесса регулярного создания снимков экрана, а также изменения их размера и сохранения на диск. Если вы пытались когда-нибудь готовить документацию или делать слайды, демонстрирующие поведение приложения, утилита немало вас порадует: вам слишком хорошо знакомо это нудное занятие. В *Gimp* приходится опускаться до постоянного кликания мышью, в командной строке нужно нанизывать друг за другом множество утилит в надежде, что всё пойдёт как надо. Милле грацие – большое спасибо Флавио Гарджуло (Flavio Gargiulo), создателю *GtkShots*.

Программа состоит из двух частей, и обе написаны на Python. Тут есть утилита командной строки *Pyshots*, которая делает основную работу, а сам *GtkShots* – надстройка над *Pyshots*, позволяющая удобно выставлять опции. Вы можете просто запустить файл **GtkShots.py** после распаковки архива, без всякой установки программы.

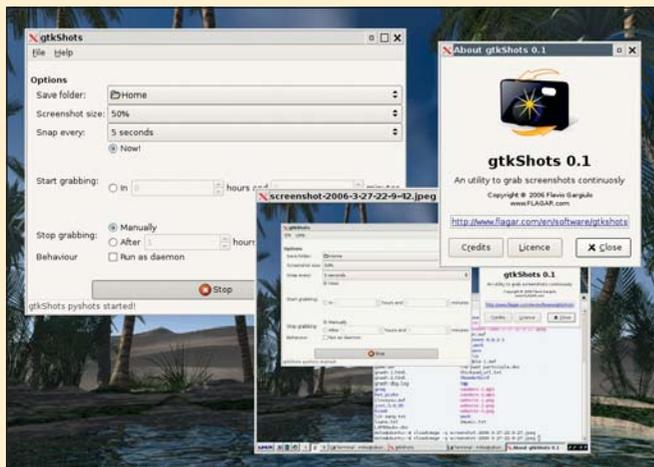
По умолчанию *GtkShots* будет скидывать снимки экрана в текущий каталог, но вы можете направить их в любое другое место. Можно также масштабировать получаемые изображения (это полезно, если вам достаточно общего вида, а детали не нужны) и устанавливать интер-



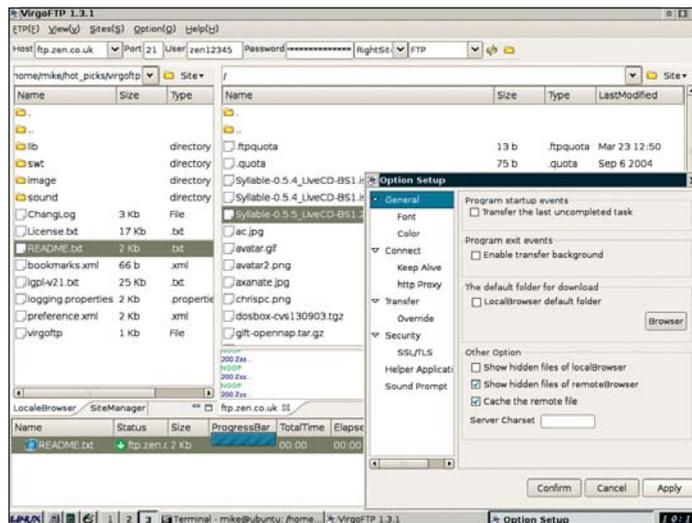
вал между снимками (от пяти секунд до дня).

Дополнительно, можете выставить время пуска и отключения процесса съема экрана. Когда вы нажмете кнопку Start, *GtkShots* примется в фоновом режиме создавать снимки экрана, с заданным вами интервалом, и сохранять их на диск. Единственная неприятность – снимки могут быть сохранены только в формате JPEG. Мы надеемся, что разработчики в конце концов добавят поддержку PNG, для создания снимков без потери качества.

Спектр применений *GtkShots* гораздо шире, чем просто создание документации. Например, вы можете настроить компьютер общего доступа на создание снимков каждые 20 секунд, а затем выкладывать их в папку, доступную через web. Таким образом, вы сможете отслеживать использование системы, не забывая сетевой канал, как в случае с VNC или X – просто поглядывайте иногда на экран, все ли в порядке. Даже если вы не следите за системой постоянно, у вас, тем не менее, остается история, по которой легко отследить возможные проблемы или подозрительную активность. Для задач документирования и мониторинга эта утилита – компактный и аккуратный помощник.



Хотите убедиться, что ваши сотрудники работают, а не читают Slashdot?



К счастью, *VirgoFTP* не связан с Джоном Вирго (John Virgo).

FTP-КЛИЕНТ

# VirgoFTP

• ВЕРСИЯ 1.3.1 • WEB <http://sourceforge.net/projects/qftp/>

**Протокол передачи файлов (FTP – File Transfer Protocol)**, возможно, седая древность – в Интернете он с 1973 года, но всё ещё используется в качестве простого и надежного способа передачи файлов. Несмотря на то, что почти каждая инсталляция Linux включает почтенную утилиту командной строки *ftp*, для графического интерфейса разработано несколько отличных клиентов, например, *GFTP*. Новый клиент носит название *VirgoFTP*, а его целью является эмуляция интерфейса *CuteFTP* и поддержка Linux, OS X и Solaris.

Наш клиент с астрологическим именем написан на Java, поэтому вам понадобится среда JRE 1.5 (или новее) от Sun. Она включена во многие дистрибутивы, и ее можно скачать с <http://java.sun.com>. Если ваша версия Java установлена правильно, запуск *VirgoFTP* не составит проблем:

- 1) Распакуйте архив **VirgoFTP\_1.3.1\_linux\_gtk\_x86.zip**
- 2) Сделайте исполняемым скрипт *VirgoFTP* (**chmod +x VirgoFTP**)
- 3) Пуск!

Интерфейс написан с использованием *SWT (Standard Widget Toolkit)*, который под Linux имеет вид *GTK 2*. Поэтому графический интерфейс хорошо сочетается с *Gnome* и *Xfce*.

Интерфейс *VirgoFTP* не особенно ласкает глаз: приятно разве что простенькая заставка. Основное окно состоит из традиционных трёх панелей. В левой показано локальное дерево каталогов, в правой – список файлов на удалённом сервере, а внизу находится список файлов, ожидающих закачки. Файлы помещаются

в очередь закачки правым щелчком мыши, при этом *VirgoFTP* пытается предсказать, сколько времени займет каждая закачка. Полезная возможность – закладки, позволяющие работать с несколькими FTP-серверами одновременно; клиент при этом помнит параметры авторизации, так что вам не понадобится всякий раз вводить их.

*VirgoFTP* обладает разумными возможностями настройки, от общих установок интерфейса (например, шрифты и цвета) до режима передачи (ASCII или бинарный). Имеется также возможность настраивать количество и интервалы повторных попыток, а продвинутые пользователи будут довольны обилием установок, влияющих на скорость закачки и позволяющих использовать сеть по максимуму. Однако, поскольку онлайн-документации на данный момент не существует, хотелось бы видеть в диалоге настроек описание некоторых опций из тех, что посложнее.

Следует также упомянуть поддержку прокси-серверов SOCKS, активный и пассивный режимы передачи, а также способность возобновить прерванную передачу данных – всё это в *VirgoFTP* есть. По возможностям данный клиент проигрывает *GFTP*, но для типичных повседневных задач это не критично: *VirgoFTP* – опрятный, покладистый и мультиплатформенный FTP-клиент, выполняющий свою работу более чем успешно. Попробуйте его, если вам нужно нечто простое и удобное.

# HOTPICKS ПОВТОРНЫЙ ВИЗИТ

ПРОВОДНИК ДЛЯ ПРОТОКОЛА SMB

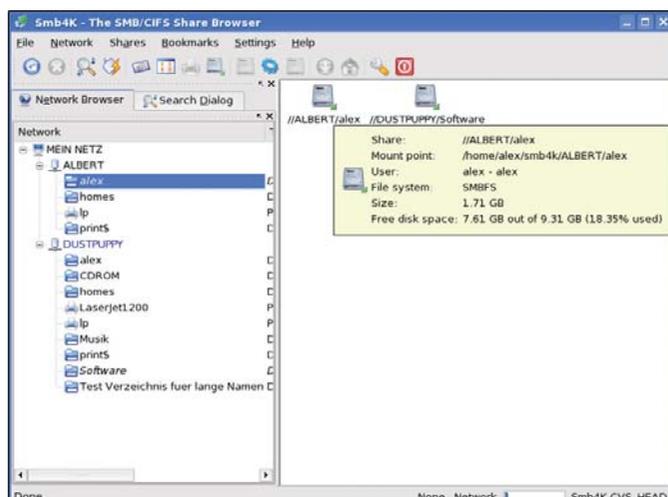
## Smb4K

- **ВЕРСИЯ** 0.7.0 beta
- **WEB** <http://smb4k.berlios.de>

С момента последнего обзора версии 0.5.2 в LXF69 HotPicks разработчики Smb4K были погружены в разработку новых функций и обновлений. Smb4K — это браузер совместно используемых ресурсов для протоколов SMB/CIFS, написанный для KDE. Он позволяет получать доступ к дискам Windows из удобного графического интерфейса, не прибегая к командной строке. Smb4K использует известный пакет программ Samba, и в нашем прошлом обзоре в номере LXF69 мы восхищались его стабильностью и производительностью.

Нам открывается приятный компактный интерфейс, удачно использующий панели изменяемых размеров и закладки для упрощения немалых возможностей в одно окно. Левая панель содержит дерево виджетов, позволяющее выполнять навигацию по сети, а в правой — список ресурсов. Вы можете искать ресурсы в сети, расставляя закладки на часто посещаемых, а также настраивать различные опции SMB и CIFS в диалоге настроек.

Примечательные новые возможности программы включают хранение пароля в



Опрятный интерфейс Smb4K упрощает использование программы.

менеджере KDE *KWallet*; ускорение поиска командой *Samba net*; дополнительный модуль для *Konqueror*; поддержку *Mount.cifs* и *Umount.cifs*, позволяющих монтировать ресурсы обычным пользователям; а также синхронизацию локальной и удалённой копий с помощью *Rsync*. Большая работа проделана над пользовательским интерфейсом, в частности, виджетом браузера ресурсов, а также исправлено множест-

во ошибок в авторизации и работе с закладками.

Приятно, что новые возможности не повлияли на надёжность и производительность Smb4K. Впрочем, каждый релиз увеличивает функциональность, что оправдывает обновление. Попробуйте эту программу, если вы пользуетесь KDE, и в вашей сети есть компьютеры под управлением Windows.

## ОПЕРАЦИОННАЯ СИСТЕМА

# ReactOS

- **ВЕРСИЯ** 0.3-SVN
- **WEB** [www.reactos.org](http://www.reactos.org)

ReactOS, обзор которого мы в последний раз делали в LXF61, в последние несколько месяцев переживал тяжелые времена. Один из постоянных разработчиков этого клона Windows с открытым кодом озвучил соображения о практиках инженерного анализа программ, подстрекнув крайне деструктивные дебаты в сообществе — о том, нужно ли дизассемблировать бинарные файлы Windows, заимствуя



ReactOS запустил *Paint Shop Pro*, а давно ли команда радовалась тому, что хоть *Notepad* работает!

оттуда идеи при перекодировании. Из-за этого часть разработчиков покинула проект, и в данный момент ReactOS перегруппировывается с целью аудита кода и внедрения новых практик реинжиниринга, чтобы предотвратить попадание в проект частей кода Windows.

Несмотря на весь этот хаос, ReactOS с момента нашего последнего обзора версии 0.2.4 сделал немало скачков. Большая часть разработки сосредоточена на низкоуровневых деталях реализации, таких, как ядро и синхронизация с последним релизом *Wine*, но главные прибавления версии 0.3.0 — стабильный вариант сетевого стека и поддержка USB. Наиболее серьезные достижения наблюдаются в поддержке приложений: ранее ReactOS мог запускать только несколько небольших приложений Windows, а теперь в нём могут работать *OpenOffice.org*, *Flash Player*, *SeaMonkey* от *Mozilla*, *MIRC*, *Unreal Tournament*, *Visual*

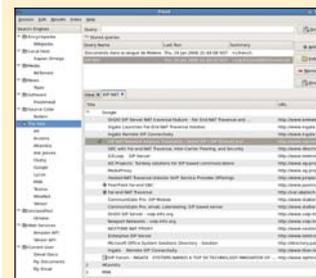
*Basic 6.0* и растущее число приложений для повседневной работы. Один из разработчиков даже заставил работать бета-версию *Internet Explorer 7*!

Мы надеемся, что разработка ReactOS продолжится в прежнем темпе. Проект пострадал из-за некорректных предположений о якобы содержащемся в нём «краденом» коде Windows, но на самом деле проблемы были вызваны отсутствием методологии реинжиниринга — а это совсем другое дело. Несмотря ни на что, ReactOS обладает невероятным потенциалом: свободная, открытая операционная система, способная запускать популярные приложения Windows, может сотворить чудеса на рынке настольных систем, и каждый ее релиз шагает далеко вперед. Если вы ни разу не пробовали эту систему, сейчас для этого наступило самое подходящее время. **LXF**

## ТАКЖЕ ВЫПУЩЕНЫ

Новые и обновленные программы, заслуживающие внимания...

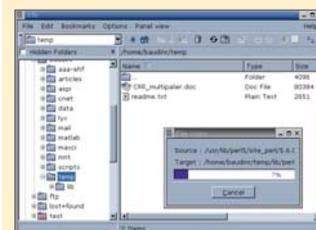
- **Kaffeine 0.8** Медиа-проигрыватель KDE <http://kaffeine.sourceforge.net>
- **Delsafe 0.2.0** Восстановление удалённых файлов <http://delsafe.cjb.net>
- **Rabbit 3.0** Прокси-сервер для web [www.khelekore.org/rabbit](http://www.khelekore.org/rabbit)
- **DVD-Baker 0.7.40** Создание слайд-шоу для DVD <http://dvd-baker.sourceforge.net>
- **Pinot 0.45** Система метапоиска для локальных и удалённых файлов <http://pinot.berlios.de>



- **GPass 0.5.1** Менеджер паролей для Gnome <http://projects.netlab.jp/gpass>
- **Hellanzb 0.9** Скачивание файлов USENET [www.hellanzb.com/trac](http://www.hellanzb.com/trac)
- **Alpy 0.1.3** Привязки Python для библиотеки создания игр Allegro <http://pyallegro.sf.net>
- **KEuroCalc 0.9.5** Конвертер валют <http://opensource.bureau-cornavin.com/keurocalc>
- **ECromedos 1.0** Издательская система для web и печати на основе XML [www.ecromedos.net](http://www.ecromedos.net)
- **Automanic 0.1.1** Гонки на выживание <http://automanic.sf.net>



- **Aria2 0.3.1** Менеджер закачки файлов <http://aria2.sourceforge.net>
- **WzdFTPd 0.7.0** Модульный FTP-сервер [www.wzdftpd.net](http://www.wzdftpd.net)
- **Subsonic 2.6** Медиа-стример для web <http://subsonic.sourceforge.net>
- **Fox 1.6.0** Библиотека GUI для C++ [www.fox-toolkit.com](http://www.fox-toolkit.com)



# Что такое...

# XGL?

Рабочий стол похож на кремовый тортик?  
Именно этого и хочет Пол Хадсон...

## >> ОК, я знаю, что такое X и догадываюсь, что 'gl' – от OpenGL...

Эй, не делайте за меня мою работу! Да, Xgl – это X Window System, работающая поверх OpenGL, графическая система с аппаратным 3D-акселератором.

## >> И что в ней хорошего?

Большую часть времени крутая графическая карта от Nvidia/ATI в вашем компьютере простаивает: с прорисовкой вашего 2D рабочего стола она справляется одной левой. Но она способна на большее, что мы и наблюдаем в Mac OS X: аппаратная прорисовка экрана создает феерические эффекты на рабочем столе.

## >> Похоже, что решение само ищет себе задачу.

Согласен, яркие эффекты уже утратили новизну, но вы должны признать, что это быстрее способ произвести впечатление на пользователей систем помельче. Кроме того, Xgl имеет и другие преимущества. Во-первых, Xgl формирует изображение «за кадром», а затем в один ход отображает его на экране, без всяких мерцаний и разрывов синхронизации при перетаскивании окон по экрану. Кроме того, вы получаете правильную обработку теней, так как Xgl поддерживает настоящее наслаивание окон.

## >> Смаживает на расширение Composite, выпущенное давно, но нигде по-настоящему не заработавшее.

Вот именно, Composite это самое и делало. И ничего больше; а Xgl содержит всю функциональность Composite, плюс еще многое другое. Опять же, Вы правы: Composite так и не достиг 100% стабильности, и, насколько мы знаем, не включен по умолчанию ни в одном дистрибутиве. Проблема в том, что изменения от Composite были скорее внешними для X, а не происходили в ядре, а значит, Xgl им не догнать.

## >> Что еще умеет Xgl?

В Xgl ваш рабочий стол проецируется на грань куба, то есть вы можете переключаться с одного виртуального стола на другой, просто вращая куб вокруг оси. Для больших эстетов, Xgl позволит масштабировать экран, что-

бы поближе разглядеть его элементы; окна у него колеблются при перемещении; имеется даже функция Exposure в стиле OS X – уменьшить ваши окна так, чтобы все они поместились на экран и легко было выбрать то, с которым нужно работать.

## >> Вращающиеся кубы... колеблющиеся окна... падающие тени... зачем? Перевод ресурсов!

Оказывается, заставив видеокарту взять на себя всю эту работу, мы получаем ускорение, потому что она освобож-

дает ваш CPU от каждодневного тяжелого труда по прорисовке GUI. Легко доказать, что эти возможности действительно приносят реальную пользу: некоторым пользователям трудно разобраться в виртуальных рабочих столах, и представление их как граней куба поможет облегчить понимание.

Аналогично, функция Exposure поможет людям найти нужное приложение, а масштабирование – удобное решение для программ или web-страниц, использующих мелкие шрифты. Только эффект медузоподобного колебания окон лишен практического смысла, но вы должны признать, что он добавляет шарма рабочему столу Linux. Есть еще одна по-настоящему полезная возможность, о которой мы не упомянули: комбинация Alt+Tab отображает экранный снимок каждого приложения, а не просто его иконку, и гораздо легче найти искомое.

## >> Раз X-сервер занимается подобной магией, он явно идет вразрез с основной философией Unix? Ну, что каждое приложение делает свою работу и делает ее хорошо...

Не будем упрощать. Xgl – это X на OpenGL, но на этом участие X и кончается. Поверх Xgl находится оконный менеджер-компоновщик под названием Compiz, вот он-то и создает все эффекты. В сущности, Compiz играет роль KWin и Metacity, и

хорошо играет. Более того, Compiz позволит вам выбрать, какие эффекты вы хотите включить, то есть вы можете отказаться от колебания окон и использовать только те эффекты, которые вам полезны. Также он предоставляет гибкую настройку – если хотите додекаэдр вместо куба для виртуальных столов, просто поменяйте параметры!

## >> Какие дистрибутивы могут работать с Xgl?

Xgl пока что новая технология, поэтому имеет слабую поддержку в дистрибутивах. Прямо сейчас вы можете использовать Xgl в Ubuntu Dapper Drake, SUSE 10.1 и SUSE Linux Desktop 10. Мы ожидаем, что поддержка Xgl будет нарастать, в первую очередь благодаря желанию пользователей видеть красивые эффекты в своем дистрибутиве. Поддержка в Debian ожидается не позднее 2010 года.

## >> А если я просто хочу попробовать Xgl?

Некоторые добрые люди уже выпустили Live CD с Xgl в дистрибутиве Koroгаа (см. Distrowatch, LXF79). Он базируется на Gentoo, но загружается в полноценный Xgl с полной поддержкой карт Nvidia и ATI. Вы можете установить Koroгаа с Live CD, но дистрибутив еще не вышел из бета-статуса, поэтому мы рекомендуем пока воздержаться от этого. Мы ожидаем, что Ubuntu Dapper Drake Live CD до известной степени

«ДАЖЕ EXPOSE ОТ MAC OS X НЕ ВЫГЛЯДИТ ТАК КРАСИВО, КАК ВЕРСИЯ XGL.»

поддержит Xgl, но, вероятно, без предустановленных драйверов видеокарт, и Xgl станет тормозить.

## >> Чудеса! Я тут подумал, что... постоите...SUSE, Ubuntu... а как насчет Fedora?

Ах да, Fedora. Разработчикам Fedora не сильно понравился Xgl. В частности, они столкнулись с тремя проблемами: во-первых, последние месяцы разработки Xgl проходили внутри Novell, без какой-либо поддержки со стороны сообщества. Во-вторых, Xgl не просто изменяет несколько строк в X, а вносит значительные изменения. И, наконец, Xgl в качестве оконного менеджера использует Compiz, а не KWin и Metacity.

Поэтому разработчики Fedora создали альтернативу Xgl и назвали ее AIGLX, и это значительный шаг на пути эволюции к X на базе OpenGL – она оставляет нетронутой архитектуру драйверов, позволяя вам переключаться между GL- и не-GL-режимами.

## >> Неужто произошло ветвление X на OpenGL?

Это возможно, но маловероятно. Команды разработчиков Xgl и AIGLX вместе работают во многих областях, а также обмениваются кодами. Есть надежда, что скоро они так



сблизятся, что останется потребность только в одном из них, и произойдет слияние проектов.

Однако сейчас это определено два разных проекта. Фирма Nvidia заняла странную позицию: она активно препятствует принятию Xgl, аргументируя, что все 3D-эффекты, осуществляемые с помощью Xgl, можно достичь с помощью эволюционного пути, выбранного разработчиками AIGLX. Так или иначе, все разработчики Xgl и AIGLX и Nvidia работают в одном направлении: все они хотят ускорить прорисовку X, переложив работу на оборудование, и в конечном счете Linux от этого выиграет. То есть в любом случае мы гарантированно получим превосходное 3D-решение.

**>> Значит, Fedora включит AIGLX в очередной дистрибутив?**

Он уже здесь! Fedora Core 5 поставляется с готовым к использованию AIGLX, и если у вас есть DVD этого номера журнала, вы можете попробовать его в действии хоть сейчас. FC5 сумела опередить выпуск SUSE 10.1 больше чем на месяц, и вы, наверно, обнаружите, что AIGLX уже завоевывает сообщество.

**>> Версия Fedora с такими же 3D-эффектами?**

AIGLX работает поверх существующей инфраструктуры, поэтому, в отличие от Xgl, Compiz не вмешивается в работу системы. Однако новый код Metacity содержит кое-какие базовые эффекты, а так как архитектура X не поменялась, то дело за тем, чтобы добавить в Metacity новые. Некоторые хакеры уже ухитрились посадить Compiz поверх AIGLX, и он неплохо работает.

**>> Можно ли сравнить Xgl/ AIGLX с OS X и Windows Vista?**

Ну, Windows Vista еще не вышла (и не будет доступна до 2007 года), поэтому в плане доступности X на OpenGL безоговорочно побеждает! В плане графических возможностей, X на OpenGL тоже опережает других кандидатов. Рабочий стол в виде куба – оригинальная идея, и даже OS X Expose не так красиво выглядит, как ее Xgl-реализация. Значит, у Linux появилась еще одна привлекательная черта, и она привлечет тех пользователей, которые любят работать в приятной обстановке.

**>> Куда же мы движемся?**

Будущее зависит от разработчиков, которые хотят сделать надстройку над архитектурой X – 3D-эффекты, которые они готовят для Compiz, определяют, действительно ли получится ускорение в работе или это не более чем красоты.

Однако разработчикам X остается еще много работы, в особенности в области улучшения совместимости с видеокартами. Кроме поддержки старых карт, мы хотели бы видеть появление, например, пиксельных шейдеров – если вам случалось перетаскивать виджет Dashboard на рабочем столе OS X, вы поймете, почему! А может, мыждемся двусторонних окон, чтобы на обратной стороне что-нибудь прятать – лично мы хотели бы, чтобы туда убрались диалоги Свойства всех приложений. Короче, будущее определено за 3D: Xgl/ AIGLX уже здесь, хотя об окончательном стандарте можно лишь гадать. **LXF**



# Ангел Apache

Многие из нас воспринимают лидерство Apache среди серверов как нечто само собой разумеющееся. Брайан Белендорф (Brian Behlendorf) рассказывает *Linux Format* о том, как сложилось такое положение дел.

**Т**рудится ли он на музыкальном фестивале Burning Man далеко в пустыне Блэк-Рок, Невада, или над web-дизайном или ПО для контроля версий в Сан-Франциско, Брайан Белендорф – всегда новатор, переживший безумие *fin de siècle*, сгубившее так много Интернет-предпринимателей в конце девяностых. Его вера в открытое сообщество и в открытую модель разработки помогла ему добиться успеха в одной из первых фирм, занимающихся web-дизайном, Organic Online, а затем – в CollabNet, основанной в 1999 с целью продажи ПО с открытым кодом крупным предприятиям.

**Грэму Моррисону** удалось поговорить с Брайаном о том, как *Apache* возник из первичного бульона NCSA, и о том, какую роль будет играть Linux в будущем в разработке программного обеспечения.

**Linux Format: Не могли бы Вы назвать Вашу роль в CollabNet?**

**Брайан Белендорф:** Технический директор (CTO). Я не указываю в моей визитной карточке, что я – основатель.

**LXF: А почему на сайте CollabNet Ваше имя – второе?**

**ББ:** Ну, на самом деле главный (CEO) – Билл Портелли [Bill Portelli]. Я нашел его через три месяца после основания фирмы, так что он – со-основатель. Я был счастлив найти человека с деловым чутьем, который сумел найти людей для работы по продажам и проделать всю муторную процедуру создания фирмы CollabNet – а мне дал возможность обдумать, что мы будем продавать и как нам поведать об этом всему остальному миру, и убедиться, что продукт, который мы выпускаем, востребован.

**LXF: И откуда же все началось?**

**ББ:** Если немного перевести стрелки назад, в самые первые дни существования *Apache*, то начинали мы это все в количестве нескольких человек с создания web-сайтов,



и нам был нужен [сервер, который мы могли бы контролировать]. Я очень быстро передал разработку тем, у кого это лучше получалось.

**LXF: Ну, так говорит практически каждый разработчик!**

**ББ:** [Смеется] Наверное, открытому сообществу свойственна патологическая скром-

никогда не было штатных сотрудников, там всегда все делали добровольцы. И я был командой инфраструктуры, я был тем парнем, которому, когда машина зависала, надо было ехать на место, нажимать кнопку перезагрузки и заставлять ее снова работать.

Занимаясь этим, я все больше ощущал недовольство – потому что понимал, что мы

ной разработки бизнес-плана. На самом деле, если кто так не поступал, его и крутым не считали.

Мы решили собирать идеи, и эта (CollabNet) была одной из них. Заключалась она в том, чтобы собрать все эти инструменты и предлагать их в качестве сервиса фирмам, работающим с открытыми проектами. Одни из первых переговоров мы

**ОБ ОТКРЫТОЙ РАЗРАБОТКЕ НА ОСНОВЕ СОТРУДНИЧЕСТВА**

«Казалось, что работа над проектом должна иметь форму беседы.»

ность. Я расширил сферу деятельности, уделяя больше внимания внешним системам, поскольку я был системным администратором – я должен был быть уверен, что почта работает хорошо, что с деревьями CVS все в порядке и повреждений нет. Сбор сообщений об ошибках работал, и я создавал проекты, добавляя пользовательские учетные записи и увязывая все это проводами, скотчем, скриптами оболочки – ну, и хакерствовал по этому поводу, потому что на такую работу жалко тратить время.

И всегда приходилось что-то делать даже в свободное время, потому что до CollabNet я работал в фирме, разрабатывающей web-сайты, под названием Organic [Online], а в *Apache*

с командой могли бы сесть и написать что-нибудь получше с самого начала. Или даже не совсем сначала, но хотя бы пересобрать куски, используя настоящую систему отслеживания пользователей, разрешений и проектов, и превращая все эти инструменты в нечто более тесно интегрированное. Это было бы круто и могло бы содействовать развитию *Apache*; а еще это могло бы упростить для многих фирм начало работы над открытыми проектами.

**LXF: Так и родилась CollabNet?**

**ББ:** Где-то году в 1998 я устал от Organic и решил поискать что-нибудь новенькое. Я поплакался Тиму О'Рейли (Tim O'Reilly), и он мне сказал: «А не приехать ли тебе к нам и не стать предпринимателем, проживающим по месту службы?» Это были 98–99 годы – деньки в стиле «давай-давай» в Кремниевой Долине – каждая проблема казалась деловой возможностью, достойной немедлен-

провели с Sun, которые хотели создать новехонькие сообщества в стиле *Apache* вокруг первого NetBeans, а затем – *OpenOffice.org*. Вот эти проекты мы запустили в 1999, и до сих пор они весьма активны. Если вы посмотрите на *OOo*, на инструменты, которые поддерживают эту общность – для CVS, отслеживания ошибок, публикации контента, координация релизов и т.п. – все эти инструменты – CollabNet. Sun нам платит, и мы занимаемся этой работой на общественном сайте.

**LXF: Как Вам с CollabNet удается теперь убеждать непосвященных в живучести движения Open Source?**

**ББ:** Во-первых, я рассказываю, что 11 лет назад те из нас, кто пользовался web-сервером NCSA, делали это не в качестве хобби. Не все, но большинство из нас использовало этот сервер, чтобы начать свой бизнес. >>



« Не важно — web-дизайнерскую компанию, как это было у меня, или Internet Movie Database, или web-мастер из MIT... Все мы использовали его для бизнеса, и мы собрались вместе, чтобы написать код.

**LXF: И как Вы сумели разглядеть потенциал движения так рано?**

**ББ:** Подозреваю, что я просто не знал ничего лучше. Я был двадцатилетним почти- выпускником Университета Беркли, и львиную долю свободного времени проводил, исследуя потрясающе крутую штуку под названием Сеть. Я был очарован работой системы — казалось, в ней нет центра управления. У моего провайдера Prodigy не было ничего похожего ни на шлюз, ни на центрального координатора, и тем не менее все работало. Я просто отпал, я читал ночи напролет все, что мог найти, просматривал ссылки на рабочие группы IETF. В итоге я присоединился к двум парням, рабочей группе HTTP, и сидел и смотрел, как Тим Бернерс-Ли (Tim Berners-Lee) и Мартин Глисон (Martin Gleeson) обсуждают способы общения Mosaic с web-сервером CERN. Или о рабочей группе HTML — должен или не должен тэг <img> быть контейнером?

По-моему, я не знал ничего лучше, но мне казалось, что работа над дизайном должна иметь форму беседы, а не то, что кто-то с горы провозглашает десять заповедей. Это было очень прочное впечатление, и в ранние дни Apache многие из нас принимали активное участие в рабочих группах HTTP. Нам нужно было вести свой бизнес, нам нужен был web-сервер получше, мы переживали, что Netscape способствует «утечке мозгов» из NCSA, и мы объединяли наши усилия и думали: «Эй, глянем-ка, нельзя ли сделать лучше. Может быть, нам все равно придется покупать коммерческую версию, но пока это лучше, чем вообще

ничего».

Однако мне думается, что доля идеализма тоже присутствовала. Мы в Apache очень хотели увидеть эталон применения стандарта HTTP, который мог бы считаться эталоном качества. Нас беспокоило, что какая-нибудь фирма сможет стать владельцем и клиентской, и серверной стороны web. Случись такое, это было бы равнозначно власти Microsoft на рабочих столах. И побудительной причиной вовсе не была неприязнь к корпорации, причина была в том, что нам нравилась эта штукавина под названием «сеть», без шлюзов, без заглушек, и мы хотели, чтобы она оставалась такой и впредь. Мы и подумать не могли, что 70% рынка станут нашими...

**LXF: Вы выбрали лицензию под влиянием таких мыслей?**

**ББ:** Мы обсудили этот вопрос на самой ранней стадии. Изначальная лицензия [Apache] на базе кода NCSA была невероятно либе-



мы верили в пряник, а не в кнут. Лицензия GPL гласит: «Вы должны сотрудничать, вы должны делиться своим кодом», а лицензия BSD гласит: «Делайте, что хотите», но мы чувствовали, что фирмы в конце кон-

росы о смысле лицензии, если я использую приложение с лицензией GPL, а оно обращается к другому приложению. Надо ли открывать доступ ко всему, или только к той части, которая относится к GPL?

**ПРО GPL V3**

«Фирмы воздерживаются от ПО под GPL из-за ее неопределенности.»

ральна — она гласила: «Вот вам код, делайте с ним все, что хотите. Можете создавать производные, можете использовать его с любыми целями, но если вы его передадите кому-то еще, не обвиняйте нас, если он не будет работать. Только обязательно сообщите остальным, что код, который вы им передаете, берет начало в NCSA».

Эта лицензия казалась нам справедливой. И поэтому лицензия Apache в основном использовала ту же концепцию. То же самое с BSD. Даже в те дни, в самом начале, мы думали: «А вдруг какая-нибудь фирма возьмет Apache и вставит его в какую-нибудь коммерческую разработку?» А потом решили: «Ну и что? На самом деле, это будет даже хорошо, потому что поможет укрепить позиции HTTP как открытого стандарта. Используя Apache, они не смогут найти отговорки для отказа соответствовать стандарту». Им не удастся сослаться на излишнюю сложность — как это делали большинство фирм. И более того,

цов снова обратятся к нам и поделятся своими заплатками, и даже своими улучшениями.

**LXF: Каким, по Вашему мнению, будет воздействие GPL версии 3 на ПО с открытым кодом?**

**ББ:** Я всегда предпочитал лицензии не в стиле GPL, а в стиле Беркли — это как раз «кнут против пряника». Боюсь, что любая формулировка, расширяющая степень открытости кода, только усложнит процесс привлечения новых участников в движение Open Source. Мне симпатична позиция [Ричарда] Столмена. Думаю, что в конечном итоге — лет через 30–50, когда программы будут в каждой дверной ручке и мы начнем программировать жизнь на биологическом уровне — доступ к исходному коду будет равнозначен гражданским правам. Так что насчет отдаленного будущего, по моему, он попадает в точку.

А в недалеком будущем мне бы хотелось привлечь как можно больше людей и убедить их в преимуществах открытого доступа к коду. Я предпочитаю лицензии попроще. Предложенное решение [для GPL v3] — что если у приложения имеется ссылка, говорящая: «загрузите код», то эта ссылка должна быть сохранена — по моему, верное. Но мне кажется, что возникнут воп-

Фирмы пытаются понять, много ли им придется отдавать, и держатся подальше от ПО с лицензией GPL из-за ее крайней неопределенности. И мне думается, что многим это не дает присоединиться к нам. Надеюсь, что GPL v3 исправит это положение вещей.

**LXF: А Вас не беспокоило, что фирмы могли бы переделать Apache под свои стандарты?**

**ББ:** Ну, сделали бы они это, и что? Пусть Netscape или даже Microsoft взяли бы код, мы бы не переживали, потому что наличие процесса открытой разработки и открытой лицензии означает, что все равно большая часть деятельности вернется к нам. С масштабом Интернета нам и не нужно было, чтобы такое количество народа кинулось помогать нам с усовершенствованиями.

Мы сделали еще одну вещь, мы подумали: «Что это за комбинация процессов, отношений и инструментов, которая помогла нам создать наш web-сервер?» Мы реплицировали ее в другие программные проекты и в 1998 создали Фонд Программного Обеспечения Apache (Apache Software Foundation). Сейчас в работе 25 различных проектов высшего уровня. Многие из них возвращаются в CollabNet. В 1998-м я видел эту потребность в более понятных компаниям инструментах, потребность в понимании



методов разработки или ориентированность на открытость, прозрачность.

**LXF:** Как Вы думаете, куда движется web?

**ББ:** Сейчас разрабатывается много интересных приложений, и большое внимание уделяется совместному написанию приложений. Мне очень нравится программа для поиска квартир, соединение Craigslist с Google Maps. Это интересно. Это признание факта, что ценность базы данных не создается одним человеком в стиле советского командно-административного подхода к экономике, нет, она динамически генерируется через объединение деятельности многих и многих людей в более интересную комбинацию. Одна из главнейших ценностей Amazon — его читательские комментарии; главная ценность Wikipedia в том, что она создается огромным числом разных людей. Это не точка зрения Джимми Уэйлса [Jimmy Wales, основатель Wikipedia] на каждую словарную статью; это — сочетание тысяч точек зрения. Я думаю, самые интересные приложения будут как раз слиянием снизу доверху таких вот данных. Мы такого еще не видели.

**LXF:** А у Вас идеи имеются?

**ББ:** [Смеется — мы поняли это в смысле «Знаю, но не скажу».]

**LXF:** Приемлемы ли, по-Вашему, коммерческие web-продукты типа Flash?

**ББ:** У меня предубеждение против них: эти ребята будут целую вечность создавать плагин для Linux. А для FreeBSD, на которой работаю я, и того дольше. Без исходного же кода всегда сложно предотвращать зависания и исправлять прочие проблемы. А ведь примерно одна из двадцати анимаций Flash либо подвешивает компьютер, либо его тормозит.

Проблема не в Macromedia per se [самой по себе — прим. пер.]. В качестве среды программирования мы выбираем Ajax как более естественный интерфейс. Анимация Flash обладает фиксированным окном, в котором производятся действия, а анимация Ajax (подобно HTML), похоже, умеет вписываться в то окно, которое вы сами предоставляете, и это дает больше гибкости.

**LXF:** Решение с открытым кодом потребует много времени...

**ББ:** Ну, это инженерный анализ. Даже если это и возможно, все равно вы рискуете вызвать гнев DMCA. Ajax, с другой стороны, создан с применением всех открытых технологий, и когда у вас есть две эквивалентные системы, выигрывает обычно та, у которой открытый код.

Если посмотреть на ситуацию шире, то мы вряд ли увидим конец проприетарного ПО. Как и Apple, мы продаем проприетарное ПО, которое нравится людям, но мы обязаны постоянно двигаться против течения. Никому не хочется оказаться в положении Netscape, продававшей web-серверы — конкуренты Apache. Нужна бизнес-модель, хотя бы и просто набор сервисов, но предлагающая больше, чем люди могут получить самостоятельно. Интересно будет взглянуть, куда пойдет индустрия ПО в целом в ответ на это.

**LXF:** Вы считаете, что патенты станут большей проблемой?

**ББ:** Вот что самое смешное: знаете, какая фирма больше всего пострадала из-за патентов в прошлом году?

**LXF:** Гм... уж не Microsoft ли?

**ББ:** Нет, всего лишь Eolas с \$500-миллионным патентом [Апелляционный суд США частично отменил вердикт и вынес постановление о пересмотре дела]. А вторая по счету крупная фирма? RIM BlackBerry [сумма, которую RIM без суда выплатила NTP Inc, превышала сумму по делу Eolas].

Я очень рад, что есть организации вроде Европейского Фонда Свободного ПО (Free Software Foundation Europe) и других, которые борются против введения более жестких законов о патентах, и нам в Штатах надо больше двигаться в том же направлении. Думаю, что главные соратники в этой борьбе — крупные держатели патентов на интеллектуальную собственность (IP), которые поймут, что если слишком закрутить гайки, это будет игра для дураков.

Я не против утверждения, что иногда патенты имеют смысл. Бывают яркие идеи, реализуемые методом долгих проб и ошибок, и должен быть вознагражден изобретатель, а не обязательно фирма, которая платит ему зарплату. Нужно в какой-то степени защитить инвестиции в исследования и разработку. Но самые новаторские достижения в ПО за последние десять лет не являются запатентованными технологиями. Они не являются результатом многолетнего сидения в лаборатории, [они] пришли из сообщества динамической разработки.

**LXF:** Считаете ли Вы, что разница в развитии американского и европейского ПО увеличится?

**ББ:** Меня интересует, как пойдут дела с интеллектуальной собственностью в Индии [у CollabNet имеется офис в Ченнае]. Я был там на прошлой неделе и прочел статью о некой фирме Cipla, объявившей, что они готовы начать производство миллионов доз Tamiflu, лекарства от гриппа. Roche



Pharmaceuticals, владелец патента на Tamiflu, ежегодно может выпускать только ограниченное количество лекарства, а время поставки, если сделать заказ в Roche сегодня, составляет 18 месяцев. Cipla заявила, что это — вопрос здоровья нации, угроза пандемии, поэтому они и начали выпуск лекарства. [Roche продает лекарство по всему миру, но не обладает патентом на Tamiflu в Индии.]

Интересно будет посмотреть, что примет по этому вопросу индийское правительство, потому что оно не может получить свою вакцину от гриппа, а у них есть миллиард человек, о которых им надо беспокоиться, в то время как Roche может изготовить только 12 миллионов доз в год. В некоторых местах патентное законодательство не будет изменяться постепенно. Кое-где настолько уверены в преимуществах более жесткого патентного законодательства, что им нужен прямо-таки Перл-Харбор, чтобы они передумали [налет на Перл-Харбор решил вопрос о вступлении США во Вторую мировую войну, — прим. ред.]. Мне бы не

хотелось ничего подобного, но может случиться, что ситуация с Tamiflu заставит людей осознать, что жесткое патентное законодательство может стоить жизней.

**LXF:** Последнее: Вы все еще играете электронную музыку?

**ББ:** В стиле Джона Пила [John Peel] — я играю на вечеринках, для друзей, где собирается от 20 человек до...ну, самое большее в прошлом году было человек 600. Это такие вечеринки в стиле андерграунд, где мы разбиваем лагерь, устанавливаем аппаратуру в горах или на пляже, и играем. **LXF**

Другие высказывания Брайана о работе с Sun, HP и, э-э... Министерством обороны — на сайте [www.linuxformat.co.uk/mag/behendorf.html](http://www.linuxformat.co.uk/mag/behendorf.html).

# Libre Graphics Meeting

Adobe, Corel, Quark и другим стоит оглянуться – свободное программное обеспечение вот-вот догонит их продукцию. Ник Вейч посетил первую конференцию Libre Graphics, знаковое событие для всего мира open source.

**L**inux. У большинства людей это слово ассоциируется с безумно высокотехнологичным миром Интернета. В голову немедленно приходят образы «лезвийных» серверов, кластеров, Apache и потоков кода. И пингины. В конце девяностых доминирование свободного ПО среди технологий, определяющих развитие Интернета, стало почти неизбежным – не только Linux, но весь массив технологий, обеспечивающих развитие сети, полагается на открытый код и открытые стандарты.

Но Linux – вовсе не платформа, ориентированная всего на одну задачу, и вводная конференция Libre Graphics, проходившая на юге Франции, в Лионе, в школе Supérieur Chimie Physique Electronique (CPE), это отчётливо продемонстрировала.

Организовал конференцию разработчик Gimp Дейв Нири (Dave Neary). Он затеял просто встречу с другими Gimp-программистами, но конференция довольно быстро переросла в рассмотрение графических инструментов вообще – разработчики других проектов ухватились за возможность повидать коллег, обменяться идеями и обсудить общие проблемы.

Три дня место проведения мероприятия было забито программистами, пользователями и просто зеваками. Презентации в главном зале прошли с аншлагом и затронули новые технологии и обновление проектов и тренинги на тему, как можно помочь проектам.

## Прорыв

Хотя полезным было все, неформальное общение посетителей между заседаниями, вероятно, в наибольшей степени способствовало успеху. В коридорах, во время перекуров на улице, в обед и, конечно же, за ужином шли разговоры об углублении сотрудничества, объединении знаний и ресурсов для решения общих проблем, обмен дружескими советами в диапазоне от того, какие библиотеки использовать, до эффективных приёмов работы.

Впрочем, это был не просто «слёт умников»: многие из представленных проектов – действительно серьёзные графических инструменты. И большинство из них – кроссплатформенные: Inkscape, Gimp, Scribus, Blender...

Далее мы подробнее рассмотрим некоторые из этих приложений, продвигающих Open Source как выбор для профессиональной графики и дизайна. Сложившийся рынок принадлежит таким фирмам, как Adobe, но молодёжь, несомненно, наступает на пятки. И вполне возможно, когда-нибудь, в не столь далёком будущем, эта конференция будет рассматриваться как поворотная точка, когда графические инструменты с открытым кодом стали взрослыми...



Говорили и об обновлении проектов, и о том, как лучше им помочь.



Смешались в кучу... пользователи, разработчики.



Libre graphics



**Интервью LXF**  
Тема: Libre Graphics Meeting  
Дата: 17 марта 2006 г., 11.49  
Место: Перед зданием CPE  
Лицо: Дейв Нири, организатор конференции

**LinuxFormat:** Итак, почему вы организовали LGM?

**Дейв Нири (ДН):** Ну, есть и другие конференции, но они фокусируются не на том. Это мероприятия вроде Fosdem [Free and Open Source Software Developers Meeting, встреча разработчиков свободного ПО, — прим. пер.], слишком общего, или Akademy [всемирная конференция сообщества KDE, — прим. пер.]. Но Gimp, да и все графические приложения несколько выпадают из этого — Blender не использует GTK или Qt, а Gimp — хотя и приложение Gnome, но фактически не связан с этим проектом.

Gimp используется и в KDE, Scribus работает на Gnome. Приложения, представленные здесь, кросс-платформенные — нам нужно большее, чем специализированная конференция Linux. К тому же, у графики свои причуды: мы должны думать о «людях в оранжевых очках». Люди, в которых мы заинтересованы, наши пользователи — графические дизайнеры и арт-директора, они далеки от сферы открытого ПО. Нам нужно делать что-то конкретное, чтобы «зацепить» этот сорт людей.

**LXF:** То есть конференция задумана, чтобы убедить людей использовать открытые инструменты?

**ДН:** Мы преследуем две цели: первая — собрать вместе участников этих проектов и наметить наше общее будущее — согласованное направление развития для всех этих приложений. Вторая — продемонстрировать открытые программы публике, которая, возможно, вообще о них не знает, людям, которые работают с графикой. Нам нужно показать, что [даже если] мы не удовлетворим все их нужды прямо сейчас, у нас есть функциональные, стабильные и многообещающие приложения. И что у нас есть сообщество вокруг этих приложений, очень дружелюбное. Нам нужно больше художников, вовлеченных в свободное ПО.

**LXF:** Что мешает людям использовать свободные программы для серьезной графики?

**ДН:** Причин несколько. Есть неписанный кодекс поведения, порой не совсем понятный. Это барьер сообщества. Есть технологические барьеры. Есть вещи, на которые распространяется понятие «интеллектуальная собственность». Это такая штука, которая мне ненавистна, но под действие авторских прав, торговых марок, патентов много чего подпадает. То есть вещи вроде каталога цветов Pantone, защищенного авторскими правами. Третий барьер — рабочая сила. Кое-что мы просто не делаем. Мы говорим о художниках — почему художники так охотно пользуются нашими программами. Есть вещи, которые Gimp не делает, но они нужны художникам, кинопродюсерам и людям, занимающимся допечатной подготовкой.

**LXF:** Некоторые считают, что кросс-платформенная природа этих инструментов — это плохо; что свободное ПО должно оставаться на свободной платформе.

**ДН:** Да, есть такие люди. Но я не из их числа. Я думаю, что наша цель — чтобы все использовали свободные программы, включая операционную систему. Тут вы можете спросить: «А не идет ли портирование приложений GNU на Windows вразрез с этой целью?» А по-моему, будь я пользователем Windows, я не стал бы ничего менять одним махом. Если кто-то даст мне CD с текстовым процессором, звуковым редактором, графическими инструментами и web-браузером, я могу использовать всё это в Windows и не беспокоиться об откате назад, если мне что-то не понравится. И вот вы вдруг получаете кого-то, кто использует свободные программы, но работает под Windows. Всё, что ему остаётся сделать — сменить операционную систему. Вы помогаете людям становиться на путь просвещения.



**ЗАМЕТКА НА ПОЛЯХ:**  
Если вам для работы требуется корректное управление цветом, обратите внимание на проект Little CMS по адресу [www.littlecms.com](http://www.littlecms.com)

# Scribus

Открытая настольная издательская система становится профессиональной.

**В**о многом, Scribus – это стремительная история успеха открытого графического инструмента. Да, *Gimp* восхитителен, *Inkscape* обладает потенциалом – но *Scribus* поставил перед собой задачу, появившись ниоткуда, стать серьёзным, профессиональным решением для настольных издательских систем (ИС), и, следуя традиционным стандартам разработки программ, в два счёта этого достиг.

Фактически, последний выпуск *Scribus* (1.3.3) предоставляет возможности, превосходящие стандарты профессионального программного обеспечения для ИС, например, генерацию штрих-кода без привлечения стороннего ПО. Для коммерческой ИС такое немислимо – за пять лет пройти путь от первой строки кода до надёжного, профессионального инструмента.

## Типы печати

Как и со всеми освещёнными здесь проектами, похоже, что выбор открытой модели разработки способствовал изрядной прозорливости *Scribus*. Достаточно стандартная история – нужный человек оказался в нужное время в нужном месте.

Везение это или нет, но команда *Scribus* сумела собрать вокруг себя очень талантливых и компетентных людей из мира печатных изданий.



Да, Scribus умеет выполнять цветоделение, используя последние версии ghostscript.



В Scribus 1.3.3 уже есть ряд простых эффектов изображения, но будущие версии могут включить GEGE.

*Scribus* увидел свет как проект немецкого программиста-самоучки Франца Шмида [Franz Schmid]. Его первоначальная гениальная идея была создать программу вёрстки, базирующуюся на технологическом процессе PDF. Еще в 2001 году было видно, что PDF может стать *lingua franca* [принятый язык межнационального общения, – прим. пер.] для печати, и так оно и вышло. Почти все крупные издательства, а также и принтеры, теперь работают преимущественно в формате PDF, который генерируется, проектируется и выводится прямо на пластины для типографии.

Шмид, очевидно, прекрасно разобрался в спецификации PDF, потому что *Scribus* выполняет фантастическую работу по генерации этих файлов – даже обрабатывает ряд интерактивных и презентационных функций, которые не используются для печати и практически не генерируются другими программами, за исключением полной версии Adobe Reader. Мы надеемся, что и LinuxFormat помог популяризации *Scribus*, поскольку мы включили его в наш раздел **HotPicks** (это один из немногих проектов, фигурировавших там неоднократно, в *LXF36* и *LXF50*).

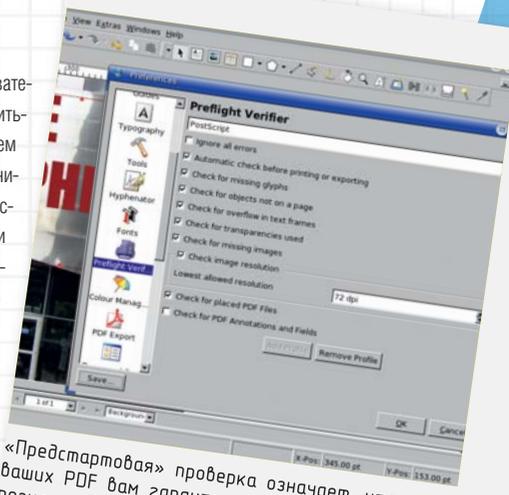
Однако по мере взросления программы маленький проект Шмида становился всё сложнее и запутаннее. Это само по себе привело к множеству изменений – значительную часть кода пришлось переписать или, по крайней мере, переместить. То, что имело смысл для небольшого проекта, работающего с маленькими документами, может не годиться, если пользователи намерены «отстукивать» 200-страничные книги.

## Усложнение

С последней серией 1.3 пользователи *Scribus* теперь могут насладиться цветовой заливкой, управлением цветом, правильным цветоделением и т.п. В общем, *Scribus* вырастает в новую систему обработки текста; тут значительно увеличена скорость обновления изображений и добавлена поддержка Unicode. Полезные возможности простираются аж до назначения поднаборов глифов [glyph – визуальный образ символа шрифта, – прим. пер.]. Много народу страдало по версии *Scribus* для

командной строки, чтобы конвертировать документы *Scribus* в PDF без посредства графического интерфейса, и что вы думаете? Это не такая уж далёкая перспектива!

Чувствуется, что по крайней мере в этом сегменте настольных систем программы с открытым кодом постепенно меняют позицию «как ни странно полезных» на звание профессиональных инструментов. На Libre Graphics команда продемонстрировала *Le Tigre* – созданную в *Scribus* еженедельную французскую газету, выходящую тиражом более 12 тыс. экземпляров. Она не первая и, конечно, не последняя.



«Предстартовая» проверка означает, что от ваших PDF вам гарантирован превосходный результат.

## ЗАМЕТКА НА ПОЛЯХ:

*Scribus* теперь умеет устанавливать поля на ваших документах автоматически, через CUPS. В диалоге настроек документа щёлкните на кнопке **Printer Margins** и укажите нужный принтер.

**Интервью LXF**  
Тема: Scribus  
Дата: 17 марта 2006 г., 18:14  
Место: большой чулан под лестницей, здание CPE  
Лица: основные разработчики Scribus

**LinuxFormat: В чём причина успеха Scribus?**  
Питер Линнелл [Peter Linnell] (ПЛ): Думаю, в том, что на командном уровне мы работаем очень хорошо...

Крейг Ринджер [Craig Ringer] (КР): Scribus – довольно закрытая система разработки. Есть основная команда, поддерживаемая добровольцами и переводчиками; некоторые добровольцы, поддерживаемые добровольцами в команду. Отбор довольно жёсткий. Анонимный доступ к внутреннему CVS не предоставляется.

ПЛ: У нас есть внутренний CVS для команды. Насчет кода мы прямо-таки параноики. Внутренний CVS – тот, с которым мы можем экспериментировать, чтобы люди могли доверять коду, выполняя реальную работу. Мы пытаемся поддерживать систему, где «слепок» CVS всегда можно собрать и использовать. Конечно, ошибки всё равно проскальзывают, но это уже совсем другое, чем пригодность для работы вообще.

**LXF: Всё-таки Scribus прошёл длинный путь за короткое время – намного быстрее многих других проектов.**  
ПЛ: Мы немного старше, чем обычные хамеры, работающие с открытым кодом. У Крейга значительный производственный опыт, Льюис много лет проработал в профессиональной допечатной индустрии. У нас довольно много людей, занимающихся издательской деятельностью, которые способны протестировать различные функции программы – либо пользовательский интерфейс, либо качество результата, либо цвета, либо PDF. Также у нас есть пользователи, располагающие высококачественным печатным оборудованием, и либо мы даём им файлы, либо они печатают свои собственные тесты. Это помогает нам сохранять высокий уровень качества.

Другая причина – команда разработки очень гармонична. Требуется некоторый опыт в допечатной подготовке, чтобы знать, что делать. Например, мы не допустим, чтобы вы получили «ложный» курсив (когда буквы преобразуются в курсив на экране, вместо того чтобы использовать специально разработанную версию шрифта) – мы не допустим этого, потому что это уродство. Этого не будет. Scribus очень прилежно разработана версия шрифта – мы не допустим этого, они действительно присутствуют и что глифы работают.

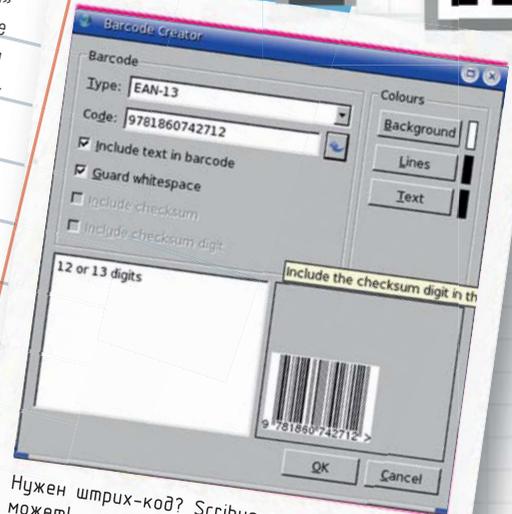
**LXF: То есть вы не терпите ошибок, по крайней мере, серьёзных?**  
КР: Вывод PDF у нас очень надёжный. Мы исключили проблемы, приводящие к созданию плохого PDF. Если такое случится, мы сосредоточим на этом всё внимание, всё остановим и исправим.

ПЛ: Была у нас подобная проблема в 1.2, очень странная ошибка: когда вы устанавливали градиент определённым способом, все просто чернело. Обнаружив ошибку, мы исправили её за шесть часов. В 1.4 нам «посчастливилось» получить некоторую разновидность автоматизированной регрессии. По мере усложнения кода становится всё труднее по-настоящему тестировать все функции, потому что их слишком много. В прежние времена я мог пробежаться по ним за выходные. Теперь этого уже не сделать. Теперь над ошибками работает много людей – ищут их, проверяют исправления, чтобы убедиться, что всё откорректировано правильно.

КР: В процессе очистки всплывают проблемы, скрытые за двойным кодом, или код, который выполняется дважды. Я переместил буквально тысячи строк кода. При переходе на другую модель – будь то возможность конвертировать в PDF из командной строки, или запускать как сервер, или использовать меньше памяти – обнаружилось, что уйма кода располагалась там, где его было удобно прописать когда-то. Почистив код, мы можем сделать его быстрее и эффективнее.

**LXF: Над чем вы сейчас работаете?**  
КР: Над очисткой и перемещением кода... будет значительное повышение скорости. Пока что мы сделали новую тему отмены последнего действия [undo], предстартовую проверку, цветовое колесо... Кое-что пытаемся сделать более дружественным к пользователю. Сейчас мы используем отдельные стили параграфов и стили строк. Но скоро появится новый менеджер стилей, который их объединит, плюс у нас будут стили символов.

ПЛ: Над крупным улучшением поддержки EPS [Encapsulated PostScript; формат для печати на PostScript-устройствах, прим. пер.]. Версия 1.3 – это путь к 1.4, и Scribus 1.4 будет инструментом профессионального класса; в нём будет всё необходимое для выполнения верстки практически любого вида. Я думаю, Scribus некоторые планки взял. Считалось, что открытое приложение никогда не сможет работать с заливкой цветом, но мы этого добились. На подходе ещё одна вещь – улучшенная поддержка сносок в больших документах, и т.п. Верите или нет, но сейчас LaTeX или OpenOffice.org, похоже, лучше всех поддерживают большие документы. Также предусмотрены спуски [imposition; расстановка полос на печатной форме в нужной для брошюровки последовательности – прим. пер.] Спуски делать будет не сложнее, чем распечатать А3: согнуть, скрепить, и готово. Следующий шаг – добраться до книг, тонкой переплётной работы и прочего: это совершенно другой мир, с весьма специфическими инструментами, стоящими от 5000 долларов. Его мы трогать не собираемся. Пока не собираемся.



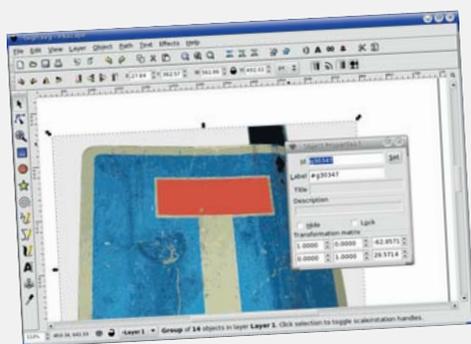
Нужен штрих-код? Scribus теперь и это может!

# Inkscape

Задаёт стандарты будущего графического дизайна.

**И**ногда жесткие решения себя оправдывают. Стандарт SVG, предназначение которого, будем надеяться, — стать повсеместным для векторной графики, как JPEG для фотографий, никогда не разрабатывался как внутренний формат. Базирование структуры на XML делает его очень гибким и пригодным для самых разнообразных целей. Но на внутреннем уровне работа с XML — мучение, заметно снижающее эффективность. Поэтому вполне естественно, что многие разработчики графических пакетов реализуют поддержку SVG только как функцию окончательной обработки. Некоторые, со скрипом, могут и импортировать SVG, но это, как правило, немногим больше, чем поверхностная трансляция — попробуйте сохранить тот же файл, и, вполне возможно, заметите искажение.

Группа разработчиков, однако, решила, что миру нужно слияние традиционных SVG-редакторов в «программистском» стиле (которые непосредственно воздействуют на XML и позволяют программистам изменять свойства узлов и т.д.) и мира инструментов рисования. Результат — *Inkscape* — добавил еще одну яркую историю успеха открытых графических программ и подтвердил идею, что открытые стандарты и свободная мысль могут изменить мир. По крайней мере, мир графики.



Трассировка раstra в Inkscape в настоящее время — лучшее из имеющегося в Linux, благодаря использованию другого открытого кода.

Проект стартовал под именем *Sodipodi*. Критической точкой явилось несоответствие мнений о том, в каком направлении его продолжать. Это вполне обычное явление: интернет-феномен под названием *Firefox* возник, когда двое разработчиков покинули проект Mozilla. А в данном случае четверо из основных участников *Sodipodi* решили отделить код и создать проект, ныне известный как *Inkscape*.

## Чистовой чертёж

Проект придерживался этого направления, и в вопросах редактирования SVG он настолько вырвался вперёд на любой платформе, что вряд ли не сможет стать основным инструментом создания векторной графики в этом стандарте.

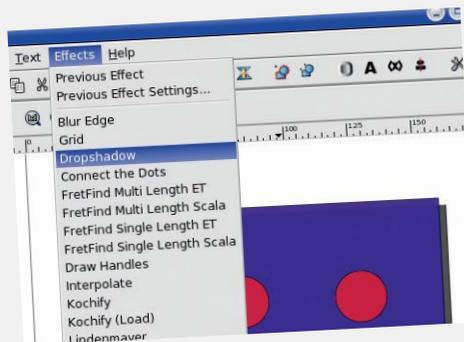
Как и большинство других рассмотренных здесь инструментов, *Inkscape* является кросс-платформенным. Одна и та же версия доступна под Linux, Windows и Mac, а в мире графики это даёт огромные преимущества. Действующее сообщество художников, несомненно, для профессионального дизайнера использует Mac. Доступность *Inkscape* на этой платформе не только открывает

программе рынок, но также обеспечивает полезнейшую обратную связь от пользователей-профессионалов. Кое-кто еще цепляется за идею, что *Inkscape* используют только потому, что он бесплатный, но это становится всё меньше похоже на правду.

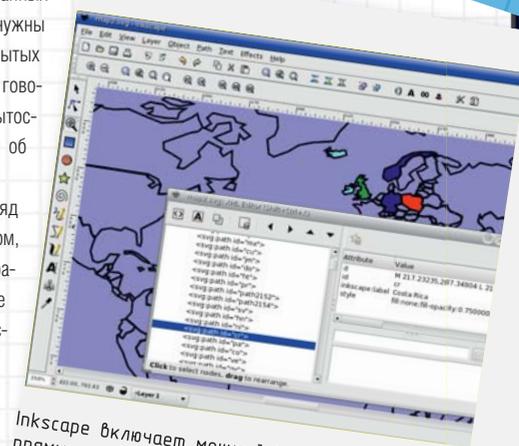
Профессиональные

дизайнеры не отказывают себе в самонаивнейших специализированных инструментах, если они им нужны для работы — и выбор ими открытых инструментов вроде *Inkscape* говорит скорее о надёжности, открытости и высоких стандартах, чем об отсутствии ценника.

Отсюда также следует, что ряд людей, интересующихся проектом, включая и часть основных разработчиков, считают себя скорее художниками, чем программистами. И хотя можно подумать, что они просто пишут то, что им интересно, команда *Inkscape* отлично понимает различных пользователей, которых хочет привлечь, и различные их нужды.



Экспериментальные эффекты, управляемые скриптами, можно получить, разрешив их использование в настройках приложения.



*Inkscape* включает мощный XML-редактор для прямых манипуляций с элементами.

## ЗАМЕТКА НА ПОЛЯХ:

Команда *Inkscape* (и многие другие разработчики) подчёркивали, сколько усилий тратится, чтобы обеспечить сборку их кода. «Никогда не фиксируйте [в CVS] ничего, способного сломать сборку», говорит команда.

Интервью LXF  
Тема: Inkscape  
Дата: 17 марта 2006 г., 14:43  
Место: аудитория в лионском СРЕ  
Лицо: Йон Круз - разработчик Inkscape



**LXF: Что привело вас в Inkscape?**

**Джон Круз (ДК):** Просто раскол *Sodipodi* произошёл, как раз когда я хотел активнее включиться в движение *open source*; я уже давал им советы насчет всяких трюков в Win32. Они сказали, что будут создавать самый лучший SVG-редактор – и в техническом, и в художественном смысле. Хотелось привлечь сообщество и разработчиков открытого кода, и хотелось так все организовать, чтобы не было единственной точки управления. На мой взгляд, разработчики ядра *Inkscape* выбрали правильные методы. Если у вас есть хорошая идея, то реализуйте её и добавьте в проект. Если есть проблемы, люди за ними присмотрят. То есть получается очень простая организация. Это проект, и каждый работает на проект – это не код одного человека.

**LXF: А может, люди просто хотят заниматься тем, что им интересно?**

**ДК:** Основная масса участников проекта старается сделать довольно серьёзные части программы. Мы пишем для пользователей. Вначале пользователями были одни разработчики, но со временем пользовательская база выросла. Вот в чём у меня есть некоторый опыт – так это в изучении требований. Люди приносят идеи о том, как программу можно использовать. Порой они просят о нужных с их точки зрения вещах, но не могут объяснить, что в конечном итоге достигается.

**LXF: То есть вы руководствуетесь тем, как пользователи в действительности используют программу, а не просто вставляете функции? Нетипично!**

**ДК:** Да, и я думаю, это одна из вещей, за которые команду критикуют. Я думаю, это критика из разряда «Они – простые инженеры, что они понимают?». Но некоторые из наших разработчиков – художники, и они очень хорошо ладят с интерфейсом.

**LXF: Какие функции появятся в следующем году?**

**ДК:** Мы думаем о просмотре свойств – преобразовании его в нечто большее, чем просмотрщик объектов, а также об улучшении диалога слоёв. Это очень нужно. Люди продолжают работать над этим и тратят много времени, чтобы сделать всё как следует. Будет здорово, но мы не хотим выпускать это в недоделанном виде.

**LXF: Займетесь ли вы слоями? Сейчас Inkscape, похоже, использует тот же подход, что и OpenOffice.org. Слои – это такая вещь, важность которых понимаешь не раньше, чем начнешь их использовать.**

**ДК:** *OpenOffice.org* и подобные приложения должны делать хорошо каждый фрагмент. А мы фокусируемся на SVG, и мы должны уметь «вылизать» эту единственную вещь, причем превосходно. Мы также занимаемся переделкой пользовательского интерфейса и целевыми профилями. Например, если вы хотите создать SVG Tiny, для телефонов и мобильных устройств, мы можем упростить вашу задачу: вы выполните экспорт и получите совместимый SVG. Firefox выпустил новую версию с поддержкой SVG. И стал качественно иным.

**LXF: Здесь явно выплывает много сопутствующих идей, например, Cairo.**

**ДК:** Ну, их может быть намного больше. Например, вы могли бы нацелить *Inkscape* на следующую версию рабочего стола Gnome и всюду использовать векторы. Чем легче создавать [векторные изображения], тем шире использование. Уже сейчас, создав XHTML-страницу со встроенным SVG, вы сможете добиться, чтобы это работало в Firefox со скриптами и DOM, и можно делать довольно сложные вещи. Сейчас делается много интересного.

**LXF: Всё же не так много инструментов, ориентированных на SVG. Программные средства есть, но не такие артистичные, как Inkscape.**

**ДК:** Да, и у нас есть цель и опыт. У нас хороший круг людей... SVG – очень понятный стандарт, и в его правильности очень легко убедиться. Я знаю людей, постоянно работающих с *Inkscape* – если что-то не так, мы тут же услышим об этом.



# Gimp

Номер один в своей области – серьёзная графическая мощь.

**П**опросите сотню средне-статистических пользователей Linux назвать графическое приложение – и смело можно держать пари, что в ответ они выпалют одно слово: *Gimp*.

Прошло уже более десяти лет с тех пор, когда Спенсер Кимболл (Spencer Kimball) объявил о проекте создания открытого приложения, способного соперничать с профессиональными (и дорогими) альтернативами для Mac и Windows. За это время появились бесчисленные дополнительные модули, обогатившие функции, и пользовательский интерфейс был полностью перестроен.

*Gimp* прошёл длинный путь, но его всё ещё критикуют за недостаток ряда ключевых функций и медленный темп разработки. Заявляют даже, что *Gimp* фактически тормозит развитие: его разительное первенство в редактировании растровой графики под Linux означает, что соперников ему бояться нечего – а следовательно, нет выбора и конкуренции. Хотя на самом деле причина в чрезвычайной сложности *Gimp*: достижение заметных улучшений требует уймы времени и усилий.

## День GEGL

Хотя это и не часть собственно *Gimp*, *GEGL* (Generic Graphics Library – общая графическая библиотека) оказывает значительное влияние на направление будущего развития *Gimp*. *GEGL* пока что находится на этапе

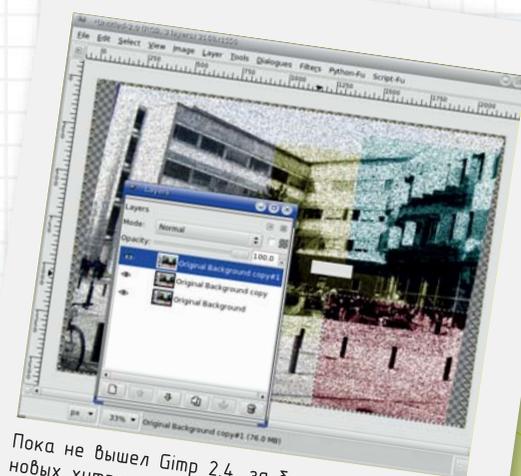
разработки релиз-кандидата, и проектируется как автономная библиотека для работы с графикой «по запросу».

Управляемость по запросу – наиболее заметная особенность системы. Она означает, что окончательная обработка выполняется именно в нужный момент (т.е. непосредственно перед выводом на экран), а не последовательно через ряд шагов. Преимущество здесь в том, что на этом этапе можно достичь очень хорошей оптимизации – вместо того, чтобы обрабатывать каждый пиксел и затем отбрасывать ненужные, трансформация коснется только конкретного участка.

*GEGL* также умеет управляться с различными глубинами цвета, так что она пригодна для широкого круга задач по обработке изображений. Конечно, *GEGL* будет очень полезна и для других приложений. Команда Scribus, например, очень заинтересовалась потенциалом *GEGL* и тем, что она может сделать для обработки изображений в издательском приложении.



SIOX позволит вырезать контуры намного проще – только используйте бета-версию *Gimp* 2.3, а не расширения версии 2.2.



Пока не вышел *Gimp* 2.4, за большинство новых хитростей отвечают фильтры.

## Всё включено

Основная работа, идущая в выпускаемых версиях *Gimp*, заключается в чистке и подборке перед долгожданным релизом 2.4 – так что мы займемся дополнениями, обеспечивающими внедрение интересных новых разработок.

Мы рассматривали SIOX (Simple Interactive Object Extraction – простое интерактивное извлечение объектов) в *LXF75*, но стоит остановиться на нем ещё раз. С этим замечательным инструментом можно отделить изображение переднего плана от фона, всего пару раз шаркнув мышкой. Каждый бит сгенерированного контура столь хорош, как будто вы кропотливо отслеживали каждую точку очертания самостоятельно.

SIOX будет интегрирован в финальный релиз *Gimp* 2.4, но уже сейчас доступен как дополнительный модуль. Из-за ограничений *Gimp* 2.2 он не столь эффективен,

как будет в финальном релизе – если вы серьезно намерены его попробовать, лучше использовать разрабатываемую версию *Gimp* (имеется на нашем диске), пока не выйдет версия 2.4. Дополнительная информация по SIOX – на сайте [www.siox.org](http://www.siox.org).



Митч Хаммерер (Mitch Natterer) объясняет, чем так хороша *GEGL*! (фото Романа Йооста (Roman Joost))

## ЗАМЕТКА НА ПОЛЯХ:

*Gimp*, подобно большинству рассматриваемых здесь инструментов, является кросс-платформенным – есть версии, доступные также для Mac OS X и Windows!

Интервью LXF  
Тема: Gimp  
Дата: 17 марта 2006 г., 15:17  
Место: Лион, на улице  
Лица: Митч Намперер и Свен Нойманн, «хранители» Gimp

**LinuxFormat: Вы довольны нынешним положением дел в Gimp?**  
**Свен Нойманн (СН):** Вообще-то я рад был бы видеть его прошедшим

большой путь. В прошлом у нас были грандиозные планы, но пока завершить их не удалось. Есть много вещей, о которых просят люди – скажем, поддержка большей глубины цвета или других цветовых схем, например, СМУК – мы их уже планировали. Но у нас недостаточно ресурсов, чтобы всё это реализовать.

Проделана огромная работа по исправлению ошибок и реализации мелких улучшений. Мы хотим добиться, чтобы функции, уже доступные пользователям Gimp, можно было найти и использовать, не читая руководства – его и так никто не читает.

**LXF: Многих удивит, что такой важный проект имеет всего 30–40 участников. Это всегда так было, или раньше у вас было больше людей?**  
**СН:** Gimp всегда был проектом, на который тратят свои силы всего несколько человек. Хотя эти люди всё время меняются.

Одна из проблем – то, что многие из участников-долгожителей, отлично знающих код, повзрослели: они уже не студенты, и у них есть работа и личная жизнь. Они просто не могут посвящать проекту так много времени, как пару лет назад. Возможно, необходимо вливание новой крови.

**LXF: Новым участникам потребуется немало времени, чтобы освоиться со столь сложным кодом.**  
**СН:** Ну, если людям это интересно, они могут поработать с дополнительными модулями. В большинстве из них легко разобраться – они в основном оформлены в одном С-файле. Трудности особой нет – и есть похожие модули и примеры для иллюстрации.

Конечно, работа с ядром и внесение фундаментальных изменений в пользовательский интерфейс или ввод новой концепции – довольно сложная задача, если имеешь дело с большим объемом кода. С другой стороны, за последние несколько лет мы значительно улучшили кодовую базу. Выполнили большую работу по очистке и, так сказать, вынесению за скобки. То, что было несколькими сотнями файлов в одном каталоге, теперь разделено и лучше организовано.

**LXF: Кое-кто мог бы предположить, что вы всей командой чуть ли не год отдыхали, потому что набор функций за этот год почти не изменился.**  
**СН:** Мы хотели [давным-давно] выпустить релиз 2.4. Мы пообещали добавить в него некоторые функции и приступили к работе над ними. А теперь мы в той ситуации, что есть пара инструментов, просто не готовых к выпуску.

Мы должны или выбросить их, или доделать. Одна из таких вещей – управление цветом. Мы решили, что действительно хотим, чтобы эти функции вошли в следующий релиз, так что будем работать до победного конца.

**LXF: Вы упомянули управление цветом и СМУК – какие ещё функции вы могли бы сделать доступными в ближайшее время?**  
**СН:** СМУК не будет использоваться для хранения данных о пикселях. Но как только мы как следует реализуем управление цветом, люди получат многое из того, что они просили. Я также думаю, что наиболее важная из отсутствующих в Gimp функция в данный момент – поддержка высокой глубины цвета. Всё больше и больше камер предоставляют такие данные, и мы не должны просто отбрасывать их при импорте.

**LXF: Думаете ли вы, что Gimp помогла бы спонсорская помощь или включение в в какую-нибудь из именуемых премиальных схем?**  
**СН:** Вот этим, в частности, я хотел бы заняться в следующих месяцах. Составить список того, что действительно требуется сделать в Gimp, и прикинуть, куда обращаться и с кем говорить. Мы хотели бы его опубликовать, и не только в Bugzilla. Я хотел бы, чтобы все знали: мы ищем разработчиков, и видели, что нам нужно доделать. Это могло бы привлечь больше людей.

С другой стороны, премии порой создают трудности, потому что для разработчиков, которые с нами уже давно, сотрудничество с людьми, которые работают за плату, может стать демотивирующим фактором. Мы хотели бы, чтобы довольными были все. Я уверен, что это возможно.



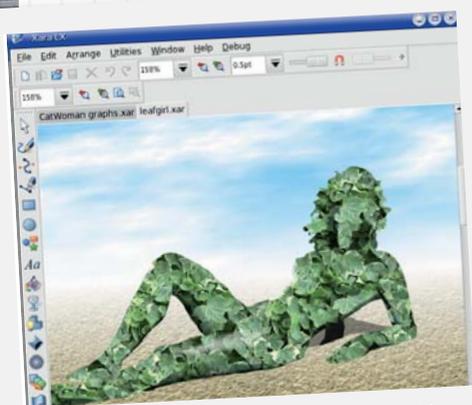
# Xara

Профессиональный инструмент дизайна становится свободным.

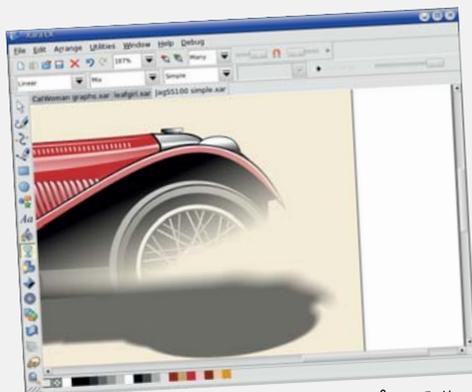
**Н**е больше ли радуется движение Open Source одно проприетарное приложение, раскаявшееся и оставившее путь закрытому кода, чем 99 приложений-праведников? Если да, то на Xara должны изливаться потоки любви. Хотя эта программа всю жизнь разрабатываемая частной британской фирмой, она много лет лицензировалась компанией Corel, распространявшей её под названием CorelXara.

Вы можете подумать, что для «освобождения» приложения достаточно опубликовать исходный код и позволить каждому им пользоваться. Для некоторых приложений так оно и есть, но проприетарное ПО – в частности, настольные приложения – очень часто используют сторонний инструментариум и библиотеки, и связанный с ними код открыть нельзя. Исключение этого кода может быть весьма кропотливой задачей, и, очевидно, оставит приложение неполным: тут какие-то подпрограммы загрузки и сохранения, там куски алгоритмов – не говоря уж о целых инструментариумах GUI.

Одна из самых знаменитых функций Xara – движок предпросмотра в режиме реального времени. Протащите прозрачный градиент по вашему рисунку, и экран обновится в реальном времени по мере протаскивания – никаких проб и ошибок при получении желаемого эффекта!



Открытие кода Xara может привести в Linux множество новых художников.



«Живые» эффекты в Xara экономят время и очень популярны среди дизайнеров.

Обратная связь в режиме реального времени оберегает от массы трудностей и делает приложение более отзывчивым на требования пользователей. Это одна из областей программы, которая пока ещё не открыта. Понятно, что разработчики Xara делают всё неспешно и аккуратно.

## Стык культур?

Когда Xara решила на переход, некоторые роптали, что это может закончиться конкуренцией с *Inkscape* за лидерство среди дизайнерских приложений для Linux, но они упустили главное: хотя общего у проектов много, они, по большому счёту, ориентированы на разные вещи. У Xara длинная история инноваторского и разностороннего инструмента для иллюстраций и работ для Интернета, но она не нацелена исключительно на работу с SVG, как *Inkscape*. Не похоже, что в Xara когда-нибудь будет встроен XML-редактор, как это сделано в *Inkscape* – хотя сейчас это открытая программа, и как знать? Очевидно одно: разработчики *Inkscape*

и Xara стремятся к сотрудничеству. На конференции LGM они охотно обменивались идеями и работали над способом облегчить передачу данных между своими приложениями.

Чарльз Мойр (Charles Moir), владелец Xara, также спонсирует разработку «суперконвертера». Идея знакома сторонникам открытого кода – создать промежуточный формат для графики и набор инструментов для преобразования всевозможных форматов в промежуточный и промежуточного – в другие форматы. Сказать это проще, чем сделать, потому что для достижения настоящей эффективности требуются способы сохранения или, по крайней мере, корректной интерпретации характеристик каждого формата, зачастую уникальных. Проект получил название Chromista, и первый его разработчик – Эрик Вильгельм (Eric Wilhelm). Вы можете узнать больше о

целях проекта на его сайте: <http://scratchcomputing.com/projects/uber-converter>.

Что же касается самой Xara, продолжается работа по реализации всех функций проприетарной версии в открытом коде. Обещаны регулярные выпуски кода и двоичных пакетов, и команда тратит уйму усилий, чтобы включить функцию «Save» (Сохранить). Дело хорошее! **LXF**

**ЗАМЕТКА НА ПОЛЯХ:**  
Наконец свободна! Код Xara был открыт 17 марта. Свежая версия программы – на нашем диске!

Интервью LXF  
Тема: Хага  
Дата: 17 марта 2006 г., 12:14  
Место: аудитория в CPE  
Субъект: Нейл Хоу, технический директор Хага

**LinuxFormat:** Почему, спустя более 10 лет, вы открыли код Хага?

**Нейл Хоу (НХ):** Основная причина – хочется видеть, что проект способен достичь намного большего, чем есть сейчас. Проблема у нас в том, что мы – очень маленькая фирма с ограниченными ресурсами. Нам все говорят: какой *Xara Extreme* хороший инструмент для рисования! – и мы просто хотим выпустить его побольше. Я думаю, что придав ему кросс-платформность, обеспечив доступ и пользователям Linux, и пользователям Mac, а также сделав код открытым, мы сможем этого добиться. Еще один фактор – мы недостаточно быстро развивали продукт. Я уверен, что каждая фирма-разработчик ПО планирует выпускать что-то побольше, чем в состоянии когда-либо осуществить. Мы не исключение – разве что, как я подозреваю, наш список ещё длиннее – и с имеющимися ресурсами мы не чувствовали сил для достаточно быстрого движения.



**LXF:** То есть выходит, что даже будь у вас превосходные идеи, вы не смогли бы их реализовать?  
**НХ:** Да, мы можем сделать не так уж много. Мы хотим видеть инструмент, добившийся определённого успеха по сравнению с подобными от Adobe и Microsoft. Если мы получим поддержку сообщества Open Source, думается, мы сможем достичь более быстрого прогресса.

**LXF:** Вы говорите об открытии кода Хага?  
**НХ:** Да! Это было трудным решением для такой фирмы, как наша, – взять инструмент, с нашими технологиями, который мы собирали так много лет – годы усилий были потрачены на него; это значительный шаг. Тут и причина, почему мы не приняли подход «большого взрыва» – мы не просто говорим «Всё это теперь открыто». Мы осуществляем открытие в два этапа, чтобы посмотреть, как проходит первый этап, сформировать сообщество, продвинуть его вперёд, а затем полностью выпустить код.

**LXF:** Технология, оставшаяся закрытой – движок рендеринга. Есть ли ещё компоненты, которые вы пока придержите?  
**НХ:** Среди них, код комбинирования фигур – код для пересечения контуров – это всё в одной и той же библиотеке. Так что закрыты пока движок и эта часть кода, они находятся в одной двоичной библиотеке, и мы поставляем ее вместе с исходным кодом, который должен с ней собираться. Точно сказать не могу, но в терминах объёма исходного кода, сейчас открыто примерно 90%.

**LXF:** Что, как вы ожидаете, будет первым эффектом от открытия кода, и чего бы вам хотелось?  
**НХ:** Мы хотели бы получить сообщество, и оно у нас уже есть. В последние месяцы мы даже частным порядком приглашали разработчиков «со стороны», чтобы сделать порт [для Linux] – мы просто открыли им доступ к коду еще до публичного выпуска. И это дало старт сообществу, если хотите, и мы рады наблюдать, как сообщество растёт, и видим, что разработчики открытого кода помогают нам завершить порт как можно быстрее. Это первая цель – сделать его столь же функциональным и надёжным, как версия для Windows, с которой мы начали.

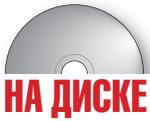
**LXF:** Сколько времени, по вашей оценке, на это потребуется?  
**НХ:** Сложно сказать. Первоначально прогресс был медленнее, чем я ожидал. Но это в значительной степени потому, что портирование – основа всего, само оно ничего не выполняет, а просто предоставляет доступ к другим функциям. Примерно с середины января начался действительный взлёт. Например, на этой неделе Алекс (один из разработчиков) включил два инструмента за два дня – инструмент «свободное выделение» и инструмент смешивания. Так что мы надеемся, что темпы развития сохранятся или даже возрастут.

**LXF:** Вы сказали, что у вас огромные планы. Как вы думаете, на этом этапе вы в состоянии реализовать новые функции помимо собственно портирования?  
**НХ:** Мы установили первую веху в функциональности *Xifra*, пока без того, что мы не можем открыть, как я уже упоминал. Но открытый код есть открытый код, и мы не собираемся указывать людям, что можно делать, а что нельзя. Если кто-то приходит с желанием работать над продуктом, пишет, например, новый инструмент и хочет внести его в код, мы не скажем ему «Извините, мы принимаем только заплатки».



Коллеги из других проектов поражены возможностями Хага.

# iFolder за десять минут



Забудьте о резервном копировании: Пол Хадсон расскажет о том, как синхронизировать файлы по сети под Linux.

## ЧТО НУЖНО:

- SUSE Linux 10.0
- Gnome
- Mono

## РЕЗУЛЬТАТ:

- Синхронизация файлов по сети в реальном времени
- Возможность совместной работы нескольких пользователей
- Надежный и безопасный способ восстановить важные файлы в случае аварии



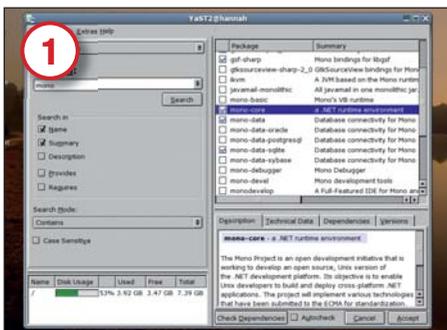
Поднимите руки, кто делает резервные копии всего содержимого своего компьютера каждую ночь. Хорошо, вы, на заднем ряду, опустите руку – мы все знаем, что вы шутите. Нет ничего хуже, чем поломка винчестера и потеря данных, однако даже самые собранные люди редко создают резервные копии чаще, чем раз в неделю. Благодаря легендарной надежности Linux многие пользователи вообще не делают этого, и мы у себя в редакции не являемся исключением.

С тех пор, как у многих людей в домах появилось более одного компьютера, стало доступным новое средство для резервного копирования: iFolder. С его помощью можно синхронизировать файлы между компьютерами, объединенными сетью, и неважно, находятся ли они в разных частях вашего дома, офиса, или всей планеты. Более того, вы можете разрешить всем пользователям, имеющим доступ к вашей директории, присылать вам сделанные изменения и синхронизировать их друг с другом.

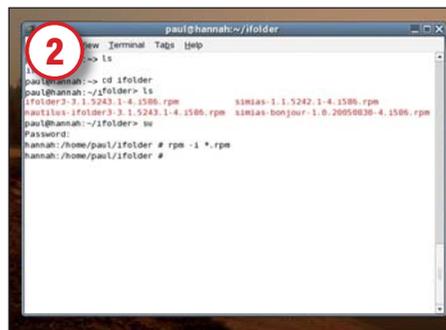
Мы выложили на диск необходимые приложения в виде RPM-пакетов для SUSE 10.0, которые вы найдете в каталоге **Magazine/iFolder**. Чтобы воспользоваться этим руководством, вам, кроме диска, понадобятся два компьютера, работающих под управлением SUSE 10.0. Итак, сегодня в роли двух пользователей выступают: Пол с компьютером «hannah» (коричневый фон) и Грэм с «susannah» (синий фон).

Имейте в виду, что *iFolder* может некорректно работать при включенном брандмауэре из стандартной поставки SUSE. В данном случае проще всего было бы отключить брандмауэр, но лучше перенастроить его правила так, чтобы *iFolder* имел доступ к сети, а остальные порты оставались бы под защитой.

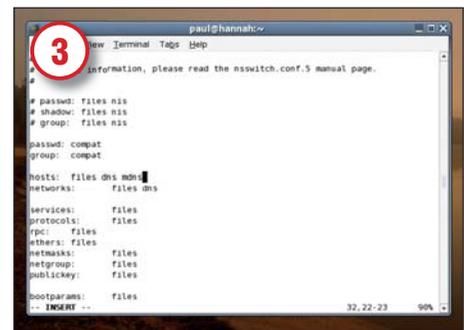
## НА СТАРТ, ВНИМАНИЕ... КАК ОБМЕНИВАТЬСЯ ФАЙЛАМИ ПО СЕТИ?



Сперва решим проблему зависимостей: это Gnome и Mono. В управлении пакетами *Yast* выберите **gtk-sharp**, **libgdiplus**, **mono-core**, **mono-data**, **mono-web** и **nautilus**. Программа предложит установить также ряд других зависимостей – просто согласитесь.



Теперь нужно установить *iFolder*. Создайте отдельный каталог в своей домашней директории и скопируйте туда четыре RPM-пакета с нашего диска (**Magazine/iFolder**). Перейдите в этот каталог (**cd <имя\_каталога>**), перейдите в режим администратора (**su**) и дайте команду **rpm -i \*.rpm**, которая установит в систему все четыре пакета.



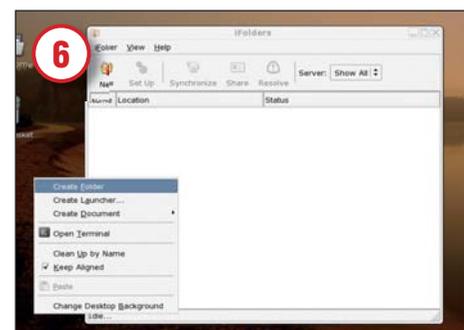
Вам понадобится включить протокол Apple Bonjour, с помощью которого компьютеры находят друг друга по сети. Отредактируйте файл **/etc/nsswitch.conf** и измените строку «hosts:files dns» на «hosts:files dns mdns». Сохраните файл и дайте команду **rcnscd restart**.



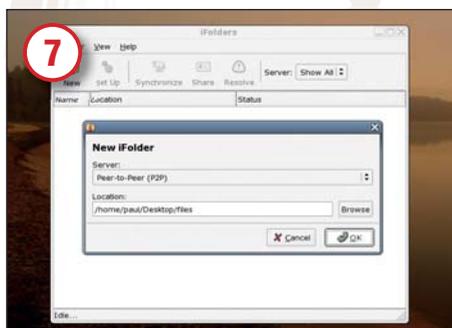
Теперь можно запустить *iFolder*. Закройте терминал и перейдите в меню Приложения > Интернет > Обмен данными, где вы найдете иконку iFolder 3. Иконка появится в системной лотке (см. окно с иконками *iFolder* сверху справа для дополнительной информации о значении иконок). Дважды щелкните по ней для открытия окна настроек.



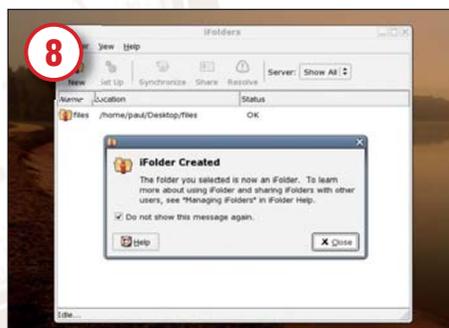
Так как вы запускаете *iFolder* впервые, перед вами представит пустое окно – вы не даете никому доступа к вашим папкам, так же как и никто не дает вам доступа к своим.



Создайте новую папку на рабочем столе, щелкнув по нему правой кнопкой мыши и выбрав соответствующий пункт меню. Назовите папку **files**. Сюда мы будем помещать файлы, для которых требуется общий доступ.



Чтобы создать новую разделяемую папку *iFolder*, щелкните на **New** (Создать), после чего выберите пункт **Найти** (Browse). Появится диалог выбора файлов и папок. Найдите и выделите созданную ранее папку **files** и нажмите **OK**, а затем еще раз **OK**, чтобы закрыть первое диалоговое окно.



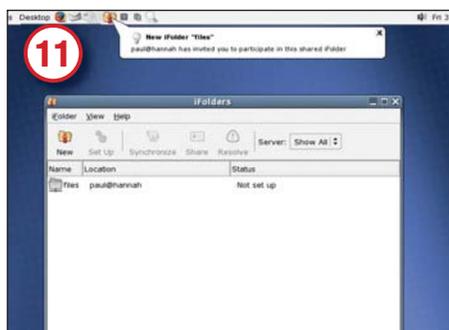
Проверяем. После создания папки *iFolder* появится экран с предупреждением. Закройте его, не забыв указать «Больше не показывать это предупреждение». В главном окне теперь появится папка **files**, статус которой будет достаточно часто обновляться (сообщения «**OK**» и «**Проверка изменений**»).



Запустите *iFolder* на втором компьютере и сразу вернитесь к первому. Щелкните правой кнопкой мыши по папке **files** и выберите **Share with...** (Сделать общим...). В появившемся окне щелкните по кнопке **Add** (Добавить).



Теперь мы добавляем второй компьютер. Появится окно **Select Users** (Выбрать пользователей), и вы заметите, что программа обнаружила такого на удаленном компьютере – волшебство Bonjour! Выберите пользователя и добавьте его в список. Если у вас есть другие компьютеры, здесь можно добавить всех пользователей.



В это время на втором компьютере появится всплывающее сообщение, приглашающее присоединиться к разделяемой папке *iFolder*. Вы также увидите, что в вашем окне настроек появился пункт *iFolder* с комментарием «**Not set up**» («**Не настроено**»).



Настроим общий доступ. Щелкните правой кнопкой мыши по папке **files** и выберите **Set up** (Настроить). Вам нужно будет выбрать адрес для локальной папки **files**. По умолчанию предлагается ваш рабочий стол. Нас это устраивает, поэтому просто нажмем **OK**.



Комментарий к разделяемой папке **files** в окне настроек изменится на «**Waiting for connect**» («**Ожидание соединения**»), затем на «**Waiting for synchronise**» («**Ожидание синхронизации**»), после чего в итоге вы увидите сообщение «**paul@hannah has joined the iFolder 'files'**» («**paul@hannah присоединился к разделяемой папке 'files'**»).



На рабочем столе второго компьютера теперь имеется папка **files**. Создайте простой текстовый файл, написав в него «**Hello, world!**» и сохраните его под именем **test.txt** в эту папку. Теперь вызовите окно *iFolder* и щелкните кнопку **Synchronise**.



Когда *iFolder* заметит новый файл **test.txt**, он разошлет его всем компьютерам, которые имеют разделенный доступ к папке **files**. Когда вы вернетесь к первому компьютеру и откроете эту папку локально, то увидите что там появился наш тестовый файл. Готово! **LXF**

# Что на уме у Novell?

LXF ИНТЕРВЬЮ



Novell открывает SUSE, потом увлекается Gnome, а сейчас возжелала уподобиться Швейцарии? Грег Манкузи-Унгаро комментирует происходящее.

**Г**регу Манкузи-Унгаро (директор Novell по маркетингу Linux и Open Source) неоднократно приходилось отвечать на вызов. Его задача – не только сманивать клиентов с Windows, он должен еще и всячески удерживать пользователей SUSE от перехода на другие дистрибутивы. А прежде всего – какие бы шаги ни предпринимала его компания, он должен привлекать на ее сторо-

ну глас сообщества Open Source.

**Майк Сондерс** встретился с Грегом, чтобы поговорить о стратегии Novell и обсудить перспективы ПО с открытым исходным кодом со специалистом по маркетингу. Это не хакер из сообщества OSS, это – бизнесмен, который работает в атмосфере конкуренции, а не вытягивает пустышки. Нужна ли Linux дискуссия по корпоративным вопросам? Решать вам.

**Linux Format:** В ноябре Novell сообщила о планах сократить около 600 рабочих мест. Как это повлияло на развитие Linux и ПО с открытым кодом?

**Грег Манкузи-Унгаро:** Ясно, что невозможно провести сокращение и при этом не затронуть морального аспекта. Я думаю, люди, работающие в такой фирме, как Novell, по-настоящему гордятся своей работой. Но я также думаю, что шок от происшедшего быстро уступит место ощущению нового направления. У нас произошли серьезные стратегические изменения, но команда Linux по большей части не поменяла направления деятельности и даже ускорила темп работы в результате нашей переориентации.

**LXF:** В каких областях?

**ГМУ:** Ну, если взглянуть на деятельность Novell и на то, как проводилось обновление [сокращение штатов], то заметно: стратегические направления деятельности, включая и Linux, в основном не были затронуты. Это не значит, что изменения

их не коснулись вообще, но командам было сказано: «То, что вы делаете – основа нашего успеха». И все, что происходит с проектом OpenSUSE, с нашей платформой для предприятий аналогично тому, что происходит в секторе рабочих столов с Hula и Mono... Я думаю, был дан очень хороший посыл [этим направлениям]. Мы тщательно продумываем все, что делаем, и мы очень, очень преданы Linux.

**LXF:** Явный конкурент OpenSUSE в сообществе – это Fedora Core...

**ГМУ:** OpenSUSE часто сравнивают с Fedora Core, но это совершенно разные проекты. Одно из отличий OpenSUSE от Fedora – то, что мы закончили работу над ним и выпустили дистрибутив.

По поводу Fedora Core шутят, что работа над ним никогда не закончится – он хорош уже такой, какой есть, и таким он и останется. А мы постарались пройти шаг дальше и сказать: «Мы завершили работу над дистрибутивом, отправили его пользователям Linux по всему миру, и эта версия достаточно законченная».

Это очень дисциплинирует: фактически, одним из добавлений, которые мы внесли в проект OpenSUSE, стал вклад ответственности в наши тесты эргономичности, через сайт [www.betterdesktop.org](http://www.betterdesktop.org). И когда вы соединяете все вместе, то получаете постоянно растущий дистрибутив, благодаря этому ускоряется процесс его принятия.

[В противоположность этому] Fedora действительно разрабатывается для инженерного сообщества. Иногда заявляют, что есть узкий круг людей, работающих с Fedora, и широкий круг людей, пользующихся Fedora. Мы знаем, что на самом деле он еще немного шире. Менталитет OpenSUSE в том, что имеется группа людей, создающих дистрибутив под названием SUSE Linux, и огромная группа людей пользуется SUSE Linux. Так что у нас имеется настоящая клиентская база, настоящая база пользователей, потому что мы делаем Linux, отличающийся по характеру от его собственных разработчиков.

**LXF:** Red Hat прекратила поставки коробочных версий Red Hat Linux после Red Hat 9, а Novell все еще продает ее.

**Повлияет ли бесплатный OpenSUSE на объем продаж SUSE Linux?**

**ГМУ:** Динамика продаж определенно изменилась, но это – часть плана. Приходится спросить себя, что лучше – продать 150,000, 200,000, да любое число – коробочных комплектов, или продать их поменьше, но зато в десятки раз увеличить пользовательскую базу? Мы в Novell приняли решение содействовать переходу на Linux и создавать базу пользователей Linux любой географии. Оказалось, что переходу на использование Linux мешает невозможность получить Linux. Не проблемы с использованием, не набор свойств, не установка – просто физическая невозможность его найти. Всегда есть те, кто говорит: «Мне нужны книги, мне нужна поддержка, и мне нужно, чтобы у меня под

Грег высказывается о Китае, AppArmor и открытых стандартах на [www.linuxformat.co.uk/mag/mancusi.html](http://www.linuxformat.co.uk/mag/mancusi.html).

рукой было то, что я могу установить». Это очень удобно для определенной части наших пользователей, и именно поэтому мы и продолжаем этим заниматься.

С начала 2006 года, мы приступаем к показу нашей инфраструктуры разработки, чтобы люди могли принести свой исходный код в проект OpenSUSE, и собрать его для получения совместимых с SUSE Linux пакетов. Так что сейчас, спонтанно, возникла ситуация, когда все разработчики приложений, которые хотят соединить свои продукты с линией SUSE (могут это сделать).

**LXF: У Ubuntu на подходе дистрибутив для предприятий, Darper Drake, с пятилетней поддержкой. А есть ли продукт SUSE, рассчитанный на столь же долгий период?**

**ГМУ:** У нас есть продукт с семилетней гарантией – SUSE Linux Enterprise Server. А что касается Ubuntu, так они копируют нашу модель, просто добавляя к ней продукт для предприятия. У нас продукт для предприятия существовал всегда, поэтому многие компании смотрят на Novell и признают нашу мудрость. Так что ничего менять в SUSE Linux нам не придется – продукт уровня предприятия у нас существует уже долгие годы.

**LXF: На сайтах фирмы Novell в последнее время идет дискуссия: может ли фирма, занимающаяся Linux, стать следующей Microsoft...**

**ГМУ:** Ну, ведь Red Hat в одном из своих первых проспектов заявила: «Наша цель – стать Microsoft от Linux». Многие при этом скривились! Я не думаю, что это наша цель. Мне представляется, что идеалом, к которому стоит стремиться, является Швейцария, то есть нация, чье единство ни у кого не вызывает вопросов. Надо стать народом, склонным к технической точности, народом нейтральным и в то же время лидером. Надо занимать должности и рассматривать проблемы, и при этом надо стараться – и вести себя, подобно послу в сообществе.

Вот так Novell представляет себе свою роль. Я не знаю, надо ли быть Microsoft от чего бы то ни было.

**LXF: Как Вы думаете, станет ли какая-либо фирма эквивалентом Microsoft для Linux, если это – открытая ОС, с которой можно делать что угодно?**

**ГМУ:** Ну, если в одно прекрасное утро мы проснемся и скажем: «Ух ты, теперь Novell – это Microsoft от Linux» или «Red Hat – это Microsoft от Linux», тут движе-

нию Linux и конец. Надо говорить, что Novell – фирма, способствующая успеху Linux. Но Linux и так пользуется доверием в сообществе, а Novell делает так, что Linux работает для крупных предприятий. Это совсем другое дело. Контроль Microsoft повсеместен; эта компания может изменять менталитет целых наций, и мне кажется, не в лучшую сторону.

Все, что Novell может сделать, и все, что Novell действительно хочет делать – это констатировать реальное положение дел и отстаивать наши интересы и точку зрения. И мы понимаем, что мы таким образом воздействуем на Linux, но мы не владеем Linux.

Вот Microsoft владеет всем: землей, дорогами, поездами. А мы – нет, и мы это знаем, и это значит, что нам надо больше полагаться на роль сотрудничества – и нам это нравится.

**LXF: SUSE традиционно был дистрибутивом, построенным на KDE. Как на это повлияет включение и поддержка Gnome в SUSE Linux?**

**ГМУ:** SUSE всегда включал и KDE и Gnome, и мы будем поддерживать обоих по мере развития нашего продукта для предприятия. Для нас работа с партнерами – рациональная работа – была весьма интересной, чем-то вроде вызова, но мы

**LXF: А были другие причины выбрать Gnome?**

**ГМУ:** С позиций коммерческой перспективы, когда вы смотрите на основные дистрибутивы, вы видите, что в качестве стандартного рабочего стола выбирают Gnome. Так что у Novell был шанс выбрать одно из двух – либо вовлечь всю промышленность в дебаты на тему «Gnome против KDE», либо взять Gnome и дебаты прекратить. Мы считаем, что наш выбор – лучший выбор, но мы не откажем в поддержке тем нашим клиентам, которые предпочитают KDE; мы приняли свое решение, чтобы и сообществу, и нашим предприятиям, и нашим поставщикам оборудования и программ стало ясно, [для какого рабочего стола] сертифицировать свою продукцию.

**LXF: Как идет внутренний переход Novell на Linux?**

**ГМУ:** Очень хорошо. Пару лет назад Novell приобрела несколько фирм Linux [Ximian и SUSE], и в то же время приняла решение – беспрецедентное для фирмы такой величины – перевести все на Linux. Проект начался с группы из двенадцати или 18 человек, которые согласились стать подо-

Визитка LXF

Грег Манкузи-Унгаро

Грег пришел в Lotus поработать над Jazz, оригинальным бизнес-пакетом Macintosh, и засиделся там на 14 лет. В 2001 он перешел в Ximian, а теперь возглавляет отдел маркетинга по Linux ПО с открытым кодом в компании Novell.



ВОЗРАСТ	48
НАЦИОНАЛЬНОСТЬ	Американец
СТАЖ РАБОТЫ С LINUX	5+
ЯЗЫКИ ПРОГРАММИРОВАНИЯ	0
КОЛИЧЕСТВО ПК	1
ДНЕВНАЯ НОРМА КОФЕ	0 чашек
ПАР САНДАЛИЙ	2
ПРЯМАЯ РЕЧЬ: «Сообщество Linux говорит нам, правы мы или нет!»	

TOP TRUMPS COURTESY: WINNING MOVES UK LTD

ПРО ДОМИНИРОВАНИЕ

«Если мы проснемся и скажем: «Ух ты, теперь Novell – это Microsoft от Linux», тут движению Linux и конец.»

приняли решение, что в следующем поколении продуктов для предприятия оболочкой рабочего стола по умолчанию будет Gnome. Но по ряду причин мы не прекратим включать в наши продукты KDE и поддерживать его.

Когда в Novell обращаются фирмы и говорят, что собираются создать классное приложение KDE для SUSE Linux, мы отвечаем: «Отлично!». Когда они говорят, что хотят создать приложение под Gnome, мы опять же говорим: «Отлично!». И когда фирмы спрашивают нас, каким рабочим столом им пользоваться, мы теперь можем дать им ответ. Раньше мы не могли им ответить – мы говорили: «Это вам решать», и это, конечно, не помогало. А вот теперь мы знаем ответ – рабочий стол по умолчанию – Gnome.

пытными кроликами. В пятницу они ушли домой, в понедельник вернулись на работу, а их машины заменили машинами, на которых был только Linux. Это было ужасно! Просто кошмар. Но нам надо было узнать, каково это – перевести всю крупную организацию на Linux. Если Linux добьется успеха, то не потому, что где-то поблизости есть эксперт Linux, а потому, что UI [пользовательский интерфейс, – прим. пер.], конструкция ПО и работа ПО оправдывают ожидания пользователей.

Сейчас почти половина фирмы работает исключительно на Linux, и мы довольны тем, что мы сделали. Мы многому научились – наша методика помощи фирмам переходить на Linux так богата благодаря нашему собственному опыту. Нельзя предлагать своему клиенту делать то, чего ты сам делать не хочешь!

**LXF: Предвидите ли Вы консолидацию дистрибутивов?**

**ГМУ:** Иногда мне кажется, что грядет консолидация, а иногда – что будет специализация. Возникает вопрос: Что такое дистрибутив? На сайте OpenSUSE есть несколько интересных проектов, например, поддержка SUPER и PowerPC. Либо эти проекты встанут на ноги и начнут расти, либо

пожмутся к основному дереву. Действительно обнадеживает то, что появляются люди, берут код и делают то, что им подходит.

На данном этапе, мне кажется, есть три основных поставщика, обеспечивающих нужды потребителя по Linux. Я думаю, что работа, которой занимаются Ubuntu и Linspire, весьма интересна, но им не хватает масштаба, поэтому они, возможно, найдут способ, как сумели Mandriva и Suse, увеличить размах, расширить долю на рынке. Так что консолидация будет, но и диверсификация – тоже. **LXF**

# Дегустация свободы

На суд общественности вынесена новая версия GNU General Public License **Ричарда Столмена** (Richard Stallman). **Грэм Моррисон** пробует документ на вкус и ожидает вспышки дебатов.

**Б**лагодаря Владетелю Столмену многие оценили разницу между правом на свободу выражения (free as in speech) и бесплатным пивом (free as in beer). Linux меняет отношение человека к работе на компьютере, предоставляя всю полноту власти пользователю. Но ещё важнее то, что Linux бросает вызов нашим представлениям о программном обеспечении и о том, как сбалансировать права его пользователей и нужды его создателей.

Просветительский процесс подкрепляется GPL, той самой лицензией, под которой выпущено большинство программ с открытым исходным кодом, и той самой лицензией, которую Линус Торвалдс избрал для ядра Linux. Это решение – союз ядра Linux и GPL, защищающей философию открытости – мы принимаем как должное. Но на этот раз Столмен, стоящий за GPL, намерен пойти дальше, а Линус рискует остаться в стороне.

## Новая GNU

*Свобода использования программы для любой цели.*

*Свобода изучения принципа действия программы и изменения её в соответствии со своими нуждами.*

*Свобода распространения копий программы для помощи своим ближним.*

*Свобода совершенствования програм-*

*мы и публикации своих усовершенствований на благо всего Сообщества.*

Вы, наверное, поняли из постоянного, подобно заклинанию, употребления слова «свобода», что эти слова взяты из определения свободного программного обеспечения по версии Фонда Свободного ПО (Free Software Foundation, FSF).

FSF является держателем GPL и гарантом её соблюдения. Фонд был основан Столменом в 1985 г. как очаг распространения философии свободного ПО, от найма программистов для разработки программ с открытым исходным кодом до написания свободной лицензии универсального применения. С 1985 г. FSF постоянно набирала силу, росла вместе с Linux, и, подобно Linux, теперь весьма влиятельна.

Но с момента принятия второй версии GPL в 1991 г. прошло целых 15 лет. В то время web-серверов просто не было, как не было и технологий DRM.

По мере развития технологий в GPL начали проявляться слабые места – или новые возможности, в зависимости от точки зрения: например, лазерка для ASP (application service providers, поставщиков цифровых услуг). ASP-приложения работают только в сети, а значит, пользователь не загружает их вовсе. Во время разработки GPL2 такой сценарий не предусматривался (идея Apache ещё только блеснула в мозгу Брайана Белендорфа), а ведь он означает, что поскольку нет бинарных дистрибутивов

(серверные приложения не компилируются, они только интерпретируются браузером), то нет и объектного кода, а есть только исходный. Любой создатель ASP, решившийся опубликовать своё приложение по GPL2, рискует тем, что конкуренты воспользуются его программой и, расширив функциональность, не вернут Сообществу ничего взамен. Это идёт вразрез с желанием FSF, которой очень хочется, чтобы Google и Amazon смогли опубликовать свои исходные коды, не опасаясь конкурентов.

В наши дни на повестке дня FSF стоят две стратегические задачи – патенты и DRM (Digital Rights Management), система защиты авторских прав (или, по расшифровке FSF, Digital Restrictions Management, система цифровых запретов). FSF убеждает своих членов объявить бойкот дискам HD-DVD и Blue-ray, утверждая, что эти диски являются «атакой на свободное ПО, поскольку свободное ПО может никогда не прочесть эти диски. Коммерческие интриги ограничивают развитие технологий и должны быть признаны незаконными». Времена меняются и, чтобы поспеть за ними, универсальная лицензия должна меняться тоже.

Вот в таком горниле идей мы нашли первую редакцию GPL-3. Третья Генеральная Общественная Лицензия GNU написана не только для устранения проблем, обнаруженных в GPL2, но и как попытка достойно встретить вызовы будущего. Первая редакция была вывешена на виртуальной доске

объявлений в январе, и с тех пор повсюду не утихают дебаты – от списков рассылки ядра Linux до коридоров власти Red Hat и IBM.

## Как это выглядит?

Эта лицензия затронет всех нас, даже при том, что разработчики ядра Linux не самые горячие её приверженцы, и именно поэтому так важно принять участие – время пришло!

Рассмотрим главные различия между второй версией и первым вариантом третьей. Документ состоит из 19 разделов плюс преамбула. Если вы рассчитывали, что новая версия будет написана простым и понятным языком, то вы будете разочарованы. Эбен Моуглен (Eben Moglen) – генеральный юристконсульт FSF и соавтор Столмена по первой редакции лицензии – утверждает, что эта сложность неизбежна при защите прав обладателя лицензии. Терминология по сравнению со второй версией лицензии определённо изменилась в сторону юридической специфики – но сущность и цели в основном сохранились.

## Избегая вулканов Преамбула

Обе редакции GPL начинаются со следующего утверждения: «Когда мы говорим о свободном ПО, мы имеем в виду свободу, а не цену». Оно расставляет акценты и проясняет намерения GPL, рисуя окруже-

## ЧТО ГОВОРЯТ В СООБЩЕСТВЕ

Разработчики свободного ПО могут остаться и с GPL 2, если захотят. Итак – кто за старую версию, а кому больше нравится GPL 3?



Решение команды ядра Linux пока не принято и, как мы обсудим на стр. 55, разработчики вполне могут остаться на 2-й версии. Ядро может вообще не перейти на GPL 3, если FSF не смягчит своё отношение к DRM. Однако другие проекты не столь категоричны.

Отзывы команды Debian – общепризнанного ценителя свободы ПО – чрезвычайно позитивны. Бранден Робинсон (Branden Robinson), лидер проекта Debian, сказал: «Изменения, большие и малые, впечатляют. Короче, мне это нравится». Слышны и другие одобрительные голоса, включая весьма неожиданный от Sun Microsystems, о возможности издания её почтенной ОС Solaris по GPL 3.

Джонатан Шварц (Jonathan Schwartz), президент Sun, записал в своём сетевом дневнике: «...нам нужно сделать всё возможное для эффективного взаимодействия между Linux и OpenSolaris. (Зачем изобретать велосипед с технологиями вроде DTrace и ZFS – или Grub и Xen?)». Возможность включения технологий Solaris в другие проекты GPL 3 – несомненно, весомое приобретение, это серьёзный повод для Линуса подумать о переходе на новую лицензию.

Труд разработчиков зависит от инструментов, которыми они пользуются, но пока рано говорить, перейдут ли проекты вроде Qt на GPL3. В данный момент трудно понять, как Qt, доступную на множестве частных и встроенных плат-

форм, можно совместить с жёсткой позицией, занятой в отношении DRM и патентов.

Диапазон программ, распространяемых по новой лицензии, не особо изменится. Пользователи будут надёжнее защищены от лицензионных судебных тяжб (вспомните случай с SCO), но могут столкнуться с трудностями в получении мультимедиа-проигрывателей, доступных в других ОС. Будем реалистами – пользователи не собираются отказываться от HD-DVD или Blue-ray только потому, что FSF считает их подрывом свободы. Это всё равно, что стоять зимой на улице, когда ваши друзья в домашнем тепле наслаждаются просмотром трилогии «Звёздные войны» в версии с высоким разрешением.



ILLUSTRATION BY DANIEL MACKIE

## РАЗВИТИЕ ПРОЦЕССА ПРИНЯТИЯ GPL3



ние сплошь из извергающихся вулканов и монстров, рыщущих за добычей – именно таков ландшафт, на фоне которого GPL вынуждена выживать и приспосабливаться к вечно-изменяемому облику ПО. Тон высказываний сдержан, но ощущается некоторая снисходительность: «Разработчики, пользующиеся GNU GPL, защищают ваши права в две ступени: (1) утверждают авторское право на программное обеспечение и (2) предоставляют вам настоящую Лицензию, дающую законное право копировать, распространять и/или модифицировать данное программное обеспечение».

Два главных вражины FSF предьявлены в конце преамбулы (чтоб далее в лицензии их отдубасить): это DRM и патенты. Судите сами, вот строчки из соответствующих параграфов: «...GPL гарантирует, что программное обеспечение, находящееся под её защитой, не будет подвергнуто, и не может подвергаться другому программному обеспечению, цифровым запретам, обход которых запрещён» и «...GPL объявляет, что любой

валет эту мысль: если вы поставляете объектный код, то должны сделать доступным исходный, а также уведомить пользователя о его праве доступа к исходному коду. Этой политики придерживалась и вторая версия GPL, но здесь даётся точное определение, а не намёк.

### Ограничение прав

#### (3) Система цифровых запретов

Во втором разделе, «Основные полномочия», ничего нового нет, зато третий – целиком новый: в GPL2 раздел под названием «Система цифровых запретов» отсутствовал. Такое определение принято в пику корпоративному термину – «система цифровых прав». FSF понимает значение слова «права» здесь как «...пропаганда, принятая с целью вынудить вас рассматривать вопрос с точки зрения тех немногих, что налагают ограничения, игнорируя права множества тех, на кого ограничения налагаются». Если дерзкая замена слова «права» на «запреты» не прояснит для

различия лицензиара» предоставляют FSF широчайшую свободу действий в достижении своих целей.

Мнение Линуса Торвалдса в отношении DRM полностью противоположно. В 2003 г. он писал в список рассылки ядра Linux: «Я считаю, что DRM насколько не мешают Linux!». Линус хотел быть прагматичным, защищая право любого человека использовать Linux так, как он пожелает. Но первая редакция GPL3 вызвала бурное обсуждение в том же самом списке рассылки, и Линус по-прежнему в гуще дебатов.

### Выражения неточны?

#### (5) Распространение модифицированных версий исходных кодов

Не дав вам и дух перевести, следующий раздел принимается за ASP, упомянутых выше. Раздел 5[2], «Распространение модифицированных версий исходных кодов», вызвал почти столько же комментариев,

Собственно говоря, на web-сайте FSF и во всём Интернете была высказана изрядная доза критики по поводу расплывчатости формулировок этого раздела. Например, в конце раздела (с) GPL делает попытку принудить ASP-разработчиков открыто объявлять о том, какое программное обеспечение GPL они использовали и какие изменения в него внесли: «Если интерфейс предоставляет список пользовательских команд или опций, наподобие меню, то команда отображения данной информации должна бросаться в глаза в этом списке. В противном случае, модифицированная программа должна отображать данную информацию при загрузке – за исключением случаев, когда Программа работает в интерактивном режиме и не может отображать такую информацию при загрузке».

Понятия «меню» и «загрузочный экран» могут быть истолкованы весьма широко. Не нужно быть юристом, чтобы найти множество способов уклониться от предоставления информации об использовании чужом программном коде.

Далее в этом разделе проект пытается ещё более ограничить поставщиков, использующих чужой код: «...если вы распространяете эти программы в сочетании с защищённым (Лицензией) программным обеспечением, независимо от формы сочетания, то вся совокупность должна подчиняться настоящей Лицензии, чьи правила должны быть распространены на всю программу, и, следовательно, на любую её часть».

Другими словами, если вы выполняете приложение GPL3 через сеть, то должны предоставить не только исходный код, но и все модификации, предпринятые для расширения функциональности. Это должно помочь фирмам, обеспокоенным возможным использованием раскрытого исходного кода конкурентами и возможными модификациями без ведома первичного правообладателя.

### О патентах

#### (6) Распространение без исходного кода

Проект лицензии охватывает распространение без исходных текстов, и условия лицензии в основном те же, что в разделе 3 второй версии, с одним существенным отличием. Тут, наконец, в предпоследнем параграфе, выступает на сцену второй враг FSF: «Распространение сопутствующих исходных кодов в соответствии с этим разделом должно осуществляться в общедоступном формате, не быть обременено патентом, и не должно требовать специального пароля или ключа для распаковки, просмотра или копирования».

## «СТОЛМЕН, СТОЯЩИЙ ЗА GPL, НАМЕРЕН ПОЙТИ ДАЛЬШЕ.»



патент должен быть лицензирован для всеобщего свободного использования или не лицензирован вообще». Однако!

### О декодировании

#### (1) Исходный код

В данном разделе начинается детальный разбор прав разработчиков и пользователей. Первое замеченное отличие от версии 2 – разграничение исходного и объектного кодов, проведённое в первом параграфе первого раздела: «Понятие «исходный код» произведения [программы] означает предпочтительную форму для внесения модификаций в нее. Понятие «объектный код» означает любую форму представления программы, кроме исходного кода».

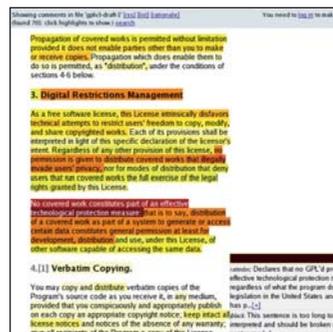
Это важно, так как разъясняет зависимость объектного кода от исходного – вещь, очевидная для большинства из нас, но не столь очевидная для большинства рядовых пользователей – и проект лицензии разви-

вас отношения FSF к DRM, то уже с первой строки третьего раздела сомнения отпадут. «Как лицензия свободного программного обеспечения, настоящая Лицензия всячески осуждает попытки технически ограничить свободу пользователей копировать, модифицировать и распространять работы, защищённые авторским правом».

Под DRM обычно подразумевается использование цифровых ключей, встроенных в программы для предоставления определённых функций (обычно, медиа-содержимого) исключительно покупателям.

По условиям текущей версии GPL продавец может раскрыть исходный код, но не обязан раскрывать цифровые ключи. В GPL3 Столмен и Ко делают попытку заставить продавца вместе с исходным кодом приложения предоставлять исходный код генератора ключей. Слова «любое из положений лицензии должно рассматриваться в свете этого специального заявления о намере-

сколько DRM, но так как формулировки в этом разделе помягче, особой поляризации мнений не произошло. FSF планирует скорее убеждать ASP-разработчиков, чем подчинять их своей воле, и стимулировать их переход на GPL3.



Прокомментировать GPL может каждый. При добавке очередного примечания к тексту фон текста темнеет: от белого, через жёлтый и красный, до чёрного.

Не обременено патентом? Гораздо проще записать это в проекте документа, чем выполнить для любой разработки с открытым исходным кодом; но формулировка не означает, что вы должны выискивать, не нарушает ли ваша работа чьих-либо патентов — это принимается как данность.

В документе, разъясняющем многие изменения в новом проекте, FSF пишет, что у неё нет желания проводить столь же жёсткую политику против патентов, как против DRM.

## Смелый шаг

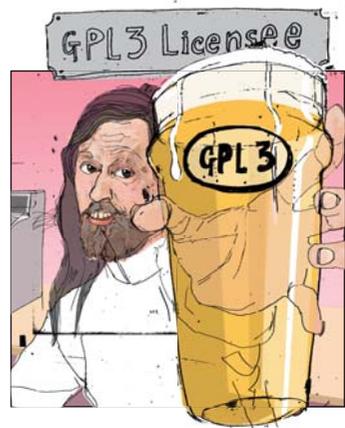
### (11) Лицензирование патентов

Фактически FSF хочет добиться, чтобы вы включали в исходный код разрешения использовать сопутствующие патенты. Это мощный шаг вперёд, его детали уточняются в разделе 11, где разработчикам предлагается обеспечить лицензию на любые патенты, которые могут потребоваться для

заведомо связанную с лицензией на патент, то вы должны принять меры для защиты пользователей программы от возможных юридических преследований, от которых ваша лицензия защищает вас». Это выглядит переложением ответственности за судебные преследования на распространителя. Применительно к Linux это могут быть, например, Red Hat или Novell, но они уже сейчас предоставляют подобные юридические гарантии.

## Драчка!

Что же в новом проекте не нравится Линусу? Его последний протест, направленный в список рассылки ядра Linux, касается той самой строчки в разделе 6, о том, что исходный код не должен требовать «специального пароля или цифрового ключа для распаковки, просмотра или копирования». Вот слова Линуса: «Я считаю безумием требовать от людей опубликовать свои сек-



DRM должны регулироваться медиа-лицензией, так как они касаются музыки, а не программного обеспечения. «Причина, по которой GPL2 оказалась столь успешной в противостоянии с авторскими правами ограничительного типа, заключается в том, что она делала доступным «интересный материал» на условиях, которые запрещали ограничения доступа к нему в дальней-

GPL и некоторых наиболее спорных её вопросах. Если вы серьёзно настроены принять участие в обсуждении GPL3, обратите внимание на те разделы, которых мы здесь не коснулись, включая раздел 8 (Прекращение действия), раздел 9 (Не контракт) и раздел 12 (Свобода или смерть программы).

В FSF хотя, чтобы перемены, предлагаемые в новой версии, со всеми мыслимыми последствиями и политическими уклонами, обсуждались в кафе и столовых, на электронных досках объявлений и в списках рассылки Сообщества свободного ПО. На сайте FSF GPL3 (<http://gplv3.fsf.org>) можно найти исчерпывающие сведения по данному вопросу и страничку для комментариев по поводу различных аспектов первой редакции. Страничка эта сделана весьма хитроумно: цвет фона отдельных слов или предложений лицензии меняется по мере накопления примечаний к ним — от бледно-жёлтого у мало комментируемых мест,



# «НЕЛЬЗЯ ДОПУСТИТЬ, ЧТОБЫ ДЕЛО СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПОГРЯЗЛО В ТЯЖБАХ.»

работы с программным обеспечением «... любому, получившему любую версию программы, предоставлять для любых версий защищаемой (Лицензией) программы, в полном объёме все виды деятельности, разрешённые или подразумеваемые настоящей Лицензией, такие, как установка, эксплуатация и распространение экземпляров программы и использование её результатов».

Основное желание FSF — добиться вашего отказа от контроля, если вы владеете патентом на существенную часть функциональности в любом исходном коде программы, которую вы распространяете под GPL-3. «Данная патентная лицензия не-эксклюзивная, не подразумевающая отчислений и всемирная, и распространяется на все патенты, которыми вы обладаете или на которые обладаете правом сублицензирования, на момент распространения защищаемой программы или в будущем».

Если вы подумали, что таким образом GPL хочет дистанцироваться от исходного кода и любых проблем с патентами, вы окажетесь правы. Патентные тяжбы становятся прибыльным бизнесом, и очень важно уберечь свободное ПО от погружения в трясину судебных разбирательств.

Последний параграф 11 раздела предельно проясняет суть дела. «Если вы распространяете защищаемую программу,

ретные ключи, например. Я не могу на это пойти. И я сомневаюсь в том, что конверсия GPL3 коснётся ядра, так как лично я не собираюсь перелицензировать свои коды».

Как известно, отношение Линуса к вопросу DRM в GPL3 многим показалось неправильным, но сомнений нет — переход ядра Linux под GPL3 может состояться только в случае значительного смягчения позиции, занятой авторами в отношении DRM.

Позже Линус уточнил свою позицию —

«... Я считаю, что тема анти-DRM гораздо ближе к лицензиям Creative Commons (CC), чем к лицензиям на ПО».

Любопытно, что Линус ввел в дебаты тему CC-лицензий, если учесть, что в февральском интервью Столмен отверг их все...

## Ваше слово

Теперь у вас наверняка сложилось мнение о важнейших аспектах первого чтения новой

через красный до чёрного. Пестрее всех, естественно, разделы о DRM.

Если вас не устраивает направление, в котором движется GPL-3 — выскажите своё мнение. Комментарии будут обсуждены и приобщены к делу. Пусть Сообщество узнает вашу точку зрения — еще не поздно все изменить! **LXF**

## В GPL3 – ЧЕРЕЗ ПИВНУЮ

Вам не до всяких там лицензирования? Помните: именно сейчас обсуждаются три ключевых вопроса...



### • Система цифровых прав/запретов

Действие DRM основывается на зашифрованном компоненте для определения персональных прав пользователя. В основном DRM используются в загружаемых музыкальных файлах, когда ваши личные данные в зашифрованном виде заносятся в сам файл, в плеер и компьютер.

FSF и Ричард Столмен считают этот метод контрпродуктивным и хотят убедить разработчиков ПО никогда не использовать DRM в GPL-коде. Они планируют добиться этого требованием предоставить «...любые шифровальные или авторизационные коды, необходимые для установки и/или эксплуатации исходного кода программы», как указано в первом разделе проекта GPL3. То есть если разработчики используют шифрование для обеспечения DRM в своих приложениях, то обязаны также публиковать исходные коды генераторов цифровых ключей.

В результате каждый сможет создать собственные ключи, что сильно затруднит применение DRM. Это может нанести ущерб Linux-компаниям вроде Tivo, которые, соблюдая требования телекомпаний, работают с зашифрованным ТВ-контентом.

### • Патенты

Отношение к патентам более терпимое, но если разработчик обладает патентом на программу, лицензируемую под GPL-3, то должен безоговорочно передать патентную лицензию любому обладателю любого экземпляра программы.

Это требование сохраняется для любой версии, модифицированной или полученной из кода-оригинала в любой точке мира. Дистрибьюторам напоминает о необходимости принять меры к защите пользователей от возможных судебных преследований.

### • ASP

В случае публикации своих исходных кодов по GPL2, поставщики услуг типа Amazon и Google рискуют потерей конкурентоспособности. Их соперники смогли бы безнаказанно модифицировать исходные коды, так как пользователи сетевых приложений, в сущности, загружают исходные коды через браузеры. GPL3 пытается заделать эту брешь, требуя предоставлять информацию об авторе исходного кода и упрощая для пользователя процесс загрузки приложения целиком.

# IBM WEBSPHERE COMMUNITY EDITION CONTEST 2006

**К**орпорация IBM и компания LinuxCenter.Ru представляют вашему вниманию конкурс IBM WebSphere Contest 2006!

Это мероприятие нацелено на разработчиков, инструкторов, а также всех, кому интересны технологии Java и Open Source. Что нужно для участия? В первую очередь – установить в своей системе IBM WebSphere Application Server Community Edition (WAS CE). Если вы регулярно читаете наш журнал, то уже наверняка знакомы с этим открытым сертифицированным сервером J2EE-приложений, базирующемся на Apache Geronimo. Если же вы по каким-то причинам пропустили апрельский номер – не отчаивайтесь: всю необходимую для быстрого старта информацию можно найти во врезке. Дистрибутив IBM WebSphere Application Server Community Edition можно найти на диске из LXF 5(79): *Сторона 2: Websphere CE/kick-start\_ocr.iso*. Перед установкой сервера данный ISO-образ необходимо предельно записать на CD.

Основная задача конкурса – популяризация IBM WebSphere Application Server Community Edition/Apache Geronimo как открытого и эффективного средства для решения широкого круга различных задач, поэтому мы приветствуем не только готовые к использованию приложения, но и инструменты для интеграции WAS CE с другим свободным ПО (в первую

очередь, Eclipse), проверки компонентов, шаблоны приложений WAS CE/ Geronimo и так далее. Конечно, ваш творческий полет не ограничивается написанием кода – мы будем рады видеть методические материалы: серьезные статьи, планы учебных курсов, технические задания для учебного проекта и так далее.

Победители конкурса будут определены Экспертным советом, в который войдут представители ведущих софтверных компаний. Подведение итогов конкурса и торжественная церемония награждения победителей пройдет на выставке LinuxLand/SofTool'06 (Москва, ВВЦ, 26-29 сентября 2006 года). Победители конкурса в каждой из номинаций получат ценные призы, предоставленные организаторами конкурса:

- ноутбуки от IBM
- программное обеспечение, литература и атрибутика от LinuxCenter.Ru
- подписка на журнал Linux Format от редакции журнала.

Статьи и другие методические работы лауреатов конкурса будут опубликованы на сайте IBM developerWorks и на страницах генерального информационного спонсора – журнала Linux Format. Адрес в интернете: [www.linuxformat.ru/contest](http://www.linuxformat.ru/contest)

Впереди – целое лето, а как показывает опыт программы Google Summer of Code, за это время можно успеть многое. Желаем удачи в ваших начинаниях!

## WAS CE – РУКОВОДСТВО К БЫСТРОМУ СТАРТУ

Для того, чтобы начать работу с текущей версией WAS CE (прежде всего, запустить этот сервер), необходимо:

>> **Использовать одну из следующих операционных систем:**

- Red Hat Enterprise Linux 3/4
- SUSE Linux Enterprise Server v9
- Windows 2003
- Windows XP SP2

Это те системы, на которых WAS CE прошел сертификацию на соответствие спецификациям J2EE 1.4, хотя сервер может работать и на других платформах.

>> **Установить JDK 1.4.2\_9 и старше, но не JDK 1.5.**

Для запуска готовых приложений достаточно иметь Java Runtime Environment (JRE). На компакт-диске Java Kick-start находится JRE 1.4.2 от IBM, но можно использовать и JRE от Sun Microsystems.

Для разработки приложений необходимо наличие JDK 1.4.2 любого из вышеперечисленных производителей. Sun JDK (J2SE 1.4.2) можно бесплатно загрузить по адресу

<http://java.sun.com/javase/downloads/index.html>

После установки JDK имеет смысл добавить в переменную окружения PATH каталог `<jdk_install_dir>/bin`.

Хотя это и не относится непосредственно к WAS CE, многие Java-приложения используют переменную окружения JAVA\_HOME. Ей можно присвоить значение в виде имени каталога установки JDK.

Теперь можно приступать к установке сервера с диска Java Kick-start. Последнюю версию WAS CE можно также загрузить с сайта IBM по адресу:

[http://www.ibm.com/developerworks/downloads/ws/wasce/?S\\_TACT=105AGX28&S\\_CMP=DLMAIN](http://www.ibm.com/developerworks/downloads/ws/wasce/?S_TACT=105AGX28&S_CMP=DLMAIN)

Разработчики могут обращаться к документации как по WAS CE (входит в состав дистрибутива), так и по Geronimo. Документация по Geronimo доступна по адресу

<http://geronimo.apache.org/>.

Статьи на русском языке, объясняющие различные аспекты установки и использования Geronimo, можно найти также по адресу:

<http://www-128.ibm.com/developerworks/ru/>

## УСЛОВИЯ КОНКУРСА

### ОБЩИЕ ПОЛОЖЕНИЯ

1. Участвовать в конкурсе «IBM WebSphere Contest 2006» может любой человек, ознакомившийся с настоящими условиями и согласный с ними, за исключением сотрудников IBM, LinuxCenter.Ru, журнала Linux Format и членов их семей.
2. Учредителями конкурса выступают корпорация IBM и компания LinuxCenter.Ru, генеральным информационным спонсором – журнал Linux Format.
3. Конкурс проводится в период с 10 мая по 10 сентября 2006 года. Поведение итогов конкурса и церемония награждения победителей состоится в ходе выставки LinuxLand/Softool'2006 (Москва, ВВЦ, 26-29 сентября 2006 года). Итоги конкурса будут также опубликованы в ноябрьском номере журнала Linux Format.
4. Победители конкурса в каждой из номинации определяются Экспертной комиссией, состав которой утверждается Учредителями конкурса. Решение Экспертной комиссии является окончательным. Денежный эквивалент призов не выплачивается.

### ПОРЯДОК ПРЕДСТАВЛЕНИЯ РАБОТ

1. Присланные на конкурс работы должны попадать в одну из трех номинаций:
  - **Методическое и информационное обеспечение.** Это может быть большая, серьезная статья с детальным рассмотрением конкретных

особенностей WAS CE версии 1.0 и выше; план учебного курса с учебными примерами и системой контроля усвояемости материала; грамотное техническое задание для учебного проекта: проработанная структура сайта на эту тему и т.п.

### Инструментарий

Это могут быть расширения для Eclipse, облегчающие работу с WAS CE, утилиты для проверки компонентов, перекодировки дескрипторов для перехода на WAS CE с других J2EE-серверов, шаблоны (templates) для разработки приложений с использованием WAS CE.

### • Решения.

Это готовые приложения, компоненты и любые законченные и готовые для использования фрагменты кода, устанавливаемые на серверах WAS CE/Geronimo.

2. Имущественные авторские права на работы, представленные в номинации 1, должны принадлежать участнику конкурса. Это подразумевает, что статья или любая другая методическая разработка не может быть опубликована в печатных или сетевых изданиях (журналах, сборниках, сайтах и т.п.), ограничивающих право автора на размещение материала в других источниках. Код, содержащийся в статьях или методических

разработках, должен быть представлен в форме, удобной для проверки его работоспособности и эффективности, то есть в виде полных исходных текстов и сценариев для сборки.

3. Работы, представленные в номинациях 2-3, должны распространяться на условиях, одобренных Open Source Initiative, то есть по какой-либо открытой лицензии. Распространение кода на тех же условиях, что и WAS CE/Geronimo – приветствуется.
4. Конкурсные работы следует высылать на адрес электронной почты: [contest@linuxformat.ru](mailto:contest@linuxformat.ru). Допускается (а в случае с работами, попадающими в номинации 2-3 – приветствуется) публикация конкурсных материалов на web-сайтах, не ограничивающих имущественные права автора (SourceForge.net, домашние страницы авторов и т.п.). В этом случае на адрес [contest@linuxformat.ru](mailto:contest@linuxformat.ru) следует выслать лишь ссылку на такой сайт с кратким описанием представляемой работы.

### ПОРЯДОК КОНКУРСНОГО ОТБОРА

1. Представленные на конкурс работы будут оцениваться по следующим критериям:
  - 1) актуальность и востребованность (в том числе потенциальная – по мнению экспертов);
  - 2) качество решения и его оформления, включая документацию;
  - 3) степень универсальности решения.

• Contribute a Story • Contribute a Link • View Links • Contact Sid • RSS •

Topics

pdfkit - the PDF Toolkit

pdfkit the pdf toolkit

If PDF is electronic paper, then pdfkit is an electronic staple-remover, hole-punch, binder, secret-decoder-ring, and X-Ray-glasses. Pdfkit is a command-line tool for doing everyday things with PDF documents. Keep one in the top drawer of your desktop and use it to:

- Merge PDF Documents
- Split PDF Pages into a New Document
- Decrypt Input as Necessary (Password Required)
- Encrypt Output as Desired
- Split PDF Pages with FDF Data and/or Flatten Forms

PDFkit allows you to manipulate PDF easily and neatly. It does not require Acrobat, and it runs on Windows, Linux, Mac OS X, FreeBSD and Solaris.

User Functions

SWFTTOOLS  
SWF manipulation and generation utilities

ABOUT  
NEWS  
DOWNLOAD  
DOCUMENTATION  
FAQ

swfc, font2swf

- Manual for swfc, @PreView
- File specification for swfc
- Man Page for swfc
- Man Page for font2swf

swfcombine, swfextract

- Man Page for swfcombine
- Man Page for swfextract
- Short introduction to swfcombine (PostScript)

g2swf, m2swf, swf2swf, swf2pdf, font2swf, img2swf, mp3swf

MAILING LIST

- Man Page for g2swf
- Man Page for m2swf
- Man Page for swf2swf
- Man Page for swf2pdf
- Man Page for font2swf

swfdump, swfstrings, swfinfo

- Man Page for swfdump
- Man Page for swfstrings
- Man Page for swfinfo

Other Documentation

- SWF Tools Wiki

# Инструменты для работы с Flash и PDF

Говоря о Libre Graphics, нельзя не упомянуть неизменных компаньонов традиционных графических форматов – Flash и PDF. Считается, что для работы с ними необходимо приобретать дорогостоящие коммерческие пакеты, предназначенные только для Windows и Mac OS X. А вот и нет! Большинство «бытовых» задач обработки Flash-роликов и PDF-документов прекрасно решаются свободными инструментами. Рассказывает Сергей Яремчук.

## ЧАСТЬ 1. SWFTOOLS – ПАКЕТ ДЛЯ РАБОТЫ С FLASH



Технологии интерактивной web-анимации, основанные на Flash и первоначально разработанные компанией Macromedia, за короткое время завоевали популярность у дизайнеров: они решили немало проблем из области мультимедийного представления информации. При сравнительно небольшом размере Flash позволяет использовать звуковые и графические эффекты, придать сайтам динамичность; а самое главное – специальные программы и модули для работы с Flash на клиентской стороне доступны практически для всех операционных систем и современных web-браузеров. Наиболее популярный инструмент – Macromedia Flash MX. Увы, при всевозможных достоинствах есть у него и существенный недостаток: цена, вынуждающая подумать об альтернативе. На сегодня для операционной системы GNU/Linux разработано большое количество инструментов, позволяющих создавать и работать с Flash. Познакомимся с одним из них.

### Пакет SWFTools

Пакет *SWFTools* (<http://www.quiss.org/swftools>) содержит несколько утилит, при помощи которых можно создавать Flash-файлы и манипулировать уже готовыми. Пакет компилируется и работает под Solaris, Linux, FreeBSD, OpenBSD, HP-UX, MacOS X и Windows 98/ME/2000/XP. Распространяется под лицензией GNU GPL.

Установка обычно никаких проблем не вызывает; кроме того, *SWFTools* включен в большинство дистрибутивов Linux. Например, пользователи ALT Linux могут установить *SWFTools* при помощи **apt-get**:

```
#apt-get install swftools
```

В случае самостоятельной сборки, для получения всех возможностей предварительно потребуется добавить следующие библиотеки:

- **freetype** – <http://www.freetype.org>
- **t1lib** – <ftp://sunsite.unc.edu/pub/linux/libs/graphics/>
- **jpeglib** – <http://www.ijg.org/files/jpegsrc.v6b.tar.gz>
- и библиотеки **avifile** – <http://avifile.sourceforge.net>

Если чего-то не хватит, сценарий *configure* выдаст соответствующее сообщение; но те, кого не интересует отсутс-

твующий функционал (например, сценарий жалуется на отсутствие *ungif*, а вы не планируете работать с GIF-файлами), сообщение могут смело проигнорировать.

### Конвертеры

Это наиболее многочисленная категория утилит, входящая в комплект *SWFTools*: их целых семь. При помощи любой из них можно преобразовать различные файловые форматы в SWF (Small Web Format). Например, для конвертирования PDF в SWF нужно выполнить команду:

```
$ pdf2swf manual.pdf manual.swf
processing PDF page 4 (612x792:0:0) (move:0:0)
```

В итоге будет получен SWF-файл, при просмотре которого в web-браузере будут показаны периодически сменяющие друг друга страницы документа. Если не указать имя выходного файла, результат попадет в **stdout**. При использовании некоторых утилит из состава *SWFTools* для генерирования графики на web-сервере могут пригодиться опции **-C** или **--cgi**, генерирующие дополнительный http-заголовок.

Можно обработать лишь часть документа, указав номера избранных страниц с помощью опции **--pages**:

```
$ pdf2swf --pages 1,2,5-10 manual.pdf manual.swf
```

Чтобы получить SWF из набора PNG-файлов, сохраненных в одном каталоге, используется следующая команда:

```
$ png2swf *.png -o file.swf
```

В результирующем ролике каждому PNG-файлу будет отведен отдельный кадр. Предусмотрены дополнительные параметры, например, сжатие (**--zlib**) – выигрыш достигает 10% объема результирующего файла (впрочем, PNG и так не самый тяжелый формат); размеры окна (**-X** и **-Y**); и, при помощи **--rate**, количество кадров в секунду (число в диапазоне от 1 до 256). Числа должны подбираться с учетом формулы 1/значение, т.е. чем меньше указанное число, тем больше будет выставлен интервал времени либо размер кадра.

```
$ png2swf -r 1 -X 50 -Y 50 --zlib *.png -o file.swf
```

Аналогичные опции имеет утилита *gif2swf*, позволяющая проделать те же процедуры с изображениями в формате GIF:

```
$ gif2swf -X 100 -Y 100 *.gif -o file.swf
```

Утилита для работы с JPEG-файлами, *jpeg2swf*, в работе напоминает предыдущую, но имеет еще две дополнительные опции: **--mx** – для использования сжатия Flash MX H.263 и **--quality** для установки качества изображения, в диапазоне от 1 (хуже) до 100 (лучше).

```
$ jpeg2swf --quality 60 -X 200 -Y 100 *.jpg -o file.swf
```

При помощи *font2swf* можно изготвить Flash-файл из файла шрифтов:

```
$ font2swf arial.ttf -o font.swf
```

Поддерживаются все распространенные форматы шрифтов: TTF, AFM, PFA, PFB и другие. К сожалению, невозможно указать кодировку, поэтому в результирующем файле локализованные шрифты выводятся неправильно (*рис. 1*).

Следующая утилита – *wav2swf*. По названию ясно, что она позволяет получить SWF из звуковых файлов в формате WAV. При кодировании используются кодеки LAME. В общем случае вызов утилиты прост:

```
$ wav2swf -v voice.wav
```

В результате в текущем каталоге появится файл **output.swf**, при запуске которого в браузере зазвучит аудио. С помощью дополнительных опций можно изменить битрейт (**--bitrate**), по умолчанию равный 32, частоту дискретизации (**--samplerate**), время повторения фрагмента (**--loop**) и другие параметры. Например:

```
$ wav2swf -v -loop 5 --definesound voice2.wav -voice.swf
```

И, наконец, назначение последнего конвертера, *avi2swf*, тоже понятно – он преобразует AVI в SWF. Для кодирования используется кодек H.263. Принцип работы с утилитой аналогичный:

```
$ avi2swf file.avi -o output.swf
```

Для уменьшения размера конечного файла можно поступить с качеством: изменить 80 (значение по умолчанию) на 60, а также выбрать масштаб как 50% исходного. Вместо кодирования всего файла, ограничимся первыми 100 кадрами:

```
$ avi2swf --num=100 --scale=50 --quality=60 file.avi -o
```

```
output.swf
```

## Утилиты для извлечения информации из Flash-файла

Самая простая в использовании утилита – *swfstrings*, извлекающая текст из swf-файла:

```
$ swfstrings filename.swf
```

В результате на экран выведется текст, содержащийся в файле. Учитывая количество сопутствующего мусора, вывод лучше будет перенаправить в другой файл:

```
$ swfstrings filename.swf > text.txt
```

Утилита *swfextract* способна на большее. Она может извлекать все объекты, содержащиеся в swf-файле: отдельные кадры, шрифты и звук. Если вызвать ее без параметров, вы получите список объектов, которые содержит файл:

```
$ swfextract file.swf
```

Objects in file file:

```
[-i] 190 Shapes: ID(s) 1, 3, 5, 18, 20, 24, 27, 31, 35, 37, 39, 41, 43, 45-47, 51, 52, 54, 56, 58, ...
```

```
[-i] 46 MovieClips: ID(s) 2, 4, 21, 57, 61, 72, 81, 89, 102, 107, 110, 114, 117, 119, 133, 136, ...
```

```
[-j] 74 JPEGs: ID(s) 55, 59, 69, 74, 82, 91, 99, 105, 112, 120, 122, 124, 126, 128, 134, 141, ...
```

```
[-p] 14 PNGs: ID(s) 17, 23, 34, 36, 38, 40, 42, 96, 243, 437, 451, 455, 457, 463
```

```
[-F] 10 Fonts: ID(s) 6, 12, 15, 25, 28, 32, 94, 183, 312, 380
```

```
[-f] 1 Frame: ID(s) 0
```

```
[-m] 1 MP3 Soundstream
```

Обратите внимание на буквы в квадратных скобках, которые идут в начале каждой строки. Они специфицируют объект извлечения. Например, нужно добыть звук:

```
$ swfextract -m file.swf
```

```
NOTICE Writing mp3 data to output.mp3
```

Как видите, все просто. Вывод в WAV-файл возможен, но его необходимо обязательно указывать вместе с опцией **-o**.

Попробуем теперь извлечь PNG-изображения, исключая пустые фреймы (**--hollow**):

```
$ swfextract --pngs 17,23,34,36,38,40 file.swf --hollow
```

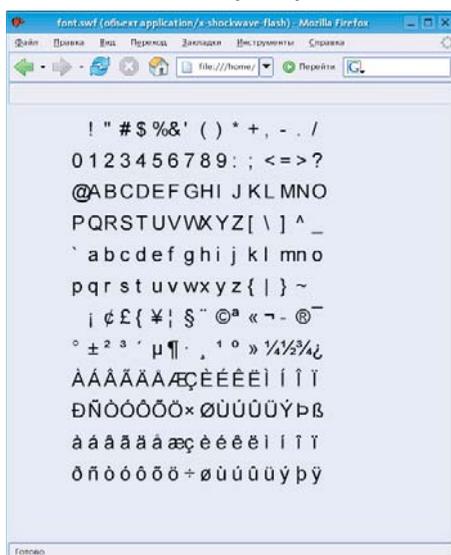
В результате в текущем каталоге появятся файлы вида **pic17.png**, **pic23.png** и т.д. А отдельный фрейм извлекается командой

```
$ swfextract -P -i 2 file.swf -o frame2.swf
```

либо

```
$ swfextract --frame 2 file.swf -o frame2.swf
```

Рис.1 Результат работы font2swf.



Утилита *swfdump* выводит информацию о swf-файле. Для вывода всей служебной информации наберите:

```
$ swfdump --full file.swf
```

```
[HEADER] File version: 6
```

```
[HEADER] File size: 3091206
```

```
[HEADER] Frame rate: 18.000000
```

```
[HEADER] Frame count: 7165
```

```
[HEADER] Movie width: 550.00
```

```
[HEADER] Movie height: 400.00
```

```
[009] 3 SETBACKGROUNDCOLOR (00/00/00)
```

```
[002] 32 DEFINESHAPE defines id 0001
```

```
l fillstyles(01) linestyles(00)
```

```
l 1 ) SOLID 000000ff
```

```
l
```

```
l fill: 00/01 line:00 - moveTo 88.50 15.50
```

```
l fill: 00/01 line:00 - lineTo -88.50 15.50
```

```
l fill: 00/01 line:00 - lineTo -88.50 -15.50
```

```
l fill: 00/01 line:00 - lineTo 88.50 -15.50
```

```
l fill: 00/01 line:00 - lineTo 88.50 15.50
```

При помощи *swfdump* можно также генерировать HTML-код, пригодный для непосредственного включения ролика в web-страницу:

```
$ swfdump --html file.swf
```

```
<OBJECT CLASSID="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
```

```
WIDTH="550"
```

```
HEIGHT="400"
```

```
CODEBASE="http://active.macromedia.com/flash5/cabs/swflash.cab#version=6.0,23.0">
```

```
<PARAM NAME="MOVIE" VALUE="file.swf">
```

```
<PARAM NAME="PLAY" VALUE="true">
```

```
<PARAM NAME="LOOP" VALUE="true">
```

```
<PARAM NAME="QUALITY" VALUE="high">
```

```
<EMBED SRC="file.swf" WIDTH="550" HEIGHT="400"
```

```
PLAY="true" ALIGN="" LOOP="true" QUALITY="high"
```

```
TYPE="application/x-shockwave-flash"
```

```
PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
```

```
</EMBED>
```

```
</OBJECT>
```

Кроме того, при помощи параметра **--text** можно извлечь текстовые поля Flash-файла (подобно *swfstrings*):

```
$ swfdump --text file.swf
```

## Манипулирование данными

Этот раздел представлен всего одной утилитой – *swfbox*, при помощи которой можно пересчитать параметры SWF-файла, а заодно уменьшить его размер. Данные о файле можно получить, запустив утилиту без параметров.

```
$ swfbox file.swf
```

```
Original Movie Size: 550.00 x 400.00 :0.00 :0.00
```

Возможна оптимизация, при помощи опции **-O**.

```
$ swfbox -O file.swf -o optimiz.swf
```

Признаться, мне редко удавалось оптимизировать SWF-файл больше, чем на 1 %.

Результат оптимизации можно просмотреть, используя параметр **-S**.

```
$ swfbox -S file.swf
```

```
{
```

```
{frame 0}
```

```
}
```

```
{
```

```
{frame 1}
```

```
{ID5 {0.00 50.00 550.00 50.00 0.00 348.40 550.00
```

```
348.40}}
```

```
{ID7 {76.50 121.60 115.60 121.60 76.50 132.65 115.60 132.65}}
```

и т.д.

## Объединение Flash-файлов

Все описанные утилиты перекодировали файлы в SWF из разных форматов, оптимизировали, извлекали служебную информацию. Теперь осталось только объединить результаты их работы. Для этой цели предназначена утилита *swfcombine*, позволяющая «склеить» два или более SWF-файла в один. Один из файлов обязательно должен быть определен как главный – *master*, остальные подключаются по мере необходимости как подчиненные – *slaves*. Кроме того, при помощи *swfcombine* можно сжимать либо распаковывать SWF-файлы. В этом случае при использовании одного файла необходима опция **--d (--dummy)**, не требующая указания файла *slave*. Для сжатия наберите команду:

```
$ swfcombine -dz file.swf -o file_new.swf
```

Распаковка происходит аналогично:

```
$ swfcombine -d file.swf -o file_new.swf
```

Теперь соединим два файла вместе. В духе Энди Ченнела, попробуем наложить звук на изображение:

```
$ swfcombine -o combined.swf picture.swf slave=sound.swf
```

Если количество *slave*-файлов велико, их можно предварительно объединить, чтобы оценить результат и упростить подбор параметров.

```
$ swfcombine --cat name1=slavefile1.swf name2=slavefile2.swf -o slave_res.swf
```

Можно добавить масштабирование (**--scale**) результирующего файла и количество кадров в секунду (**--rate**).

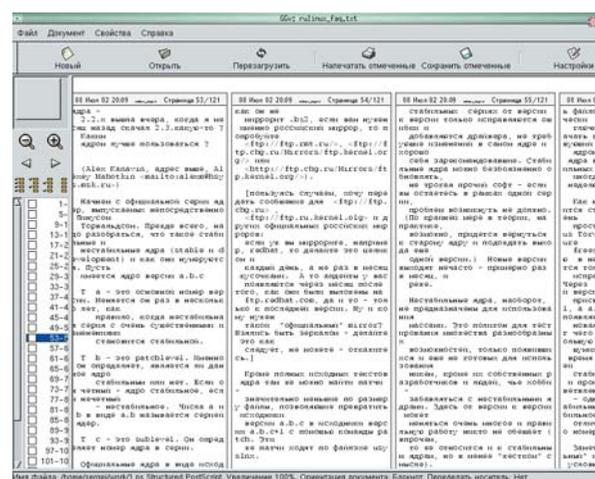
И, наконец, последняя утилита – *swfc*. С ее помощью можно скомпилировать сценарий *sc* в готовый Flash-ролик. Делается это очень просто (пример скрипта можно найти в подкаталоге *doc* пакета *swftools*):

```
$ swfc fileformat.sc -o file.swf
```

Инструменты, входящие в состав пакета *SWFTools*, позволяют выполнить большую часть операций по работе с файлами в формате Flash. Многие опции унифицированы, а значит, пара-тройка экспериментов – и вы уже полностью освоили работу с пакетом.



Рисунок 2. Утилита IPDF (см. следующую страницу) позволяет создать PDF-документ с закладками.



## ЧАСТЬ 2. РАБОТАЕМ С PDF В LINUX

### Работаем с PDF в Linux

Работу с большим количеством документов трудно представить без формата PDF, который стал особенно популярен с развитием Интернета и распространением электронной документации. Для создания этих документов пользователи Windows зачастую применяют отнюдь не самые дешевые инструменты. Однако сторонники GNU/Linux могут выполнять практически те же задачи абсолютно бесплатно.

Такие популярные приложения, как настольная издательская система *Scribus*, *GSView*, *OpenOffice.org* и другие, позволяют экспортировать в PDF вновь созданный или открытый документ любого поддерживаемого этими приложениями формата. Кроме того, существуют утилиты, позволяющие выводить документ на псевдопринтер, который также создает PDF-файл. К сожалению, получившийся документ не всегда отвечает всем требованиям, да и подход этот, несмотря на кажущуюся простоту, имеет смысл при работе с небольшим количеством документов. При одновременной обработке нескольких файлов или желании автоматизировать процесс создания документов удобнее использовать специальные утилиты.

### Pdftk-инструмент для выполнения ежедневных задач

Порывшись на специализированных сайтах, можно найти десятки утилит для работы с PDF, но первое место по функциональности принадлежит *pdftk* (<http://www.accesspdf.com/pdftk/>) Сида Стюарда [Sid Steward]. С его помощью можно легко объединить несколько документов в один; провести обратную операцию, разделив документ по страницам; расшифровать/зашифровать PDF-файл; заполнить данные форм, добавить или удалить вложения; восстановить поврежденные документы; и так далее. Утилита распространяется по лицензии GNU GPL, работает под Windows, Linux, Mac OS X, FreeBSD и Solaris. Для компиляции в различных дистрибутивах Linux или операционных системах в архиве имеются подготовленные Makefile. Например, для сборки под Red Hat/Fedora нужно ввести следующую команду:

```
$ make -f Makefile.RedHat
```

Справку об утилите можно получить при помощи *pdftk --help*. В общем случае, команда запуска выглядит так *pdftk исходный\_файл(ы) опции выходной\_файл пароль*

Например, следующей командой можно объединить два файла в третий — **outfile.pdf**.

```
$ pdftk in1.pdf in2.pdf cat output outfile.pdf
```

Можно также использовать следующую конструкцию:

```
$ pdftk A=in1.pdf B=in2.pdf cat A B output outfile.pdf
```

Такая запись удобна, если один из входных файлов требует отдельных параметров — например, пароля на раскрытие:

```
$ pdftk A=secured.pdf B=in2.pdf input_pw A=foopass cat A B output outfile.pdf
```

При большом количестве файлов удобно пользоваться маской:

```
$ pdftk *.pdf cat output outfile.pdf
```

Возможна и обратная операция. «Разложим» **example.pdf** на файлы с именем **Page\_<номер\_страницы>.pdf**:

```
$ pdftk example.pdf burst output Page%03d.pdf
```

Исходные файлы должны быть в формате PDF, хотя для некоторых операций понадобятся файлы и в других форматах. Например, в файл формата PDF можно добавить вложение, аналогичное вложению в почтовое сообщение:

```
$ pdftk form.pdf attach_files attachfile.tex output new_file_with_attach.pdf
```

Количество файлов вложений и их формат не ограничены. По умолчанию, все вложения будут помещены на верхнем уровне документа, но, используя инструкцию **to\_page**, можно указать конкретную страницу:

```
$ pdftk in.pdf attach_files table1.html to_page 6 output out.pdf
```

Чтобы, наоборот, извлечь присоединенные файлы — допустим, в подкаталог **source**, выполните:

```
$ pdftk attachment.pdf unpack_files output ~/source
```

Также можно добавить в исходный файл любой фон, который может использоваться в качестве водяного знака. Новый фон должен быть обязательно помещен в файл формата PDF (как это сделать, читайте дальше). Например, добавим к исходному файлу *example.pdf* фон, содержащийся в файле **background.pdf**:

```
$ pdftk example.pdf background background.pdf output result.pdf
```

При помощи *pdftk* можно попробовать восстановить испорченный документ (если повреждения не катастрофичны) — здесь каких-либо специальных опций не предусмотрено:

```
$ pdftk broken.pdf output fixed.pdf
```

— или сохранить метаданные в файл **info.txt**:

```
$ pdftk example.pdf dump_data output info.txt
```

Последняя возможность, о которой хотелось бы рассказать — заполнение заранее подготовленного файла-шаблона данными из специального файла FPF файла или стандартного потока ввода (STDIN):

```
$ pdftk form.pdf fill_form eg.fdf output edit.pdf
```

### Утилита Indexed PDF Creator

Утилита *pdftk* предназначена для работы с уже готовыми PDF-файлами, а вот *IPDF (Indexed PDF Creator)* Стива Слейвена (Steve Slaven) позволяет создавать индексированные PDF-документы из текстовых файлов. Домашняя страница проекта расположена по адресу <http://hoopajoo.net/>. Для установки потребуется библиотека PDFlib (<http://www.pdflib.com/>), свободная для личного употребления. Пользователи дистрибутива ALTLinux могут найти ее в репозитории *Sisyphus*. Основное назначение *IPDF* — помощь в создании безбумажного производства с использованием простых текстовых файлов, которые можно в любую минуту превратить в документ, поддерживающий индексацию и поиск. При запуске без аргументов утилита ожидает данных из STDIN и направляет результат в стандартный поток вывода (STDOUT).

При создании нового документа доступен ряд параметров — рассмотрим, например, параметры настройки. Создадим файл **report.pdf** из исходного **report.txt**:

```
$ ipdf report.txt -o report.pdf
```

Правда? Теперь установим размер документа A2, портретную ориентацию и шрифт Times-Roman с кеглем 12:

```
$ ipdf report.txt -o report.pdf -m A2-p -F Times-Roman -f 12
```

Установим размер верхнего (-T) и левого (-L) полей документов и максимальное количество строк (-I).

```
$ ipdf report.txt -o report.pdf -T 25 -L 25 -I 50
```

Если надо, добавьте информацию об авторе и заголовок документа:

```
$ ipdf report.txt -o report.pdf -a Jaremchuk -t "Report Document"
```

Документ можно украсить фоновым изображением, например, логотипом фирмы (в качестве фона пока поддерживаются только файлы в формате TIFF):

```
$ ipdf report.txt -o report.pdf -b background.tiff
```

При постоянном использовании какого-либо параметра можно записать его в конфигурационный файл **/etc/ipdf.conf** или **~/ipdf.conf**. Любой из параметров допускает текущее переопределение через командную строку.

Например:

```
page_mode      A3-p
font_size      12
font_face      Courier
top_margin     40
left_margin    40
chars_per_line 135
lines_per_page 66
author        Jaremchuk
```

А теперь — самое интересное. В настоящее время *Indexed PDF Creator* поддерживает три типа индексирования т.е. создания закладок в PDF документе: абсолютное (-I), относительное (-I) и использующее регулярные выражения (-X). Одновременно разрешается использовать не более одного типа индексирования.

Абсолютная индексация применяется, когда заранее известно количество страниц. Параметр выглядит так:

```
-I СТРАНИЦА:СТРОКА[:СТАРТ:СТОП:МУЛЬТИИНДЕКС:ПОЗИЦИЯ]
```

Например, чтобы сделать индекс в 1-й, 2-й и 3-й строке, по 2-й строке, даем такую команду:

```
$ ipdf -o report.pdf -I 1:2 -I 2:2 -I 3:2 report.txt
```

Относительная индексация применяется, когда известно имя полей, но количество страниц различно. Формат ее следующий:

```
-i СТРОКА[:СТАРТ:СТРОП:МУЛЬТИИНДЕКС:ПОЗИЦИЯ]
```

Например, создадим индекс для 1-й и 3-й строки, в первой использовав символы от 11 до 20 (рис.2):

```
$ ipdf -o report.pdf -i 1:11:20 -i 3 report.txt
```

И, наконец, при помощи регулярных выражений можно автоматически создать индексы, по любым правилам.

```
$ ipdf -mletter-p -f12 -l65 -x '^Page??' -o result.pdf
```

### Пакет GhostScript

В пакет *GhostScript* (<http://www.cs.wisc.edu/~ghost/>) входит большое количество утилит, при помощи которых можно выполнить практически все необходимые операции по работе с форматами PostScript и PDF. *GhostScript* можно вызывать, набрав *gs* в окне терминала. Возможна интерактивная работа с утилитой, когда все команды вводятся по мере необходимости, а также командная. Например, просто набрав **gs file.ps**, вы можете просмотреть указанный файл, а добавив опцию **-sDEVICE=<devname>** — преобразовать его в другой формат или подготовить для выдачи на принтер определенной марки. С помощью опции **-sOutputFile=** можно указать имя выходного файла. Вот так файл PostScript конвертируется в рисунок с расширением *bmp*:

```
gs -sDEVICE=bmp16m -sOutputFile=test.bmp test.ps.
```

А теперь создадим файл **filename.pdf** из исходного PostScript-файла **filename.ps**:

```
$ gs -q -dNOPAUSE -sDEVICE=pdfwrite -sOutputFile="filename.pdf" "filename.ps"
```

Для преобразования PS в PDF в состав пакета входит специальный скрипт **ps2pdf**. Используется он незатейливо:

```
$ ps2pdf input.ps output.pdf
```

Скрипт **ps2pdf** поддерживает все параметры **gs**. Например, *dCompatibility* позволяет создать PDF-файл, совместимый с требуемой версией PDF. Для создания документа, совместимого с версией 1.4, используем следующую команду:

```
$ ps2pdf -dCompatibilityLevel=1.4 input.ps output.pdf
```

Для удобства можно воспользоваться отдельными скриптами, генерирующими PDF-файл заданной версии. Вот, например, конвертирование PS в PDF 1.4:

```
$ ps2pdf14 input.ps output.pdf
```

*GhostScript* поддерживает около 75 опций, и рассмотреть их все в пределах одной статьи не представляется возможным. Например, при помощи **-dOptimize** можно создать оптимизированный документ, при запросе отдельных страниц которого сервер будет выдавать пользователю именно требуемую страницу, а не весь файл целиком.

```
$ ps2pdf -dOptimize=true report.ps report.pdf
```

Можно применить компрессию, а заодно позволить использовать более качественные шрифты:

```
$ ps2pdf -dUseFlateCompression=true EmbedAllFonts=true report.ps report.pdf
```

## Текст в PS/PDF при помощи **enscript**

Небольшая утилита *GNU enscript* (<http://people.ssh.fi/mtr/genscript/>) позволит с легкостью получить из обычного текстового файла PostScript или PDF. Делается это так:

```
$ enscript filename.txt --output=filename.ps
```

Утилита позволяет задать шрифт (по умолчанию используется Courier 10 pt), установить перенос слов и другие параметры. Например, установим перенос слов, шрифт в выходном документе изменим на Times-Roman кеглем 12 и организуем вывод информации на страницу в две колонки:

```
$ enscript -2 --fontTimes-Roman12 --word-wrap --output=filename.ps filename.txt
```

## Утилита **a2ps**

Это более мощная утилита, с помощью которой можно подготовить документ к выводу на печать в формате PostScript или в файл. *a2ps* расшифровывается как Any to PostScript («Что угодно – в PostScript»), домашняя страница проекта <http://www.inf.enst.fr/~demaille/a2ps/>. Утилита небольшая, а опций у нее много, о некоторых из них мы поговорим подробнее. Представьте ситуацию: вы, как прогрессивный человек, написали себе «шпору» на экзамен на компьютере, но, естественно, в формате A4 нести ее с собой не совсем удобно. Значит, надо уменьшить шрифт и разбить документ на маленькие части. Именно это и делает следующая команда:

```
a2ps --columns=4 --font-size=10 -j -encoding=koi8 shpora.txt -o shpora.ps
```

Мы установили количество колонок текста равным четырем, кегль шрифта – 10, параметр **-j** означает рамку вокруг текста. Еще указаны необходимая кодировка, исходный файл и, с помощью флага **-o**, выходной. На вход можно подать практически все что угодно, хотя при использовании опции **-Z** или **-delegate** файлы делегируются другому приложению для предварительной обработки. Выходной формат – только PostScript, но как превратить его в PDF, вы уже знаете. При помощи *a2ps* можно также распечатать результат на принтере. Например, выведем содержимое и оглавление каталога, на принтер **prnt1**:

```
$ a2ps -P prnt1 --toc work/*
```

При выводе результата работы команды обратите внимание на строки вида.

```
[manual.pdf (pdf, делегирован pdf2ps): страниц: 35, листов: 18
```

```
[pdfioipe-20051114.tar.gz (gzip, делегирован gzip-a2ps): завершился с ошибкой.
```

```
[16_04.tar.bz2 (bzip2, делегирован bzip2-a2ps): завершился с ошибкой. Пройгнорировано]
```

```
[lone.txt (простой): страниц: 3, листов: 2
```

В них показано, какие из приложений должны использоваться для делегирования файла. Если на момент запуска такое приложение в системе отсутствует, то вывод будет завершен с ошибкой. Обычные текстовые файлы, как видите, не требуют внешнего приложения. Попробуем преобразовать файл *test.pdf* в книжный формат и сохранить результат в **book.ps**:

```
# a2ps --book --delegate=on test.pdf -o book.ps
```

```
[test.pdf (pdf, делегирован pdf2ps): страниц: 11, листов: 11
```

```
[Всего: страниц: 11, листов: 11] сохранен в файл `book.ps
```

При помощи *a2ps* можно вывести содержимое файлов просто на экран.

```
$ a2ps -P display sample.ps sample.html
```

А можно обработать почтовый ящик и сохранить результат в файле **mail.ps**, по 4 страницы на листе:

```
$ a2ps --mail -4 mailbox -o mail.ps
```

Параметров много; все указывать не обязательно. Просмотреть настройки по умолчанию можно при помощи команды:

```
$ a2ps --list=defaults
```

## Конвертер **pstoedit**

Теперь решим обратную проблему. С помощью утилиты *pstoedit* (<http://www.pstoedit.net/>) можно конвертировать файлы из форматов PostScript или PDF в любую из более чем 30 других векторных форматов: Flash, CAD, LaTeX, Sketch, Mathematica, Kontour, ImageMagick (а значит, еще и во все форматы поддерживаемые этим пакетом), формат любого устройства GhostScript, LightWave 3D, апплетов Java 1/2, Adobe Illustrator, WMF и Enhanced Windows Meta Files (EMF) и многих других. Список поддерживаемых форматов можно получить, набрав *pstoedit -help*. Кроме того, на сайте доступны дополнительные shareware-модули, позволяющие работать с некоторыми другими форматами.

Несмотря на большое разнообразие опций, освоиться с работой довольно просто. Например, конвертируем файл PostScript в PDF:

```
$ pstoedit -f pdf inputfile.ps outfile.pdf
```

Используемый в этом случае драйвер обеспечивает только самые простые операции, связанные с конвертированием текста; рисунки не передаются. Поэтому для серьезных документов рекомендуется использовать опцию **-f gs:pdfwrite**.

```
$ pstoedit -f gs:pdfwrite inputfile.ps outfile.pdf
```

А вот конвертирование в формат MetaPost:

```
$ pstoedit -f mpost test.ps test.mp
```

Аналогично производится конвертирование в любой другой формат. Некоторые драйвера поддерживают дополнительные опции, которые необходимо задавать в виде **-f "format:option1 -option2"**. Кроме того, доступны и все опции, необходимые при конвертировании. Например, установим в новом документе шрифт Times (по умолчанию используется Courier) и конвертируем только первую страницу документа:

```
$ pstoedit -page 1 -df Times -f gs:pdfwrite inputfile.ps outfile.pdf
```

При помощи опции *split* можно разложить многостраничный документ на страницы. В имени выходного файла обязательно должно стоять **%d**, которое будет заменено соответствующим номером:

```
$ pstoedit -split -df Times -f gs:pdfwrite inputfile.ps out%d.pdf
```

Следующая команда установит размер страницы исходного документа равным A3.

```
$ pstoedit -pagesize a3 -f gs:pdfwrite inputfile.ps outfile.pdf
```

## Изображения в PDF

Хотя описанные выше утилиты умеют конвертировать файлы изображений в формат PDF, легче будет воспользоваться специальными утилитами. В первую очередь это не развивавшийся с 2004 года *PDFKreator* (<http://www.kraus.tk/projects/pdfkreator/>) (рис. 3), позволяющий создать PDF-документ из графических файлов в форматах JPEG, PNG, BMP, TIFF и GIF.

Существует также небольшая написанная на C утилита *img2pdf* ([http://pobox.com/~newt/greg\\_software.html](http://pobox.com/~newt/greg_software.html)), с помощью которой можно получить PDF-файл из рисунков в форматах PNG, TIFF и JPEG. Пользоваться *img2pdf* очень просто:

```
$ img2pdf *.tiff
```

В результате из всех файлов с расширением TIFF, находящихся в текущем каталоге, будет создан PDF-файл **output.pdf**. Возможности указать другое имя не предусмотрено. Поэтому при повторном использовании необходимо убедиться, что ранее созданный файл не пропадет. Кроме того, по умолчанию генерируются страницы размера Letter. Чтобы на выходе получить формат A4 (других не предусмотрено), необходимо перекомпилировать *img2pdf*, предварительно заменив в 108-й строке:

```
currPage = panda_newpage(outputdoc, panda_pagesize_usletter);
```

аргумент **panda\_pagesize\_usletter** на **panda\_pagesize\_a4**. При необходимости использования обеих версий удобно сохранить их под разными именами, например, **img2pdf\_letter** и **img2pdfa4**.

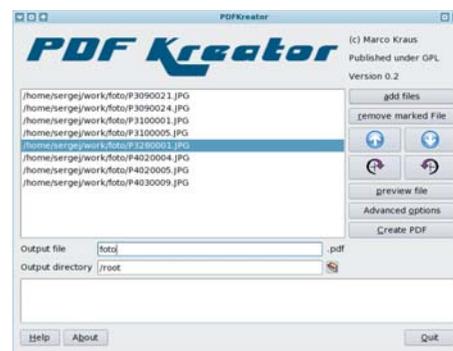
Столь же удобен в применении скрипт **makepdf.sh**, создающий PDF-файлы из файлов изображений, который можно найти на сайте <http://download.novell.com/>. По умолчанию поддерживается PNG; чтобы использовать другой формат, достаточно изменить переменную *imagetype*. К примеру, JPG устанавливается так.

```
imagetype="jpg"
```

Для преобразования используется *ImageMagick* и *pdftk*, так что они должны быть установлены в системе.

Как видите, с PDF-документами в Linux можно делать практически все. И не надо платить большие деньги – все это и многое другое доступно совершенно бесплатно. **LXF**

Рисунок 3. С помощью *PDFKreator* легко конвертировать изображения в PDF.






-  [Free software installer](#)
-  [Application migrator](#)
-  [Application backup](#)
-  [Web hosting](#)

[Version Française](#)

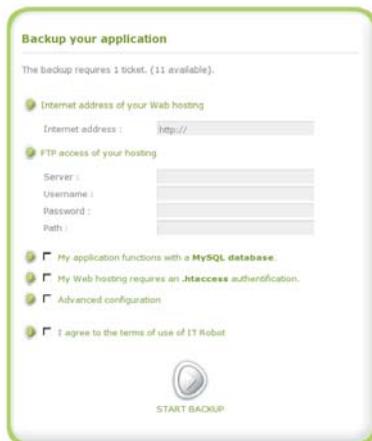
# WEB 2.0

## Новые перспективы

### Final Experimental Phase (\*beta) of the IT Robot

IT Robot services are currently in the final experimentation phase. The goal of the experimental phase is to test developed system and to gather every element ensuring optimal operation services and configuration, prior to the IT Robot's official opening. | [More information](#)

Хотя мы уделили достаточно внимание феномену WEB 2.0 в апрельском номере нашего журнала, шумиха вокруг нового подхода к работе в Сети по-прежнему не утихает. Сегодня **Илья Шпаньков** расскажет еще об одном решении, соединяющем в себе современные технологии и принципы Open Source.



**В** наш обиход уже прочно вошёл термин «Web 2.0», применяемый по отношению к web-ориентированному программному обеспечению. Работа с приложениями, устанавливаемыми не на компьютере пользователя, а на удалённом сервере, обладает вполне реальными преимуществами как для пользователей, так и для самих разработчиков. Пользователь освобождается от таких забот, как своевременное обновление программы, её корректная работа в операционной системе и интеграция с другими приложениями, да в конце концов просто не загромождает свой компьютер массой файлов и папок, используя весь объём жёсткого диска исключительно для хранения рабочих материалов и личных архивов. Авторы подобных программ тоже в выигрыше: нет нужды тратить время и силы на распространение своих разрабо-

ток, организацию службы тех.поддержки (которой часто приходится решать проблемы пользователей просто «вслепую»), следить за своевременной установкой патчей и выполнять массу прочих более мелких, но не менее важных обязанностей. Именно эти положительные стороны нового направления интернет-технологий способствуют бурному росту числа программ, подпадающих под критерии «Web 2.0».

Подобные разработки уже сами по себе отмечены печатью новаторства, но ещё более интересные результаты получаются при их синтезе с другими современными направлениями софтверостроения. И особый простор для творчества предоставляет сочетание новых интернет-технологий с принципами разработки свободного программного обеспечения. Примером может служить заканчивающийся период бета-тестирования ресурс Use-it-Now (<http://www.use-it-now.com/>), созданный швейцарс-

кой компанией KALIX Network при поддержке Free IT Foundation. В основе идеи лежит попытка объединить принципы WEB 2.0 и набирающую всё большую популярность технологию LAMP с целью максимального упрощения и унификации процесса установки и администрирования web-ориентированного программного обеспечения. Термин LAMP расшифровывается, как Linux+Apache+MySQL+PHP и применяется к комплексным программным решениям, построенным с использованием вышеперечисленного ПО. Данная технология позволяет создавать развёрнутые web-сервисы, способные обеспечивать выполнение широкого круга задач, начиная от обычных веб-сайтов и форумов и заканчивая онлайн-магазинами или корпоративной рабочей средой. Объединение этих возможностей с концепцией WEB 2.0 значительно расширяет функциональность отдельно взятых технологий, что и заставило разработ-

чиков из KALIX взяться за реализацию проекта, и сегодня уже можно оценить, насколько успешно они справились с поставленными задачами.

Сами авторы называют свою разработку «IT Robot», в обязанности которого на данный момент входит четыре функции: автоматическая установка программного обеспечения на указанный сервер (или сервер авторов проекта), обеспечение миграции с одного ПО (или сервера) на другое, организация архивирования и создания резервных копий уже существующих в сети ресурсов, а также предоставление свободного места на сервере проекта для размещения своих программ. Основная задача — это, конечно, автоматическая установка приложений, которых сейчас насчитывается в списке более 20: все они являются популярными свободно распространяемыми программами, позволяющими создавать веб-сайты, форумы, организовывать групповую работу и многое другое. Для того, чтобы не запутаться в функциональной принадлежности того или иного приложения, авторы ресурса подготовили краткие аннотации для каждого из них. Собственно установка выбранной программы занимает пару минут, из которых львиная доля уходит на подготовку в виде указания нескольких важных параметров, необходимых для дальнейшего функционирования вновь создаваемого ресурса. После завершения процедуры установки вы становитесь обладателем и полноправным администратором полностью настроенного и работоспособного приложения.

Можете создавать базу пользователей и приглашать друзей-знакомых-коллег к работе с новоиспечённым веб-ресурсом.

Остальные функции IT-робота «родились» в процессе реализации основной идеи как вспомогательные, что, впрочем, не мешает использовать их отдельно от остальных. Если, например, вы решили воспользоваться услугами IT-робота для создания своего нового веб-сайта, но при этом не хотите потерять уже существующие данные, функция миграции окажется как нельзя кстати. Подготовка занимает чуть больше времени, чем при простой установке нового ПО, так как в данном случае вам требуется указать ровно в два раза больше параметров (источника миграции и места назначения), но перенос всех необходимых данных занимает считанные секунды. Разработчики IT-робота гарантируют, что все ваши уже существующие архивы, пользовательские базы данных и прочие ценные материалы, потерять которые совсем не хочется, будут аккуратно переселены на новое место жительства. И никакого волшебства здесь нет: одним из важнейших признаков технологии LAMP является унификация структуры данных ресурсов, что и позволяет в автоматическом режиме переносить пользовательские настройки, архивы, базы данных и прочее из одного приложения в другое. По такой же простой схеме происходит и перенос одного и того же ресурса на другой сервер.

Также одной из важнейших процедур при работе с сетевыми ресурсами является архивирование или резервное копирование

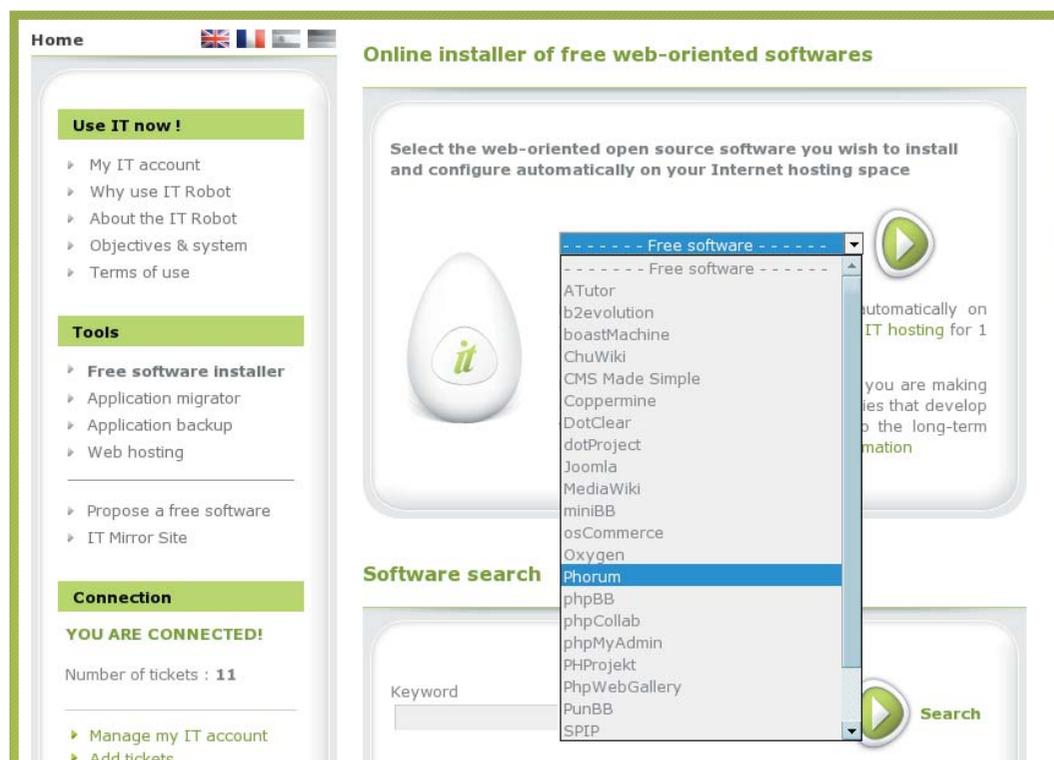
накопленных за время работы данных. С помощью IT-робота данная процедура осуществляется легко: после указания источника все важные файлы копируются в сжатом виде на ваш локальный компьютер. Допустимый объем сохраняемых материалов может составлять до 1 гигабайта данных в течении одной сессии. Таким образом, с помощью IT-робота можно не только за считанные минуты создавать с нуля полностью работоспособный ресурс, но и осуществлять практически все важнейшие функции администрирования, от тонкой настройки установленного программного обеспечения до создания резервных копий данных. Чтобы уж совсем достичь полной независимости от внешних условий, разработчики IT-робота предоставляют возможность аренды свободного дискового пространства на своих серверах — от 100 мегабайт и выше в зависимости от потребностей пользователя. При этом данная услуга предлагается в комплекте со всеми популярными функциями, включая FTP-доступ и возможность работы с использованием защищённых протоколов.

Примечательно, что проект Use-it-Now базируется исключительно на свободном ПО: к установке с помощью IT-робота предлагается только программное обеспечение, распространяемое под лицензиями GPL или BSD, да и исходный код всего проекта после окончания бета-тестирования также будет доступен всем желающим на свободной основе. При этом все поступления от работы сервиса, который ввиду масштабовности просто физически не может предо-

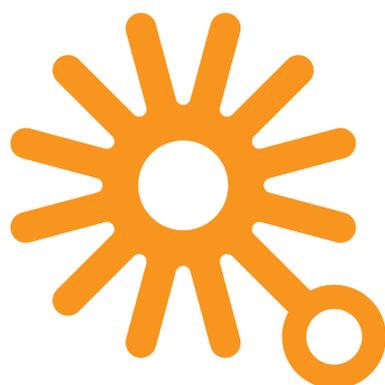


ставляться на бесплатной основе, будут использоваться не только на поддержание проекта в рабочем состоянии, но и распределяться между командами разработчиков того самого свободного ПО, используемого в процессе функционирования ресурса. Таким образом, успешность данного проекта будет ещё одним подтверждением того, что на разработке и распространении программного обеспечения с открытыми исходными кодами можно и нужно зарабатывать вполне реальные деньги, а также позволит оказать реальную помощь всему сообществу разработчиков и пользователей Free Software в целом. Вполне вероятно, что открытые исходные тексты IT-робота станут своего рода стартовой площадкой для новых, ещё более глобальных проектов.

В целом можно сказать, что разработчики Use-it-Now решили поставленные задачи вполне успешно: автоматизация установки и обслуживания web-ориентированного программного обеспечения с помощью IT-робота позволяет даже самому неискущённому в компьютерной грамоте человеку заниматься созданием и управлением довольно сложными сетевыми ресурсами, что в очередной раз разрушает миф о сложности Free Software для обычного пользователя. Более того, не последнюю роль в этой простоте сыграл именно тот факт, что авторы проекта используют открытое ПО, позволяющее с помощью небольших изменений исходного кода получать качественно новые программные продукты. Можно не сомневаться, что подобный проект — лишь начало бурного расцвета как аналогичных по функциональности ресурсов, так и других, ещё более интересных решений. **LXF**



Т е х н о л о г и я   с ч а с т ь я



**SUN**RADIO.RU

сетевое радио под ключ на базе Linux ✖ новое будущее вашей компании

pr@sunradio.ru +7 812 955 76 70 www.sunradio.ru

# Учебники >>

Наши эксперты помогут вам с любым приложением Linux

## ЗАЧЕМ ПОСЛЕ GOOD ДОБАВЛЯЮТ ENOUGH?



Евгений Балдин подумывает о написании серии статей про TeX.

«Как отдохнул?» – «Классно». «Как зарплата?» – «Жить можно». Есть разница? Как лучше? Что-то делается с душой и удовольствием, а что-то суровая и независимая от индивидуальности реальность. С написанием программ та же ситуация.

В коммерческом программном бизнесе торжествует идея «good enough». Невозможно же вылизывать программный продукт до бесконечности, кроме того улучшения можно реализовать в следующей версии. Здесь цель вовсе не качество, а прибыль. Затраченный ресурс необходимо окупить. Сверхприбыль оправдывает всё.

А пользователь? А что пользователь? О нём заботятся. «Простому пользователю» «good enough» качество в самый раз. Зачем думать, когда можно мышкой ткнуть? Зачем автоматизировать, когда можно ткнуть можно много раз?

«good enough», собственно говоря, имеет право на жизнь. Когда ресурсы ограничены, то делается не так как надо, а так чтобы заработало. Но если программа востребована и если её можно усовершенствовать для своих личных нужд, то «enough» достаточно быстро выпадает. Свойство свободных программ в том, что у них есть шанс стать совершенными.

P.S. Если хотите увидеть совершенство, то посмотрите в сторону emacs :)

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

Познакомившись с командной строкой, вы наверняка начнете использовать для управления процессами утилиту `ps`. Вызвав ее без параметров, вы получите список процессов текущей сессии. Если вы не выполняете ничего «на фоне», в нем будет всего две строки: `Bash` (или что вы там используете) и сам `ps`. Это не слишком информативно, поэтому многие люди используют `ps` их, чтобы получить список процессов, которые прина-

## В ЭТОМ ВЫПУСКЕ ВЫ УЗНАЕТЕ:

### 66 ПЕРВЫЕ ШАГИ

Энди Ченнел научит вас сводить видео и звук

### 70 OOO BASIC

Используйте макросы, чтобы автоматизировать решение задач в OpenOffice.org. **Марк Бейн** начнет с *Writer*

### 74 HARDCORE LINUX

**Крис Браун** будет мучать вас регулярными выражениями, пока не заболит голова.

### 78 PHP

Добавить многоязыковую поддержку? **Пол Хадсон** знает, что такое Gettext

### 80 ШИФРОВАНИЕ

**Грэм Моррисон** теперь устанавливает Linux только на зашифрованный раздел

### 84 БЫСТРАЯ ЗАГРУЗКА

Вжииик! Ускорьтесь вместе с **Майком Сондерсом**

### 88 QT/KDE: ЧАСТЬ 3

**Андрей Боровский** научит рисовать пользовательский интерфейс одной мышью

### 92 ПРОГРАММИРОВАНИЕ ДЛЯ LINUX

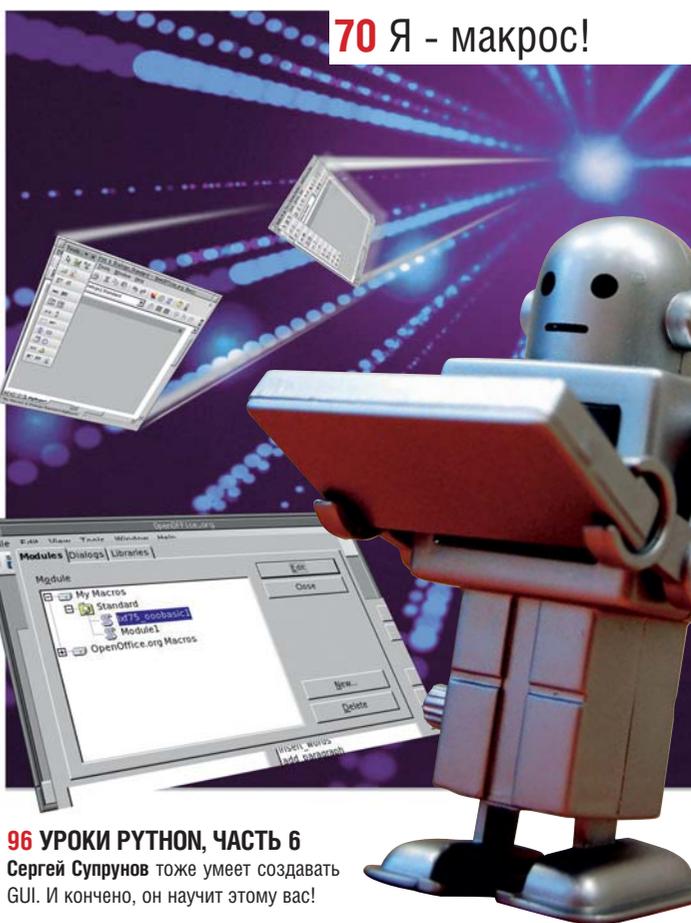
Не нравится KDE? **Андрей Боровский** научит писать программы, не вылезая из консоли

НОВАЯ СЕРИЯ

## СОВЕТ МЕСЯЦА

длежат им или даже `ps aux`, чтобы просмотреть все процессы, работающие в системе.

Мы уже использовали `ps` в LXF72 для отстрела зомби, кроме этого, утилиту можно легко комбинировать с `grep`, например: `ps auxl | grep konqueror`. Как правило, с помощью этой команды находится «подвисший» процесс, которому затем отправляется сигнал 9: `kill -9 pid.PID` – это идентификатор процесса, он-то (в числе проче-



## 70 Я - макрос!

### 96 УРОКИ PYTHON, ЧАСТЬ 6

**Сергей Супрунов** тоже умеет создавать GUI. И конечно, он научит этому вас!

### 100 METAPOST, ЧАСТЬ 5

**Евгений Балдин** готов поделиться с вами секретами мастерства

## УБИВАЯ ВРЕМЯ

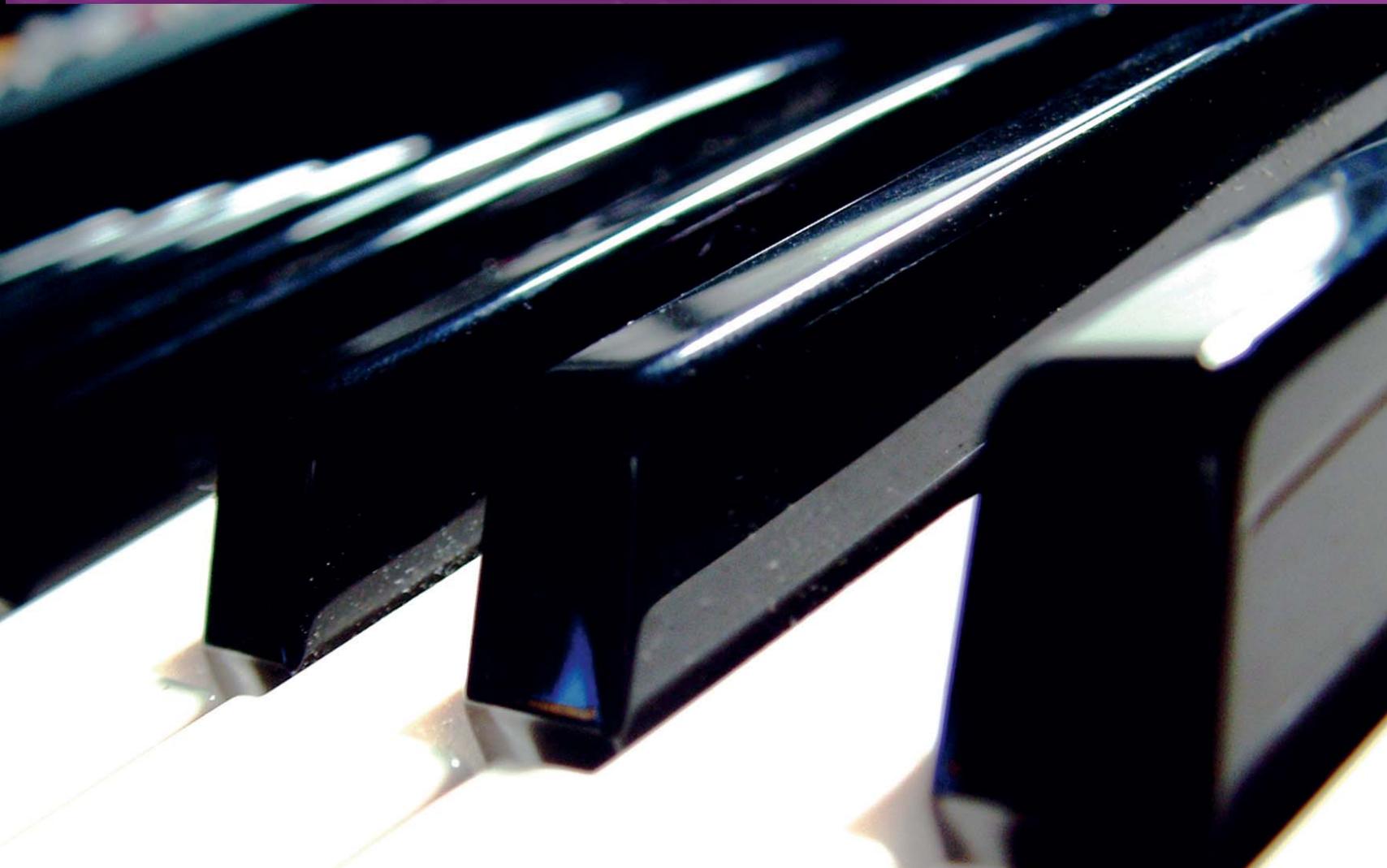
го) и выводится командой `ps`.

Существует и другой способ решения этой задачи – воспользуйтесь командой `pidof`, чтобы получить идентификатор нужного вам (запущенного) процесса. Например, вы можете набрать: `pidof konqueror`. Результат будет выглядеть примерно так:

```
pidof konqueror
18380 18021 24825 13081 6478 6473 6472
```

Вы видите 7 запущенных экземпляров *Konqueror* – и у каждого свой PID. Чем больше номер, тем моложе процесс. В нашем примере, чтобы завершить, последний запущенный экземпляр *Konqueror*, нужно набрать `kill -9 18830`.

Одно из преимуществ `pidof` – возможность получить PID даже в случае, когда вы не можете (или не хотите) анализировать вывод команды `ps`. Это прекрасный инструмент для использования в сценариях, завершающих процессы или изменяющих приоритеты.



ПЕРВЫЕ ШАГИ. СЕРИЯ «LINUX ДЛЯ НОВИЧКОВ»

# Звук: создаем саундтрек к видеозаписи



- Kino 0.8.0
- Audacity 1.2.4b

Добавьте музыку, голос диктора или и то и другое к свои фильмам при помощи *Kino* и *MainActor*. Энди Ченнел расскажет и покажет.

**МЕСЯЦ  
НАЗАД**



Мы подготовили собственный подкаст в Audacity.



Как правило, звук – это та самая часть домашнего видео, о которой не вспоминают до тех пор... пока не услышат результат собственными ушами. Если вы хотите добавить к своим домашним фильмам музыку или голос за кадром, то вам будет несложно приобрести некоторые навыки редактирования и монтажа в открытых программах, а результат не заставит себя ждать.

Для Linux существует несколько приложений для видеомонтажа, как любительского, так и профессионального, однако такого ультра-простого, как *iMovie* для Mac или *SonicDVD* для Windows, до сих пор нет. Для этого руководства можно воспользоваться *Kino* – открытым приложением, которое входит в состав многих дистрибутивов и/или *MainActor*, который стоит денег. Версия 5.5 продается по цене \$199, однако фирма-разработчик ([www.mainconcept.com](http://www.mainconcept.com)) предлагает загрузить бесплатную демо-версию программы, которой вы также можете воспользоваться для этой статьи. Последняя версия *Kino* – 0.8.0 – доступна, как обычно, через менеджер пакетов вашего дистрибутива. Если ее там нет, то вы можете загрузить программу с веб-сайта [www.kinodv.org](http://www.kinodv.org) или взять с нашего

диска. В версии 0.8.0 разработчики добавили новые эффекты и расширения, которые больше не нужно устанавливать отдельно.

Будем считать, что ваш компьютер удовлетворяет базовым требованиям и имеет устройства захвата изображения (то есть у вас есть цифровая видеокамера и интерфейс FireWire), которые настроены и готовы к работе. Устройства с интерфейсом FireWire уже достаточно давно работают без проблем под Linux. Все современные дистрибутивы поддерживают эту шину и работают с ней «на лету».

## ОБ АВТОРСКОМ ПРАВЕ

Если вы читали статью о подкастинге из прошлого номера, то должны помнить, что существуют строгие правила, запрещающие публичную трансляцию защищенного контента, в качестве которого могут выступать музыка и литературные произведения.

## ЧАСТЬ 1. ГОТОВИМ САУНДТРЕК В KINO

**В SUSE 10.0, Kino распознано мою камеру Sony DCR-TRV265E** сразу при подключении, но мне все же понадобилось поменять некоторые настройки программы, которые могут пригодиться и вам (Правка > Настроить).

Для начала убедитесь, что во вкладке Display активирован отрисовщик GDK. В противном случае записанное вами видео будет мерцать, так как в окне предпросмотра будут играть сразу две копии фильма. Во вкладке Defaults выберите из списка 44.1 kHz Stereo. Это стандартная частота дискретизации для большинства устройств, тем не менее, проверьте инструкцию к своей камере, так как некоторые все же требуют 48 kHz. По умолчанию Kino записывает звук на частоте 32 kHz, что очень странно и создает раздражающие дефекты в звуке. Проверьте также остальные настройки. В Великобритании (и в России) значение Normalisation должно быть PAL, а соотношение сторон – 4:3, если у вас, конечно, не экзотическая камера.

Теперь все настроено, и вы можете снимать кино или использовать уже готовые записи. Вы можете прочитать полезные советы о захвате видео по FireWire и редактировании в Kino в моей статье из LXF65 ([www.linuxformat.co.uk/pdfs/LXF65.tut\\_begin.pdf](http://www.linuxformat.co.uk/pdfs/LXF65.tut_begin.pdf)).

Когда дело доходит до сохранения фильма, Kino предлагает три формата файлов, и ваш выбор должен зависеть от того, что вы собираетесь делать с отснятым материалом. Формат RAW DV пригодится, если вы хотите залить фильм обратно на камеру по FireWire или записать на DVD, но он отнимает больше дискового пространства, чем остальные две разновидности AVI и, кроме этого, несколько сложен в обработке. Два варианта AVI, которые поддерживает Kino, позволяют получить небольшие файлы за счет сжатия с потерей качества. Этот вариант больше подходит для видео-дисков (VCD) и короткометражных фильмов для Интернета.

### Склеиваем фрагменты

Записанное видео обычно разбивается на отдельные клипы, которые легко обрезать и пересортировать, после чего на готовый видеоряд можно наложить звук. В общих словах, работа с видео предполагает наличие отдельных клипов исходной записи, между которыми добавляются эффекты перехода. Чтобы нормально работать со звуком, вам нужно будет склеить все фрагменты в один большой клип. Сперва, перейдите на вкладку Edit (Правка) в правой части окна программы и убедитесь, что самый первый

клип выделен на панели слева. Теперь щелкните значок Join Current And Next Scenes (Объединить со следующим фрагментом) на панели инструментов (значок выглядит как горизонтальная линия со сходящимися стрелками) и первые два клипа будут склеены в один. Нажимайте эту кнопку до тех пор, пока у вас не получится один большой клип, а панель воспроизведения не будет одной непрерывной серой полоской (см. рисунки «До» и «После» внизу страницы).

Kino вносит изменения без потери первоначального качества, поэтому ваша оригинальная запись останется неизменной при сохранении, однако после добавления музыки и экспорта готового проекта отделить звук от видео будет уже не так просто. В этой связи, я советую сделать резервную копию звуковой дорожки, экспортировав ее в отдельный WAV-файл (Export > Audio). Можно конечно выбрать и другой формат, но так как Kino может импортировать только WAV-файлы, я рекомендую остановиться на нем. Задайте имя и расположение файла, выберите частоту дискретизации звука (лучше оставить ее без изменения, если вы не собираетесь менять ее намеренно) и нажмите кнопку Export.

### Маэстро, музыку!

Теперь мы можем добавить к записи некоторый музыкальный аккомпанемент. Если вы хотите использовать для него MP3-файл, то вам нужно будет сперва конвертировать его в WAV. Для этих целей существует прекрасный аудиоредактор Audacity, который вы найдете на нашем диске. Когда вы подготовите музыку, перейдите в Kino, в раздел FX и удостоверьтесь, что вкладка Overwrite настроена на работу со всеми фрагментами записи (опция Limit To: не выбрана), а настройки Advanced Options не задействованы.

Теперь вся работа будет происходить в правой части окна. Загляните во вкладку Audio Transition. Не переживайте – мы не будем делать звуковые эффекты перехода. Нас интересует выпадающий список вверху панели, где нужно выбрать значение Mix. Появившееся поле позволит нам указать Kino на расположение файла с музыкой, который будет добавлен к оригинальному звуку видеозаписи. Если последний вам не нужен, выберите из списка Dub вместо Mix.

Просто щелкните по кнопке со значком папки и выберите нужный WAV-файл. Когда файл загружен, вы можете установить его гром-

## АУДИО-ЭФФЕКТЫ

Как Audacity, так и MainActor предлагают массу эффектов для обработки звука, и некоторые из них могут пригодиться при редактировании видео. Такие эффекты обычно распадаются на три категории: задержки, фильтры и переходы.

- К эффектам, основанным на задержках, относятся хор (chorus, увеличивает диапазон частот звука), эхо (echo), реверберация (reverb) и фазер (phaser). Это специальные эффекты, которые, как правило, не используются при обработке домашнего видео.

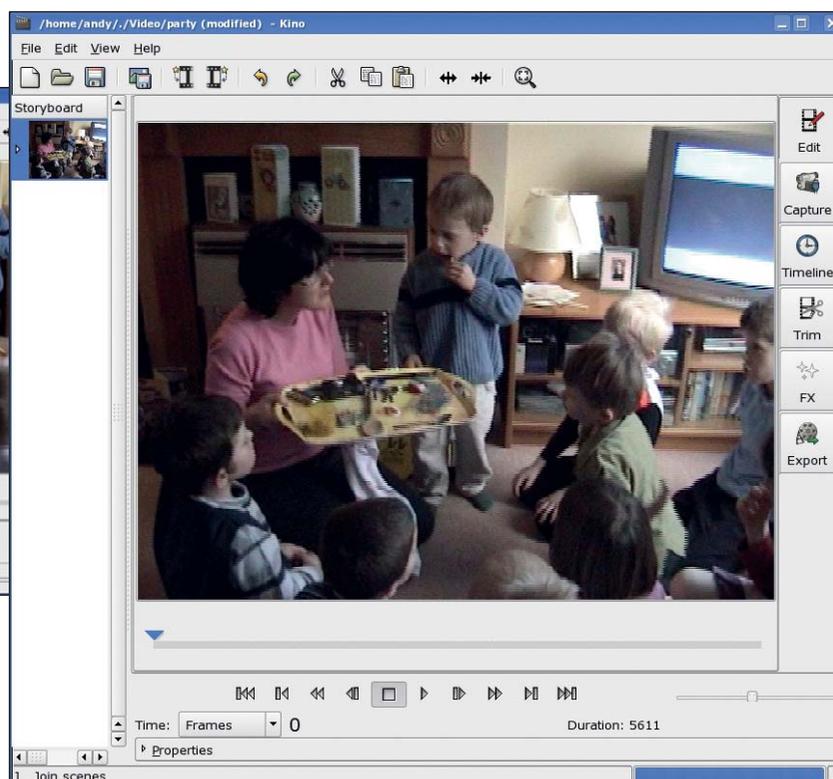
- Фильтры оказывают влияние на акустические качества записи. Например, фильтр High Band Pass удалит из записи все частоты до заданного порога. Это очень полезная функция, которая поможет вырезать нежелательный шум или гул в фильме о вашем холодильнике. Фильтр Low Pass действует по тому же принципу, только срезают не низкие, а высокие частоты, что может понадобиться при редактировании записей Витаса, дабы утихомирить чересчур мощный вокал.

- Переходы позволяют оформить «стыки» между разными записями. В действительности, здесь есть лишь один нужный параметр: скорость перехода. Можно сделать так, чтобы звук менялся более плавно, чем картинка или наоборот.

Также как и в киносъемке или фотографии, лучше делать звуковые эффекты незаметными. Если зритель начнет обращать внимание на эффекты, это означает, что вы переборщили. Если вы этого и добивались – флаг в руки.



До и после: как видно из панели Storyboard в левой части окна Kino, все клипы сведены или «склеены» в один большой кусок, готовый для наложения звука.



«костяк по отношению к «родному» звуку видеозаписи (имеется в виду, что вы не выбрали **Dub** как способ наложения и исходный звук есть) с помощью движка под именем файла, а также сдвинуть начало музыки по отношению к видео на определенное число кадров. Перед тем, как нажать **Render** и наложить музыку на видеозапись, проверьте результат своей работы с помощью кнопки **Preview**. В случае, если длительность музыкального файла превышает длину видеозаписи, звук будет попросту обрезан по окончании видеоряда. Это можно исправить только вручную, отредактировав музыку в *Audacity*, укоротив ее до нужных размеров и добавив плавное затухание в конце.

Это действие можно проделать и без сведения клипов в один кусок, однако результат будет тот же самый – один большой клип с наложенным звуком, а заниматься разбиением на фрагменты вручную удобнее в самом начале. Для того, чтобы наложить звук на серию клипов, повторите опи-

санную процедуру, только убедитесь что на вкладке **Overwrite** указан диапазон кадров: от первого до последнего. Номер последнего кадра можно узнать, щелкнув по последнему клипу и запомнив число в поле **To**.

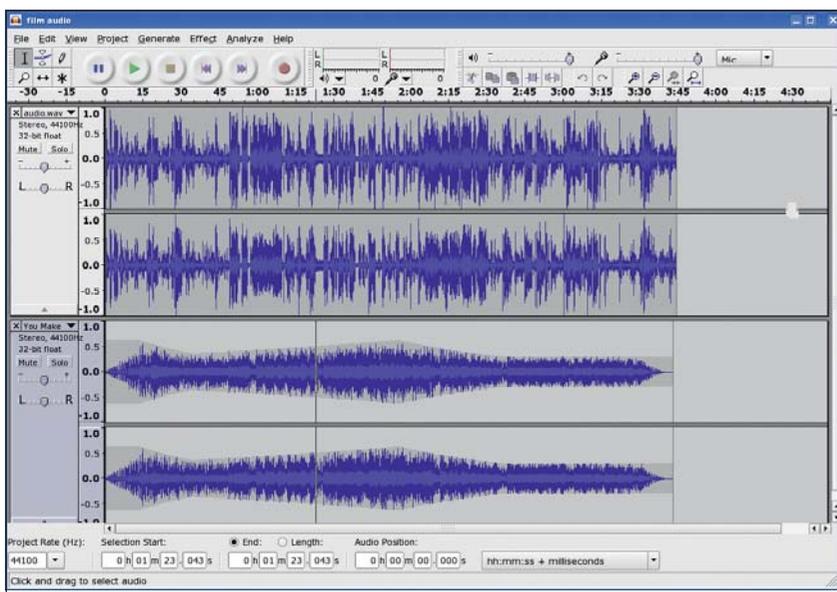
### Расставьте акценты!

Если помимо музыки вам нужен голос за кадром, вы можете повторить все описанные действия для еще одной дорожки, содержащей голос, но в данном случае это слишком грубый и неудобный способ. Звук будет иметь постоянную громкость и может мешать в отдельных частях фильма, а отредактировать его будет невозможно.

Более гибкий метод подразумевает экспорт звука из оригинальной видеозаписи в отдельный файл и наложение на него голоса в *Audacity*, после чего готовый звук вставляется в *Kino* в режиме **Dub**. Как мы знаем, этот режим позволяет затереть любой существующий звук, в отличие от **Mix**, когда звук просто добавляется. Делается все следующим образом. В прошлом номере я уже писал о приглушении звука, когда громкость одной дорожки связана с событиями на другой, и теперь все, в двух словах, должно быть так: вы загружаете исходный звук в *Audacity*, записываете свой голос на отдельную дорожку, и затем меняете ее громкость с помощью инструмента **Envelope**. Когда вы добьетесь нужного результата, экспортируйте всю работу в один WAV-файл, затем вернитесь в раздел **FX** редактора *Kino* и вставьте этот файл в режиме **Dub**. Как и было обещано, старый звук будет заменен новым. Если вы растягивали, сжимали (имеется в виду длительность звучания, а не размер файла) или меняли скорость звука, новая дорожка может не совпасть с видеорядом, но мы всего лишь добавили новый звучащий слой и настроили громкость, поэтому проблем быть не должно.

Может кому-то это покажется излишним, но следует сказать, что *Audacity* поставляется с набором высококачественных эффектов, которые облегчают добавление новых дорожек – таких как закадровый голос – и позволяют править правый и левый канал отдельно. Это удобно когда, к примеру, двое людей в кадре ведут беседу. Если голос одного человека немного сместить в левый канал, а другого в правый, появится нужное ощущение пространства.

Не забывайте сохранять все версии записей в папке вашего видеопроекта. Это будет очень кстати, когда тестовый просмотр выявит недостатки в громкости, и вам нужно будет их исправить.



Так выглядит наложение голоса на музыку в *Audacity*. Вы видите, что я приглушил музыку на нижней дорожке в тех местах, где на верхней ведется важный диалог.

## ЧАСТЬ 2. ПРОДВИНУТОЕ РЕДАКТИРОВАНИЕ В MAINACTOR

Если вам нужен более сложный и изощренный саундтрек, то самое время обратиться к *MainActor 5.5* (для технарей также хорошо подойдет *Cinelegra*). *MainActor* представляет собой проприетарное приложение от компании *MainConcept*. Программа обладает множеством функций и эффектов класса high-end и способна на гораздо большее, чем редактирование домашнего видео.

По скриншотам на соседней странице видно, что программа *MainActor* объединяет в себе подходы *Kino* и *Audacity*: она разделяет звук и видео по отдельным дорожкам. Эта деталь, на первый взгляд незначительная, сильно влияет на результат всей работы.

В *MainActor* нам нужно сделать две вещи. Сначала мы повторим полученные в *Kino* и *Audacity* эффекты и наложим голос диктора, не забыв настроить громкость в зависимости от действия фильма. Затем мы вставим в основную запись тихий и короткий фрагмент, а звук оставим исходный.

Но для начала нам нужен какой-то материал. *MainActor* предлагает собственный инструмент захвата DV, но поскольку мы уже загрузили видео через *Kino*, воспользуемся уже готовым фильмом. В окне **Browser** перейдите на вкладку **Project** и щелкните первую кнопку на панели инструментов – появится диалог выбора файлов. Здесь можно импортировать разнообразные материалы: видеозаписи, звук и фотографии. Все добавленные в проект данные доступны в разделе **Project > Media**. Загрузив нужные файлы, разместите образец видео в окне **Timeline**. Чтобы облегчить монтаж фрагментов, поместите их на две видеодорожки в шахматном порядке. Когда весь материал окажется на своем месте, можно будет приступить к работе со звуком.

### Регулируем громкость

По умолчанию, *MainActor* выделяет звук из каждой видеозаписи на отдельную дорожку, которая помечена как **Audio 1**. Это позволяет программе синхронизировать переходы между видео со звуковыми переходами (когда один звуковой фрагмент плавно переходит в другой). Все изменения относятся только к текущему выделенному клипу. Если мы хотим приглушить звук видеодорожки, к примеру, на 20%, нужно сначала щелкнуть правой кнопкой мыши по клипу и выбрать пункт **Show Parameters (Показать параметры)**. В окне эффектов теперь появится две записи, для звука и видео соответственно. Закройте раздел **video**. В разделе под названием **Audio Source (Звуковая составляющая)** есть три движка: **Master Volume**, **Left Volume** и **Right Volume**, отвечающие за общую громкость и за громкость левого и правого каналов соответственно. Значение громкости выражается числом в диапазоне от 0.00 до 1.00. Передвижение маленькой точки влево по любому из этих движков уменьшит громкость, а вправо – увеличит. Чтобы уменьшить громкость на 20%, установим значение движка равным 0.80.

Для более точной настройки используйте небольшую стрелку-указатель, за которой скрывается редактор кривых. Красную линию с точкой посередине можно двигать вверх-вниз, что скажется на уровне громкости.

Если требуется изменить громкость плавно, например, от 0.00 до 1.00, то вам нужно будет создать дополнительную контрольную точку. Переместите первоначальную точку-узел в нижний левый край графика. Теперь щелкните правой кнопкой мыши в любом месте редактора и выберите пункт **Add Key (Добавить узел)**. Новая контрольная точка



В левом верхнем углу окна *MainActor* выделен редактор кривых, который контролирует громкость звука каждого клипа.

появятся в самой правой части графика, поэтому вы ее сначала не заметите. Используйте движки подстройки громкости, и вы сразу обнаружите новую точку. Перетащите ее в правый верхний угол графика, и между двумя точками будет создан плавный переход. В окне *Preview* включите воспроизведение и оцените результат.

Последовательные узловые точки на кривой позволяют изменять громкость звука на всем протяжении клипа. Теперь мы знаем, как делается приглушение в *MainActor*.

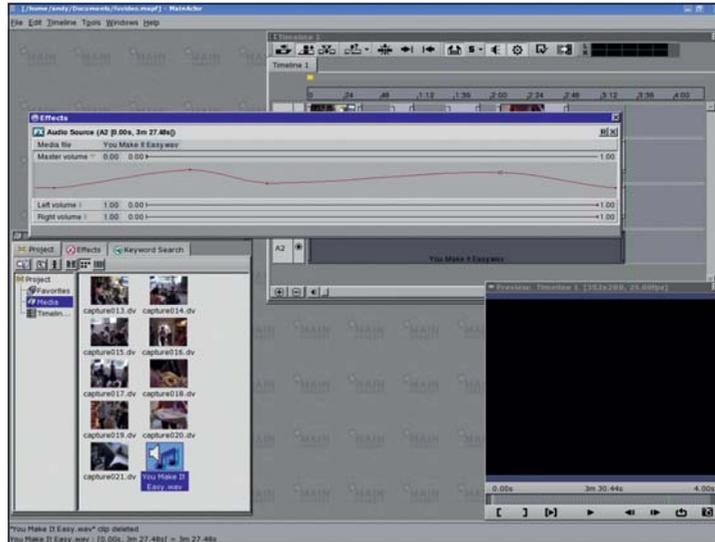
### Приглушаем записи

Чтобы проверить это на практике, мы создадим дорожку с музыкой, которую мы приглушим в тех местах, когда в фильме ведется диалог. Щелкните правой кнопкой мыши по окну *Timeline* и выберите пункт *Add Audio Track* (Добавить звуковую дорожку). Появится пустая дорожка, в которую вы можете сразу же загрузить нужный файл с музыкой. Я опять рекомендую пользоваться форматом WAV, несмотря на то, что *MainActor* понимает и другие форматы аудио (кроме MP3). Перетащите файл из окна с клипами на новую дорожку. Если звук превышает видео по длине, переместите указатель текущей позиции по шкале до того места, где кончается видео и, убедившись, что выделена именно нужная дорожка, нажмите *K* на клавиатуре или щелкните значок *Make Cut At Current Position* (Разрезать в текущей позиции) на панели инструментов *Timeline*. Таким образом вы разделите звуковую дорожку на две части; щелкните правой кнопкой мыши по второму куску и выберите *Delete*, чтобы избавиться от него.

Подобно тому, как мы изменяли громкость звука в предыдущей части, вызовите через правый щелчок мыши окно настройки громкости для нашей новой дорожки. Благодаря тому, что окна в *MainActor* можно растягивать, есть смысл увеличить окно с движками громкости вправо на максимальную ширину для более точной подстройки кривых. Измените красную кривую описанным выше способом. Начните со значения 0.00, продолжите плавным нарастанием по мере развития фильма, сбросьте громкость до трудноразличимых звуков и ближе к финалу медленно наращивайте громкость, после чего сделайте плавное затухание перед самым концом (см. рисунок кривой на рисунке выше справа). Проиграйте результат и смотрите на свою работу в окне *Preview*.

### Добавляем видеоклип

Вторая часть работы предполагает вставку одного небольшого видеоклипа (крупный план) в основной (общий план), причем звуковое сопровождение не должно меняться. К счастью, *MainActor* позволяет делать это с легкостью. Начнем с того, что перенесем два клипа на шкалу *Timeline*. *MainActor* поместит их на разные дорожки, что соответствует нашей задаче: общий план должен быть на первой дорожке, а крупный — на второй. Щелкните правой кнопкой мыши по второму клипу и выберите *Detach Audio From Video* (Отделить звук от видео). Так как нам не нужен этот звук, щелкните по нему и нажмите *Delete*.



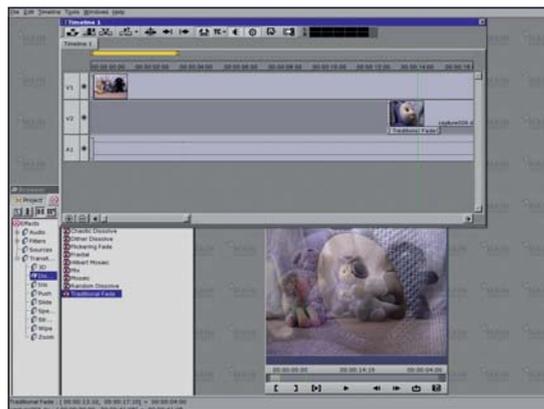
Красная кривая наиболее наглядно показывает изменение громкости по всей длине проекта.

В итоге у вас должно остаться два видеоклипа и одна дорожка со звуком, причем больший клип должен быть на первой дорожке, а меньший — на второй. Для того, чтобы все сработало, убедитесь, что вторая кнопка на панели инструментов *Timeline* — *Overwrite Mode* (Режим замещения) включена. Включите ее мышью или нажмите *2* на клавиатуре. Перетаскивайте вставленный клип вдоль второй дорожки до тех пор, пока он не займет нужное место под общим планом. Теперь, когда вы проиграете смонтированный фрагмент, картинка в нужный момент покажет крупный план, а звук при этом останется прежним.

Вы можете делать любое количество вставок по вашему желанию. Можно немного усложнить проект и перетащить с панели *FX* эффект перехода между клипами. Поместите его в начале вставки с крупным планом и тогда камера перейдет к нему не резко, а, например, через приятный эффект растворения. Поиграйте с эффектами и не бойтесь экспериментировать.

После того, как все исправления внесены, звук отшлифован и эффекты выбраны, щелкните кнопку экспорта на панели *Timeline* (находится в правой части панели инструментов) и выберите нужный формат экспорта. У вас должен получиться один файл, который можно смотреть в любом медиаплеере или записать на VCD или DVD.

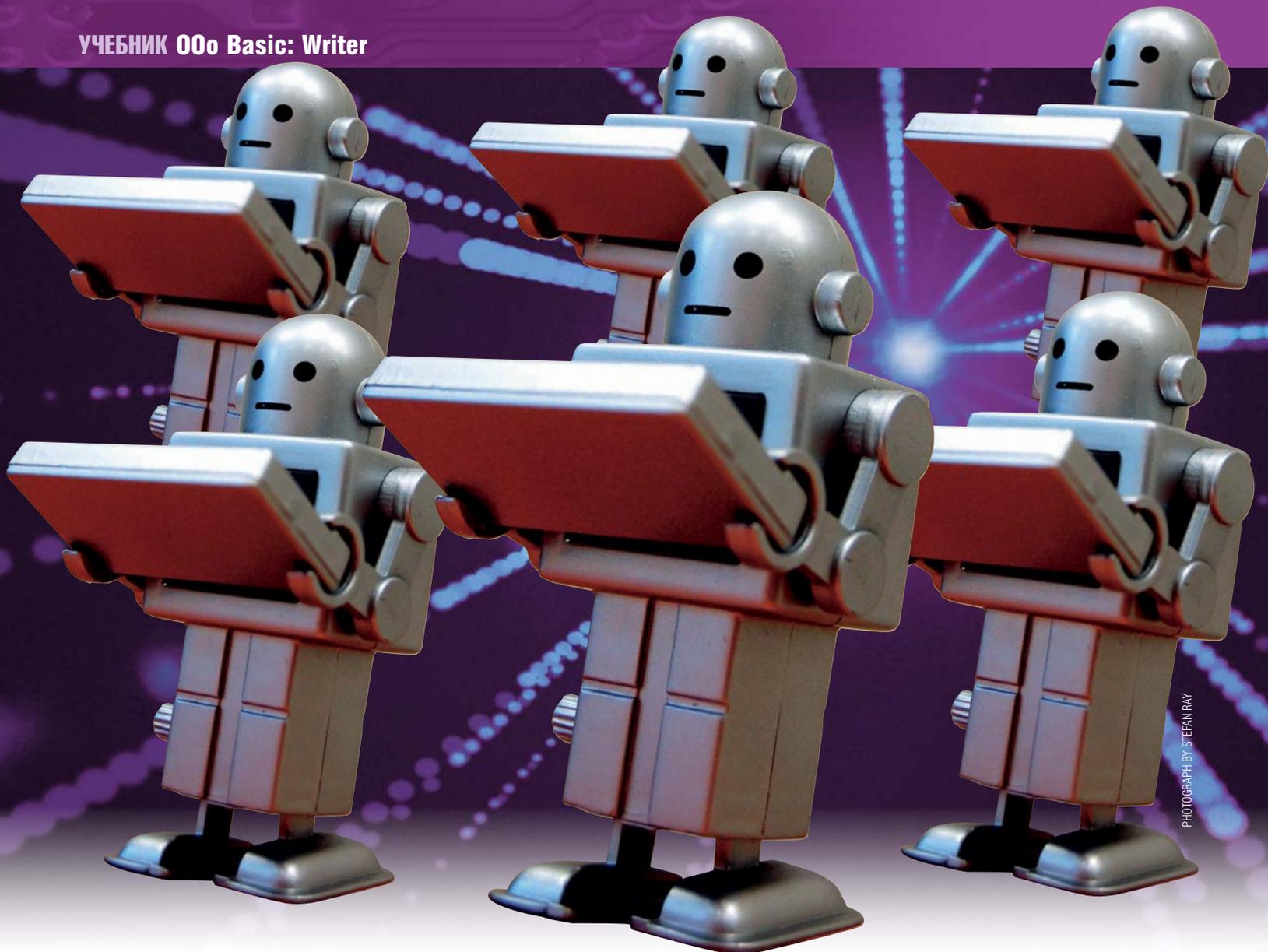
На уроках по режиссерской работе всегда говорится о том, что публика скорее стерпит неприятную картинку, чем отвратительный звук. Поэтому очень важно, чтобы фильм о дне рождения вашего малыша не был омрачен неистовым ребенком, который своими пронзительными криками дополняет завывания *Crazy Frog* на заднем плане. Желаю удачного видеомонтажа! **LXF**



Телепузик готовится к своему бенефису. На экране пойман эффект перехода от общего к крупному плану.

**ЧЕРЕЗ МЕСЯЦ**

Мы узнаем секреты безопасного интернет-серфинга: использование кеша, cookies и другое.



PHOTOGRAPH BY STEFAN RAY

АВТОМАТИЗАЦИЯ OPENOFFICE.ORG СЕРИЯ: «СЦЕНАРИИ НА BASIC»

# OOo Basic Макросы в Writer'e

**ЧАСТЬ 1** Марк Бэйн смеется над ничтожествами, которые все еще не умеют пользоваться скриптами в текстовом процессоре. А вы сейчас научитесь!



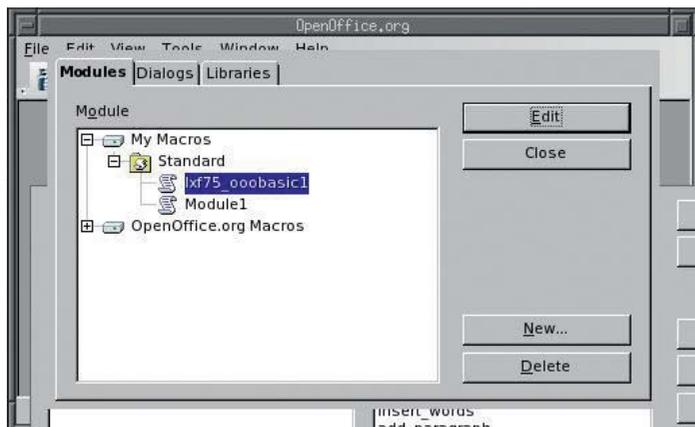
**НА ДИСКЕ**

• Исходные тексты примеров.



Недавно произошло важное событие – заключен очередной контракт в пользу *OpenOffice.org*. Это должно принести нам много радости: офисный пакет *OOo* появился на свет в

период зрелости движения Open Source, то есть не так давно, поэтому чем больше людей им пользуются, тем ближе знакомятся с открытыми программами. Однако многие пользователи пакета не знают, что в *OpenOffice.org* встроен мощный язык программирования – и в этой новой серии статей мы, счастливики, научимся им пользоваться. Язык называется *OpenOffice.org Basic*, и в первой части мы используем его в текстовом процессоре *OOo Writer*.

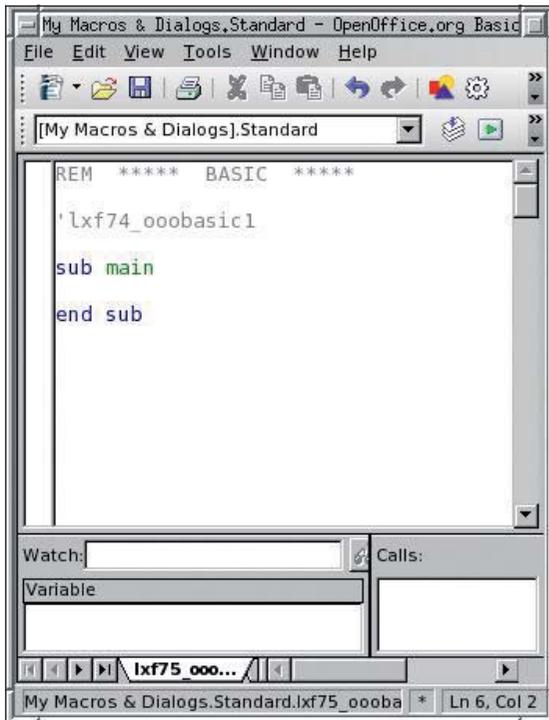


Доступ к макросам, диалоговым окнам и библиотекам через Macro Organizer.

## Basic: основы основ

Вам, наверное, интересно, с чего бы это пользователю *OOo Writer* интересоваться программированием. Ответ прост: автоматизация. Представьте, что вам нужно готовить ежедневные отчеты, куда нужно включать информацию об использовании дискового пространства или список активных пользователей. Это – несложная работа: можно использовать команды *who* и *df*, а потом вставить результаты в *Writer*. Тем не менее это скучно и требует времени – пять минут здесь, пять минут там. Не правда ли, было бы здорово, если бы *Writer* делал всю работу сам, а вы бы шли в пивную немного раньше?

В этом учебнике вы узнаете, что в *OpenOffice.org Basic* можно писать макросы для выполнения любых задач, от открытия документов *Writer* и



### В OpenOffice.org есть простой редактор Basic.

вставки внешних объектов до вывода диалоговых окон, работы с динамическими данными и многого другого. Для этой статьи я использовал *OpenOffice.org* 2.0 (1.9.79) под Linux и версию 1.1.4 под Windows (просто — я лишь хотел проверить, как все заработает).

*OpenOffice.org* Basic очень похож на любую другую разновидность этого языка. Впервые я стал использовать Basic на Sinclair ZX81 в начале восьмидесятых. Сегодня существует множество версий Basic — Visual Basic, Gambas и другие. Каждая имеет все ту же командную структуру, и поэтому различаются они, как правило, лишь нюансами. Тем не менее, имейте в виду, что с этого момента под словом Basic я понимаю именно *OpenOffice.org* Basic. Не рассчитывайте, что приведенный в качестве примера код так сразу заработает в другой версии Basic.

Наверно, вы хотите, чтобы я подробно рассказал о функциях и подпрограммах, переменных и объектах и лишь затем приступил к делу? Нет? Вы хотите по-простому во всем разобраться и получить результат? Хорошо, вот простой пример кода, который создает пустой документ *Writer*. Используйте **Macro Organizer** для того, чтобы создать новый модуль (см. врезку вверху справа на этой странице), затем впишите туда следующий код:

```
Sub Main
loadNewFile
End Sub
Sub LoadNewFile
dim doc as object
dim desk as object
dim url as string
dim args()
desk = CreateUnoService("com.sun.star.frame.Desktop")
url = "private:factory/sWriter"
doc = desk.loadComponentFromUrl(url, "_blank", 0, args())
End Sub
```

Теперь вы можете нажать кнопку Run BASIC (Выполнить код BASIC) на панели инструментов и увидеть обещанный новый документ.

Приведенный в примере код позволяет понять, каким образом можно загрузить любой файл. Важным параметром является URL, например:

```
url = "file:///home/bainm/test.odt"
```

Более того, вы можете создать более сложный сценарий, когда скрипт пытается открыть файл и в случае неудачи создает новый документ.

```
Sub LoadNewFile (optional myFile as string)
dim doc as object
dim desk as object
dim url as string
dim Dummy()
if isMissing(myFile) then
myFile = "private:factory/sWriter"
end if
desk = CreateUnoService("com.sun.star.frame.Desktop")
url = myFile
doc = desk.loadComponentFromUrl(url, "_blank", 0, Dummy())
End Sub
```

Если вы знакомы с Basic, то узнаете общую структуру — мы создали две подпрограммы. Первая (**Main**) нужен для управления действиями макроса. Вторая (**LoadNewFile**) делает всю работу. Она определяет новые переменные (**doc**, **desk**, **url** и **args**), затем создает UNO (Universal Network Object — универсальный сетевой объект), который дает нам доступ к свойствам объектов *Writer*.

### Изменяем документ

От пустого файла мало проку, к тому же его можно легко создать вручную, нажав **Ctrl+N**. Давайте теперь напишем код для изменения документа:

```
Sub Insert_words
dim doc as object
dim cursor as object
doc=thisComponent
cursor=doc.text.createTextCursor
cursor.string="Hello World"
End Sub
```

Вам также понадобится изменить подпрограмму **Main**:

```
Sub Main
loadNewFile
insert_words
End Sub
```

Видите, как просто управлять *Writer* при помощи кода! Чтобы извлечь из этого пользу, напишем фрагмент, который будет воспринимать введенный текст как абзац и помещать его в файл.

```
Sub Add_paragraph (myText as String)
dim doc as object
dim cursor as object
doc=thisComponent
cursor=doc.text.createTextCursor
cursor.gotoEnd(False)
doc.text.insertControlCharacter(cursor, _
com.sun.star.text.ControlCharacter.PARAGRAPH_BREAK,
False)
doc.text.insertControlCharacter(cursor, _
com.sun.star.text.ControlCharacter.PARAGRAPH_BREAK,
False)
cursor.string = myText
End Sub
```

В данном случае скрипт перемещает курсор в конец документа, создает новый абзац и помещает в него введенный текст. Например:

```
Sub Main
loadNewFile
add_paragraph("This is my first paragraph.")
add_paragraph("This is my second paragraph.")
End Sub
```

Знаю, знаю, и в этом случае будет быстрее набрать текст в *Writer*, но вы уже, наверное, придумали, как можно использовать этот код — он пригодится, если добавить в него подпрограмму загрузки внешних данных.

Если вы уже пользовались какой-либо разновидностью Basic, следующий код покажется вам знакомым:

```
Sub Load_report_file(myFile as String)
```

## РЕДАКТОР МАКРОСОВ

В OOo предусмотрен собственный инструмент для доступа и редактирования макросов, диалоговых окон и даже библиотек. При необходимости вы можете создавать, изменять и удалять новые объекты. Если вы достаточно храбры (звучит несколько самоуверенно), то можете редактировать уже готовые встроенные макросы OOo. Способ запуска редактора макросов, зависит от вашей версии *OpenOffice.org*. В версии 1.1.4 следует найти пункт **Tools > Macros > Macro... > Organizer**. В версии 2.0 путь будет немного другим: **Tools > Macros > Organize Macros > OpenOffice.org Basic > Organizer**.



```

<< dim filenumber As Integer
dim iLine As String
dim pText As String
filenumber = Freefile
open myFile For Input As filenumber
while not EOF(filenumber)
Line Input #filenumber, iLine
if (iLine <> "") then
pText = pText & iLine
else
add_paragraph(pText)
pText=""
end if
wend
if (PText<> "") then
add_paragraph(pText)
end if
close #filenumber
End Sub
    
```

Этот код работает с определенным файлом, сканируя его на предмет полных абзацев, которые определяется по наличию пустой строки. Когда скрипт находит такой абзац, он вставляет его в новый документ, если не находит – продолжает поиск до конца файла.

Некоторые моменты в этом коде все же нужно прояснить. Начнем с использования значения **Freefile**. Оператор **open** подразумевает, что каждому открытому файлу вы можете присвоить уникальный порядковый номер. Это может быть любое число на ваше усмотрение, но вы должны обязательно его запомнить, чтобы оно потом не повторилось (это станет важным, когда у вас будет открыто более одного файла). Вместо этого можно использовать **Freefile**, и тогда файлу будет просто дан очередной порядковый номер.

Вы также можете спросить, зачем мы второй раз используем выражение **add\_paragraph** за пределами блока **while...wend**. Это нужно для того, чтобы скрипт не пропускал в файлах последние параграфы, после которых нет пустой строки.

Хорошо, теперь можно использовать этот код для загрузки текста из любых файлов. Вот пример:

```

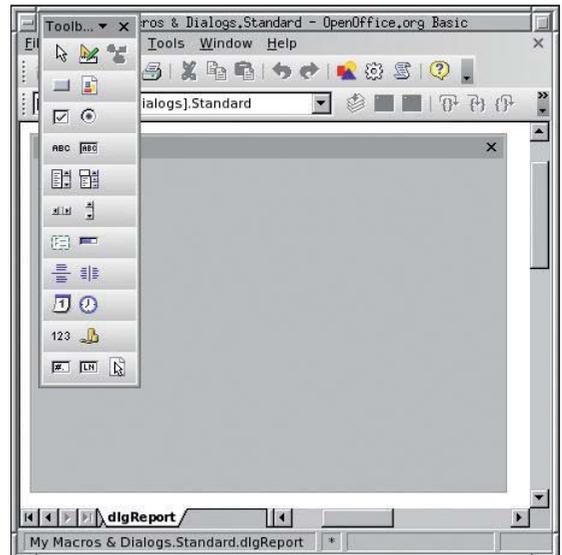
Sub Load_report_simple
dim rep_dir as String
rep_dir = "-/articles/lxf75_OOobasic1/demo/"
load_report_file(rep_dir & "manager_header.txt")
load_report_file(rep_dir & "body.txt")
End Sub
    
```

Даже сейчас метод ручного ввода текста в *Writer* имеет преимущество, и вы можете самостоятельно сохранить или распечатать документ. И все же наш код станет гораздо полезнее, если в нем появится возможность выбора:

```

Sub Load_report (optional reportType as integer)
const rep_dir as string = "-/articles/lxf75_OOobasic1/demo/"
if isMissing(reportType) then
reportType = 1
end if
select case reportType
case 1
load_report_file(rep_dir & "manager_header.txt")
case 2
load_report_file(rep_dir & "contractor_header.txt")
end select
load_report_file(rep_dir & "body.txt")
End Sub
    
```

Теперь загружаться будет один из двух файлов (**manager\_header.txt** или **contractor\_header.txt**), в зависимости от значения переменной **reportType**. Тем не менее в обоих случаях в текст будет вставлен файл **body.txt**. Вы можете изменить это, отредактировав подпрограмму **Main** так:



**Organizer** можно использовать не только для написания кода, но и для создания новых диалоговых окон с набором кнопок.

```

Sub Main
loadNewFile
load_report(1)
End Sub
    
```

или так:

```

Sub Main
loadNewFile
load_report(2)
End Sub
    
```

Я уверен, что на этом этапе вы заметите несовершенство данного метода: каждый раз, когда вам нужен разный результат, вам необходимо править код. Нам нужен более элегантный способ выбирать нужный файл. Сейчас мы создадим диалоговое окно, контролирующее вывод скрипта.

### Создаем диалоговое окно

Для этой цели нам понадобится редактор макросов. До того, как щелкнуть кнопку **New**, перейдите на вкладку **Dialogs** (Диалоговые элементы). В случае, если вы задали новому диалогу имя **dlgReport**, код будет выглядеть следующим образом:

```

dim dlgReport as object
Sub DlgReport_show
basicLibraries.loadLibrary("Tools")
dlgReport = loadDialog("Standard","dlgReport")
dlgReport.execute()
End Sub
    
```

Не забудьте также изменить и подпрограмму **Main**:

```

Sub Main
dlgReport_show
End Sub
    
```

Диалоговое окно пока еще ничего не делает (нажмите **Esc**, чтобы закрыть его), но мы можем добавить на него элементы управления – кнопки и списки.

Наш диалог даст нам возможность контролировать тип отчета, который будет открыт в *Writer*. Нам понадобится поле со списком и кнопка. Создайте их, выбрав соответствующие значки на панели инструментов и нарисовав нужные объекты.

Через редактор свойств задайте объектам имена **lstReport** и **btnReport** соответственно, а также напишите какой-нибудь текст на кнопке (предлагаю слова «Создать отчет»).

Теперь мы заполним список нужными данными при помощи встроенного метода **OOo** – **additem**. Вам может прийти в голову следующая конструкция:

```

Sub DlgReport_show
    
```

### СВЯЗЬ КОДА С ОБЪЕКТАМИ

Возможно, вы уже имели дело с языками, где код автоматически привязывается к определенным объектам. Это мог быть Delphi, Kylix, Gambas, к примеру (или даже Visual Basic). В *OpenOffice.org* вы должны сначала написать код, а потом вручную ассоциировать его с нужным объектом (например, кнопкой). Это можно сделать на вкладке **Events (События)** в окне настроек объекта (оно вызывается через правый щелчок по объекту, при выборе пункта **Properties**)

**СОВЕТЫ**

- Язык OpenOffice.org Basic нечувствителен к регистру символов. Поэтому переменная **myVariable** – это то же самое, что и **myvariable** и даже **MYVARIABLE**. Не имеет значения, что вы выберете, просто старайтесь придерживаться какого-либо одного написания.
- Для того, чтобы закрыть диалоговое окно, достаточно нажать клавишу **Esc**.
- После того, как вы создали кнопку и написали для нее код, не забудьте зайти в свойства кнопки и ассоциировать этот код с ней.

```
basicLibraries.loadLibrary("Tools")
dlgReport = loadDialog("Standard", "dlgReport")
dlgReport.lstReport.AddItem("Managers", 0)
dlgReport.lstReport.AddItem("Contractors", 1)
dlgReport.execute()
End Sub
```

Выглядит логично, но *OpenOffice.org* Basic работает немного по-другому. Вместо приведенного выше, вам понадобится следующий код:

```
dim lstReport as object
sub DlgReport_show
basicLibraries.loadLibrary("Tools")
dlgReport = loadDialog("Standard", "dlgReport")
lstReport = dlgReport.getControl("lstReport")
lstReport.AddItem("Managers", 0)
lstReport.AddItem("Contractors", 1)
dlgReport.execute()
End Sub
```

Заметьте, что список (list box) необходимо определить как уникальный объект. Воспользоваться им можно только при помощи метода **getControl** диалогового окна. Также, мы объявили переменную **lstReport** как глобальную – это значит, что ее можно использовать в любой подпрограмме (вы это еще заметите).

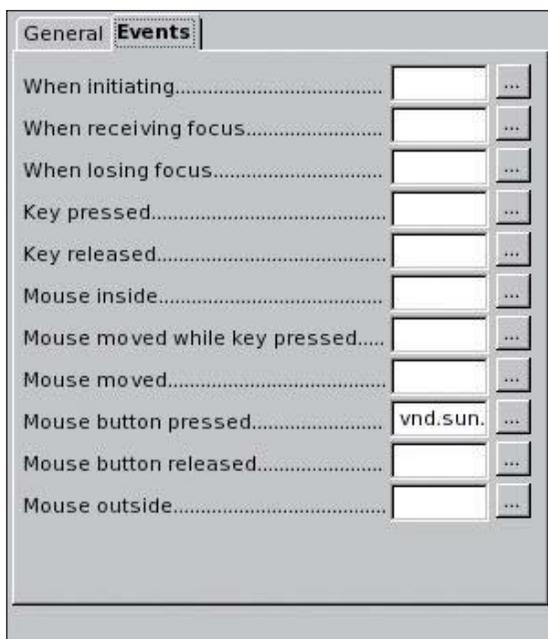
**Заряжаем кнопку**

Теперь мы напишем код, который будет выполняться при нажатии на кнопку:

```
Sub BtnReport_Click
loadNewFile
load_report(lstReport.selectedItemPos)
End Sub
```

Я уверен, что теперь вы запустили основной макрос и обнаружили, что поле со списком работает, но при нажатии на кнопку ничего не происходит. Скорее всего, с ней пока не ассоциирован никакой код. Перейдите в окно свойств кнопки и на вкладке **Events (События)** выберите подпрограмму, которая должна выполняться при нажатии (для закрытия окна используйте **Esc**).

Теперь у вас есть полнофункциональный инструмент для контроля над создаваемыми документами.



Воспользуйтесь окном настроек (Properties) кнопки, чтобы назначить ей выполняемый код.

**Идем дальше**

Никто не будет спорить с тем, что сделанную работу нужно сохранить. Во всех приведенных примерах мы создавали документы автоматически, но сохраняли их вручную. Поскольку эта статья посвящена автоматизации, мы обратимся к коду, который позволит сохранить созданные файлы. Попробуйте это:

```
Sub SaveMyFile (fileUrl as string)
dim params()
doc.saveAsUrl("file:" & fileUrl, params())
doc.close(true)
End Sub
```

Возможно, нам понадобится всего лишь распечатать документ, не сохраняя его. Для этого добавим новую подпрограмму:

```
Sub PrintMyFile
dim params()
doc.print(params())
doc.close(true)
End Sub
```

Хорошо, мы разобрались с тем, как работать с документами *Writer* и как извлекать данные из внешних файлов. Все это имело отношение к статической информации, но, что любопытно, мы можем работать и с динамическими данными. Воспользуемся встроенным в *OpenOffice.org* методом *SystemShellExecute*:

```
Sub RunCommand (command as string)
dim svc as object
svc = createUnoService("com.sun.star.system.SystemShellExecute")
svc.execute(command, "", 0)
End Sub
Sub BtnReport_Click
const tmpfile as string = "/tmp/myfile.tmp"
loadNewFile
load_report(lstReport.getSelecteditemPos())
runCommand("df > " & tmpfile)
load_report_file(tmpfile)
End Sub
```

В этом фрагменте скрипт передает оболочке Linux команду *df* и сохраняет ее вывод в файл (в данном случае, в **/tmp/myfile.tmp**). Затем содержимое этого файла загружается в новый документ *Writer*, где оно выглядит примерно так:

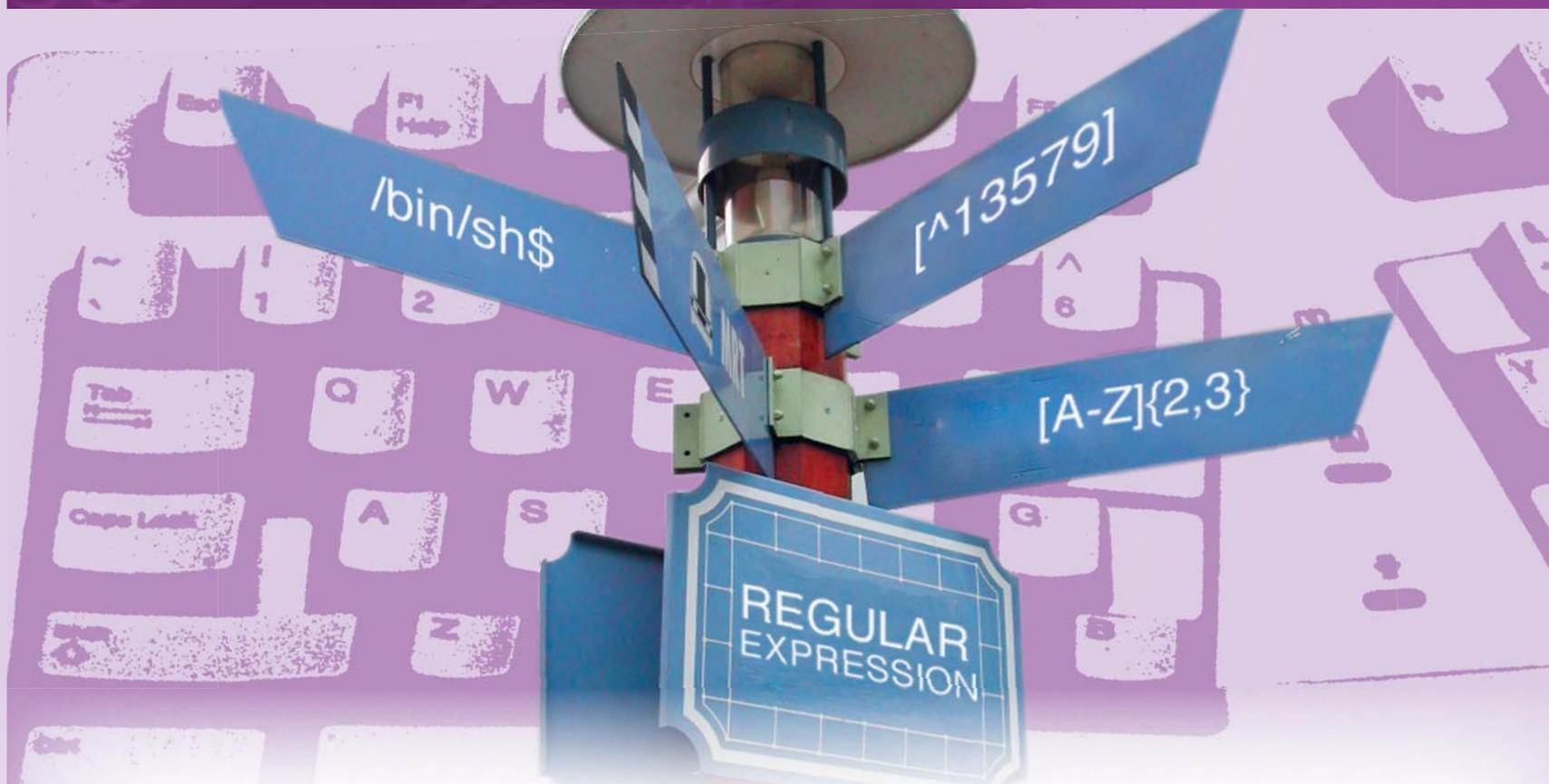
```
'Filesystem 1K-blocks Used Available Use% Mounted on
/dev/hda3 3470204 3089264 201816 94% /
/dev/hda4 1510060 1064572 368780 75% /opt
/dev/hda1 4593600 3732708 860892 82% /
WINDOWS.'
```

Эти данные очень полезны, но выглядят они не слишком привлекательно – было бы лучше увидеть все в виде таблицы. В директории **Magazine** на нашем диске вы найдете полный вариант кода для загрузки данных из файла в таблицу *Writer* – эта функция сэкономит вам массу времени, которое вы сможете потратить на прием горячительных напитков.

Этот код слишком велик, чтобы печатать его прямо здесь, но изучив его копию на диске, вы узнаете о новых полезных функциях Basic: например, *Chr* (возвращает ASCII-код целого числа), *Array* (создает массив из строковых переменных) и *ubound* (выдает число элементов в массиве). А теперь домашнее задание: изучите подпрограмму *btnReport\_Click* и выясните, из чего состоит командная переменная и как она передается оболочке. До новых встреч! **LXF**

**ЧЕРЕЗ МЕСЯЦ**

Мы углубимся в OpenOffice.org Basic и займемся электронными таблицами.



**HARDCORE LINUX** УЧЕБНИК ДЛЯ ПРОДВИНУТЫХ ПОЛЬЗОВАТЕЛЕЙ

# Обработка текста: регулярные выражения

Освоив регулярные выражения, принимайтесь за изучение геномов с д-ром **Крисом Брауном** (Chris Brown). А может, и без него.

**МЕСЯЦ  
НАЗАД**



Как важно знать основы! Мы поэтому прошлись по Bash и изучили управление заданиями, просмотр каталогов и переменные среды.



Доброе утро. Милости просим на мастер-класс по регулярным выражениям. От вас потребуется внимание, так что не болтайте там сзади, не валяйте дурака и повремените с чтением почты. Прежде чем начать, я хочу убедиться, что все могут найти обратную косую черту на клавиатуре. Все? И фигурные скобки тоже? Хорошо. Итак, приступим.

Регулярные выражения (также известные как регексы — от англ. regular expressions) — это способ написания определенных текстовых шаблонов. Они существуют в Linux (а до этого в Unix) уже давно. Самое раннее упоминание о них мне удалось найти в map-странице древнего редактора Ed, входившей в справочник по шестой редакции Unix. Этот документ датирован 15 января 1973 года, так что сами регулярные выражения и того древнее. Регексы используются во многих Linux-утилитах: они служат для поиска и замены в редакторах, как типа Vi, так и посложнее, OpenOffice.org; их можно найти в фильтрах вроде grep, sed и awk, они применяются в Apache для перезаписи URL и обеспечивают мощь языков Perl и PHP. И есть множество библиотек для использования регулярных выражений в разных языках программирования, вплоть до ASP.NET.

Регулярные выражения приспособляются под самые разные вещи. С ними вы сможете:

- Удалять комментарии из файлов конфигурации.
- Находить пустые параграфы в документах OpenOffice.org.
- Проверять, что указанная строка есть корректный IP-адрес.
- Извлекать адреса электронной почты из текстовых файлов.
- Выделять год из строки с датой.
- Искать палиндромы в списке слов.
- Искать специфические участки ДНК в геноме.

Насчет последнего примера: существует целая индустрия, применяющая Perl и регулярные выражения для обработки биоинформации. (К сожалению, с генетикой я вам не помогу — я доктор в области теоретической физики. Но у O'Reilly есть три книги по этому вопросу!)

Для начала следует осознать, что регулярные выражения отнюдь не то же самое, что и безразличные символы (wildcards), распознаваемые командными оболочками. Как пример последних, приведу такую строку:

```
rm tutorial*.v[0-2]
```

Оболочка будет искать файлы, соответствующие данному шаблону — их имена начинаются на tutorial и заканчиваются на v0, v1 или v2.

Регулярные выражения и универсальные (безразличные) знаки записываются в том числе и общими метасимволами (например, \* и []). Но смысл их различен, как и контекст их применения. Универсальные замещающие знаки обычно используются для поиска имен файлов, а регулярные выражения — для выбора текста в файле или строке, обрабатываемой программой.

Впрочем, перейдем к мастер-классу. Буду считать, что вы хорошо знакомы хотя бы с утилитой grep. По умолчанию она выводит строки, соответствующие некому выражению. Например, команда

```
grep xen ~/book/chapter5.txt
```

выведет все строки файла **chapter5.txt**, где обнаружится слово **xen**. Буквы здесь не имеют особого значения — это просто набор литер **x**, **e** и **n**. Регулярное выражение правильное, но очень простенькое: искать фиксированные строки таким способом нет смысла. Допустим теперь, что вы хотите проверить несколько файлов, чтобы узнать, в каких содержится слово **xen**. Достаточно будет следующей командой:

```
grep -l xen ~/book/chapter[0-9].txt
```

Опция **-l** говорит «не надо мне строк – покажите имена файлов, содержащих данную строку». Учтите, что строка **~/book/chapter[0-9].txt** здесь – не регулярное выражение *grep*, а шаблон файлов, обрабатываемых оболочкой.

### Возьмемся за метасимволы

Пора познакомиться с настоящими регулярными выражениями – вы должны были уже проснуться. Приведенная ниже таблица содержит элементы регулярных выражений, которыми мы займемся. Рассмотрим выражение:

```
grep -l '[Xx]en' ~/book/chapter[0-9].txt
```

Заметьте, я поместил регулярное выражение в одинарные кавычки для того, чтобы оболочка не интерпретировала **[Xx]** как замещение имени файла. Мы ведь хотим передать аргумент **[Xx]en** в *grep*, чтобы именно *grep* обработал **].**

Употребление метасимволов, интерпретируемых как надо – почти искусство, и без него в мире синтаксиса программных оболочек и регулярных выражений не прожить. Можно составить выражение, соответствующее любой заглавной гласной букве (**[AUOIE]**) или цифре (**[0123456789]** – охватить все цифры можно и короче – **[0-9]**; аналогично описываются все малые латинские буквы – **[a-z]**).

Имейте в виду: что бы ни было помещено внутрь квадратных скобок, соответствует оно только одному символу в тексте. Добавление **^** в начале списка символов инвертирует смысл выражения, то есть регексу **[^0-9]** соответствует любой символ, кроме цифры.

Символы **^** и **\$** привязывают искомое значение к началу и концу строки соответственно. Например, выражению **^login** соответствует любая строка, начинающаяся со слова **login**, а **[0-9]\$** – любая строка, оканчивающаяся цифрой.

Поясно на примере. Есть множество файлов конфигурации, где строки комментариев начинаются с символа **#**. Иногда в гуще комментариев трудно отыскать нужные параметры. Регулярное выражение отберет для вас незакомментированные строки командой:

```
grep -v `^#`
```

Опция **-v** велит *grep* печатать только те строки, которые НЕ соответствуют указанному регулярному выражению. То есть мы сказали *grep*: «не печатай строки, начинающиеся с **#**». Имеется и альтернативное решение:

```
grep `^[^#]` /etc/ssh/ssh_config
```

Оно говорит «печатай строки, не начинающиеся с **#**». Попробовав воспроизвести оба примера, вы увидите, что они не эквивалентны. Отличие – в отношении к пустым строкам. Ни одно из приведенных регулярных выражений не соответствует пустой строке, однако в первом варианте пустые строки отображаются, а во втором – нет. Обратите

## «ИСПОЛЬЗОВАНИЕ МЕТАСИМВОЛОВ – ВАЖНЕЙШЕЕ ИСКУССТВО ПРИ SHELL-ПРОГРАММИРОВАНИИ.»

внимание, что во втором варианте символы **^** обозначают совершенно разные вещи.

Кстати о пустых строках. Им соответствует регекс **^\$**. Так, команда

```
grep -v `^$` somefile
```

выведет все непустые строки файла.

### Расширьте свои знания

Одна из вещей, отличающих любителя от профессионала, заключается в умении отличить простое регулярное выражение от расширенного. Когда-то *grep* распознавал лишь простые регексы, а для расширенных имелась отдельная утилита, *egrep*. Зато GNU-версия *grep* распознает оба типа; по умолчанию она ищет только простые регексы, но если указать ключ **-E**, то *grep* заработает и с составными выражениями. Например, обычные круглые скобки **( и )** не имеют какого-то особого смысла в простых регексах – они обозначают просто самих себя. В



## 13 САМЫХ ИСПОЛЪЗУЕМЫХ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Это список наиболее привычных выражений, которые мы использовали в наших уроках

Выражение	Описание	Пример	Что означает пример
a	Обычные символы обозначают сами себя	apple	Строка «apple»
[...]	Любой символ, заключенный в []	[02468]	Любая из цифр 0,2,6,8
[^...]	Любой символ, не заключенный в []	[^13579]	Любой символ кроме нечетных цифр
[x-x]	Диапазон символов	[A-Z]	Любая большая латинская буква
.	Любой одиночный символ	c.t	cut, cat, c9t и т.д.
^	Начало строки	^[0-9]	Строки, начинающаяся с цифры
\$	Конец строки	/bin/sh\$	Строки, кончающиеся /bin/sh
*	Ноль или больше предшествующих символов	[a-z]*	Любая последовательность символов в нижнем регистре или ничего
?	Ноль или один предшествующий символ	https?://	http:// и https://
+	Один или больше предшествующих символов	T+	T, TT, TTT, TTTT и так далее
{n}	n повторов предыдущих символов	[0-9]{3}	Последовательность из трех цифр, например 124, 111, 743
{n,}	n или больше предыдущих символов	0{3,}	000, 0000, 00000 и т.д.
{n,m}	От n до m предыдущих символов	[A-Z]{2,3}	Строки вроде AB, ABC, YY, ZZZ

расширенных же регексах круглые скобки используются для группировки частей регулярного выражения, как в арифметике – (a+b)\*c. Чтобы расширенный регекс понимал символы круглых скобок как таковые, перед ними ставится обратный слэш – \ (будет соответствовать открывающей скобке. Если вы еще не совсем запутались, добавлю, что некоторые программы (включая GNU-версию *grep*) по умолчанию используют простые регексы, где обратный слэш как раз включает особое значение символа. Так что приходится постоянно держать в голове: включает обратный слэш специальное значение какого-либо символа или отключает его.

Далее в таблице идут модификаторы-повторители – внимание! Сам по себе такой модификатор ничему не соответствует, он лишь указывает, сколько раз предшествующее регулярное выражение должно повторяться. Например, **[0-9]\*** соответствует нулю или более цифр, а **[A-Z]+** – одной или более заглавной латинской буквы. Очень часто встречающийся пример: **.\*** обозначает «любое количество чего угодно». Так, выражение **login.\*failed** соответствует любой строке, содержащей слово **login**, за которым через любое количество вторгающихся символов следует слово **failed** [в предыдущей фразе содержится распространенная неточность. Вместо «...слово failed» следует читать «последовательность символов failed», то есть строки «login failed», «login abcfailed» и даже «login\_not\_failed» одинаково удовлетворяют шаблону, – прим.ред.].

### Регексы шаг за шагом

Для упражнения в использовании повторяющих модификаторов создадим регулярное выражение, распознающее IP-адреса, типа 192.168.0.42. Правильный IP-адрес описывается как «четыре десятичных числа от 0 до 255, разделенные точками». К сожалению, мы не можем напрямую перевести это описание на язык регулярных выражений, так как в них невозможны арифметические сравнения. В первом приближении скажем так: «IP-адрес – это четыре десятичных числа длиной от одной до трех цифр, разделенных точками».

Это действительно соответствует реальным IP-адресам, а заодно и несуществующим, вроде 123.456.789.0. Что ж, давайте построим регекс шаг за шагом. Во-первых, **[0-9]{1,3}** соответствует числу из одной-трех цифр. Для добавления литерала . (точка) нужно предварить его косой чертой, чтобы отключить его специальное значение: **[0-9]{1,3}\.**

Далее, нужно сказать: «Повторить всю группу три раза». Приставим еще один повторитель: **[0-9]{1,3}\.[3]**. Однако это выражение не будет работать так, как нам надо, поскольку повторитель применяется только к последнему символу (здесь это .). Значит, надо окружить повторяющееся выражение круглыми скобками: **([0-9]{1,3}\.)\{3}**. Помните, что используя *grep* без ключа **-E** (то есть в режиме простых регексов), вы должны включать специальное значение скобок, предварив каждую из них обратным слэшем: **\([0-9]\{1,3}\.\)\{3\}**. Наконец, завершим выражение, приставив описание последней группы цифр IP-адреса, следующим образом: **([0-9]{1,3}\.[3])[0-9]{1,3}**. Головка не болит?

### Извлечение адресов e-mail

Мы созрели для примера посложнее. Вот задача: у меня есть множество текстовых файлов, содержащих адреса электронной почты. Сам адрес может находиться внутри другого текста, например так: «Спроси Андрея (**andy@example.net**), знает ли он...»

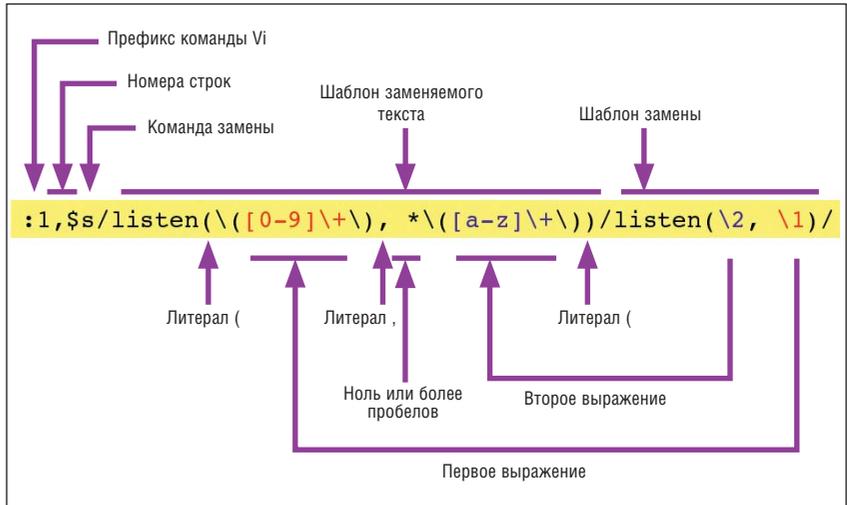
Я хочу извлечь все уникальные адреса из этих файлов. Разобьем задачу на четыре шага:

- **Шаг 1:** Опишем обычным человеческим языком, как выглядит адрес электронной почты.
- **Шаг 2:** Создадим регекс, соответствующий описанию адреса.
- **Шаг 3:** Воспользуемся *grep* с этим регексом для получения списка адресов.
- **Шаг 4:** Напишем небольшую обертку для представления законченного решения.

Сложнее всего первый шаг. Вот простые попытки формализовать адрес электронной почты:

**БЕЗУПРЕЧНАЯ ЛОГИКА РЕГЕКСОВ**

Не очевидно, почему **[^#]** не соответствует пустой строке. Но ведь пустые строки не начинаются с #, верно? Фактически, после начала строки должен быть хотя бы один символ, чтобы сравнить его с #.



Вскрытие регулярного выражения. Крови нет, но синтаксиса много.

- 1 Строка из одного или более заглавного или строчного литерала (имя пользователя)
- 2 Литерал @
- 3 Один или более доменов. Каждый домен – это строка литералов, оканчивающаяся точкой
- 4 Домен первого уровня, содержащий две или три буквы

Теперь перейдем ко второму шагу – поочередному переводу этого описания на язык регулярных выражений. Первая часть адреса переводится как **[a-zA-Z]+**. Вторая – как **@** (это не метасимвол, он просто обозначает сам себя).

Над третьей частью придется призадуматься. Регекс, соответствующий одному домену, может быть таким: **[a-zA-Z]+\.** – сюда попадают все слова с точкой, например, *example*. (здесь \ используется для описания точки). Однако доменов бывает и несколько, поэтому нужно добавить сюда **+**, после чего получится **([a-zA-Z]+\.)+**. Этому выражению соответствует и *example*, и *foo.example*, и *foo.bar.example*.

Четвертая часть требует повторителей (см. таблицу) и может быть записана как **[a-zA-Z]{2,3}**.

Соединив все части, получим **[a-zA-Z]+\@([a-zA-Z]+\.)+[a-zA-Z]{2,3}**.

Конечно, вариант у нас упрощенный, ведь имена и домены могут содержать цифры и другие символы. Кроме того, наше выражение включает адреса с несуществующими доменами вроде *хуз:andy@example.xyz*. [Более того, строго поставленная задача поиска адресов электронной почты в принципе не разрешима с помощью регулярных выражений. Однако в практических целях этим можно пренебречь, – прим.ред.]

Обычный подход – начать с более или менее работающего регекса, а затем немного его подправить. Как правило, первые 10% усилий затрачиваются на выражение, распознающее 90% случаев. Остальные усилия тратятся на оставшиеся 10%.

Согласно шагу 3 мы должны подsunуть наше выражение утилите *grep* и обработать все текстовые файлы (оканчивающиеся на *.txt*):

```
grep '[a-zA-Z]+\@([a-zA-Z]+\.)+[a-zA-Z]{2,3}' *.txt
```

Однако это еще далеко не все. Во-первых, необходимо добавить опцию **-E**, поскольку мы писали регекс в расширенном формате. Во-вторых, *grep* обычно выводит строки, содержащие участки, удовлетворяющие шаблону; чтобы показывался только адрес (то есть часть, соответствующая регексу), нужно передать ему параметр **-o**. В-третьих, при обработке множества файлов *grep* будет печатать имя файла перед каждым совпадением. Для отключения вывода имени файла есть опция **-h**.

Вот что получится в итоге:

```
grep -E -o -h '[a-zA-Z]+\@([a-zA-Z]+\.)+[a-zA-Z]{2,3}' *.txt
```

Эта команда выдаст список вроде следующего:

```
santa@northpole.com
isaac@trinity.cam.ac.uk
jrrtolkien@exeter.ox.ac.uk
bill@millssoft.uk.com
isaac@trinity.cam.ac.uk
bill@MILLSOFT.UK.COM
```

Я умышленно составил такой список, чтобы показать ряд возможных проблем. Во-первых, домены регистронезависимы, и иногда их пишут заглавными буквами. Нельзя считать два адреса разными только потому, что они отличаются регистром. Во-вторых, адрес может встретиться в списке не один раз.

## ЖАДНЫЕ РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

Да, регексы очень жадные – они употребляют строку до последнего символа. Точнее, они прикладывают шаблон к самому левому вхождению и прихватывают по максимуму. Рассмотрим действие регекса `t+` на строку `'aatttaatttttaa'`. `t+` соответствует `'t'`, `'tt'`, `'ttt'` и так далее. В соответствии с правилами, сравнение начнется с первого `t` в строке и продолжится сколько сможет. Таким образом, результат – первая подстрока `'ttt'`, хотя дальше есть более длинная последовательность из `t`.

Если нас устраивает обычное поведение `grep`, то есть вывод строк, содержащих совпадения, то никаких проблем не возникает, нам важен только факт наличия. Однако если требуется заместить подстроку, соответствующую регексу, все усложняется.

Например, нужно добыть имена пользователей из файла `/etc/passwd`. Каждая строка этого файла отвечает за одну учетную запись; строки имеют следующий вид:

```
chris:x:1000:100:Chris Brown:/home/chris:/bin/bash
```

Наша задача – ликвидировать все, что после двоеточия. Полагаясь на алчность регекспов, можно сделать это командой

```
sed 's/:.*//' /etc/passwd
```

Здесь старый шаблон – `:.*`, а новый пуст. Таким образом, удалится все, что соответствует регексу, то есть от первого двоеточия до конца строки.

## Небольшая пост-обработка

Перейдем к последнему шагу 4. Регулярных выражений он не касается: просто приведем полученные результаты к удобочитаемому виду. Для начала переведем все адреса в нижний регистр. К счастью, это легко делается с помощью `tr`:

```
grep -E -o -h `[a-zA-Z]+@[([a-zA-Z]+\.)+[a-zA-Z]{2,3}]*.txt | tr A-Z a-z
```

Хотя легко предположить, что параметры `tr` – регулярные выражения, это не так. Мы просто приказываем `tr` преобразовать символы `A-Z` в `a-z`, то есть привести их к нижнему регистру.

Следующий шаг – сортировка. Направим вывод в `sort`:

```
grep -E -o -h `[a-zA-Z]+@[([a-zA-Z]+\.)+[a-zA-Z]{2,3}]*.txt | tr A-Z a-z | sort
```

Теперь наш список выглядит так:

```
bill@millisoft.uk.com
bill@millisoft.uk.com
isaac@trinity.cam.ac.uk
isaac@trinity.cam.ac.uk
jrrtorkien@exeter.ox.ac.uk
santa@northpole.com
```

Наконец, воспользуемся `uniq` для удаления повторяющихся строк. Вот окончательный ответ:

```
grep -E -o -h `[a-zA-Z]+@[([a-zA-Z]+\.)+[a-zA-Z]{2,3}]*.txt | tr A-Z a-z | sort | uniq
```

А вот и результат:

```
bill@millisoft.uk.com
isaac@trinity.cam.ac.uk
jrrtorkien@exeter.ox.ac.uk
santa@northpole.com
```

Теперь вы знаете, как пользоваться регулярными выражениями в `grep` и делать простую обработку результатов. Если вы все поняли – поздравляем! Вы заслужили погоны специалиста по регексам. **LXF**

## ЧЕРЕЗ МЕСЯЦ

Следующий учебник – по итальянской моде: про DansGuardian рассказывает Марко Фиоретти [Marco Fioretti].

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

# SUPERMICRO® РЕВОЛЮЦИЯ В СЕРВЕРОСТРОЕНИИ



Серверы TRINITY на базе платформ SUPERMICRO 2-Way Dual Core AMD Opteron (2-х процессорные двухядерные конфигурации)

Производительность двухядерных процессоров превышает одноядерные процессоры на 70 - 90 %. Заказывая 2-х процессорную двухядерную конфигурацию Вы получаете производительность 4-х процессорного сервера по цене 2-х процессорного.

В начале июля компания ТРИНИТИ представила серверные системы на базе двухядерных процессоров AMD Opteron серии 200. На сегодня доступны двухпроцессорные системы на базе платформ Supermicro:

**Trinity Revolution** На базе Supermicro® H8DA8 # 17181



Case: Supermicro CSE-743S1-650w/ 8xHS HDD  
CPU: 2 x AMD Opteron 265 Dual-Core  
RAM: 2GB DDR PC3200 ECC REG  
HDD: 2 x 73GB SCSI

Гарантия 3 года. Цена от: **\$ 4669**

**Trinity Revolution** На базе Supermicro® H8DAE # 17190



Case: Supermicro CSE-743S2-760w/ 8xHS HDD  
CPU: 2 x AMD Opteron 265 Dual-Core  
RAM: 2GB DDR PC3200 ECC RE  
RAID: LSI MegaRAID 320-1+BBU  
HDD: 3 x 73GB SCSI, RAID5

Гарантия 3 года. Цена от: **\$ 5289**

**Trinity Revolution** На базе Supermicro® H8DAE # 17191



Case: Supermicro CSE-743S2-760w/ 8xHS HDD  
CPU: 2 x AMD Opteron 275 Dual-Core  
RAM: 4GB DDR PC3200 ECC REG  
RAID: LSI MegaRAID 320-2x+BBU  
HDD: 6 x 73GB SCSI

Гарантия 3 года. Цена от: **\$ 8989**

**Trinity Revolution** На базе Supermicro® AS1020A-8 (H8DAR-8) # 17192



Case: Supermicro CS812S-420w/ 3xHS HDD  
CPU: 2 x AMD Opteron 275 Dual-Core  
RAM: 2GB DDR PC3200 ECC REG  
RAID: LSI MegaRAID 320-1+BBU  
HDD: 2 x 73GB SCSI

Гарантия 3 года. Цена от: **\$ 6619**

**Trinity Revolution** На базе Supermicro® AS1020A-T (H8DAR-T) # 17193



Case: Supermicro CS813T-500w/ 4xHS HDD  
CPU: 2 x AMD Opteron 265 Dual-Core  
RAM: 1GB DDR PC3200 ECC REG  
HDD: 4 x 200GB SATA

Гарантия 3 года. Цена от: **\$ 4719**

Специальное предложение подписчикам  
**LINUX FORMAT**  
предъявите этот купон  
и Вы получите скидку

**3%**

**TRINITY**  
CORPORATE IT PROJECTS

(812) 327-5960  
(095) 232-9230  
[www.trinitygroup.ru](http://www.trinitygroup.ru)

Любые вопросы по серверам и системам хранения данных на форуме: [www.3nity.ru](http://www.3nity.ru)

ПОДРОБНО О РАСШИРЕНИЯХ ПРОГРАММИРОВАНИЕ СЦЕНАРИЕВ

# PHP: перевод с помощью Gettext

Пол Хадсон показывает простой способ добавить поддержку множества языков в ваши сценарии. Powodyenia!

МЕСЯЦ  
НАЗАД



Мы узнали, как отразится на ваших скриптах переход на PHP6



История всего программирования вплоть до нашего времени — это история постоянного изобретения колеса. На самом деле это довольно неприятно, так как если вы сейчас захотите посадить в каждое колесо по белке, то у вас быстро закончатся белки. Так почему же мы до сих пор продолжаем заново переписывать базовые концепции и решения?

Если вам претит писать нечто, уже кем-то написанное (кроме случаев, когда вы хотите узнать, как это работает) — значит настало время отступить и позволить другим людям сделать работу для вас. В этой первой из посвящённых расширениям PHP статей мы рассмотрим Gettext, позволяющий создавать приложения с поддержкой множества языков. Итак, вперёд!

## Множество переводов

В прошлой статье мы с вами убедились, что поддержка Unicode в PHP далека от совершенства, но это не значит, что программы на PHP вообще нельзя переводить на другие языки. Фактически, в PHP есть прекрасная поддержка системы GNU Gettext, которая позволяет перевести все строки в вашем проекте так, чтобы пользователи работали с ним на своём родном языке. Вероятно, вы подумали: «Ну, это не сложно, потребуется примерно такой код», — и написали нечто вроде:

```
include "lang/spanish.php";
print $welcome;
```

Такая программа будет прекрасно работать, но её очень трудно поддерживать — вам потребуется великолепная память, чтобы запомнить, что же содержится в переменной **\$welcome**, особенно когда число строк, нуждающихся в переводе, перевалит за тысячу.

Но есть способ лучше! Gettext позволяет вам использовать строковые константы на английском языке прямо внутри программы, а затем ищет их и заменяет на переводные аналоги прямо в процессе выполнения, в зависимости от настроек пользователя.

Поддержка Gettext по умолчанию отсутствует в PHP, так что вам потребуется достать последнюю версию исходного кода и во время конфигурации указать опцию **--with-gettext** в числе других опций, настраивающих, какие расширения вы собираетесь использовать. После выполнения команд **make** и **make install** вы будете готовы к последней проверке. Запустите **php -m** и убедитесь, что модуль Gettext установлен.

Кроме того, чтобы использовать Gettext в своих приложениях, вам потребуются пакеты **gettext** и **gettext-devel**. Модуль **gettext** устанавливается по умолчанию во всех дистрибутивах, которые я знаю, но утилиты для его использования в своих программах входят в **gettext-devel**, так что вам придётся установить и его. В моём дистрибутиве SUSE 10.0 **gettext-devel** загадочным образом отсутствует, но я быстро нашёл нужный RPM в Google.

Теперь давайте создадим простой сценарий, который затем будем переводить. Для этого нам потребуются несколько новых функций:

**setlocale()** для того, чтобы подменить текущие языковые настройки пользователя, **bindtextdomain()** для того, чтобы указать где на жёстком диске находится файл перевода и **textdomain()**, чтобы приказать Gettext использовать этот перевод. Запишите следующий код в файл **gettext\_test.php**:

```
<?php
setlocale(LC_ALL, "pl_PL");
bindtextdomain("messages", "./locale");
textdomain("messages");
echo gettext("Hello, Linux Format!\n");
?>
```

## ¿Hables inglés?

Следующим шагом мы добавим несколько переводов нашего текста. И тут уже вступает в игру **gettext-devel**. Если у вас установлен этот пакет, значит, вы можете воспользоваться утилитой **xgettext**, которая извлекает все строки из программы и создаёт файл **messages.po**. Это тот самый файл, в котором будет жить ваш перевод, и именно его нужно отправлять команде, отвечающей за локализацию проекта, чтобы они могли сделать свою работу.

Поскольку у вас, скорее всего, нет под рукой команды локализации, вероятно вам придётся сделать и этот шаг самостоятельно (я надеюсь, что у вас найдётся польско-венгерский словарь). Для начала запустите

```
xgettext -n gettext_test.php
```

Эта программа создаст файл **messages.po**, в котором после некоторого количества неинтересной информации будет следующее:

```
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006e-03-23 10:07+0000\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"
#: text1.php:6
msgid "Hello, Linux Format!\n"
msgstr ""
```

Каждой строке **msgid** соответствует строка в нашем PHP-сценарии. Ниже расположены строчки, начинающиеся на **msgstr**, в которых нужно указывать переводы. Использование параметра **-n** привело к включению в файл комментариев, указывающих где какая строчка находится.

Мы собираемся создать три варианта переводов: pl\_PL (польский язык, на котором говорят в Польше), hu\_HU (венгерский язык, на котором говорят в Венгрии) и en\_GB (английский, на котором говорят в цивилизованной части мира). Мы уже указали в качестве области с текстами каталог **locale**, а это значит, что в этом каталоге нам нужно создать по подкаталогу для каждого поддерживаемого языка. Каждый из этих каталогов должен содержать подкаталог **LC\_MESSAGES**, в котором уже будет храниться наш PO-файл с переводом.

Итак, скопируйте файл **messages.po** в каждый из этих каталогов. Откройте файл в каталоге **locale/pl\_PL/LC\_MESSAGES** и отредактируйте его так, чтобы последними строчками были:

```
msgid "Hello, Linux Format!\n";
msgstr "Czesc, Linux Format!\n";
```

Сохраните файл и запустите команду

```
msgfmt messages.po
```

Вероятно, вы получите предупреждение о неправильной кодировке, но не беспокойтесь на этот счёт – **msgfmt** преобразует **messages.po** в **messages.mo**, который содержит бинарные версии наших строк и может быть использован **gettext**.

В каталоге **locale/hu\_HU/LC\_MESSAGES** вы, вероятно, захотите указать в качестве значения **msgstr** **"Szia, Linux Format!\n"**, а в **locale/en\_GB/LC\_MESSAGES** – **"What ho, Linux Format!\n"**. Предупреждая ваш вопрос, сразу отвечу – нет, в действительности я так не разговариваю. Запустите **msgfmt** в каждом из каталогов, так чтобы у вас в итоге получилась структура, показанная на рис. 1, где в качестве корня служит тот каталог, в котором вы работаете.

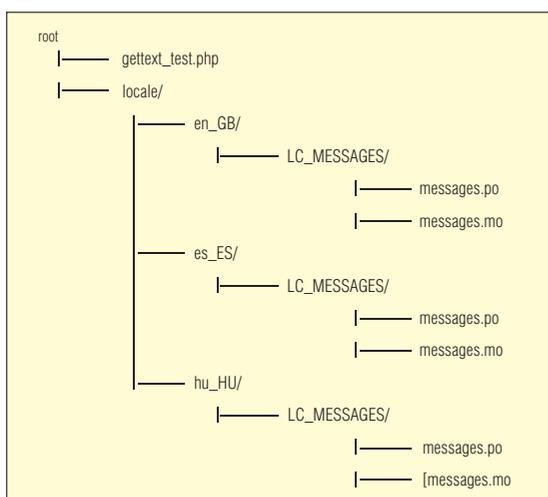
Прямо сейчас вы уже можете запустить программу и увидите работу **gettext** в действии. Если вы помните, в программе мы указали в качестве языка pl\_PL, так что если вы запустите **php gettext\_test.php**, то получите на консоли текст **"Czesc, Linux Format!"**.

Если вы измените файл **gettext\_test.php** так, что вместо pl\_PL там будет указано en\_GB, то текст на консоли соответствующим образом изменится.

## Обработка множественных форм.

Программисты ленивы. Я имею ввиду вот что – видели ли вы где-нибудь текст наподобие **«3 file(s) deleted»**? Такие вещи – это обычное дело, поскольку требуется написать несколько лишних строк кода, чтобы программа говорила «files» когда речь идёт об удалении более чем одного файла и «file» в случае удаления единственного экземпляра. Но если даже потребовать от программистов аккуратности, они скорее всего напишут что-то вроде

```
if ($deleted_files == 1) {
    echo "1 file deleted";
} else {
    echo "$deleted_files deleted";
}
```



Структура каталогов для нашей тестовой программы.

Да, это немного получше, но всего лишь немного. Проблема в том, что все три языка, которые мы используем, имеют разные правила образования множественного числа: в английском языке используются формы **file** и **files**, венгры используют одну единственную форму (**fáj!**), идёт ли речь об одном файле или о тысячах. Польский ещё сложнее – у вас есть один **plik**, два, три или четыре **pliki**, а для чисел от 5 до 21 используется форма **plików**. Это очень сложно, но **Gettext** прекрасно справляется даже здесь.

Для начала давайте добавим в наш тестовый сценарий строку, которая зависит от числа. Добавьте в конец файла **gettext\_test.php** следующий код:

```
$file_count = 0;
printf(ngettext("%d file", "%d files", $file_count), $file_
count);
echo "\n";
```

Функция **ngettext()** отвечает за образование множественного числа для выбранного языка. В качестве первого его параметра передаётся текст для единственного числа, в качестве второго – какую строку использовать для множественного числа, а в качестве третьего – число, от которого зависит выбор формы. **%d** передаётся в **gettext** и не изменяется им, так что например при использовании польского языка будет возвращено **%d pliki**, если **\$file\_count** равно 34. Затем эта строка передаётся в функцию **printf()**, которая уже подменяет **%d** на значение переменной **\$file\_count**.

Теперь нам нужно научить **Gettext** вычислять множественные формы для всех используемых языков. Итак, в **en\_GB messages.po** добавьте следующую строчку прямо перед **'Project-Id-Version'**:

```
"Plural-Forms: nplurals=2; plural=n != 1\n"
```

Это значит: в этом языке есть две формы множественного числа, множественное надо использовать, если значение числа не равно единице. Теперь нам надо добавить переводы для строк **%d file** и **%d files**, но поскольку дело касается множественного числа, то и указывать переводы надо особым образом. Добавьте в **messages.po** в каталоге **en\_GB/LC\_MESSAGES** следующий текст:

```
msgid "%d file"
msgid_plural "%d files"
msgstr[0] "I say, there is %d file!"
msgstr[1] "I say, there are %d files!"
```

Для венгерского языка описание множественной формы будет следующим:

```
"Plural-Forms: nplurals=1; plural=0"
```

Это значит: использовать одну и ту же форму для единственного и множественного числа. Перевод в конце файла будет выглядеть так:

```
msgid "%d file"
msgid_plural "%d files"
msgstr[0] "%d fájl"
```

И наконец, польский язык. У него очень сложная строка описания множественной формы, поскольку надо обработать три типа множественного числа. К счастью, дружелюбные к GNU местные жители, которые написали справку к **Gettext**, подготовили для нас правильный вариант:

```
"Plural-Forms: nplurals=3; plural=n==1 ? 0 : n%10>=2 && n%10<=4 && (n%100<10 || n%100>=20) ? 1 : 2;\n"
```

А вот перевод:

```
msgid "%d file"
msgid_plural "%d files"
msgstr[0] "%d plik"
msgstr[1] "%d pliki"
msgstr[2] "%d plików"
```

Осталось запустить **msgfmt** в каждом из каталогов с переводами и запустить сценарий на выполнение. Попробуйте менять значение **\$file\_count** и наблюдайте за результатом, особенно для польского языка.

**Gettext** – это всё, что вам нужно для того, чтобы ввести многоязычность в ваши программы. Да, придётся приложить некоторые усилия, но пользователи будут вам очень благодарны. **LXF**

## ПОДСКАЗКИ

- Если вам надоело постоянно набирать **gettext()**, просто используйте более короткое имя функции **\_()**.
- Если вы хотите установить ваш вариант перевода на уровне всей системы, то не используйте файлы под названием **messages.mo**. Вместо этого используйте название своего приложения, например **fuzz.mo**, чтобы избежать конфликта имён. Вам потребуется изменить вызовы **bindtextdomain()** и **textdomain()**, чтобы они соответствовали новому имени файла.
- Когда вы распространяете своё приложение для обычных пользователей, то вам не нужно включать в дистрибутивы PO файлы, вполне достаточно его скомпилированной MO версии. PO может потребоваться только тем, кто будет изменять перевод.
- Чтобы узнать, как правильно обрабатывать множественные числа других языков, обратитесь к онлайн-документации **Gettext** на <http://www.gnu.org/software/manual>.

## ЧЕРЕЗ МЕСЯЦ

Снова о расширениях: безопасные оболочки из PHP



МОБИЛЬНАЯ БЕЗОПАСНОСТЬ

## Шифрование Защита файловых систем

Захотим обезопасить свои данные – обойдемся и без знаменитой Энигмы. **Грэм Моррисон** покажет, как зашифровать ваш USB-брелок и данные на ноутбуке, используя Fuse, EncFuse и небольшой модуль под названием *dm-crypt*.



**НА ДИСКЕ**

- EncFS 1.31
- Fuse 2.5.2
- *dm-crypt* 0.1



О шифровании говорят многие, но мало кто им реально пользуется. Нам всем нравится идея засекретить данные, причем так, чтобы их можно было видеть как обычно, и все мы прошли через рутину баловства с *GNU Privacy Guard* и помещения наших публичных ключей в Интернет. Но судя по всему, только пользователи, обладающие важными данными, которые действительно надо прятать, или особо усердствующие насчет сетевой безопасности, пользуются шифрованием ежедневно.

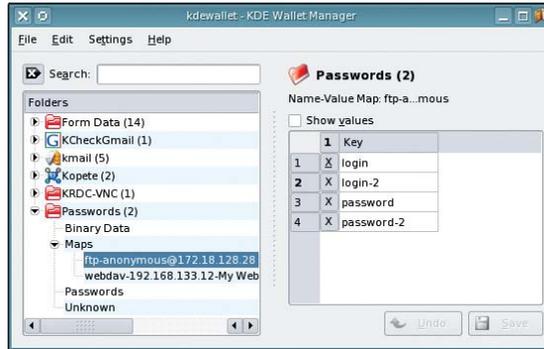
Те из нас, у кого есть переносные устройства, не должны быть столь апатичны. Довольно неприятно потерять ноутбук, но потерять вместе с ним еще и ценные данные вообще ужасно. А последствия утери жёсткого диска с конфиденциальными данными, описывающими все аспекты вашей жизни, могут быть просто катастрофичны. И даже не из-за пропажи финансовой информации вроде номеров кредитных карт и банковских счетов – адресная книга и почтовый архив могут быть много

важнее. Как это ни глупо, но большинство из нас скорее предпочитают закрывать глаза на возможность беды, нежели предпринять несколько простых предосторожностей. И один из простейших способов защитить наши данные – это зашифровать их.

Существует три основных подхода к шифрованию данных в вашей системе Linux. Первый – это зашифровать ваши файлы при помощи *GPG* или подобного инструмента. Способ хорошо работает в том случае, когда вы храните всю вашу важную информацию (типа паролей и сетевой банковской информации) в немногочисленных файлах. Когда надо будет прочитать или изменить данные, вы можете расшифровать их. На этой теории основана масса приложений типа защищенных бумажников, появившихся в течение последней пары лет. Важная информация шифруется в несколько файлов, а опирается глобальным ключом.

Расширением идеи шифрования только некоторых файлов является создание особого каталога: файлы, помещаемые в него, зашиф-

ровываются, и дешифруются при необходимости их извлечения и чтения — естественно, при вводе правильного ключа; а в остальном каталог работает как привычный компонент вашей файловой системы. Данный подход известен как файловая система с паролем, и у него множество преимуществ по сравнению с обычным шифрованием файлов; но более важно то, что он работает как обычная файловая система. После шифрования вы видите ту же файловую структуру, только имена файлов и папок выглядят бессмысленной абракадаброй. Можно распространить такое шифрование на весь USB-брелок, для уверенности в том, что если он потеряется, то информация будет нельзя извлечь.



*Kdewallet* применяет шифрование и однократный запрос пароля, чтобы обезопасить настольные приложения *Kopete* и сессию VNC, а также пароли online-форумов.

## ЧАСТЬ 1. ВИРТУАЛЬНЫЕ ЗАШИФРОВАННЫЕ КАТАЛОГИ

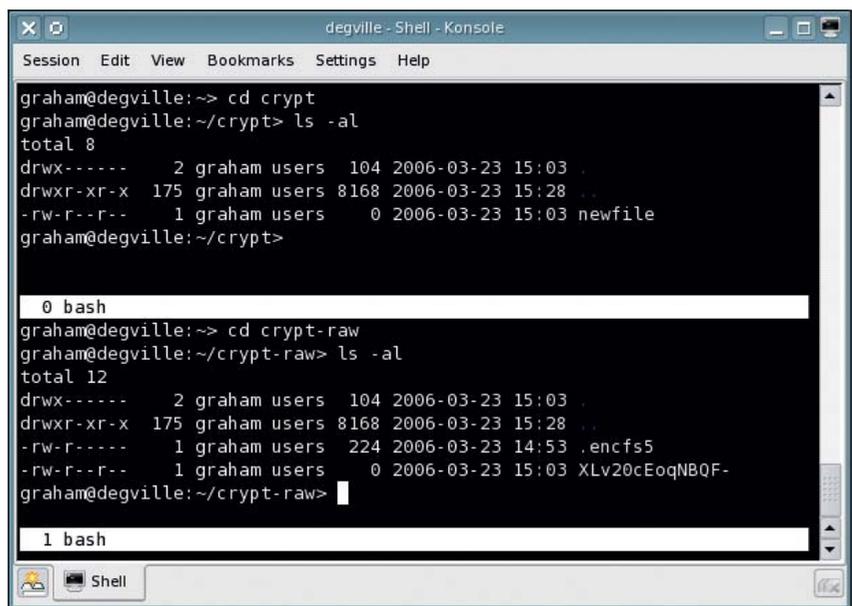
**Наиболее простой и эффективный способ создания зашифрованной области для ваших файлов** — это использовать Fuse, Файловые системы в пространстве пользователя (Filesystems in Userspace), детально рассмотренные в нашем учебнике в *LXF76*. Fuse — модуль ядра, позволяющий пользователям создавать свои собственные локальные файловые системы, в своём домашнем каталоге или на USB-брелке, например. Главное преимущество здесь в том, что можно затем открыть файловую систему носителя под другими операционными системами — особенно полезно, если вы собираетесь подключать ваш USB-брелок к другим компьютерам (а именно для этого USB-брелок и придуман).

### Fuse, легкий способ

Обычно вам не нужно беспокоиться о самостоятельной установке Fuse. Практически во всех дистрибутивах последних версий, включая SUSE, Fedora и Ubuntu, это предусмотрено. Хотя в прошлый раз при рассмотрении Fuse мы ограничились SSH, для зашифрованных файловых систем и Fuse тоже существует специально разработанный инструмент.

Этот инструмент называется *EncFS*, и благодаря Fuse решение получается весьма эффективным и гибким. Если вы не смогли найти пакет *EncFS* для вашей системы (что маловероятно), сборка из исходных текстов займет столько времени, сколько вам потребуется для того, чтобы найти и установить Fuse и связанные с ним библиотеки разработчика. Вам также понадобится пакет **rlog** (созданный автором *EncFS*), который доступен по адресу <http://arg0.net/wiki/EncFS>.

После установки *EncFS*, для создания зашифрованного каталога достаточно просто набрать **EncFS ~/crypt-raw ~/crypt**. *EncFS* сначала спросит, хотите ли вы создать два каталога, указанные в вышеприведенной команде (**crypt-raw** и **crypt**), которые, в нашем примере, расположены в домашнем каталоге. «Реальным» каталогом является **crypt-raw**, содержащий зашифрованные файлы, тогда как **crypt** содер-



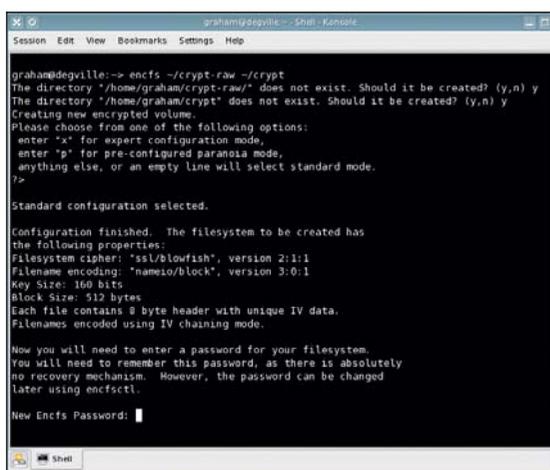
жит виртуальную файловую систему — расшифрованную версию содержимого каталога **crypt-raw**.

Следующий необходимый шаг — указание желаемого режима настройки. Выбор включает Expert Configuration (Для Эксперта), Preconfigured For The Paranoid (Для параноика) и Standard Mode (Стандартный). В большинстве случаев стандартный режим более чем достаточен — он применяет 160-битные ключи и алгоритм шифрования SSL/Blowfish от Брюса Шнайера (Bruce Schneier). Экспертный режим предоставляет также 256-битное AES-шифрование, надежное, как в Пентагоне. Размер области зашифрованных данных определяется размером блока. Пусть вы хотите сохранить одиночный байт, *EncFS* все равно создаст зашифрованный блок указанного вами размера.

Последний шаг в построении файловой системы — установка пароля для блокировки содержимого. Пароль обычно считается самым слабым местом в шифровании, так как многие люди используют один и тот же простой пароль много раз, и теоретически Blowfish особенно уязвим для атак на слабые ключи. Хотя на практике это почти невозможно, слабые ключи несомненно упрощают работу хакера, а пароли случайной природы, включающие символы и цифры наряду с буквами, труднее для взлома, чем девичья фамилия матери. Проблема в том, что если вы забудете пароль для *EncFS*, то нет инструментов для восстановления ваших данных. В результате, лучше проявить слабость, взяв запоминающийся пароль, а не недоступный для взлома.

Затем войдите в каталог **crypt** и создайте файл. Команда **touch newfile** создаст пустой файл с именем **newfile**, но вы можете легко скопировать в этот каталог данные, которые хотите защитить. Теперь, взглянув на содержимое каталога **crypt-raw**, вы увидите то же количество файлов, но их имена и содержимое будут представлять собой

Один и тот же каталог в расшифрованном (вверху) и зашифрованном (внизу) состоянии.



Выглядит не особо эффектно, зато *Fuse* и *EncFS* — эффективное и мощное решение для шифрования.



« случайный набор символов. Эта криптограмма – результат шифрования. Однако права доступа к файлу, вместе с датой и временем изменения, все еще читаемы. Это представляет небольшой риск, но зато программы резервного копирования смогут понять, когда файл изменялся в последний раз, и сделать его копию. В каталоге **raw** есть также скрытый файл, с именем **encfs5**, в нем содержится информация, облегчающая понимание содержимого исходной файловой системы для *EncFS*.

Вы можете размонтировать зашифрованную файловую систему, набрав **fusermount -u ~/crypt** – запрет на шифрование в каталог **crypt-raw**, и ваши файлы будут как в сейфе. Чтобы вновь получить к ним доступ, просто запустите первую команду, которую мы использовали для создания каталога.

## ЧАСТЬ 2. ШИФРОВАНИЕ ВСЕЙ ФАЙЛОВОЙ СИСТЕМЫ

**Для переносных устройств виртуального каталога недостаточно.** Если вы, например, потеряете ноутбук, то потеряете намного больше, чем важные данные – операционная система и используемые вами приложения станут вашими злейшими врагами. Офисные приложения при редактировании почти всегда создают копии документа, которые не бросаются вам в глаза. Также рискованно обнародовать кэш и cookies вашего браузера.

Теперь, познакомив вас с основами шифрования частей вашей файловой системы, я собираюсь перейти к следующему этапу – шифрованию целого компьютера. Шифрование всей файловой системы кардинально отличается от виртуальных папок. Проблема в загрузке зашифрованной корневой файловой системы – надо знать, в какой момент сообщить

до офисной машины позднее. Процесс намного бы упростился, если бы дистрибутив разрешал сразу создать зашифрованный раздел, но большинство дистрибутивов этого не разрешают.

### Загрузочный раздел

Перед началом работы важно уяснить, как именно происходит загрузка системы и в какой момент мы можем переключиться на зашифрованную систему. Поскольку большинство дистрибутивов Linux в настоящее время используют ядро версии 2.6, то мы сосредоточимся именно на нем. Можно зашифровать файловую систему и при использовании старой версии ядра, но этот процесс сильно отличается – обычно он основан на методе Loop-AES. Есть превосходный HOWTO, описывающий этот метод, он является частью Linux Documentation Project ([www.tldp.org](http://www.tldp.org)).

Мы воспользуемся модулем *dm-crypt*, который впервые появился в ядре версии 2.6.4. Он пользуется инфраструктурой отображения устройств (device-mapping infrastructure), чтобы создавать виртуальные блочные устройства, например, раздел жесткого диска, и почти прозрачен. Большинство ядер обеспечивают поддержку *dm-crypt* по умолчанию, и намного проще использовать ее, чем компилировать свою собственную версию. Вы можете проверить свое ядро на наличие данной поддержки, попытавшись загрузить модуль и просеять вывод **lsmod** при помощи **grep**. Например, выполнив команды **modprobe dm-crypt** и **lsmod | grep crypt** с правами root, вы должны получить нечто подобное:

```
dm_crypt 12424 0
dm_mod 58936 2 dm_crypt
```

Начиная на голом месте с новой установки, вы должны создать отдельный загрузочный раздел плюс достаточно большой основной, а также временный раздел, который будет содержать минимальную установку дистрибутива. У меня загрузочным разделом является **hda1**, основным – **hda2**, а временным – **hda3**, и, конечно, потребуются еще раздел подкачки (swap) – **hda4**. Установите минимально возможную версию редакции дистрибутива, используя **hda3** в качестве главного раздела, и не забудьте указать **hda1** в качестве загрузочного раздела (boot).

### Захватчики пространства

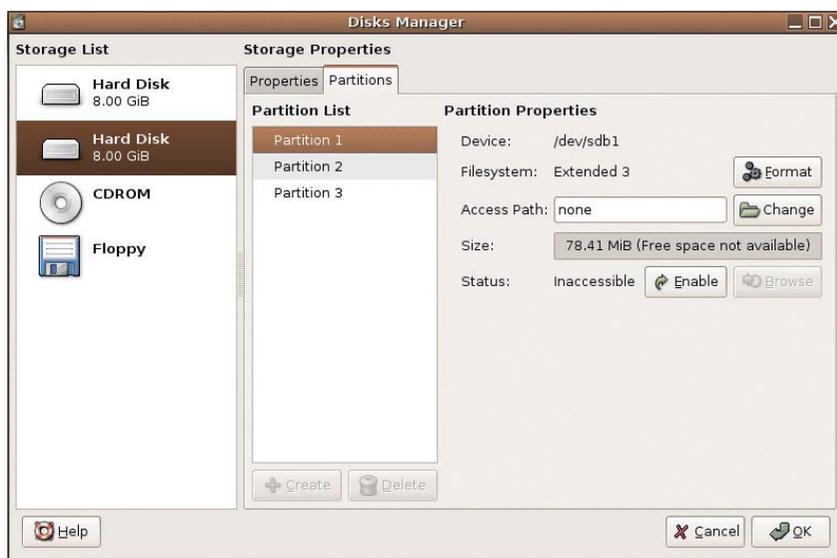
Как только установка закончится и вы получите доступ к оболочке, необходимо заполнить случайными данными раздел (**hda2**), который станет зашифрованным корневым разделом. Может, вы уже и сделали это, избавляясь от остаточных следов удаленных вами файлов, но причина заняться этим сейчас в том, что когда зашифрованная система начнет заполняться файлами, можно будет узнать, сколько места используется и сколько остается свободным, просто посмотрев двоичную карту диска. Свободное место выглядит более упорядоченно, чем имеющие случайную природу зашифрованные данные, а если записать в неиспользуемые участки диска случайные данные, то занятое место на диске определить невозможно. Просто наберите **if=/dev/urandom of=/dev/hda2**, чтобы скопировать поток случайных данных на второй раздел (все команды урока требуют привилегий администратора).

Вам понадобится установить пакет **cryptsetup**, который обычно входит в основные дистрибутивы (хотя пользователям Debian/Ubuntu придется просмотреть репозитории Universe и Multiverse). *Cryptsetup* – инструмент пространства пользователя для управления модулем *dm-crypt*. Чтобы создать зашифрованный раздел при помощи *Cryptsetup*,

# «НЕ ТАК УЖ СЛОЖНО СОЗДАТЬ СИСТЕМУ LINUX В ЗАШИФРОВАННОМ ROOT-РАЗДЕЛЕ.»

ядро, что оно управляет зашифрованной файловой системой. А если файловая система зашифрована, то как загрузить ядро в первый раз?

К счастью, это проторенный путь, и создать Linux-систему, запускающуюся с полностью зашифрованного корневого раздела, не так уж и сложно. Но потребуются принятие некоторых специальных решений, поскольку в настоящий момент нельзя просто преобразовать ваши разделы в зашифрованную версию. Мы рекомендуем использовать последнюю версию Ubuntu, в которой и опробовались все наши инструкции; впрочем, для большинства других дистрибутивов процесс весьма схож (кроме последнего шага). Вы можете также установить сильно усеченную версию Ubuntu, выбрав в меню загрузки пункт 'server', и обновить



**Нужно создать загрузочный раздел, большой раздел для зашифрованного корневого раздела и временный раздел для минимальной установки дистрибутива. Да, не забудьте раздел подкачки!**

## ШИФРУЕМ SWAP

По поводу шифрования swap-раздела много спорят. С одной стороны, swap-раздел – это снимок вашей оперативной памяти, и он может показать все, над чем вы работаете, включая документы и пароли. С другой стороны, ваша система теряет производительность, хронически шифруя нечто столь часто запрашиваемое, как swap-раздел.

Шифрование swap-раздела в Ubuntu не слишком сложно – просто создайте swap-раздел так же, как мы создали корневой раздел:

```
cryptsetup create swap /dev/hda4
```

Затем измените запись для swap в `/etc/fstab` на `/dev/mapper/swap none swap sw 0 0` и добавьте `swap /dev/hda5 /dev/random swap /etc/crypttab`. При следующей перезагрузке ваш swap-раздел будет зашифрован вместе с корневым. Предыдущее состояние swap-раздела защищено случайным паролем, который создается при каждой загрузке.

наберите то же самое, что и в нашем примере. Будет запрошен ввод пароля (pass phrase), который должен быть достаточно сложен, но запоминаем. Потеря пароля блокировки машины в корне отличается от потери пароля, блокирующего зашифрованный файл – вы потеряете ваши электронные письма, закладки, нажатые мучительным трудом последние версии... одним словом, всё.

```
Cryptsetup -y create root /dev/hdb2
```

```
Enter passphrase:
```

```
Verify passphrase:
```

Затем `dm-crypt` отобразит зашифрованный диск в `/dev/mapper/root` точно так же, как когда мы использовали зашифрованные файлы в предыдущем разделе. Но поскольку это блочное устройство, мы можем форматировать его как настоящий жесткий диск, а это необходимо сделать перед тем, как копировать на него корневую файловую систему. Так как Linux рассматривает его как обычное блочное устройство, для его форматирования вы можете использовать вашу любимую утилиту. Но мы используем командную строку, чтобы отформатировать отображенный раздел в ext3.

Вы увидите примерно такое:

```
mkfs -t ext3 /dev/mapper/root
```

```
Password:
```

```
mkfs 1.38 (30-Jun-2005)
```

```
Filesystem label=
```

```
OS type: Linux
```

```
Writing inode tables: 24/49
```

Перед тем, как скопировать наш минимальный дистрибутив с текущей корневой файловой системы на новую зашифрованную, нужно смонтировать ее в удобном месте. Чтобы это сделать, наберите `mkdir /mnt/enc_root` и `mount /dev/mapper/root /mnt/enc_root`. Скопируйте ваш старый корневой раздел в новый, используя команду вроде `cp -axv / /mnt/enc_root`.

Последний шаг – измените порядок загрузки и убедитесь, что загружаемая система знает, что делать с зашифрованной файловой системой. Первой целью является `/etc/fstab` на зашифрованной файловой системе, и запись для корневого устройства необходимо изменить на `/dev/mapper/root`, чтобы она выглядела следующим образом:

```
/dev/mapper/root / ext3 defaults,errors=remount-ro 0
```

Для пользователей Debian и Ubuntu последний шаг значительно облегчен, потому что, отредактировав пару файлов, они будут готовы к загрузке зашифрованной системы. Просто добавьте `root /dev/hda2 в /etc/crypttab` (не забыв изменить `hda2` на имя вашего зашифрованного диска), а `ramdisk = /usr/sbin/mkinitrd` в `/etc/kernel-img.conf`, и, наконец, отредактируйте `Grub (/boot/grub/menu.lst)`, изменив строку, начинающуюся с `# kopt` на

```
# kopt=root=/dev/mapper/root devfs=mount ro
```

Для внесения изменений в `Grub` также необходимо выполнить команду `dpkg-reconfigure linux-image-2.6.15-15-386`, чтобы убедиться, что используется корректная версия ядра. Если все прошло как

планировалось, после перезагрузки ваша машина запросит пароль корневого каталога, и только потом вы сможете загрузить зашифрованную систему и ввести пароль пользователя.

## Другие дистрибутивы

Пользователи дистрибутивов, основанных не на Debian, должны будут изменить скрипт `linuxrc`. Это сценарий, выполняющийся при первой загрузке системы, и ему необходим локальный доступ к модулям, используемым при шифровании. Это подразумевает копирование модулей `dm-mod.ko` `dm-crypt.ko` в загрузочный раздел (boot) и их загрузку из скрипта `linuxrc` перед ручным созданием точки монтирования `/dev/mapper/root` как мы делали ранее. Большинство дистрибутивов содержат пример скрипта для выполнения подобных вещей.

Шифрование файловой системы становится всё более важным, поскольку всё больше и больше народу используют ноутбуки и USB-брелки. SUSE – один из немногих дистрибутивов, предлагающих при установке выбрать режим использования зашифрованной файловой системы. Эта опция доступна в окне Экспертное Разбиение (Expert Partitioner), которое

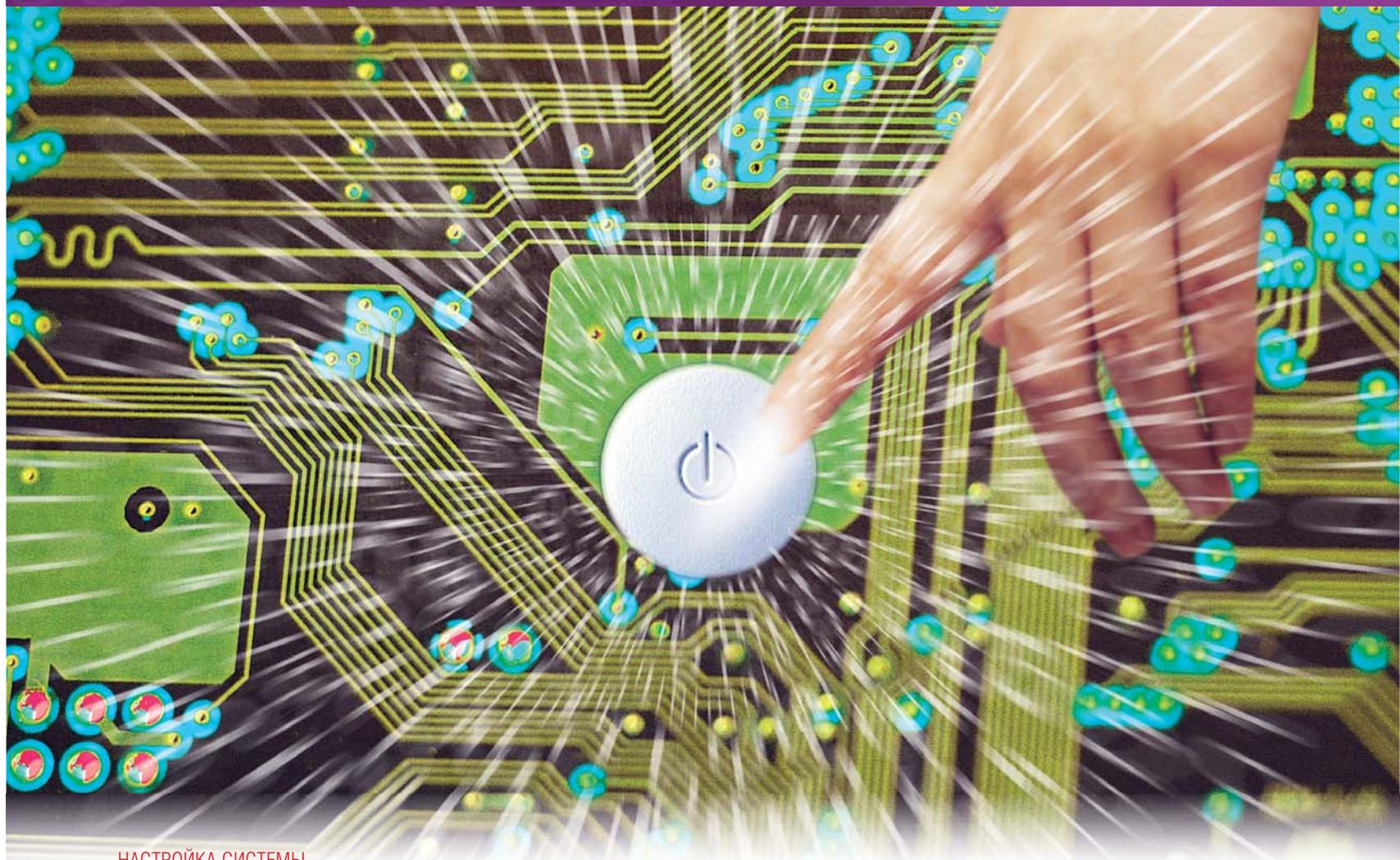
# «НИЧТО НЕ СРАВНИТСЯ СО СПОКОЙСТВИЕМ ОТ СОЗНАНИЯ СВОЕЙ БЕЗОПАСНОСТИ.»

появляется сразу после выбора раздела для установки. Для создания зашифрованного раздела просто щелкните на кнопке Редактировать (Edit) и установите галочку в пункте Шифровать файловую систему (Encrypt File System). Появится диалоговое окно, в котором дважды запрашивается один и тот же пароль – точно так же, как при создании зашифрованного раздела вручную – и вы должны ввести пароль, который будет необходим при загрузке. Для создания зашифрованного раздела и виртуальных каталогов можно также использовать Yast, если SUSE уже установлен, но корневую файловую систему вы зашифровать не сможете.

Уже недолго ждать предоставления большинством дистрибутивов режима установки в зашифрованный раздел. Но ничто не сравнится с душевным покоем, проистекающим из сознания, что вы сами позаботились о своей безопасности и все сделали правильно. Дайте знать, что у вас получилось. И не забудьте наши форумы, если вам нужен совет! **LXF**



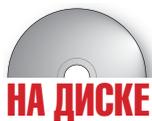
IMAGE: COURTESY DELL, INC



НАСТРОЙКА СИСТЕМЫ

# Ускорение загрузки системы

Поковырившись в настройках, вы сможете вдвое уменьшить время загрузки вашей машины, утверждает **Майк Сондерс**.



- InitNG 0.5.5
- Bootchart 0.9



Одна из популярнейших жалоб пользователей Linux – жалоба на мучительно долгую загрузку. На современной машине обычно приходится ждать 45–60 секунд, а если вы используете не самое новое оборудование, это время может возрасти до двух и более минут. Правда, препятствие невелико – благодаря абсолютной надежности Linux сбои возникают редко и нет нужды перезагружаться так часто, как под другими ОС. Но медленность загрузки досаждала пользователям машин в стиле «включил и работаю», вроде ноутбуков; и если вы хотели поберечь энергию и не держать машину все время включенной, то за время ожидания вы сильно усомнитесь в своей экологической добродетели.

Скорость загрузки не является специфичной проблемой индивидуальных компьютеров – для web-сервера, если вы хотите уменьшить время недоступности после обновления ядра, это тоже чрезвычайно важный фактор. Недоступность, например, сайта по продаже товаров в течение 30 секунд запросто разгонит покупателей и превратит потенциальные дензнаки в нахмуренные брови. К счастью, причину медленной загрузки обычно легко определить: дело в том, что многие дистрибутивы пытаются выполнять всё. Коробочная версия дистрибутива вроде Fedora охватывает пользователей всех мыслимых типов, но если вы знаете, чего хотите, то лишние службы нетрудно отсечь.

На этом уроке мы займемся специальными настройками и улучшениями, которые вы сможете приложить к своей системе Linux для сокращения времени загрузки, чтобы приступить к работе с невозмож-

ной быстротой. Приведенные советы работают независимо; вам не обязательно следовать всем, и если какой-то покажется сложным, можете его пропустить. Каждый совет также помечен степенью сложности. Даже если вы замаяхнете на мероприятия для особо продвинутых, при наличии резервной копии стартовых скриптов вам ничего не грозит (как сделать такую копию, показано ниже во врезке «Резервируем»!). Приступим...

## РЕЗЕРВИРУЕМ!

Изменения, которые мы внесем, повлияют на стартовые скрипты вашей системы Linux, расположенные в каталоге `/etc`, поэтому разумно будет прежде всего сохранить их копию. Откройте окно терминала, войдите как суперпользователь и введите

```
cp -R /etc /old-etc
```

Конфигурационные файлы вашей системы скопируются из каталога `/etc` в каталог `/old-etc`. Если что-то пойдет не так, вы сможете перекинуть содержимое `/old-etc` обратно в ваш каталог `/etc`, перезагрузиться, и все придет в норму. Если вы предприняли шаги, описанные в данном учебнике, и все работает корректно, каталог `/old-etc` можно смело удалять.

Не паникуйте, если вы допустили ошибку и появились проблемы при загрузке – вы можете запустить Live-дистрибутив вроде Damn Small Linux (присутствующий на DVD-приложении), подмонтировать ваш жесткий диск и скопировать `/old-etc` из-под него.

## УБИРАЕМ ЗАДЕРЖКУ ЗАГРУЗЧИКА

Сложность совета: простой

Если у вас мультизагрузочная система, и вы не покинули компьютер, едва нажав кнопку пуска, вы, скорее всего, коршуном бросаетесь на клавишу **Enter**, как только появится меню выбора ОС. Но если, выключив компьютер, вы идёте попить чайку (обычный способ употребить ожидание с пользой), ваша машина будет простаивать, пока загрузчик не выберет ОС автоматически. Многие дистрибутивы по умолчанию устанавливают задержку в 20–30 секунд перед загрузкой ОС, и если загрузка происходит в ваше отсутствие, лучше убрать эту задержку.

Первое, что нужно сделать — это определить используемый загрузчик. Поищите файлы с именами **/etc/grub.conf**, **/boot/grub/grub.conf** или **/boot/grub/menu.lst** — если вы нашли один из них, то вы используете *Grub* и можете изменить конфигурационные файлы, от лица суперпользователя (**root**). Откройте найденный файл в текстовом редакторе и поищите строку **timeout**, что-то вроде

```
## timeout sec
```

```
# Set a timeout, in SEC seconds, before automatically booting the default entry
```

```
# (normally the first entry defined).
```

```
timeout 30
```

Здесь установлена задержка 30 секунд перед стартом операционной системы по умолчанию (указана в строке **default**, которая расположена прямо под рассматриваемой). Измените это значение на, скажем, 5 секунд — вполне достаточно, чтобы, сидя за компьютером, увидеть меню и выбрать ОС, и не слишком задерживает неконтролируемую загрузку.

Между тем, некоторые дистрибутивы используют загрузчик *Lilo*, чьи настройки обычно находятся в **/etc/lilo.conf**. Найдите строку **timeout** (в которой указано **timeout = 20** или вроде того) и установите число 5 или меньше. Завершив изменения, запустите **/sbin/lilo** (от имени суперпользователя) и обновите загрузчик. В следующий раз, когда вы предоставите машину самой себе, меню загрузки исчезнет значительно быстрее.

## ОТКЛЮЧАЕМ НЕНУЖНЫЕ СЛУЖБЫ

Сложность совета: простой

Этот совет является наиболее эффективным способом уменьшения времени загрузки — к тому же у него низкий рейтинг сложности! При старте, ваша система запускает некоторое количество неинтерактивных программ (называемых службами — *services*), помогающих настроить дистрибутив или выполняющихся в фоновом режиме, пока вы работаете с машиной. Например, службы *udev* и *hotplug* запускаются при загрузке, но вступают в дело, только когда вы подсоединяете к своей машине устройство. Или, если на вашей машине работает сервер, например *Apache*, ваш стартовый скрипт запускает его во время загрузки.

Некоторые большие дистрибутивы запускают 15 и более служб, и каждая добавляет ко времени загрузки несколько секунд (а то и больше). Некоторые из них важны для корректной работы вашей системы, но большинство можно отключить без вредных последствий. Например, ваш дистрибутив при загрузке может запускать службу *PCMCIA/Cardbus* — очень нужную в ноутбуках, но напрасно съедающую время на настольных компьютерах, потому что скрипт опрашивает несуществующие устройства. Аккуратно отключив ненужные службы, вы можете уменьшить время загрузки примерно на 20 секунд.

Труднее всего определить, что именно подлежит пресечению. По традициям UNIX, инициализационные скрипты Linux имеют краткое имя, мало что говорящее обычному пользователю. К счастью, те же традиции UNIX означают, что это просто текстовый файл, содержащий текстовые комментарии, так что вы можете самостоятельно изучить каждый скрипт и решить, нужен он вам или нет. Все скрипты обычно хранятся в каталоге **/etc/init.d** — заглянув в него, вы увидите наборы исполняемых скриптов *Bash*. Ниже перечислены службы, без которых многие могут обойтись:

- **mountnfs, nfs и portmap** Они используются для сетевой файловой системы (NFS) вместе с утилитой поддержки *Portmap*. Если вы не монтируете каталоги с других Linux/Unix машин, расположенных в сети, или используете *Samba* для доступа к Windows-машинам, можете их отключить.
- **pcmcia** Эта служба тратит время на поиск несуществующих устройств. Если вы работаете не на ноутбуке, то она вам не нужна.
- **apmd** Система управления питанием, используемая в старых ноутбу-

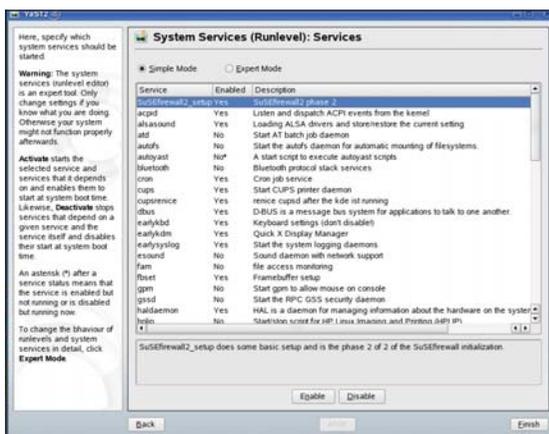
## «КРУПНЫЕ ДИСТРИБУТИВЫ ЗАПУСКАЮТ 15 И БОЛЕЕ СЛУЖБ, И КАЖДАЯ УДЛИНЯЕТ ЗАГРУЗКУ.»

ках (то есть до 2002 года выпуска). Я бы оставил ее включенной, если вы не уверены, что ваша машина поддерживает новый стандарт ACPI.

- **lvm** Система управления логическими томами (Logical Volume Manager), для организации разделов диска. Большинство пользователей настольных компьютеров запускают ее разве что в процессе установки дистрибутива, на этапе создания разделов.
- **joystick/gpm** Службы джойстика и мыши в текстовом режиме.
- **mdadm/mdadm-raid** Скрипты администрирования составных устройств. Если вы не используете RAID-массив (для настольных машин это редкость), отключите его.
- **ssh** Для удаленных подключений. При отсутствии удаленного доступа к вашей машине службу можно выключить.
- **mysql** Сервер баз данных. Некоторые дистрибутивы по умолчанию включают его, но коли вы не администратор базы данных и не разработчик — зачем он вам?
- **apache, sendmail** или **postfix** Интернет-службы для web и почты. Если они вам не нужны, то их отключение может быть очень эффективным.

Следует также просмотреть и другие скрипты (откройте их при помощи *less* или текстового редактора) и почитать комментарии в самом их начале, чтобы понять, что они делают. Если становится ясно, что вы не используете ни одну из возможностей скрипта, то его можно отключить. Отметим, что файлы **rc** и **rcS** (или аналогичные — **rc.single**, **rc.sysinit** и т.п. — прим.ред.) критичны для правильного включения и выключения машины, и их трогать не нужно!

Приняв решение, что вам нужно, а что нет, можно приступить к отключению сервисов. Выполнение этой операции зависит от дистри-



YaST в SUSE включает полезный модуль, который позволяет найти и отключить ненужные программы.



бтива. В SUSE откройте утилиту конфигурирования *Yast*, перейдите в System > System Services и отключите выбранное в списке служб (services). В Mandriva запустите Центр Управления Mandriva, перейдите в Службы (Services) и снимите выделение с тех служб, которые вам не нужны. Fedora Core и ее производные содержат утилиту Службы (Services) в меню Рабочий стол > Системные утилиты > Настройки сервера (Desktop > System Tools > Server Settings).

Если вы работаете в Debian или одной из его производных (Ubuntu, Knoppix и иже с ними), запустите терминал, войдите как суперпользователь (root) и наберите:

```
update-rc.d -f <имя службы> remove
```

Если ни один из перечисленных способов у вас не работает, остается

метод грубой силы: просто-напросто убрать стартовые скрипты вручную, позаботившись о возможности их восстановления на случай ошибки. Войдите как суперпользователь и в терминале наберите

```
mv /etc/init.d/<имя скрипта> ~
```

где <имя скрипта> – имя службы, которую вы хотите отключить (например, *rcscsi*).

Команда перенесет скрипт в ваш домашний каталог, и он не будет запускаться при загрузке; метод несложный, но может сопровождаться ошибкой 'доступ запрещен' ('permission denied') во время загрузки, означающей только то, что дистрибутив не может найти скрипты – не опасно, но нудно. Так что уж лучше пользоваться инструментами вашего дистрибутива.

## СНИЖАЕМ ПАУЗЫ ЗАСЫПАНИЯ

Сложность совета: средний

**Жизнь инициализационных скриптов дистрибутива тяжела:** они должны обеспечить работу огромного количества оборудования, различных типов пользователей и всевозможных программ, которые должны быть запущены во время загрузки. Поэтому при исследовании

ем файлы, содержащие вызовы *sleep* (вместе с временем задержки). По именам файлов определите, программы какого типа они запускают, затем откройте их в редакторе, найдите строки с командой *sleep* и удалите числа. Вряд ли стоит менять все значения на ноль, но вы спокойно можете уменьшить их вдвое (сгодится даже 0.5), перезагрузиться и проверить, что все работает нормально.

Если у вас есть время, я бы рекомендовал делать изменения в пакетном режиме, чтобы облегчить возврат к предыдущим значениям, если что-то стартует некорректно. В больших дистрибутивах, имеющих множество стартовых скриптов, сокращение задержек может привести к уменьшению времени загрузки на 10–15 секунд. И ваш дистрибутив не потребует снотворного взамен!

# «НЕ РАЗ ВО ВРЕМЯ ЗАГРУЗКИ ДИСТРИБУТИВ ФАКТИЧЕСКИ ЗАСЫПАЕТ НА ПАРУ СЕКУНД...»

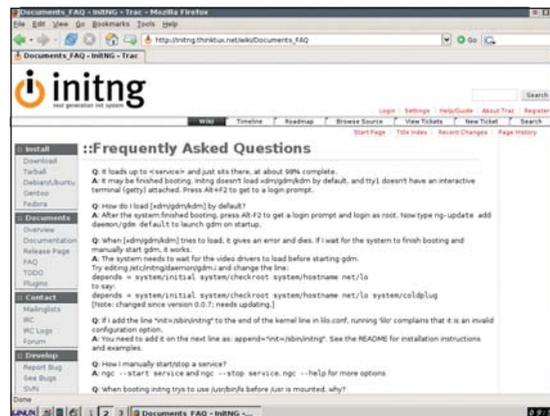
стартового скрипта вы найдете несколько хитростей и уловок. Из них наиболее распространены «спячки» (sleeps). Да, во время загрузки дистрибутив несколько раз фактически засыпает на пару секунд, чтобы убедиться, что оборудование инициализировано и другие программы определенно заработали.

В скриптах это обозначено в виде вызовов *sleep(1)*, утилиты командной строки, которая ничего не делает, кроме задержки выполнения на указанное время. (Откройте терминал и введите **sleep 5 ; ls** – список файлов будет выведен после пятисекундной паузы). Например, перед настройкой сети стартовый скрипт должен сначала убедиться, что сетевой адаптер PCMCIA/Cardbus инициализирован, и он засыпает на пару секунд, чтобы не обгонять события.

Многие из этих задержек являются излишней предосторожностью – если ваша система хорошо справляется со стандартным оборудованием, вы можете без опаски убрать их. Чтобы их найти, откройте терминал, войдите как суперпользователь (**su** или **sudo bash**) и введите

```
cd /etc/init.d
grep sleep *
```

Мы входим в каталог стартовых скриптов (*/etc/init.d*) перечисля-



**Бойтесь InitNG? Сокращение числа впаданий в спячку почти не хуже, причем безопаснее!**

## ТОНКАЯ НАСТРОЙКА СТАРТОВЫХ СКРИПТОВ

Сложность совета: продвинутый

**Большинство советов этого урока годятся практически для всех дистрибутивов и типов установки** (настольная, сервер и т.д.), но есть изменения, которые применимы только для определенных конфигураций. Если вы досконально знаете устройство своей системы и все процессы, необходимые для вашей работы, вы можете просеять стартовые скрипты сквозь сито и отфильтровать ненужные строки.

Выполните команду **top** в терминале после загрузки системы и отметьте все процессы, которые вам явно не нужны. Затем в каталоге */etc/init.d* при помощи *grep* вы можете найти строки, их запускающие, и превратить эти строки в комментарии.

Довольно трудно определить, что происходит во время загрузки и какое время отнимает каждая часть этого процесса, особенно если показана только графическая заставка или некоторые строки в текстовом режиме мелькают слишком быстро. В каталоге */var/log* найдите файлы **messages**, **syslog** и **daemon.log** с подробным описанием происшедшего во время загрузки (каждая запись в файле имеет отметку времени, которая поможет определить виновников задержки).

Если ваша система в полной гармонии с вами и вы не планируете подключение нового оборудования или драйверов ядра, ликвидация стадии загрузки *depmod* сэкономит еще пару секунд. Это утилита, которая генерирует зависимости модулей ядра; ее незачем запускать при каждой загрузке, если вы не добавляете драйвера. Выполните поиск *depmod* в */etc/init.d* (например, **grep depmod /etc/init.d**) и закомментируйте строки, где она запускается, поставив символ '#' в самом начале.

Отследите перекрытия функций в скриптах. Например, в Ubuntu скрипт *checkroot.sh* содержит секции, связанные с управлением питанием (особенно заметные, если система работает от блока питания). Но если у вас отключены APM/ACPI, то и не нужно выполнять проверки, связанные с питанием, можете их закомментировать. Здорово! Такие пересечения часто встречаются в стартовых скриптах дистрибутива, так что полезно пройти по ним, определяя назначение всего запускаемого и оценивая его необходимость.

Вы можете обнаружить, что некоторые части скрипта или служба замедляют загрузку, но их функциональность важна и вы не хотите

## ИСПРАВЛЯЕМ ПРОБЛЕМЫ

Как утверждалось в начале этого урока, сохранение резервной копии вашего каталога **/etc** является хорошей идеей на случай, если что-то пойдет не так. Так как все изменения вы производите в **/etc**, наличие его неизменяемой копии в **/old-etc** жизненно важно для исправления проблем.

А вдруг вы понаделаете такого, что система вообще не загрузится? Не бойтесь: вам помогут Live-дистрибутивы! Для восстановления вашего исходного каталога **/etc** вы можете взять практически любой Live-дистрибутив, предоставляющий доступ к

терминалу или командной строке — даже Games Knoppix с диска-приложения к этому номеру. Загрузите Live-дистрибутив с вашего DVD, откройте терминал и смонтируйте ваш жесткий диск командами

```
mkdir /mydrive
```

```
mount -t ext3 /dev/hda1 /mydrive
```

Ваш диск станет доступен в каталоге **/mydrive**. (Возможно, придется заменить **'ext3'** на **'reiserfs'** если вы используете файловую систему Reiser, а **'hda1'** может стать **'hda2'**, если вы

установили Linux во второй раздел, **'hda3'** — если в третий, и т.д.) Затем введите

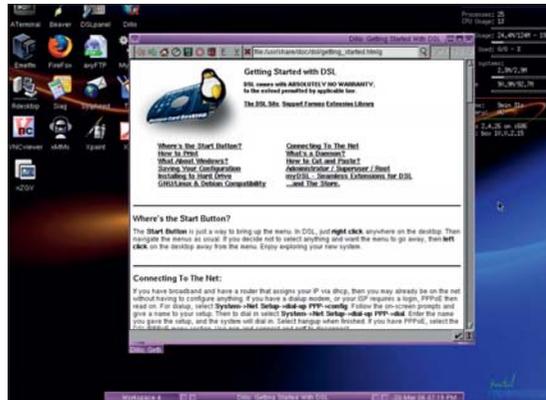
```
mv /mydrive/etc /mydrive/broken-etc
```

```
mv /mydrive/old-etc /mydrive-etc
```

```
umount /mydrive
```

и перезагрузитесь. Ваш исходный **/etc** восстановлен, а проблемный **/etc**, сорвавший загрузку, помещен в **/broken-etc**.

отключать их навсегда. Например, загрузка тормозится, пока сетевая карта пытается получить DHCP-отклик от сервера — но не отказываться же напрочь от сети! Для решения подобных проблем просмотрите каталоги **/etc/init.d** и **/etc/dhcp3** и определите, где вызывается DHCP; затем попробуйте добавить амперсанд (&) после строки его вызова, чтобы запускать его в фоновом режиме. Перезагрузитесь, и, надеюсь, ваша карта будет сконфигурирована в фоновом режиме и позволит другим стартовым скриптам выполняться беспрепятственно!



Live-дистрибутивы вроде **Damn Small Linux** (см. стр. 15) — это реаниматоры, помогающие восстановить исходные скрипты, если что-то пошло не так.

## ЗАМЕНА СТАРТОВЫХ СКРИПТОВ

Если вы прочли первый шаг, вы видели, что процесс загрузки Linux включает в себя программы фоновых служб, неинтерактивно настраивающих машину. Однако здесь есть проблема: некоторые скрипты зависают от других, и при попытке что-то изменить возникает путаница. Например, скрипт, запускающий доступ к файлам и принтерам Windows через *Samba*, должен убедиться, что сетевая карта инициализирована другим скриптом. Поэтому в настоящее время большинство дистрибутивов Linux запускают скрипты один за другим, в строго определенном порядке, чтобы все необходимое наверняка было запущено.

Но это же жутко неэффективно: почему бы не выполнять несколько скриптов сразу? К примеру, пока сетевой скрипт ожидает ответа DHCP, можно заняться звуковой картой.

Этот подход называется параллельной инициализацией, в противовес последовательному запуску инициализационных скриптов, используемому практически в каждом дистрибутиве. При запуске служб все еще нужен порядок — но он предусматривается таким, чтобы ЦП и жесткий диск подолгу не простаивали. Распараллеливание может сократить время загрузки больше чем вдвое.

*InitNG* предоставляет такую возможность. Этот проект описан в рубрике *HotPicks*, и хотя он все еще находится в срединной стадии развития, многие давние пользователи Linux говорят, что он действительно помогает. *InitNG* заменяет ваши стартовые скрипты параллельными эквивалентами, чтобы ресурсы не оставались праздными, и это приводит к умопомрачительно быстрой загрузке. Вы можете обнаружить это приложение в разделе **Система** на нашем диске или найти последнюю версию по адресу <http://InitNG.thinktux.net>. Скачайте tar-архив и установите, как показано ниже (понадобится перезагрузка):

```
tar xfv InitNG-0.5.5.tar.gz
```

```
./configure && make install
```

Отредактируйте конфигурационный файл загрузчика (т.е. **/boot/grub/menu.lst** или **/etc/lilo.conf**), добавив **init=/sbin/initNG** к строке образа ядра. Это обеспечит старт вашего дистрибутива со специальной *InitNG* версией *init* (основной процесс Linux), который в свою очередь начнет загружать службы. Строка в настройках вашего загрузчика должна походить на

```
kernel /boot/vmlinuz-2.6.10-5-686 root=/dev/hda1 ro
```

```
init=/sbin/initNG
```

(Если вы используете *Lilo*, после изменений запустите **/sbin/lilo**, для обновления загрузчика). Теперь, перекреститесь и перезагрузитесь; если все верно, то появится сообщение о загрузке. Внимательно следите за всеми сообщениями об ошибках и предупреждениями: *InitNG* с каждым выпуском становится все более гибким, но пока не справляется с особо трудными службами. Если вы обнаружите, что что-то не запустилось, введите **ngc -s** и посмотрите, что предположительно запущено и работает. Чтобы служба запускалась при каждой загрузке, используйте **ng-update add type/service**

За подробной информацией обратитесь к документации на прилагаемом диске. Утилита *InitNG* имеет много опций для конфигурирования и администрирования, так что она заслуживает опробования всех ее возможностей. Если при старте возникают проблемы, вы можете удалить строку **init=/sbin/initNG** в файле настройке загрузчика (см. выше), а также задать вопросы на полезном форуме *InitNG* по адресу <http://forum.InitNG.thinktux.net>.

Теперь ваш ПК должен быть быстрее ниндзя — ура! **LXF**

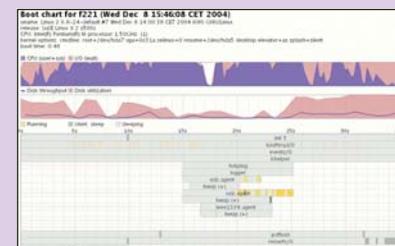
Сложность совета: продвинутый

## ДРУГИЕ РЕСУРСЫ

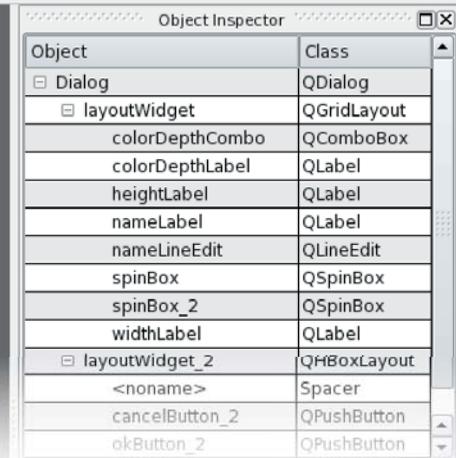
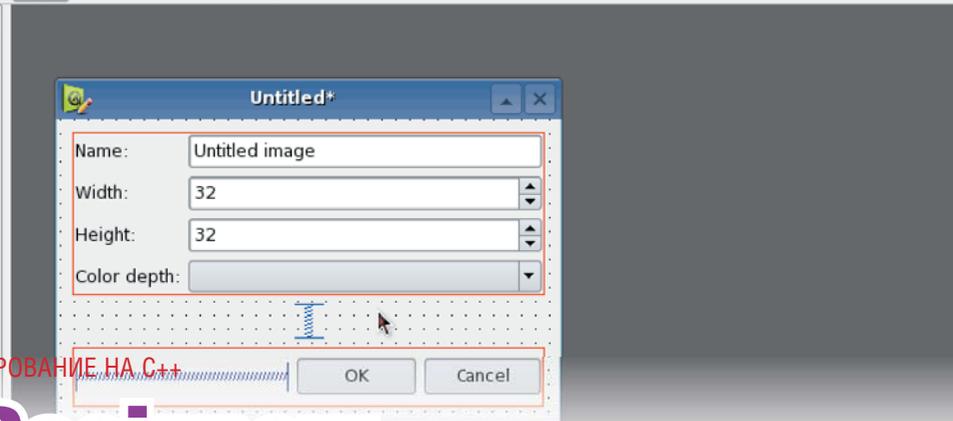
Вы можете оценить эффект от применения описанных шагов при помощи простого секундомера, но он не покажет вам, где остались узкие места. Полезной утилитой, позволяющей получить больше информации о процессах загрузки, является *Bootchart* ([www.bootchart.org](http://www.bootchart.org)), генерирующая графики ваших загрузочных процессов, где показана степень использования процессора и жесткого диска. После установки *Bootchart* вы сможете отследить провалы и паузы в ваших стартовых скриптах, сравнивая графики при каждой загрузке. На web-сайте или в разделе **Системы/Диаграммы** загрузки на нашем диске-приложении имеются исходные тексты и документация.

Ну вот, все закончилось! Вы нашли другие пути снижения времени загрузки? Или придумали хитрый трюк, экономящий еще несколько секунд? А может, зашли

в тупик, и из-за сделанных изменений время загрузки на самом деле увеличилось? Присоединяйтесь к нашему форуму по адресу [www.linuxformat.ru](http://www.linuxformat.ru)



*Bootchart* представляет ход загрузки вашей системы графически.



ПРОГРАММИРОВАНИЕ НА C++

# Qt Designer:

## КРАСИВЫЙ ИНТЕРФЕЙС ОДНОЙ МЫШЬЮ

**ЧАСТЬ 3** Компьютер был придуман для того, чтобы избавить человечество от рутины, так почему вы должны корпеть над кодом, создавая бесконечные кнопки, списки и поля ввода?

Андрей Боровский знает более легкий путь.

### МЕСЯЦ НАЗАД

Мы занимались конструированием графических интерфейсов вручную и поняли, насколько это утомительно.



В прошлой статье мы конструировали графические интерфейсы вручную. Преимуществами этого подхода являются свобода обращения с визуальными компонентами и достижение глубокого понимания механизмов работы Qt. Недостаток у ручного проектирования только один – высокая трудоемкость. Процесс проектирования графического интерфейса можно упростить и ускорить с помощью среды визуального проектирования *Qt Designer*. Следует сразу подчеркнуть, что *Qt Designer* не является полноценной IDE, такой как *Borland C++Builder* или *MS Visual C++*. *Qt Designer* не управляет процессами компиляции или отладки, а только генерирует «заготовки» исходных текстов приложения (заголовочные файлы C++ и специальные файлы с расширением **ui**). В качестве примера проектирования с помощью *Qt Designer* мы напишем простую программу-просмотрщик графических файлов.

Запустите *Qt Designer* командой **designer** или через главное меню оболочки KDE вашего дистрибутива. В главном окне выберите команду File/New. В открывшемся окне укажите тип интерфейса. Для нашей задачи подойдет интерфейс Dialog (напомню, что окно QDialog является основой интерфейса простых приложений Qt). В результате будет создана новая форма – класс **Form1** (естественно, потомок класса **QDialog**). Внешний вид *Qt Designer* в режиме редактирования формы напоминает другие визуальные редакторы (например, Delphi или Kylix). Для размещения визуального элемента в окне нужно «перетащить» мышью соответствующую иконку с панели компонентов на форму. На специальной панели можно установить значения свойств объектов, представляющих

элементы интерфейса. С помощью этой панели можно, например, задать надписи на кнопках и в заголовках окон или установить политику изменения размеров. Окно создаваемой нами программы (рис. 1) содержит три визуальных элемента: кнопку Открыть (объект **pushButton1** класса **QPushButton**), с помощью которой загружается файл, элемент **Pixmap Label** ( **pixmapLabel1**) для вывода изображения и элемент **Text Label** ( **textLabel1**) для вывода имени файла. Кнопка и

текстовая метка объединены элементом **Layout** (менеджером компоновки, с которыми мы познакомились в прошлый раз). В режиме редактирования формы область действия менеджера компоновки обводится красной каймой. Для того, чтобы связать одним менеджером компоновки несколько элементов интерфейса, необходимо выделить их (щелкая по ним мышью и удерживая нажатой клавишу **Shift**), а затем в контекстном меню выбрать одну из команд **Lay out**. Под строкой с кнопкой и меткой расположен визуальный элемент **Pixmap Label** (выделен на рисунке). Чтобы скомпоновать **Pixmap Label** и верхнюю (уже скомпонованную) строку элементов управления, мы щелкаем правой кнопкой мыши по свободному месту в окне формы и в контекстном меню выбираем команду **Lay Out Vertically** (будучи применена к форме, эта команда связывает все расположенные на ней группы элементов единым менеджером компоновки). В нашем случае с формой будет связан вертикальный менеджер компоновки, управляющий расположением верхней строки элементов управления и элементом **Pixmap Label**. В примере из предыдущей статьи мы добавляли специальную «распорку» с помощью метода **addStretch()** менеджера компоновки. Требовалось это для того, чтобы не допустить «расползания» элементов интерфейса. В нашем примере в распорке нет необходимости, так как элемент **Pixmap Label** занимает все свободное пространство окна. В тех ситуациях, когда распорка нужна, к нашим услугам визуальный элемент **Spacer** (объект класса **QSpacerItem**). В режиме редактирования формы этот элемент обозначается пружинкой. Логика работы интерфейса нашей программы очень проста: в ответ на щелчок по кнопке Открыть открывается окно выбора файла, после чего выбранное изображение выводится в окно **Pixmap Label**.

Проектирование интерфейса на этом завершено, и мы можем приступить к написанию кода. Начнем с переменных, необходимых нашей программе. Программирование сводится, в основном, к созданию оснастки (переменных и методов) класса **Form1**.

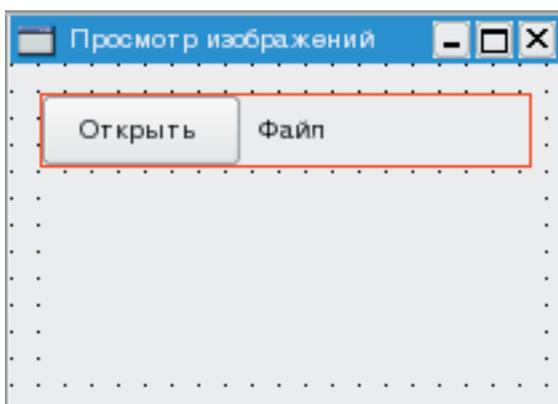
Начнем с переменных. В нашей программе нужно реализовать упомянутое выше диалоговое окно выбора файла. Мы воспользуемся стандартным окном, реализованным в классе **QFileDialog**, но для этого нам надо, во-первых, включить в проект заголовочный файл **qfiledialog.h**, а, во-вторых, объявить переменную типа **QFileDialog**. Чтобы включить в проект заголовочный файл **qfiledialog.h**, в окне **Object Explorer** переходим на вкладку **Members**, щелкаем правой кнопкой мыши по названию раздела **Includes (in Declaration)** (включения в заголовочный файл) и



### НА ДИСКЕ

• Исходные тексты примеров статьи.

Рис. 1. Окно формы в режиме редактирования.



вводим строку **qfiledialog.h**. Точно так же введем имя еще одного заголовочного файла, который понадобится нашей программе, — **qpixmap.h**. В раздел **Includes (in Implementation)** (включения в файл реализации) добавим заголовочный файл **qstring.h**, он тоже нам пригодится. В том же окне **Object Explorer** мы объявляем новую переменную-член класса **Form1**. Для этого на вкладке **Objects** выбираем **Form1**, переходим на вкладку **Members**, щелкаем правой кнопкой мыши по имени подраздела **Private** в разделе **Class Variables** (переменные класса) и в открывшемся контекстном меню выбираем (единственную в этот момент) команду **New**. В строке редактора переменных вводим

```
QFileDialog * OpenFileDialog;
```

Все сделанные изменения отображаются в окне **Object Explorer** (рис. 2), (по умолчанию это окно «припарковано» в окне **Qt Designer**). Переменную **OpenFileDialog** нужно инициализировать. Обычно инициализация выполняется в конструкторе, однако **Qt Designer** не позволяет редактировать конструктор **Form1** напрямую. Вместо этого мы должны определить специальную функцию-метод **init()**, которую будет вызывать конструктор класса. Именно в метод **init()** следует помещать весь вводимый вручную код инициализации. Заготовку метода **init()** мы создадим в окне **Object Explorer** в разделе **Functions/private**, точно так же, как мы объявляли переменную **OpenFileDialog** (то есть выберем команду **New** из контекстного меню и в открывшемся окне редактора введем строку **“init()”**). После этого должно открыться окно редактора исходных текстов, в котором следует вести код функции **init()**:

```
void Form1::init()
{
    OpenFileDialog = new QFileDialog(0);
    OpenFileDialog->setCaption(trUtf8("Открыть"));
}
```

Диалоговое окно, создаваемое объектом **QFileDialog**, может использоваться как для открытия, так и для сохранения файла (точнее, для выбора имени файла для открытия или сохранения). С помощью метода **setCaption** мы устанавливаем текст заголовка данного окна.

У метода **init** есть метод-напарник **destroy**, который вызывается деструктором объекта формы. Определять метод **destroy** в принципе не обязательно, но желательно с точки зрения правил хорошего тона:

```
void Form1::destroy()
{
    delete OpenFileDialog;
}
```

Теперь определим метод **loadFile()**, который и будет открывать графический файл для просмотра. Новый метод определяется точно также, как метод **init()**, только в строке ввода редактора методов нужно ввести **«loadFile(const QString fn)»**.

```
void Form1::loadFile(const QString fn)
{
    textLabel1->setText(trUtf8("Файл: ") + fn);
    QPixmap pm;
    if (pm.load(fn))
        pixmapLabel1->setPixmap(pm);
    else
        pixmapLabel1->setText(trUtf8("Неизвестный формат"));
}
```

Параметр **fn** служит для передачи имени файла, который нужно открыть. Для загрузки графических файлов мы используем объект класса **QPixmap**. Этот класс поддерживает работу с множеством популярных и не очень популярных графических форматов (конкретный список форматов зависит от условий компиляции **Qt**). Метод **load** объекта **QPixmap** пытается загрузить файл и возвращает значение **TRUE**, если файл загружен успешно, и **FALSE** в случае неудачи, вызванной, например, попыткой загрузить изображение в неподдерживаемом формате. Далее мы отображаем изображение с помощью метода **setPixmap** объекта **pixmapLabel1** или выводим сообщение об ошибке с помощью метода **setText** того же объекта. Возможно, это покажется неожиданным, но объект **pixmapLabel1** (элемент **Pixmap Label** с панели **Tool Box**) принадлежит тому же классу **QLabel**, что и текстовая метка.

Для реализации логики работы нашего интерфейса необходимо определить в классе **Form1** слот, который бы обрабатывал событие **clicked()** объекта **pushButton1**. **Qt Designer** предоставляет нам несколько средств для создания и редактирования слотов. Панель редактирования свойств позволяет создавать обработчики сигналов, посылаемых объектами. Если в этой панели выделить объект **pushButton1**, то на вкладке **Signal Handlers** появится список сигналов, эмитируемых этим объектом. Среди прочего мы находим сигнал **clicked**. Щелчок мышью по имени сигнала создает обработчик — новый метод класса **Form1** **pushButton1\_clicked()**. Далее открывается знакомое окно редактора, в котором можно ввести исходный текст обработчика сигнала, что мы и делаем:

```
void Form1::pushButton1_clicked()
{
    if(OpenFileDialog->exec() == QDialog::Accepted )
        loadFile(OpenFileDialog->selectedFile());
}
```

Метод **exec()** класса **OpenFileDialog** выводит на экран диалоговое окно и приостанавливает исполнение основного потока программы до тех пор, пока это окно не будет закрыто. Возвращаемое методом значение указывает, была ли нажата кнопка **OK (QDialog::Accepted)**. Имя выбранного пользователем файла можно получить с помощью метода **selectedFile()**. Новый слот автоматически связывается с соответствующим ему сигналом, однако сигнал и слот можно связывать и вручную, например, если один и тот же слот должен быть связан с несколькими сигналами. Для установки дополнительных связей сигнал-слот можно использовать специальное окно **View and Edit Connections**, которое открывается по команде меню **Edit Connections...** Это же окно можно вызывать и более простым способом: нажать клавишу **F3** (курсор приобретет вид перекрестья), навести курсор на нужный элемент формы и щелкнуть левой кнопкой мыши. При этом открывается окно **View and Edit Connections**, в котором для выбранного элемента уже подготовлена строка сигнал-слот.

На этом разработка программы закончена. Сохраним результаты работы в отдельной директории. У нас появилось два файла — **form1.ui** и **form1.ui.h**. Первый из этих файлов представляет собой описание интерфейса создаваемой программы. В документации к **Qt Designer** можно найти подробное описание формата **ui**-файлов. Помимо самого **Qt Designer**, эти файлы можно также открывать в специальной программе просмотра. Файл **form1.ui.h** содержит весь код, который мы ввели вручную, то есть слот **pushButton1\_clicked()**, а также методы **init()**, **destroy()** и **loadFile()**.

Наша следующая задача заключается в том, чтобы сгенерировать заголовочный файл и файл реализации класса **Form1**, используя описание класса из файла **form1.ui**. Генерацию файлов осуществляет утилита командной строки **uic**. Для того, чтобы получить заголовочный файл **form1.h** даем в окне терминала команду:

```
uic form1.ui -o form1.h
```

По умолчанию программа **uic** выводит результаты своей работы в стандартный поток вывода, то есть, на терминал, но с помощью ключа **-o** мы указываем программе, что вместо терминала данные нужно записывать в файл с заданным именем. Для генерации файла **form1.cpp**, ссылающегося на **form1.h**, командуем:

```
uic -impl form1.h form1.ui -o form1.cpp
```

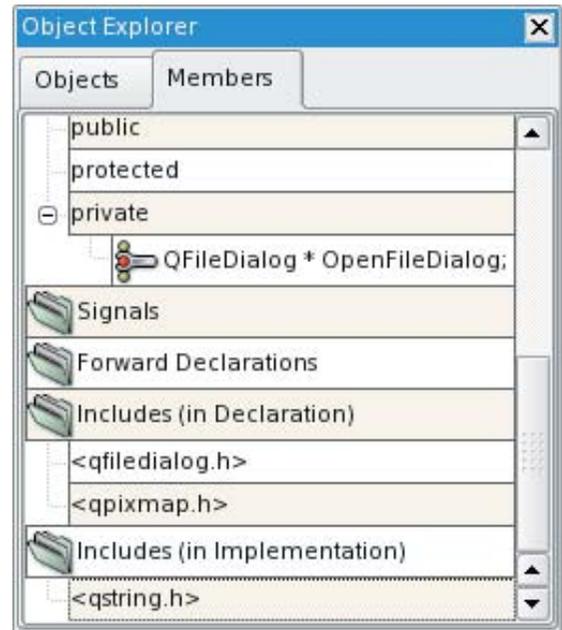


Рис. 2.  
Окно **Object Explorer**.



Команды, использующие *uic*, приводятся скорее в целях общего развития, поскольку нет никакой необходимости вызывать *uic* вручную. Утилита *qmake* в любом случае добавит в создаваемый *make*-файл вызов *uic* для генерации файлов *form1.h* и *form1.cpp*. Поскольку файлы *form1.cpp* и *form1.h* генерируются автоматически (в процессе сборки программы!), не следует пытаться вносить в них какие-либо изменения. Для ручного редактирования доступен только файл *form1.ui.h*, который включается в *form1.cpp* директивой *#include*. Как обычно, нам осталось написать только функцию *main*:

```
#include <qapplication.h>
#include "form1.h"
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    Form1 fm(0);
    app.setMainWidget(&fm);
    fm.show();
    return app.exec();
}
```

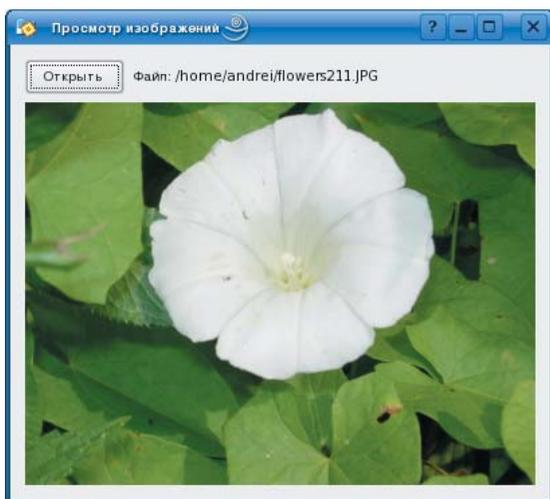


Рис. 3. Наша программа в работе.

Далее мы создаем файл проекта и *make*-файл с помощью традиционных вызовов утилиты *qmake*. На платформе Linux наша программа справится с отображением файлов BMP, JPEG (рис. 3), GIF, PNG и многими другими.

Программа из нашего примера создает только одно окно, но ничто не мешает нам создать множество разных окон (например, нестандартных диалогов) и связать их в один проект. При запуске *Qt Designer* в окне *New/Open* можно выбрать элемент *Project*, и тогда файл проекта программы (файл с расширением *.pro*) будет сгенерирован самим *Qt Designer*.

Размещение кода, открывающего графический файл, в отдельной функции позволяет нам безболезненно наращивать интерфейс программы. Допустим, мы хотим снабдить нашу программу возможностью открывать файлы с помощью **Drag and Drop**. Сделать это нам будет несложно. Добавим в метод *init()* строку:

```
setAcceptDrops(TRUE);
```

Метод *setAcceptDrops()* позволяет превратить окно приложения в приемник перетаскиваемых объектов. Далее нам нужно перекрыть два метода-обработчика событий класса *Form1*. В разделе *protected* создаем методы

```
dragEnterEvent( QDragEnterEvent * event )
```

и

```
dropEvent( QDropEvent * event )
```

Событие **DragEnterEvent** происходит, когда перетаскиваемый объект оказывается в области окна программы. Обработчик этого события должен сообщить системе, может ли программа принять перетаскиваемый объект. Наша программа принимает все объекты, содержащие данные в формате URL (универсальные идентификаторы ресурсов). Создавая обработчики событий **DragEnterEvent** и **DropEvent** следует помнить, что объекты **Drag and Drop** могут содержать несколько блоков данных в разных форматах. Код нашего обработчика состоит из одной строки:

```
void Form1::dragEnterEvent( QDragEnterEvent * event )
{
    event->accept(QUriDrag::canDecode(event));
}
```

Класс *QDragEnterEvent* содержит все необходимые сведения о событии, включая указатель на область данных перетаскиваемого объекта. Класс *QUriDrag* специально предназначенный для обра-

ботки перетаскиваемых объектов, содержащих URI, объявлен в файле *qdragobject.h* (а не *quridrag.h*, как можно было бы ожидать). Помимо класса *QUriDrag* в том же файле объявлены классы для работы с перетаскиваемым текстом и изображениями. Статический метод *QUriDrag::canDecode()* возвращает TRUE, если объект содержит данные в формате URI и FALSE в противном случае. Это значение передается методу объекта-события *accept()*, который сообщает системе, готова ли программа принять перетаскиваемый объект. Если теперь пользователь отпустит левую кнопку мыши, программе будет передано сообщение *DropEvent*, которое мы тоже должны обработать:

```
void Form1::dropEvent( QDropEvent * event )
{
    QStringList S;
    if (QUriDrag::decode(event, S))
        openFile(QUriDrag::uriToLocalFile(S.getFirst()));
}
```

Статический метод *decode* класса *QUriDrag* преобразует набор URI в набор строк с именами файлов (в общем случае пользователь может перетаскивать сразу несколько файлов) и заполняет ими объект класса *QStringList*. Наше приложение максимально упрощено, а потому перетаскивание файлов будет работать не всегда. Помимо файлового менеджера KDE, источником файлов для перетаскивания в программу в ее текущем виде может служить *Nautilus*, а вот перетаскивать имена файлов из *Firefox* и *Mozilla* не удастся. Дело тут вовсе не в несовместимости *Drag and Drop* – и *Qt*, и *Mozilla* на платформе X11 поддерживают XDND (стандартный протокол *Drag and Drop* в X Window System). Однако для передачи ссылок на файлы разные программы используют разные форматы MIME. *Mozilla* использует «фирменные» типы MIME *text/x-moz-url* и *\_NETSCAPE\_URL*, *Konqueror* – *text/uri-list*. Поскольку за устрашающими названиями скрывается обычный текстовый формат, написать перекодировщик совсем не сложно.

В заключение обзора *Qt Designer* коснемся вопроса интернационализации. В исходном тексте нашей программы мы везде добавляем вызов метода *trUtf8*. Для исходных текстов, которые *Qt Designer* генерирует автоматически, этот вызов добавляется автоматически, так что все, что нам остается сделать, – создать файлы ресурсов интернационализации и добавить код, выполняющий загрузку этих файлов и установку транслятора.

## Взаимодействие с оконной системой и подключение сторонних библиотек

Интерфейсы библиотеки *Qt* предоставляют практически всё, что только может понадобиться разработчику графического приложения, а потому у него, как правило, нет необходимости напрямую обращаться к функциям оконной системы. Если же такая необходимость возникает, следует помнить, что код, обращающийся к оконной системе напрямую, практически всегда теряет переносимость. Рассмотрим два фрагмента кода. Первый генерирует снимок формы и сохраняет его в файле *scr.png*. Он должен выполняться одинаково на всех платформах:

```
#include <qpixmap.h>
#include <qnamespace.h>
...
QPixmap pm = QPixmap::grabWindow(this->winId());
pm.save("scr.png", "PNG");
```

Метод *winId()* возвращает идентификатор окна визуального элемента. В Windows это переменная типа *Handle*, в X Window – переменная типа *Window*. Второй фрагмент вызывает функцию X Window напрямую. Для этого вызова требуется не только идентификатор окна, но и идентификатор X-дисплея:

```
#include <X11/Xlib.h>
...
Window w = this->winId()
Display * d = this->x11Display();
XClearWindow(d, w);
...
```

Метод `X11Display` определен в классе `QPaintDevice`, который является предком классов `QWidget`, `QPixmap`, `QPicture` и `QPrinter`. Как вы догадываетесь, код, использующий этот метод, не будет работать в среде Windows GDI.

Сколь бы всеобъемлющей ни была система классов *Qt*, не ею единой живы наши приложения. В реальном мире нам неизбежно придется добавлять в программы ссылки на другие библиотеки. С подключением к проекту сторонних заголовочных файлов проблем у нас не возникает, а как быть со ссылками на сами двоичные файлы библиотек? Можно, конечно, отредактировать вручную **make**-файл, сгенерированный командой **qmake**, однако в процессе отладки проекта этот файл перезаписывается многократно. Более простой и естественный способ заключается в использовании ключей самой **qmake**. Например, если нам нужно связать наше приложение с библиотеками *OpenGL* `libGL` и `libGLU`, мы даем такую команду:

```
qmake --after LIBS="-IGL -IGLU"
```

В автоматически сгенерированном **make**-файле мы увидим строку наподобие

```
LIBS = $(SUBLIBS) -L/usr/lib/ -L$(QDIR)/lib/ -L/usr/X11R6/lib/ -IGL -IGLU -lQt-mt -lXext -lX11 -lm
```

То есть в переменную `LIBS`, содержащую список подключаемых библиотек, были добавлены ключи `-IGL` и `-IGLU`. Ключ `--after` указывает, что переменная `LIBS` должна быть обработана после завершения синтаксического разбора файлов (например, файла `pro`) нашей программы. Если бы мы не указывали ключ `--after`, а скомандовали, например:

```
qmake --makefile LIBS="-IGL -IGLU"
```

переменные были бы обработаны до синтаксического разбора файлов (режим по умолчанию). В нашем случае порядок разбора не имеет значения. Ну, а если вы не желаете указывать библиотеки при каждом вызове **qmake**, можете скомандовать так:

```
qmake --project LIBS="-IGL -IGLU"
```

Теперь ключи компоновщика будут добавлены в файл проекта, из которого генерируется **make**-файл.

## Отладка

В заключение хотелось бы коснуться еще одной, важной для программистов, темы, а именно – отладки программ. Было бы странно, если бы такая развитая система разработки не предоставляла никаких дополнительных средств, упрощающих поиск и исправление ошибок. К наиболее простым и часто используемым техникам отладки следует отнести расстановку в программе функций, выводящих отладочные сведения о значении определенных переменных, о выполнении определенных условий или просто о том, что данная точка программы была выполнена. Отладочные сообщения делятся в *Qt* по трем категориям важности: обычные сообщения (**debug**), к которым относятся, например, сообщения о прохождении определенного блока программы, предупреждающие сообщения (**warning**), каковыми являются сообщения о некритичных ошибках и критические сообщения (**fatal**), которые помимо вывода информации об ошибке завершают работу программы. Соответственно трем типам сообщений, в *Qt* определено три глобальных функции: **QDebug()**, **QWarning()** и **QFatal()**. Формат вызова всех трех функций подобен формату вызова **printf(3)**. По умолчанию сообщения, создаваемые с помощью этих трех функций, выводятся в стандартный поток ошибок, однако вывод можно перенаправить с помощью функции **qInstallMsgHandler()**. В качестве примера добавим функции отладки к многострадальному приложению просмотра изображений. Прежде всего, внесем изменения в тот модуль нашей программы, в котором определена функция **main**:

```
#include <qapplication.h>
#include <qmessagebox.h>
#include <qstring.h>
#include "form1.h"
void debugMsg(QtMsgType type, const char *msg)
{
    switch (type) {
```

```
        case QtDebugMsg:
            QMessageBox::information(0, QWidget::trUtf8("Информация"),
                QString(msg));
            break;
        case QtWarningMsg:
            QMessageBox::warning(0, QWidget::trUtf8("Предупреждение"),
                QString(msg));
            break;
        case QtFatalMsg:
            QMessageBox::critical(0, QWidget::trUtf8("Критическая ошибка"),
                QString(msg));
            abort();
    }
}
int main(int argc, char *argv[])
{
    qInstallMsgHandler(debugMsg);
    QApplication app(argc, argv);
    Form1 fm(0);
    app.setMainWidget(&fm);
    fm.show();
    return app.exec();
}
```

В новом варианте программы мы, первым делом, определяем функцию **debugMsg()**. Эта функция осуществляет вывод отладочных сообщений. По умолчанию сообщения выводятся на консоль. Наш же вариант функции создает для каждого отладочного сообщения диалоговое окно `QMessageBox`, оформленное в соответствии с типом сообщения. Тип (`QtDebugMsg`, `QtWarningMsg`, `QtFatalMsg`) передается в параметре **type**, а текст сообщения – в параметре **msg**. В функции **main** мы вызываем функцию **qInstallMsgHandler()**, которая и устанавливает **debugMsg** в качестве стандартного обработчика сообщений об ошибках. Отметим еще раз, что функция **debugMsg** вызывается средой *Qt* в случае возникновения ошибки *Qt* (именно *Qt*, а не системной) и вызывать ее напрямую не следует. Теперь добавим отладочное сообщение в тело метода

```
void Form1::dragEnterEvent( QDragEnterEvent * event )
{
    if(QUriDrag::canDecode(event))
        event->accept(TRUE);
    else
        qWarning(event->format());
}
```

Теперь, если над окном приложения перетаскивается объект в неподдерживаемом формате, на экран будет выведено красивое сообщение (рис. 4).

Действительно, если ошибки в программах неизбежны, то почему бы не сделать их красивыми? Среди других элементов *Qt*, предназначенных для отладки, отметим еще два макроса:

**Q\_ASSERT()** и **Q\_CHECK\_PTR()**. Первый макрос выводит на консоль предупреждающее сообщение, если переданный ему аргумент типа **bool** имеет значение **FALSE**. Макрос упрощает **Q\_CHECK\_PTR()** проверку указателя, возвращаемого **new** и **malloc(3)**.

Мы рассмотрели все основные аспекты программирования приложений *Qt*. Пора переходить к программированию KDE, ради которого все и затевалось.

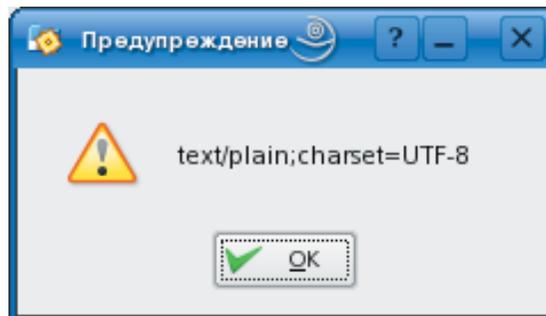
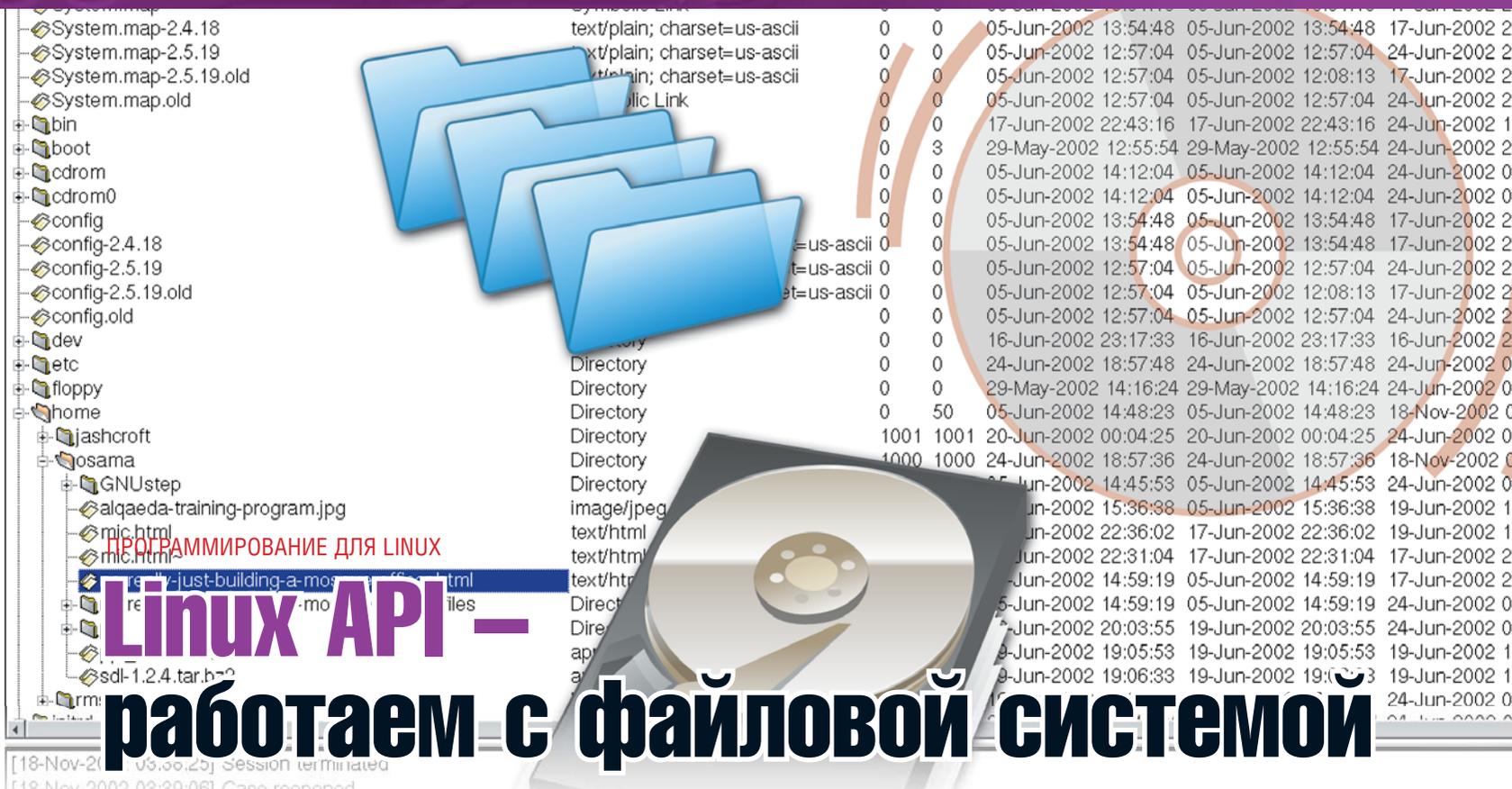


Рис. 4.  
Окно Object Explorer.

## ЧЕРЕЗ МЕСЯЦ

Мы узнаем, чем программы KDE отличаются от программ Qt.



**ЧАСТЬ 1** Как известно, в Unix все есть файл.  
 Андрей Боровский покажет, как можно извлечь из этого выгоду.



Исходные тексты примеров статьи.

Управление хранением данных на диске – одна из самых важных задач любой ОС, настолько важных, что система DOS так и называлась – дисковая операционная система. Вероятно, читатель этой статьи, желающий стать Linux-программистом, уже знает, как устроена файловая система Linux с точки зрения пользователя. Мы рассмотрим эту систему с точки зрения программиста. Один из основополагающих принципов Unix/Linux – everything is a file – в вольном переводе означает: «файлы – наше все». Linux отображает в виде файлов не только сами файлы, но и различные типы установленных в системе устройств, а также некоторые структуры данных, создаваемые в ходе работы. Возможность управлять устройством как файлом существенно упрощает решение некоторых задач, однако во многих случаях файловый интерфейс – не самый удобный, и параллельно с ним существуют другие типы программных интерфейсов для работы с устройствами. Стандартная библиотека Linux glibc предоставляет нам полный набор функций для работы с файлами, а точнее, даже два полных набора – системные вызовы Linux и реализованные на их основе функции стандартной библиотеки C (прежде, чем приступить к изучению примеров из этой статьи, рекомендуется ознакомиться с документацией к этой библиотеке). Может показаться, что системные вызовы более эффективны, чем библиотечные функции, но это не так. Как правило, чем меньше системных вызовов, тем быстрее будет работать ваша программа. Дело в том, что для выполнения своей работы системные вызовы переключают систему в режим ядра, а затем возвращаются в пользовательский режим. В прежних версиях переключение осуществлялось с помощью прерывания int 80h, в новых, где указана архитектура i686, – с помощью специальной процессорной команды, появившейся в Pentium II [1]. Переключение между режимами отнимает сравнительно много процессорного времени, поэтому библиотечные функции минимизируют количество системных вызовов, заставляя каждый такой вызов выполнять как можно больше полезной работы. И системные вызовы, и функции стандартной библиотеки C экспортируются библиотекой glibc. Функции для работы с файлами open, close, read, write и им подобные, использующие дескрипторы файлов, представляют собой обертки для соответствующих системных вызовов. Функции fopen, fclose, fread,

fwrite, fseek и другие, работающие со структурой FILE, – являются частью стандартной библиотеки. Особую роль среди системных вызовов играет вызов ioctl(2), являющийся, фактически, универсальным средством управления устройствами, представленными в виде файлов (за универсальность вызовов ioctl() называют швейцарским армейским ножом). Первым параметром функции ioctl() должен быть дескриптор открытого файла. Второй параметр – запрос или команда. Помимо этого, при вызове ioctl могут передаваться дополнительные параметры, число и типы которых зависят от значения второго параметра функции. Функция ioctl() возвращает результаты вызова в переменных, переданных по ссылке или через стек (в виде результата функции). Следует отметить, что в последнее время среди разработчиков Linux наметилась тенденция на отказ от использования ioctl(). Мы не будем останавливаться на классических примерах работы с файлами, а рассмотрим несколько примеров, специфичных именно для Unix/Linux.

**Управление файлами устройств**

В качестве примера использования системных вызовов для работы с файлами устройств мы напишем программу, копирующую аудиоданные с audio-CD в wav-файлы (так называемый «риппер»). Файлы устройств, как известно, расположены в директории /dev/. В этой директории мы найдем и устройство /dev/cdrom, представляющее первый из установленных в вашей системе CD-приводов. Прежде, чем мы приступим к написанию программы-риппера, имеет смысл обратить внимание на врезку, посвященную организации Audio CD. Работа с CD-ROM с помощью устройства /dev/cdrom обычно выполняется по следующему сценарию: открытие файла устройства, настройка параметров с помощью ioctl(2), чтение (запись) данных, закрытие устройства. Полный текст программы вы найдете на прилагаемом к журналу диске; здесь мы рассмотрим только самые интересные части, имеющие отношение к управлению устройствами-файлами. Текст программы начинается с директив включения заголовочных файлов. Файлыunistd.h и sys/fcntl.h содержат функции для работы с системными вызовами. Заголовочный файл linux/cdrom.h содержит различные константы и макросы, используемые при работе с CD-ROM, но, увы, не содержит

макросов, с помощью которых можно было бы преобразовать MSF во фреймы и обратно. Мы сами определяем соответствующие функции. Мы открываем файл устройства с помощью системного вызова `open(2)`:

```
int cdd = open("/dev/cdrom", O_RDONLY);
```

Флаг, переданный функции `open`, указывает, что файл открыт только для чтения. Дальнейший доступ к устройству будет выполняться с помощью полученного дескриптора `cdd`. В *Linux 2.4.22* каждый процесс может открыть не более 1048576 дескрипторов одновременно [2]. Нашим программам этого будет вполне достаточно. Мы предполагаем, что устройство `/dev/cdrom` установлено в системе и работает правильно, однако, в общем случае неплохо проверить значение дескриптора, возвращенное `open`, на предмет ошибки (в этом случае функция возвращает `-1`, переменная `errno` содержит дополнительный код ошибки). Вызовы `ioctl`, связанные с воспроизведением Audio CD, приведены в таблице 1.

Результат запросов `CDROM_DRIVE_STATUS` и `CDROM_DISC_STATUS` возвращается не в параметре-ссылке, а как результат функции `ioctl`. В качестве третьего аргумента в этих запросах выступает одна из констант `CDSL_XXX`, определенных в файле `cdrom.h`. Эти константы предназначены для работы с устройствами автоматической смены компакт-дисков (CD changers). В случае «однодискового» устройства следует использовать `CDSL_CURRENT`. Результатом вызова `CDROM_DRIVE_STATUS` могут быть значения `CDS_NO_DISC` (нет диска в устройстве), `CDS_DRIVE_NOT_READY` (устройство не готово), `CDS_DISC_OK` (диск обнаружен), а также некоторые другие константы из файла `cdrom.h`. Среди значений, возвращаемых вызовом `CDROM_DISC_STATUS`, следует отметить `CDS_NO_DISC` (см. выше) `CDS_AUDIO` (диск опознан как аудио) и `CDS_MIXED` (диск опознан как «смешанный»). Остальные значения соответствуют не-аудиодискам. Нижеследующий фрагмент программы проверяет, готов ли CD-дисковод к передаче данных:

```
disc_stat = ioctl(cdd, CDROM_DRIVE_STATUS, CDSL_CURRENT);
if ((disc_stat != CDS_DISC_OK) && (disc_stat != CDS_NO_INFO))
{
close(cdd);
printf("Устройство не готово\n");
return 1;
}
```

Вызовы `CDROMREADTOCHDR` и `CDROMREADTOCENTRY` предназначены для работы с оглавлением диска. Вызов `CDROMREADTOCHDR` позволяет получить данные о номере первого и последнего информационных треков на диске, а вызов `CDROMREADTOCENTRY` — данные об отдельном треке: адрес начала трека (в формате MSF или LBA), тип трека (аудио или данные) и т.п. Вызов `CDROMSUBCHNL` позволяет получить информацию о текущем состоянии устройства — находится ли диск в режиме воспроизведения, и в какой позиции выполняется чтение данных. Строка программы:

```
ioctl(cdd, CDROMREADTOCHDR, &toc);
```

заполняет переменную `toc` типа `cdrom_tochdr` данными заголовка оглавления диска. Структура `cdrom_tochdr` позволяет нам узнать количество треков на диске.

Вызов:

```
ioctl(cdd, CDROMREADTOCENTRY, &entry);
```

Вызов	Описание	Дополнительный параметр
<code>CDROM_DRIVE_STATUS</code>	Получение данных о состоянии устройства	константа <code>CDSL_XXX</code>
<code>CDROM_DISC_STATUS</code>	Получение данных о диске	константа <code>CDSL_XXX</code>
<code>CDROMREADTOCHDR</code>	Чтение заголовка оглавления диска	структура <code>cdrom_tochdr</code>
<code>CDROMREADTOCENTRY</code>	Чтение элемента оглавления диска	структура <code>cdrom_tocentry</code>
<code>CDROMSUBCHNL</code>	Чтение данных о параметрах воспроизведения	структура <code>cdrom_subchnl</code>
<code>CDROMPLAYTRKIND, CDROMPLAYMSF</code>	Воспроизведение аудиозаписи	Структуры <code>cdrom_ti</code> и <code>cdrom_msf</code>
<code>CDROMSTOP</code>	Остановка воспроизведения	значение 0
<code>CDROMPAUSE, CDROMRESUME</code>	Приостановка, возобновление воспроизведения	значение 0
<code>CDROMEJECT</code>	Открытие лотка устройства	значение 0
<code>CDROMCLOSETRAY</code>	Закрытие лотка устройства	значение 0

Таблица 1. Вызовы `ioctl`, связанные с воспроизведением Audio CD.

## AUDIO CD ИЗНУТРИ

Запись на любом компакт-диске состоит из нескольких треков. Треки нумеруются начиная с нуля (трек 0 содержит оглавление диска). Номер трека не может превышать значение 99. На аудио CD каждый музыкальный фрагмент как правило записывается на отдельном треке. На одном и том же диске могут быть записаны как аудио-треки, так и треки данных. Аудиоданные на CD записываются в 16-битном представлении с чередующимися сэмплами для правого и левого канала, с частотой дискретизации 44.1 КГц (если вы не знаете, что такое сэмплы и частота дискретизации, не волнуйтесь, для нашего примера это не принципиально).

Запись на диске разбивается на фреймы. Каждый фрейм содержит 2352 байта. Нетрудно подсчитать, что для обеспечения указанных выше характеристик цифровой записи чтение данных должно выполняться со скоростью 75 фреймов в секунду (что и соответствует однократной скорости чтения CD-ROM). С фреймами связан и один из форматов адресации на аудио CD. Адресация осуществляется в единицах MSF — минуты, секунды, фреймы — где фрейм можно рассматривать как 1/75 секунды. Другой формат адресации, связанный с логическими блоками (LBA), используется в основном при работе с не-аудиодисками.

позволяет получить информацию о заданном треке. Дополнительный параметр вызова имеет тип «указатель на структуру `cdrom_tocentry`». Перед вызовом `ioctl` мы заполняем поля `format` (формат длительности трека) и `track` (номер трека) этой структуры. В этой же структуре системный вызов возвращает информацию о выбранном треке, в том числе тип трека (аудио или данные) и длительность трека. В файле `cdrom.h` определена константа `CDROM_LEADOUT`, указывающая на условный трек, расположенный после последнего трека.

Чтение данных трека выполняется с помощью вызова:

```
ioctl(cdd, CDROMREADAUDIO, &rdaudio);
```

где `rdaudio` — структура `cdrom_read_audio`.

Наша программа считывает данные CD и записывает их в файл формата `wav`. Строка вызова программы должна выглядеть так (исполнимый файл называем `cdripper`)

```
cdripper <трек> <файл>
```

где `трек` — номер трека (первый трек, содержащий пользовательские данные имеет номер 1), `файл` — имя файла в котором будут сохранены аудиоданные в формате `wav`.

Принцип, согласно которому любой объект системы должен быть представлен в виде файла, приводит к тому, что даже дескрипторы файлов представлены в Linux в виде файлов. В директории `/dev/fd` можно увидеть файлы-ссылки с именами `0`, `1`, `2` и так далее. Эти файлы представляют дескрипторы файлов, открытых процессом, который читает директорию `/dev/fd`. Именно так, каждый процесс видит в этой директории только свои дескрипторы. Как некий артефакт из фантастического мира, директория `/dev/fd` выглядит по-разному в зависимости от того, кто на нее «смотрит» (этим свойством обладают также многие директории и файлы из виртуальной файловой системы `/proc`, которую мы рассмотрим ниже). Открытие файла ссылки из `/dev/fd` эквивалентно созданию дубликата дескриптора, который представляет файл. Например, вызов

```
fd = open("/dev/fd/1", 0);
```

присваивает `fd` дубликат дескриптора, представленного файлом `/dev/fd/1` (файлы `/dev/fd/0`, `/dev/fd/1` и `/dev/fd/2` по умолчанию соответствуют стандартным потокам ввода, вывода и ошибок).



Элемент	Тип	Содержание
cmdline	файл	Командная строка, использовавшаяся при запуске процесса
cwd	символическая ссылка	Указывает на директорию процесса
environ	файл	Список переменных окружения для данного процесса
exe	символическая ссылка	Указывает на файл, хранящий образ процесса
fd	директория	Ссылки на файлы, используемые процессом
root	гибкая ссылка	Указывает на корень файловой системы процесса
stat	файл	Различные сведения о процессе

Таблица 2. Файлы и дочерние каталоги /proc/<PID>, позволяющие получить различную информацию о процессе.

### Файловая система /proc

Помимо файловой системы /dev, в Linux есть еще один источник необычных файлов – файловая система /proc. С помощью этой файловой системы можно получить множество ценных сведений о состоянии различных устройств и системных объектов (модулей ядра, например) а также о выполняющихся процессах (собственно, отсюда и происходит ее название). Сведения об устройствах понадобятся, вероятнее всего, только всяким настраивающим/диагностическим утилитам. Мы же рассмотрим некоторые элементы системы /proc, которые могут пригодиться в программах самого разного назначения. Данные о каждом процессе хранятся в специальной поддиректории директории /proc, с именем, соответствующим численному значению идентификатора процесса. В директории процесса находятся несколько файлов и поддиректорий, из которых можно почерпнуть данные о нем (см. таблицу 2)

Если вы не root, то доступ ко многим поддиректориям процессов будет вам запрещен, но к своей собственной поддиректории процесс может получить доступ всегда. Как найти свою поддиректорию? С помощью getpid(2) процесс может узнать свой идентификатор и сконструировать путь к поддиректории, но есть и более простой способ. Помимо поддиректорий с именами, соответствующими идентификаторам процессов, каждый процесс «видит» в директории /proc поддиректорию-ссылку self, которая указывает на каталог с его данными. Использование данных из директории процесса мы рассмотрим на примере небольшой программы printenv, которая распечатывает в стандартный поток вывода полный список своих переменных окружения.

```
include <stdio.h>
#define BUF_SIZE 0x100
int main(int argc, char * argv[])
{
    char buf[BUF_SIZE];
    int len, i;
    FILE * f;
    f = fopen("/proc/self/environ", "r");
    while((len = fread(buf, 1, BUF_SIZE-1, f)) > 0)
    {
        for (i = 0; i < len; i++) if (buf[i]==0) buf[i] = 10;
        buf[len] = 0;
        printf("%s", buf);
    }
    return 0;
    fclose(f);
}
```

Строки в файле environ разделены не символами перевода строки (имеющим код 10 или 0x0A), а нулями.

### Два способа прочесть содержимое директории

Задача перечисления всех элементов директории возникает довольно часто. Стандартная библиотека Linux предоставляет два способа перечисления содержимого директории: первый способ – с помощью функции scandir и функций обратного вызова, второй – с использованием набора функций opendir, readdir, closedir. Рассмотрим два варианта программы printdir, распечатывающей содержимое директории, передаваемой ей в качестве аргумента.

```
#include <stdio.h>
#include <dirent.h>
int sel(struct dirent * d)
{
    return 1; // всегда подтверждаем
}
int main(int argc, char ** argv) {
    int i, n;
    struct dirent ** entry;
    if (argc != 2)
    {
        printf("Использование: %s <директория>\n", argv[0]);
        return 0;
    }
    n = scandir(argv[1], &entry, sel, alphasort);
    if (n < 0)
    {
        printf("Ошибка чтения директории\n");
        return 1;
    }
    for (i = 0; i < n; i++)
        printf("%s inode=%i\n", entry[i]->d_name, entry[i]->d_ino);
    return 0;
}
```

Функция scandir() создает список элементов указанной директории. Ей необходимо передать указатель на функцию обратного вызова, которая, получая данные об очередном элементе, принимает решение включить этот элемент в результирующий список. В нашем примере это функция sel. Если при очередном вызове функция sel вернет значение 0, соответствующий элемент директории не будет включен в конечный список. Последний параметр scandir – функция сортировки элементов директории. Мы используем функцию alphasort, сортирующую элементы в лексикографическом порядке.

Данные об элементах директории передаются в структурах dirent. Можно было бы ожидать, что структуры типа dirent содержат много полезной информации об элементах директории, но это не так. Кроме имени файла, dirent содержит номер inode для этого элемента (просто программистам обычно незачем знать номера inode, но чтобы наш пример как-то отличался от стандартного, мы включаем эту информацию). У структуры dirent есть еще поле d\_type типа char \*, но оно, как правило, содержит null.

Функция scandir позволяет нам получить полный отсортированный список элементов директории за один вызов. У нас есть возможность использовать низкоуровневые средства, которые могут оказаться быстрее в том случае, если сортировка файлов нам не нужна. Рассмотрим второй вариант программы:

```
#include <stdio.h>
#include <dirent.h>
int main(int argc, char ** argv)
{
    DIR * d;
    struct dirent * entry;
    if (argc != 2)
    {
```

```
printf("Использование: %s <директория>\n", argv[0]);
return 0;
}
d = opendir(argv[1]);
if (d == NULL)
{
printf("Ошибка чтения директории\n");
return 1;
}
while (entry = readdir(d))
printf("%s inode=%i\n", entry->d_name, entry->d_ino);
closedir(d);
return 0;
}
```

Этот вариант программы использует функции **opendir**, **readdir** и **closedir**, которые работают с директорией как с файлом. Функция **readdir** возвращает значение TRUE до тех пор, пока не будут прочитаны все элементы директории.

### Разреженные файлы

Unix-системы позволяют создавать файлы, логический размер которых превышает физический. Такие файлы могут быть удобны, когда необходимо отобразить какую-либо незаполненную структуру данных (например, матрицу с большим количеством нулей). Наглядным примером разреженных файлов являются знакомые всем нам файлы **core.dump**. Рассмотрим текст программы **makehole**.

```
#include <stdio.h>
#include <string.h>
#define BIG_SIZE 0x1000000
int main(int argc, char * argv[])
{
FILE * f;
f = fopen(argv[1], "w");
if (f == NULL)
{
printf("Невозможно создать файл: %s", argv[1]);
return 1;
}
fwrite(argv[1], 1, strlen(argv[1]), f);
fseek(f, BIG_SIZE, SEEK_CUR);
fwrite(argv[1], 1, strlen(argv[1]), f);
fclose(f);
}
```

Если скомпилировать эту программу под именем **makehole** и запустить

```
makehole bighole.txt
```

то на диске будет создан файл **bighole.txt**. Команда **ls -al** сообщит нам, что размер файла составляет чуть больше 16 мегабайт (см. значение константы **BIG\_SIZE** в программе). Однако с помощью команды

```
du bighole.txt
```

мы узнаем, что на диске этот файл занимает 24 байта. Причиной появления пропусков в открытом для записи файле стало смещение с помощью функции **fseek** в область после конца файла. Выход за пределы файла с помощью **fseek** – стандартный метод получения разреженных файлов. В момент вызова **fseek** в нашей программе позиция записи находится в конце файла. Флаг **SEEK\_CUR** указывает, что смещение отсчитывается от текущей позиции. Таким образом, в файле образуется пропуск, величина которого в байтах соответствует значению **BIG\_SIZE**. При чтении пустых блоков в разреженном файле функция чтения данных будет возвращать блоки, заполненные нулями.

### Блокировка областей файла

Блокировка областей файла позволяет нескольким программам совместно работать с содержимым одного и того же файла, не мешая друг другу, или, точнее, мешая друг другу испортить данные. Мы рассмотрим интерфейс блокировки областей, основанный на использовании функ-

ции **fcntl(2)**. Функция **fcntl()** тоже представляет собой нечто вроде швейцарского армейского ножа. С помощью этой функции можно манипулировать дескрипторами файлов и устанавливать рекомендательные (**advisory**) блокировки. Рекомендательными эти блокировки называются потому, что следование им является для программ, работающих с файлом, делом доброй воли. Если программа сама не использует блокировок, блокировки, установленные другими программами, не будут иметь для нее никакого эффекта. Существует возможность придать рекомендательным блокировкам **fcntl()** обязательный характер, но для этого соответствующая файловая система должна быть смонтирована со специальным ключом. Для изучения работы блокировок напишем программу **testlocks** (файл **testlocks.c**). При работе с блокировками во втором параметре функции **fcntl** передается одна из команд управления блокировками, третий же параметр должен содержать адрес структуры **lock**, в которую записывается информация о блокировке (см. таблицу 3).

Поле	Тип
<b>l_type</b>	фа Тип блокировки: записи – F_RDLCK, чтения – F_WRLCK, сброс – F_UNLCK
<b>l_whence</b>	Точка отсчета смещения
<b>l_start</b>	Начальный байт области
<b>l_len</b>	Длина области
<b>l_pid</b>	Идентификатор процесса, установившего блокировку (для GETLCK)

Таблица 3. Описание полей структуры **lock**.

Для установки блокировки мы заполняем поля структуры **lock** необходимыми значениями и вызываем **fcntl()** с командой **F\_SETLK** (установить блокировку):

```
fi.l_type = F_WRLCK;
fi.l_whence = SEEK_SET;
fi.l_start = 0;
fi.l_len = 64;
off = 0;
while (fcntl(fd, F_SETLK, &fi) == -1)
{
fcntl(fd, F_GETLK, &fi);
...
printf("байты %i – %i заблокированы процессом %i\n", off, off+64, fi.l_pid);
}
```

Если заданная область уже заблокирована, **fcntl** возвращает **-1**. С помощью команды **F\_GETLK** можно узнать идентификатор процесса, заблокировавшего данную область. Чтобы снять блокировку, мы вызываем **fcntl()** с командой **F\_SETLK** (странно, не правда ли?) и параметром **l\_type** структуры **lock**, равным **F\_UNLCK**:

```
fi.l_type = F_UNLCK;
if (fcntl(fd, F_SETLK, &fi) == -1)
printf("Ошибка разблокирования\n");
```

Скомпилируйте программу **testlocks** и запустите на выполнение сразу несколько экземпляров. Первый экземпляр **testlocks** создаст файл **testlocks.txt**. Каждый процесс заблокирует 64 байта в этом файле и сделает запись в заблокированную область. Второй, третий и все последующие экземпляры процессов сообщат, какие области файла уже заблокированы другими процессами. Завершить программу **testlocks** можно, нажав любую символьную клавишу, а затем ввод.

Файлы Linux, – это не только удобные хранилища данных. С их помощью можно решать множество задач, начиная с управления устройствами и заканчивая разграничением доступа к ресурсам. Однако, работа с файлами – далеко не единственное, что может Linux. **LXF**

### ЛИТЕРАТУРА:

- D. P. Bovet, M. Cesati, Understanding the Linux Kernel, 3rd Edition, O'Reilly, 2005
- W. R. Stevens, S. A. Rago, Advanced Programming in the UNIX® Environment: Second Edition, Addison Wesley Professional, 2005

## ЧЕРЕЗ МЕСЯЦ

Возможность разграничения доступа к ресурсам очень полезна, но Linux не была бы настоящей многозадачной системой, если бы в ней не существовало простых и эффективных методов взаимодействия между процессами. В следующей статье мы рассмотрим межпроцессное взаимодействие.



ПРОГРАММИРОВАНИЕ СЦЕНАРИЕВ

# Уроки Python



• Исходные тексты примеров к статье.

**ЧАСТЬ 6** Теперь, когда мы научились обрабатывать строки, самое время показать наше умение другим. Но кто будет смотреть сценарий «без лица», пусть даже в Konsole с красивым прозрачным фоном? Для тех, кто не Квентин Тарантино, **Сергей Супрунов** покажет, как создаются на Python графические интерфейсы.

## МЕСЯЦ НАЗАД

Мы разобрались с регулярными выражениями и показали, что Perl — не единственный язык для обработки текстов.



Чем больше работаешь с Python, тем сильнее ощущаешь его мощь и гибкость. Порой кажется, что для этого языка нет ничего невозможного. Это же можно отнести и к разработке графического интерфейса пользователя (хотя здесь Python просто использует возможности мощных библиотек). Как вы увидите, Python и в этом вопросе сохраняет простоту и эффективность, становясь весьма удобным инструментом для решения таких задач, как создание «обёрток» к различным консольным утилитам, разработка графических конфигураторов для ваших любимых инструментов, проектирование прототипов программ (куда проще за пару дней согласовать с заказчиком все «интерфейсные» моменты разрабатываемого проекта, меняя внешний вид прямо на переговорах, чем спустя месяцы напряжённой работы переделывать многомегабайтный проект на C++, только лишь потому, что заказчик счёл интерфейс слишком сложным для освоения), и т.д. Кстати, инсталлятор Gentoo, появившийся в версии 2006.0 и наделав-

ший столько шума, тоже написан на Python.

Для Python разработано множество инструментов, отличающихся по гибкости, сложности, степени интеграции с вашим окружением. В рамках одного урока сложно рассмотреть всё это многообразие даже в общих чертах. Поэтому мы ограничимся кратким знакомством с модулем Tkinter и «привязками» PyQt и PyGTK.

Есть ещё достаточно мощный и переносимый модуль wxPython, являющийся привязкой к графической библиотеке wxWidgets, который тоже заслуживает внимания, но думаю, вы сможете разобраться с ним самостоятельно.

Начнём со старого доброго Tkinter, который, несмотря на некоторую «неказистость» и плохую интеграцию с системой, всё же остаётся стандартным модулем, входящим в поставку практически всех дистрибутивов. К тому же его простота — идеальная особенность для учебных целей. А разобравшись с ним, вы без труда освоите и другие инструменты.

## Очарование простоты

Проверить наличие *Tkinter* в вашем дистрибутиве достаточно просто (он должен быть, но всякое случается): запустите *Python* и выполните команду `import Tkinter` (обратите внимание на заглавную первую букву). Если вам не повезло и вы увидели сообщение об ошибке, придётся установить этот модуль отдельно. Проконсультируйтесь с вашим менеджером пакетов, не знает ли он случайно про *Tkinter* (имя пакета, в отличие от модуля, обычно записывается только маленькими буквами). Если он окажется из партизанской семьи, то прямая дорога вам на [www.Python.org](http://www.Python.org) (который с новым дизайном смотрится очень даже приятно).

Разобравшись с модулем, приступим сразу к серьёзной работе. В качестве примера выберем такую задачу: разработать графический интерфейс для просмотра map-страниц. Хорош он тем, что логика предельно проста, и нам почти не придётся отвлекаться от основной задачи. Итак, сразу код (по мере необходимости, будем прерывать его некоторыми пояснениями):

```
#!/usr/bin/Python
# -*- coding: utf-8 -*-
import os, re
from Tkinter import *
```

Здесь мы объявили кодировку нашей локали и подключили необходимые для работы модули (не забывайте следить за регистром символов). Далее опишем класс, который будет отвечать за работу нашего приложения. Использование объектного подхода позволит в дальнейшем, если потребуется, легко создавать модифицированные приложения на базе этого класса или интегрировать его функциональность в другие объекты:

```
class ManReader:
    def getman(self, manpage, sn):
        bschar = '\b'
        tmp = os.popen('man %s %s 2>/dev/null' % (sn, manpage)).read()
        if tmp:
            tmp = re.sub(r'(\%c\1' % bschar, r'\1', tmp)
            tmp = tmp.replace('_', '%c' % bschar, ")
        else:
            tmp = 'Page "%s" in section "%s" not found' % \
                (manpage, sn or 'Auto')
        return tmp
```

Метод с украинским именем **getman()** отвечает за то, чтобы вернуть запрошенную map-страницу в виде текстовой переменной. Для этого используем уже знакомую нам функцию **popen()** модуля **os**. Нужно учитывать, что map-страницы могут содержать специальное форматирование: подсоветку и подчёркивание. Чтобы не усложнять наш пример, мы просто забываемся от ненужных символов, оставляя чистый текст. Пожалуй, нужно пояснить, как мы это делаем.

Если вы посмотрите на результат работы команды `man` в «натуральном» виде, например, сохранив его в файл, то заметите, что яркость создаётся следующим приёмом: символ выводится, стирается символом «**\b**» (Backspace) и затем снова выводится. Подчёркивание достигается выводом и последующим стиранием символа «**\_**». То есть вы можете увидеть что-то такое: **N\bNA\bAM\bME\bE = NAME, \b/\_\be\\_bt\\_bc = /etc**. С удалением подчёркиваний всё понятно. А вот для удаления «символов яркости» мы используем регулярное выражение, причём на первый фрагмент, взятый в скобки, мы можем в дальнейшем ссылаться с помощью **\1**. То есть регулярное выражение «**(.)\1**» означает два одинаковых символа, разделённых восклицательным знаком. Мы же используем символ **\b**. Обратите внимание, что его нельзя указывать непосредственно в регулярном выражении, иначе он будет трактоваться как граница слова. Вернёмся к нашим «виджетам»:

```
def pressBtn(self):
    sn = sectno.get()[0]
    if sn == ':': sn = ""
    tx.delete(1.0, END)
    tx.insert(1.0, self.getman(ent.get(), sn))
```

Это — функция-обработчик нажатия на кнопку (которую мы пока не нарисовали). Ещё ничего не понятно, но мы вернёмся к ней чуть позже, а сейчас перейдем к самому важному методу нашего класса, собственно и рисуемому рабочему окну. Разберём его подробнее:

```
def Drawface(self):
    global ent, tx, sectno
```

Объявляем некоторые переменные как глобальные, чтобы мы могли обращаться к ним из других функций (в нашем примере — из **pressBtn**).

```
win = Tk()
win.title('Просмотрщик map-страниц')
```

Так мы создаём объект, который будет являться нашим рабочим окном. Заодно задаём ему заголовок.

```
sectno = StringVar()
sectno.set('---Auto---')
```

В будущем нам понадобится эта переменная — она должна представлять собой особый объект, поэтому и создаём её как экземпляр класса **Tkinter.StringVar()**.

```
fcmd = Frame(win)
fcmd.pack(side=TOP, fill=X)
```

В основном окне создаём фрейм, который будет являться контейнером для элементов управления. Второй строчкой «упаковываем» его, т.е. указываем место расположения на главном окне. Параметры метода **pack()** означают: прикрепить элемент к верхнему краю (TOP) и растянуть по горизонтальной оси.

Здесь нужно сказать, что **pack** — не единственный метод упаковки в *Tkinter*. В ряде случаев удобнее использовать упаковку «по сетке» — **grid** (подробности можно узнать в документации). Приступаем к созданию графических элементов:

```
lbl1 = Label(fcmd, text='Read about ')
lbl1.pack(side=LEFT)
```

Это — обычная текстовая метка. Она размещается во фрейме (первый аргумент метода) и упакована по левому краю.

```
ent = Entry(fcmd)
ent.bind('<Return>', (lambda event: self.pressBtn()))
ent.pack(side=LEFT)
```

Текстовое поле ввода также упаковываем слева, сразу после **lbl1**. Метод **bind()** задаёт реакцию поля на нажатие клавиши [Enter] (a.k.a. Return) — будет вызван метод **pressBtn()**.

```
lbl2 = Label(fcmd, text=' in section ')
lbl2.pack(side=LEFT)
slt = OptionMenu(fcmd, sectno,
    '---Auto---',
    '1 User Utilities',
    '2 System Calls',
    '3 Library Functions',
    '4 Devices & Kernel Interfaces',
    '5 File Formats',
    '6 Games',
    '7 Macros & SQL Commands',
    '8 System Utilities',
    '9 X Window',
    'n Built-In commands',
    )
slt.pack(side=LEFT)
```

Ещё одна текстовая метка, и далее — выпадающий список. В нём мы перечисляем имеющиеся разделы справки, к которым будет относиться наш запрос.

```
btn1 = Button(fcmd, command=win.quit, text='Quit')
btn1.pack(side=RIGHT)
btn2 = Button(fcmd, command=self.pressBtn, text='Open')
btn2.pack(side=RIGHT)
```

Две кнопки — завершающая работу и отображающая запрошенную map-страницу. Обратите внимание, что здесь упаковка выполняется по правому краю, т.е. Quit будет самой правой, а Open — чуть левее. Важнейший параметр — **command**, задаёт функцию-обработчик. В пре-



◀ вом случае используется встроенный метод **quit**, завершающий работу, во втором — наш метод **pressBtn**. Обратите внимание, что здесь происходит не вызов метода, а даётся ссылка на него, т.е. имя указывается без скобок.

```
fview = Frame(win)
fview.pack(side=BOTTOM, fill=BOTH, expand=YES)
sb = Scrollbar(fview)
tx = Text(fview, relief=SUNKEN)
sb.config(command=tx.yview)
tx.config(yscrollcommand=sb.set)
sb.pack(side=RIGHT, fill=Y)
tx.pack(side=TOP, expand=YES, fill=BOTH)
```

Во втором фрейме, который мы прикрепляем к нижнему краю и растягиваем во все стороны, заполняя всё доступное пространство, размещается текстовое поле, где мы будем отображать содержимое map-страницы. Поскольку последняя может быть достаточно длинной, понадобится скроллинг (элемент Scrollbar). Обратите внимание на то, как методами **config()** мы обеспечиваем взаимную привязку текстового поля и полосы прокрутки. Параметр **relief** в описании текстового поля задаёт вид рамки вокруг поля.

Настало время поговорить о **pressBtn**. Как вы видели, этот метод будет выполняться в двух случаях — по щелчку на кнопке Open и при нажатии [Enter] в поле ввода. Получив цифру раздела справки и очистив текущее содержимое поля **tx** от начала (**1,0**) до конца (END), вызываем метод **getman()** и вставляем результат в **tx**.

```
win.mainloop()
```

Всё выше было просто подготовкой. А вот этой командой мы запускаем наше окно в работу. С этого момента управление передаётся Tkinter-объекту, и влиять на его поведение можно только с помощью описанных ранее обработчиков событий.

```
if __name__ == '__main__':
test = ManReader()
test.Drawface()
```

Ну, это должно быть понятно — если скрипт запускается автономно, а не экспортируется в другой, то создаём объект нашего класса и запускаем его в работу.

Фух... Кажется, мы сделали это. Результат наших трудов можно наблюдать на рисунке (Рисунок 1). Как видите, «виджеты» имеют свой уникальный дизайн и будут резко выделяться на фоне остального интерфейса, внешний вид которого вы с такой любовью выбирали среди десятков различных стилей. Но это работает (причём одинаково) и в KDE, и в Gnome, и даже в Windows и Mac OS X.

Теперь, когда мы разобрались с основами, самое время перейти к «родным» для Linux-окружения средствам — *PyQt* и *PyGTK*. Прежде чем приступить к работе, проверьте, есть ли в вашей системе нужные модули.

### Пара слов про PyGTK

Мощь библиотеки *GTK* позволяет создавать таких «гигантов графики», как *Gimp*. Отрадно, что мы можем в значительной степени воспользоваться её возможностями и в сценариях *Python*. В каталоге **/usr/**

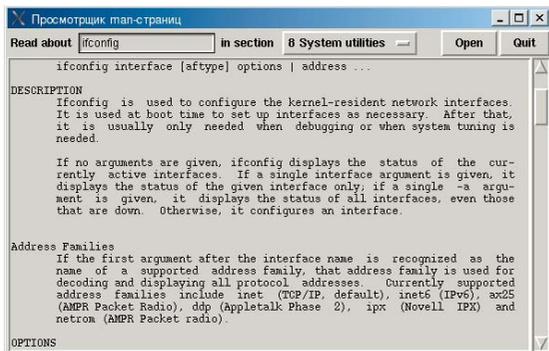


Рисунок 1. Tkinter – лучший способ сделать не так, как у других!

**share/doc/pygtk<версия>/examples** вы найдёте массу примеров её использования. Здесь приведём простейший вариант:

```
#!/usr/bin/Python
# -*- coding: utf-8 -*-
import gtk
# Создаём основное окно
win = gtk.Window()
def main():
# Задаём параметры окна (размер, заголовок и т.д.)
win.set_default_size(300, 50)
win.set_border_width(10)
win.connect('destroy', gtk.main_quit)
win.set_title('Небольшой пример')
txtvar = ""
```

Библиотека *GTK* привносит в *Python* небывалую мощь и высокий уровень интеграции в среду Gnome.

```
# Текстовая метка – не забывайте применять метод
#- show() к каждому объекту, чтобы он был видимым
txt = gtk.Label(txtvar)
txt.show()
# Кнопка. На неё «привязываем» завершение работы
btn = gtk.Button('Заккрыть')
btn.connect('pressed', lambda button: gtk.main_quit())
btn.show()
# объект Window может содержать только один элемент,
#- так что им будет HBox, который служит контейнером для остальных
box = gtk.HBox()
box.pack_start(txt)
box.pack_start(btn)
win.add(box)
box.show()
win.show()
gtk.main()
main()
```

Результат представлен на рисунке 2.

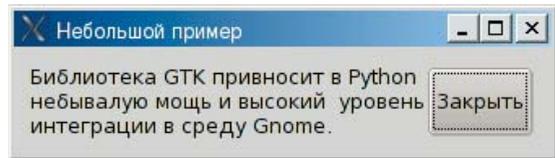


Рисунок 2. Именно так вы и должны представлять себе GTK-приложение.

### Фанатам KDE посвящается

Если для вас Linux не мыслим без KDE, то *PyQt* – как раз тот инструмент, который способен обеспечить «бесшовную» интеграцию ваших *Python*-сценариев с окружением рабочего стола. (Впрочем, *Qt* работает не только в Linux, так что ваши решения по-прежнему сохранят определённый уровень переносимости.) К тому же, для разработки интерфейса к вашим услугам *Qt Designer*. А создадим мы программу для просмотра запущенных в системе процессов. Итак, в путь!

### Делай «РАЗ»!

Первым делом, запустите Qt Designer и откройте новый проект типа Widget. Для кнопки создайте соединение с формой (Connect Signal/Slots) и объявите новый слот, назвав его showPs. В дальнейшем это будет наш обработчик нажатия кнопки. Сохранив интерфейс (Рисунок 3) под именем **psview.ui**, нужно конвертировать его в *Python*-код:

```
$ pyuic psview.ui > psview.py
```

Теперь в **psview.py** описан класс, отвечающий за наш интерфейс. Если вы посмотрите на него, то увидите, что всё не намного сложнее, чем в *Tkinter*, так что при желании интерфейс можно создать и вручную.

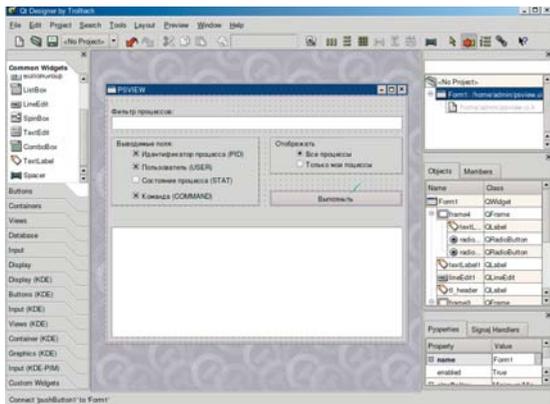


Рисунок 3. Мышкой влево, мышкой вправо – а зачем вам вообще клавиатура?

## Делай «ДВА»!

Далее, нам нужно довести всё это до ума. Непосредственное редактирование созданного сценария – самое плохое решение, поскольку вы лишитесь возможности корректировать интерфейс в *Qt Designer*, т.к. все изменения при этом будут потеряны. Поэтому правильно будет оставить *psview.py* в неприкосновенном виде, а для работы описать ещё один класс на базе созданного, пользуясь возможностями наследования. Просто создать экземпляр класса *Form1* тоже не совсем хорошо – ведь нам нужно будет изменить поведение класса, отредактировав обработчик нажатия кнопки. Полученный код с переопределённой функцией-обработчиком будет выглядеть так:

```
#!/usr/bin/Python
# -*- coding: utf-8 -*-
import os, sys
# Импортируем модуль, созданный автоматически
from psview import *
class PsForm(Form1): # наследуем от Form1
    def __init__(self):
# Помните, что при наследовании инициализацию
#- родительского класса нужно делать вручную?
        Form1.__init__(self)
    def showPs(self):
# Формируем нужные ключи команды ps
        if self.radioButton1.isChecked():
            all = 'a'
        else:
            all = ""
# Выводимые поля описываются после ключа «0»
#- Очень важно не допускать пробелов!
        fields = '0'
        if self.checkBox1.isChecked():
            fields += 'pid,'
        if self.checkBox2.isChecked():
            fields += 'user,'
        if self.checkBox3.isChecked():
            fields += 'stat,'
        if self.checkBox4.isChecked():
            fields += 'command,'
        fields = fields[:-1] # последнюю запятую – долой!
        filter = self.lineEdit1.text()
# Если есть фильтр, то заголовок придётся забирать
#- отдельной командой (head – чтобы не выводить лишнее)
        if filter:
            cmd = 'ps %s%s | grep %s | grep -v grep' % (all, fields,
            filter)
            head = os.popen('ps %s | head' % fields).readline()
            body = os.popen(cmd).read()
        else:
```

```
cmd = 'ps %s%s' % (all, fields)
pspipe = os.popen(cmd)
head = pspipe.readline() # первая строка
body = pspipe.read() # всё остальное
pspipe.close()
self.tl_header.setText(head)
self.te_body.setText(body)
```

# Создаём объект-приложение

```
app = QApplication(sys.argv)
form = PsForm()
app.setMainWidget(form)
form.show()
app.exec_loop()
```

Небольшое пояснение: чтобы при прокрутке заголовков был всегда на виду, мы вынесли его в отдельную текстовую метку (**tl\_header**). Поскольку при использовании *grep* заголовок теряется, придётся приложить чуточку усилий, чтобы всё-таки обеспечить его вывод (см. фрагмент «**if filter – else**»).

## Делай «ТРИ»!

Всё! Можно запускать нашу оболочку, не забыв сделать скрипт исполняемым (Рисунок 4). Конечно, это не верх совершенства – из множества доступных полей поддерживаются только четыре, нет проверки выражения фильтра на безопасность (а что, если пользователь введёт «**root; rm -Rf /**»?)). Впрочем, это беда любого графического интерфейса – неизбежная потеря функциональности и гибкости в угоду сомнительным удобствам. Тем не менее, наша задача была всё-таки в другом – показать пример разработки ГИП, с чем мы успешно справились.

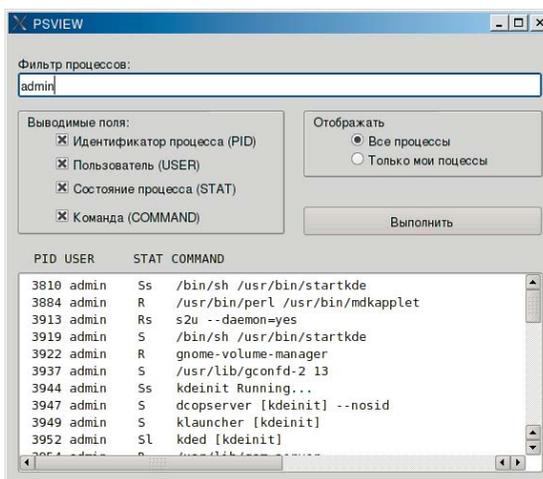


Рисунок 4. Ну, нравится мне стиль *Redmond* – ничего с собой поделаться не могу...

Как видите, изложенное сегодня – всего лишь основа. Но, надеюсь, этого будет достаточно, чтобы приступить к быстрой и эффективной разработке графических интерфейсов. На этом мы завершаем «Уроки *Python*», но не прощаемся – со следующего номера начинается серия «*Python* для профессионалов», где мы поговорим о многопоточных приложениях, обработке сетевых протоколов, взаимодействии с СУБД и библиотеке *Python Image Library*. **LXF**



**ЧЕРЕЗ  
МЕСЯЦ**

Мы рассмотрим создание приложений с графическим интерфейсом пользователя.

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

# MetaPost **Дополнительные главы**

**ЧАСТЬ 4** Всё описать невозможно, но никто не запрещает попробовать.

Евгений Балдин представляет вашему вниманию заключительную серию учебника MetaPost.

**МЕСЯЦ  
НАЗАД**



Возможности META в то или иной мере уже изложены. В этой статье будут разобраны полезные приёмы, которые можно применять при кодировании картинок и описаны некоторые из стандартных пакетов, поставляемых с *MetaPost*.

Исходники стандартных *MetaPost*-пакетов обычно находятся в директории  $\$(TEXMF)/texmf-dist/metapost/$ , а документация к ним – в директории  $\$(TEXMF)/texmf-dist/doc/metapost/$ , где  $\$(TEXMF)$  – корневая директория для дистрибутива *LaTeX*. Сказанное верно для *TeX Live*.

## Пакет boxes

*boxes* – один из самых первых пакетов общего назначения, появившихся в *MetaPost*. Его предназначение – рисовать простые диаграммы. Он проектировался как более изощрённая замена для пакета Брайана Кернигана *pic* (*man pic*).

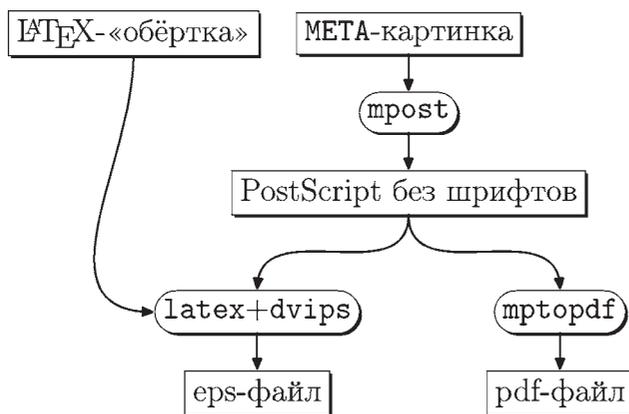
С другой стороны, функциональности этого пакета достаточно для автоматической генерации достаточно сложных зависимостей. В качестве примера разберём, как была реализована диаграмма, объясняющая действие *MetaPost*-конвейера во введении в цикл статей по *MetaPost*. (см. LXF76)

```
%Файл dop.mp
%в отличии от boxes здесь ещё определено окружение rboxit
input rboxes;
```

```
%определяем ещё один вид box'a
%параметры можно передавать и так
%expr - изолированные выражения
%text - абсолютно всё, что передаётся
\vardef drawshadowed(expr dx,dy)(text t) =
fixsize(t);
forsuffixes s=t:
fill bpath.s shifted (dx,dy);
unfill bpath.s;
drawboxed(s);
% можно было напечатать только текст
% и не рисовать рамку
% draw pic(s) withcolor red;
endfor;
enddef;

%пример boxes
\beginfig(2) ;
numeric u;u:=1mm;
%определяем box'ы
%определяем box с прямыми углами
boxit.a(btex \texttt{META}-картинка etex);
%определяем box с закруглёнными углами
rboxit.b(btex \texttt{metapost} etex);
%определяем жёсткую связь между a и b
b.n = a.s - (0,5u);
boxit.c(btex PostScript без шрифтов etex);
c.n = b.s - (0,5u);
rboxit.e(btex \texttt{latex}(+) \texttt{dvips} etex);
e.n=c.s-(20u,10u);
boxit.d(btex \LaTeX-<<обёртка>> etex);
d.e=a.w-(5u,0);
boxit.g(btex eps-файл etex);
g.n=e.s-(0,5u);
rboxit.h(btex \texttt{mptopdf} etex);
h.n=c.s-(-20u,10u);
boxit.i(btex pdf-файл etex);
i.n=h.s-(0,5u);
```

MetaPost-конвейер



```
%разрешаем зависимости и рисуем boxes
drawshaded(1/3u,-1/3u,a,b,c,d,e,g,h,i);
pickup pencircle scaled 0.3u;
%рисуем стрелки (сдвигка вида (0u,-1/3u) появилась,
%так как было определено новое окружение drawshaded)
drawarrow a.s -- b.n-(0u,-1/3u);
drawarrow b.s -- c.n-(0u,-1/3u);
drawarrow c.s{dir -90} .. {dir -90}e.n-(0u,-1/3u);
drawarrow c.s{dir -90} .. {dir -90}h.n-(0u,-1/3u);
drawarrow e.s -- g.n-(0u,-1/3u);
drawarrow h.s -- i.n-(0u,-1/3u);
drawarrow d.s{dir -90} .. {dir 0}e.w-(1/3u,0u);
endfig;
```

Обратите внимание на ещё один вид цикла – **forsuffixes**. Эта запись позволяет разбирать произвольное число передаваемых аргументов.

Алгоритм создания диаграммы следующий:

- С помощью команд **boxit**, **rboxit** или **circleit** объявляем box. Имя box'a добавляется к команде через точку как суффикс. В данном примере определено 8 box'ов от «a» до «i» включительно.

- Составляем уравнения связей. Через точку к именам box'ов можно добавить один из восьми суффиксов: **n** – север (верхняя точка), **s** – юг (нижняя точка), **w** – запад (крайняя точка рамки слева), **e** – восток (крайняя точка рамки справа) и **sw**, **bw**, **se**, **sw** – комбинации уже перечисленных суффиксов, которые соответствуют углам рамки

Уравнение связи

```
b.n = a.s - (0.5u);
```

можно описать следующим образом: северная (верхняя) точка box'a «b» находится ниже южной (нижней) точки box'a «a» на 5u. При составлении уравнения связей следует пользоваться только знаком равенства. Это не присваивание, а именно уравнение, которое требуется разрешить.

- С помощью команды **drawboxed** нарисовать «закодированную» диаграмму. При исполнении этой команды разрешается система уравнений. По умолчанию, если не указать специально, box располагается в точке (0,0).

- Нарисовать стрелки между элементами диаграммы. Можно пользоваться именами box'ов со стандартными суффиксами.

Прежде чем использовать этот пакет, необходимо прочитать соответствующий раздел в руководстве пользователя по *MetaPost* Джона Хобби **mpman.pdf**.

### Фейнмановские диаграммы

Простота *MetaPost* позволяет использовать его как базу для построений более высокого порядка. Существует несколько подобных *LaTeX*-пакетов. *mpic* уже упоминался. В этом разделе описывается ещё один старейший пакет, созданный по данной технологии.

В 1995 году Торстен Охл (Torsten Ohl) представил пакет *feynmp* для рисования фейнмановских диаграмм. Фейнмановские диаграммы используются для вычисления сумм большого числа вкладов от элементарных процессов. В своё время эта технология довольно сильно продвинула технику вычислений в физике высоких энергий. Её можно

использовать везде, где сложный процесс описывается с помощью элементарных приближений. Пакет *feynmp* – это пакет *LaTeX*, который для рисования диаграмм использует *MetaPost*.

Интересующий меня с целью извлечения электронной ширины  $J/\psi$  резонанса процесс имеет в Борновском приближении следующий вид:

Это несложная диаграмма, и если не считать метки, то для её описания требуется всего пять операторов:

```
%Файл eepsiee.tex
%пакет для рисования фейнмановских диаграмм
\usepackage{feynmp}
...
\begin{fmffile}{ee-psi-ee} %имя mp-файл
\begin{fmfgraph*}(110,62) %размер диаграммы
\fmfleft{ei,pi} %вершины-источники
\fmfright{eo,po} %исходящие вершины
\fmflabel{e^+}{ei} %метка источника e^+
\fmflabel{e^+}{pi} %метка источника e^+
\fmflabel{e^+}{po} %метка исходящей вершины
\fmflabel{e^+}{eo} %метка исходящей вершины
%линия соединяющая источники
\fmf{fermion}{ei,pi}
%линия соединяющие исходящие вершины
\fmf{fermion}{po,Jo,eo}
%Метка для начальной вершины промежуточной частицы
\fmflabel{\Gamma_{e^+e^-}}{Ji}
%Метка для конечной вершины промежуточной частицы
\fmflabel{B_{e^+e^-}}{Jo}
%Соединительная линия
\fmf{heavy,label=$J/\psi$}{Ji,Jo}
\end{fmfgraph*}
\end{fmffile}
```

Окружение **\begin{fmffile}** в качестве параметра требует имя mp-файла, в который будут писать команды META. В данном примере имя файла определено как **ee-psi-ee.mp**. Для того, чтобы получить диаграмму, описанную в файле **eepsiee.tex**, были проделаны следующие действия:

```
> latex eepsiee.tex
> mpost ee-psi-ee.mp
> latex eepsiee.tex
```

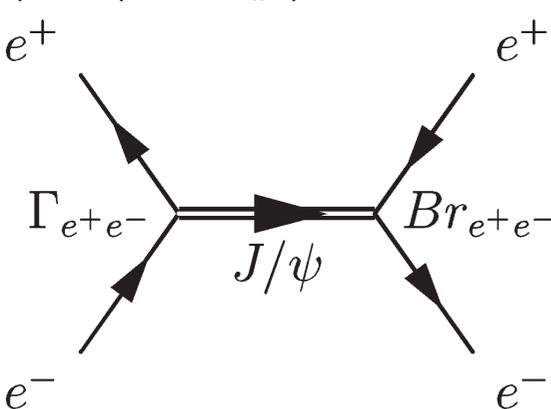
После выполнения этих команд результат можно посмотреть с помощью программы *xdvi* или преобразовать dvi-файл в PostScript или pdf.

Следующая простейшая диаграмма не имеет особого смысла. Она просто демонстрирует возможности пакета:

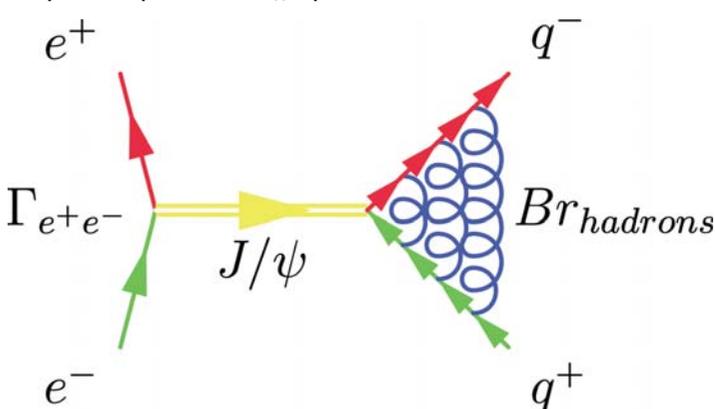
«Сотрудничество» с *MetaPost* даёт возможность *feynmp* получить доступ к цвету. Цвет можно определять точно так же, как он определяется в *MetaPost*.

```
%Файл eepsihadr.tex
\begin{fmffile}{ee-psi-hadr}
\begin{fmfgraph*}(110,62)
\fmfleft{i1,i2}
```

Простейшая фейнмановская диаграмма



Раскрашенная фейнмановская диаграмма





```

\fmfright{o1,o2}
\fmflabel{\$e^-\$}{i1}
\fmflabel{\$e^+\$}{i2}
\fmflabel{\$q^+\$}{o1}
\fmflabel{\$q^-\$}{o2}
\fmf{fermion,foreground=green}{i1,v1}
\fmf{fermion,foreground=red}{v1,i2}
\fmf{fermion,foreground=green}{o1,v2,v3,v4,v5}
\fmf{fermion,foreground=red}{v5,v6,v7,v8,o2}
%изменяем натяжение (расталкиваем вершины)
\fmf{heavy,label=\$J\psi,tension=1/3,
foreground=red+green}{v1,v5}
\fmffreeze
\fmf{gluon,foreground=blue}{v2,v8}
\fmf{gluon,foreground=blue}{v3,v7}
\fmf{gluon,foreground=blue}{v4,v6}
\fmfv{label=\$Gamma_(e^+e^-)\$}{v1}
\fmfv{label=\$Br_{\text{hadr}}\$,label.dist=0.3w}{v5}
\end{fmfgraph*}
\end{fmffile}

```

При создании диаграммы *MetaPost* пытается оптимально расположить вершины, минимизируя взвешенную сумму расстояний между ними. При оптимизации используется понятие натяжения между вершинами. По умолчанию натяжение равно 1. С помощью опции **tension** натяжение можно изменить. Чем меньше натяжение, тем сильнее расталкиваются вершины.

Обычно полученная диаграмма со значениями по умолчанию не требует вмешательства в код. В случае, если результат не устраивает, достаточно, как правило, изменить натяжение для одного, максимум для двух соединений.

Подробнее документацию по пакету *feynmp* можно найти в файле **\$(TEXMF)/texmf-dist/doc/latex/feynmf/manual.ps.gz**, где **\$(TEXMF)** – корень дерева *LaTeX* (заведомо верно для *TeXLive*).

### Фракталы

META, естественно, поддерживает рекурсию. Пользуясь этим, а также генераторами случайных **uniformdeviate** и **normaldeviate**, можно организовать фрактальную «лесопосадку».

```

%фрактальная лесопосадка
beginfig(1) ;
u:=1mm; branchrotation := 50;
offset := 180-branchrotation;
thinning := 0.7;
shortening := 0.8;
def drawit(expr p, linethickness,col) =
draw p withpen pencircle scaled linethickness withcolor col;

```

```

enddef;
%A - основание B - направление роста, n - число ветвей,
%size - толщина дерева, col - цвет
vardef tree(expr A,B,n,size,col) =
save C,D,thickness; pair C,D;
thickness := size;
C := shortening[B, A rotatedaround(B,
offset+uniformdeviate(branchrotation))];
D := shortening[B, A rotatedaround(B,
-offset-uniformdeviate(branchrotation))];
if n>0:
drawit(A-B, thickness, col);
thickness := thinning*thickness;
tree(B, C, n-1, thickness,col);
tree(B, D, n-1, thickness,col);
else:
drawit(A-B,thickness,col);
thickness := thinning*thickness;
drawit(B-C, thickness,col);
drawit(B-D, thickness,col);
fi;
enddef;

```

```

numeric nbr,nx,ny,ell,size;
color col;
nx:=10;ny:=5;
pair A;
for ix:=1 upto nx:
for iy:=1 upto ny:
nbr:=4+uniformdeviate 5;
ell:=nbr*u;
x:=ix*(1+1/20*normaldeviate);
y:=iy*(1+1/20*normaldeviate);
A:=(20u*(x+y*sqrt(2)/2),20u*y*sqrt(2)/2);
size:=ell/5;
col:=(uniformdeviate 1,uniformdeviate 1,uniformdeviate 1);
show ix,iy,A,ell,nbr,size,col;
tree(A, A+(0,ell), nbr, size,col);
endfor;
endfor;
endfig;

```

Увеличение автоматически создаваемыми картинками влечёт за собой опасность переполнения памяти. *MetaPost* создавался в то время, когда к используемой памяти относились исключительно бережно.

Если при компиляции картинки будет выдана ошибка *MetaPost capacity exceeded*, то необходимо поправить файл конфигурации

Фрактальный лес



**texmf.cnf.** Обычно этот файл можно найти в директории  $\$(\text{TEXMF})/\text{web2c}/\text{texmf.cnf}$ .

За выделяемый объём памяти для *MetaPost* отвечает переменная `main_memory.mpost`. Значение этой переменной должно быть меньше значения `main_memory`, которое определяет верхнюю границу использования памяти для всех TeX-подобных программ. Обратите внимание на то, что в конфигурационном файле может быть несколько переменных с одним и тем же именем — используется последнее значение.

В моём случае ограничение по памяти составляло 15 Мб. Очевидно, что размер используемой памяти можно безболезненно увеличить. С другой стороны, если происходит переполнение, то это, возможно, значит, что в алгоритм закралась ошибка.

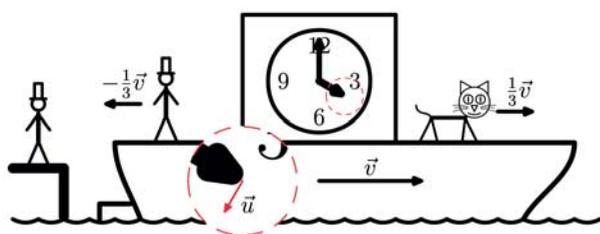
После исправления значения переменных следует заново сгенерировать форматные файлы, например, с помощью команды `texconfig init`.

## Увеличительное стекло

Бывает, в процессе создания картинку необходимо увеличить участок для лучшей детализации. *MetaPost* предоставляет средство для этого в виде команды `clip`, которая позволяет обрезать картинку по любому замкнутому пути. Ниже идёт код, который позволил увеличить кусок циферблата корабельных часов.

```
%Файл dop.mp
R:=3.6u;
path q;
%определяем форму области которую хотим вырезать
q=(-R,0)..(R,0)..cycle;
%определяем центр вырезаемой области (orig)
%и местоположение увеличенного участка (copy)
pair orig,copy;
orig=(65u,24u);copy=(45u,7.5u);
%отмечаем вырезаемую область
draw q shifted orig dashed evenly scaled 1/2u
withpen pencircle scaled 0.2u withcolor red;
picture p;
%сохраняем текущую картинку в переменной p
p:=currentpicture;
%коэффициент увеличения
numeric scale;scale:=3;
%Обрезаем картинку p по замкнутому пути q
clip p to (q shifted orig);
%чистим область, где будет нарисована увеличенная копия
fill q scaled scale shifted copy withcolor white;
%рисуем копию
draw p shifted -orig scaled scale shifted copy;
%рисуем дополнительную стрелку и метку
%на уже увеличенной копии
draw copy withpen pencircle scaled 1u withcolor red;
drawarrow copy--(copy+7u*dir -120)
withpen pencircle scaled 0.4u withcolor red;
label.lrt(btex {\vec{u}}) etex,
1/3[copy,(copy+7u*dir -120)];
```

Пользуясь этим приёмом, можно создать любой фон для какой-либо фигуры, сделать любую штриховку.



Преобразование Галилея

## Штриховка

`clip`-технология, которая упоминалась выше, позволяет создать любой вид штриховки. В стандартной поставке *MetaPost* идут пакеты, которые этим пользуются.



Пакет *hatching*

Пакет *hatching.mp* представляет из себя обычный «хак»:

```
%Файл dop.mp
%пример использования пакета hatching
input hatching;
beginfig(3);
numeric u;u:=1mm;
path q;
q=(10u,0)..(dir -135)(0u,0u){dir 135}..
(-10u,0)..(dir 90)(0u,0u){dir -90}..cycle;
hatchfill q withcolor red withcolor (25,1u,-1);
draw q;
endfig;
```

Функция `hatchfill` берёт информацию о штриховке из данных, которые следуют с декларацией `withcolor`. Сигналом, что эти данные предназначены именно для `hatchfill`, является то, что голубая компонента тройки чисел меньше 0. Красная компонента соответствует углу наклона штриховки, зелёная — расстоянию между штрихами. Подробности можно узнать в краткой документации к пакету.

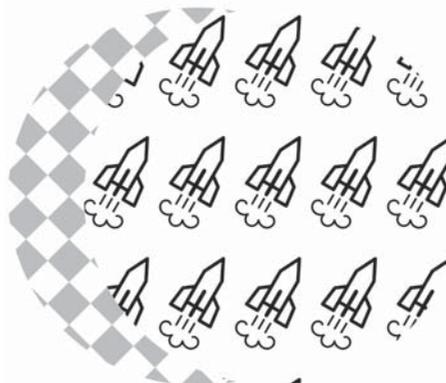
Более развитым (возможно, излишне) является пакет *mpattern*:

```
%создаём "обои" ракета
beginpattern(rocket);
begingroup;save u;
u:=1mm;
draw Rocket scaled 2/3u rotated -30;
patternbbox(-5u,-8u,5u,8u);
endgroup;
endpattern;

%создаём "обои" повернутая клетка
beginpattern(rotated_checker);
fill unitsquare scaled 4mm rotated 45 withcolor .7white;
endpattern;

beginfig(4);
numeric u; u:=1mm;
```

>>



Пакет *hatching*

```

path p;
z1=(10u,0u);
p=fullcircle scaled 50u;
%рисует фигуру в клетку
fill p withpattern rotated_checker;
unfill p shifted z1;
%рисует фигуру "в ракету"
fill p shifted z1 withpattern rocket;
endfig;
    
```

Здесь уже определены свои команды для создания штриховки. Пакет не свободен от недостатков, но он достаточно прост, и можно легко «довести» код до необходимой кондиции. Простота — это общее свойство пакетов *MetaPost*. МЕТА вынуждает писать кратко. Подробности можно найти в документации к пакету.

### Вставка eps

Пакет *exteps.mp* позволяет включить eps-картинку как единый объект. Краткая документация доступна в файле **exteps.pdf**, поставляемом с этим пакетом.

```

%Файл dop.mp
input exteps;
%Включение eps-картинки
beginfig(6);
numeric u;u:=1mm;

% базовая надпись
for alpha:=90 step 3 until 0:
label(btex LinuxFormat в России etex
scaled (5*(1+alpha/100)) rotated alpha,(0,0)
withcolor
(uniformdeviate 1,uniformdeviate 1,uniformdeviate 1);
endfor;
    
```

```

% посадим Тукса (penguin.eps) справа
begineps "penguin.eps";
% ширина картинки
width:=30u;
% сдвиг картинки от начала координат
base:= (60u,5u);
% можно нарисовать решётку на картинке
% grid := true;
% обрезание по bounding box
% clip := true;
endeps;
    
```

Добавляем  
чёрно-белых Туксов

```

% посадим Тукса (penguin.eps) слева
begineps "penguin.eps";
% поворот
angle:=90;
width:=30u;
base:= (-30u,-40u);
endeps;
endfig;
    
```

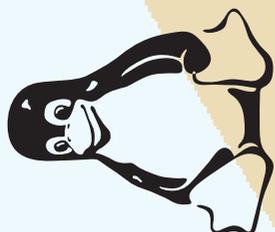
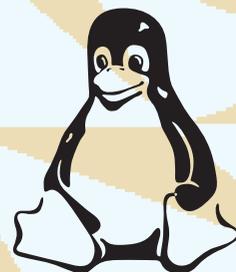
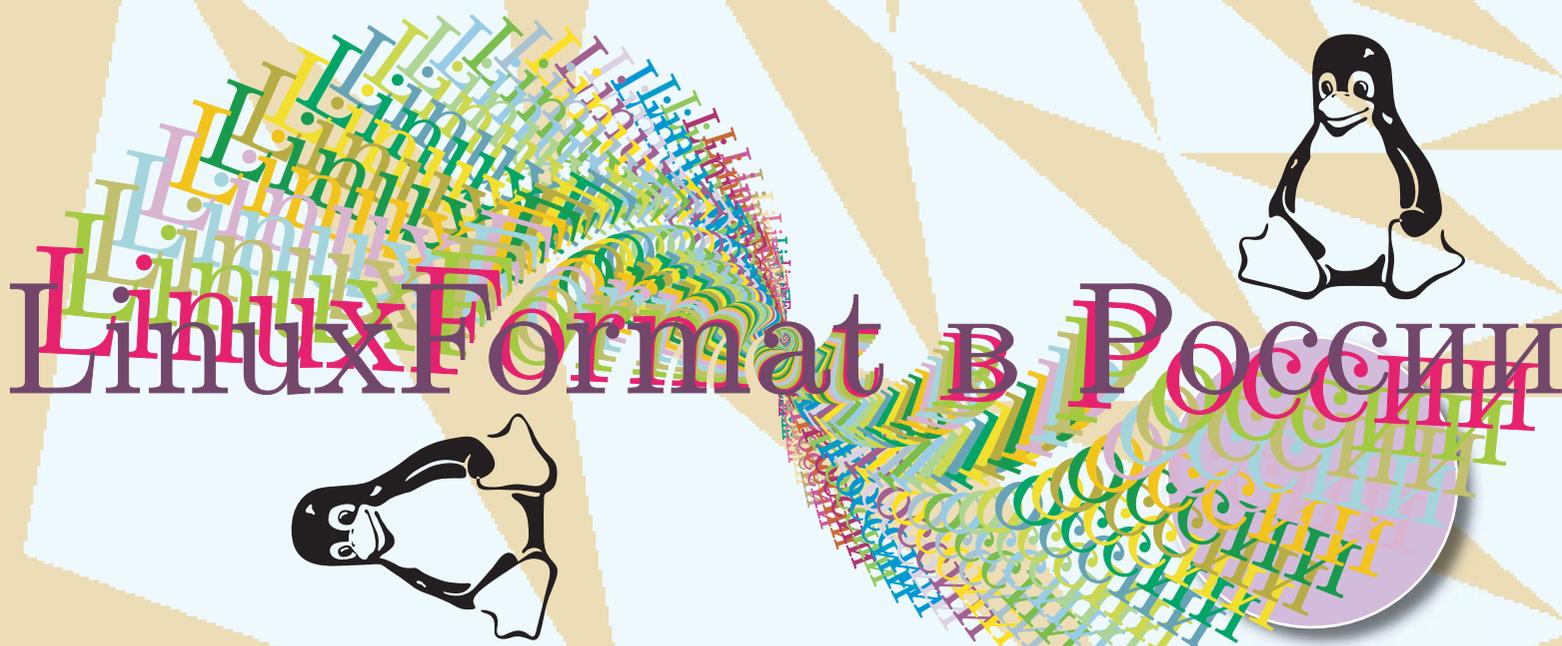
### Большие числа

Когда обсуждалась вставка меток, в качестве примера был представлен треугольник Паскаля. Из-за ограничений МЕТА на максимальный размер числа треугольник Паскаля отрисовывался до 14-й строки.

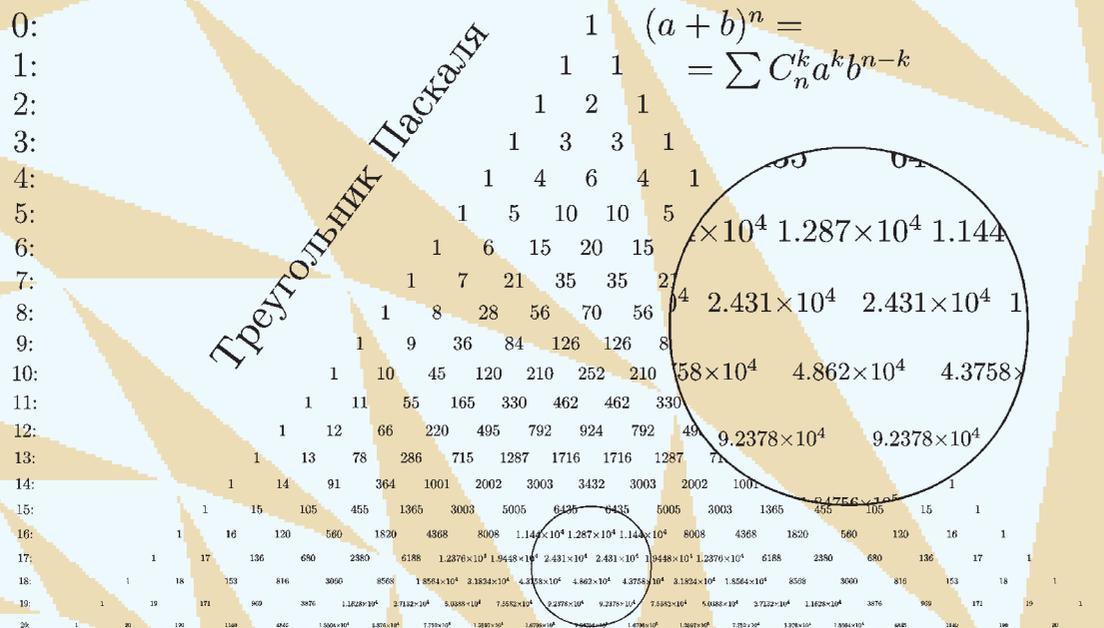
Это ограничение можно обойти, воспользовавшись пакетом *sarith.mp*. Он был создан специально для пакета *graphics.mp*, что вполне естественно, так как если при рисовании незначительных рисунков больших чисел быть не может, то при анализе данных их наличие — вполне рядовая ситуация.

```

%Файл dop.mp
%Macros for arithmetic on strings that represent big numbers
input sarith;
%Треугольник Паскаля с большими числами
beginfig(5);
numeric u;
u = 1.mm;
numeric dy,dx,x,y,i,j,sy,ds,nlast;dy:=5u;
dx:=6u;x=0;y=0;nlast:=20;
%для хранения больших чисел нужна строка
string n[][];
ds=0.04;sy=0.032;
picture z;
for i:=0 upto nlast:
dy:=dy*(1-sy);
y:=y-dy;
for j:=0 upto i:
if (j=0) or (j=i):
n[i][j]:=Sabs 1; %сложение
else:
n[i][j]:=n[i-1][j-1] Sadd n[i-1][j]; %сложение
fi
%формат вывода
z:=thelabel(format("%6g", n[i][j]),(0,0));
x:=dx*(j-i/2);
    
```



Треугольник Паскаля



```
label(z scaled (1-ds*i),(x,y));
endfor
z:=thelabel.lft(decimal(i)&"", (0,0));
label(z scaled (1-ds*i),(dx*(-nlast/2-1),y));
endfor
endfig ;
```

Для представления больших чисел используется встроенный тип *string* – строка.

Оператор	Действие
Scvnum <number>	конвертация в numeric
Sabs <number>	абсолютное значение
<number> Sadd <number>	сложение
<number> Ssub <number>	вычитание
<number> Smul <number>	умножение
<number> Sdiv <number>	деление
<number> Sleq <number>	сравнение <=
<number> Sneq <number>	сравнение <>

Документация к пакету *sarith.mp* является частью документации к *graphics.mp* (**mpgraph.pdf** от Джона Хобби).

**Макрос TEX**

Вы, наверное, уже обратили внимание, что метки в *MetaPost* являются статическими. Всё, что между *btex* и *etex*, отдаётся *LaTeX* для обработки. Это значит, что метку нельзя скомпоновать из различных кусков. Решением является немного модифицированный макрос из стандартного пакета *TEX.mp*:

```
%Файл macros.mp
vardef TEX primary s =
write "verbatim" to "mptextmp.mp";
write "input{preheader-base}" to "mptextmp.mp";
write "begin{document}" to "mptextmp.mp";
write "etex" to "mptextmp.mp";
write "btex "&s&" etex" to "mptextmp.mp";
write EOF to "mptextmp.mp";
scantokens "input mptextmp"
enddef;
```

В процессе вызова макроса *TEX* всё, что находится внутри него, записывается в **mptextmp.mp**, а затем этот файл включается в основной файл прямо во время компиляции. То есть происходит модификация программы прямо во время компиляции. Пример использования макроса приведен ниже:

```
%Файл coord.mp
numeric u;u:=2mm;
for i:=15 step 5 until 15:
label.lft(TEX("&decimal(i)&"),(-10u,i*u));
endfor;
```

При формировании ASCII-строки использовался оператор *&* для объединения. Хотя в данном случае можно было обойтись только статическими записями, но иногда этот приём может пригодиться для целей автоматизации, где скорость компиляции – не главное.

Основной минус этого способа в том, что он очень «мееедленный». Для ускорения следует оставить в **preheader-base.tex** только самые необходимые инструкции. Что-то вроде:

```
\documentclass[12pt]{article}
\usepackage[warn]{mathtext}
\usepackage[T2A]{fontenc}
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
```

**Заключение**

Естественно, в таком кратком обзоре невозможно рассказать всё. Основные понятия и приёмы уже изложены, но множество вещей выпало из обзора. В частности, совсем не рассмотрены 3D объекты, создание геометрических чертежей, параметризация пути и векторные поля в *MetaPost*. Это не смертельно, так как есть весьма качественная свободная литература, посвящённая этим предметам.

*MetaPost* достаточно легко использовать после обучения, но обучение перед использованием просто необходимо. Это свойство всех стоящих технологий, которые позволяют делать то, для чего компьютеры и существуют: автоматизировать рутинные действия, для выполнения которых не требуется задумываться. Думать же, в любом случае, prerогатива человека. **LXF**

**NOTA BENE!**  
 Вас заинтересовал *MetaPost*? Хотите с помощью этой технологии решить свою проблему? Напишите на форум *Linux Format* ([www.linuxformat.ru](http://www.linuxformat.ru)) – возможно, ваш вопрос ляжет в основу новой статьи!

# Вопрос? Ответ!

Если вы завязли в какой-то проблеме и чтение HOWTO не помогает, почему бы не написать нам? Наши эксперты помогут разобраться даже в самых сложных вопросах.

## НАШИ ЭКСПЕРТЫ

Наши эксперты найдут ответ на самый трудный ваш вопрос. Если у вас проблемы с установкой, настройкой модема, сетью или еще чем-нибудь – просто напишите нам, а обо всем остальном позаботимся мы.

Управляя Интернет-провайдером, а заодно подрабатывая редактором дисков **LXF**, **Нейл Ботвик** (Neil Bothwick) скромно зовет себя мастером на все руки.



**Сергей Владимирович Рыжиков** Генеральный директор компании «Битрикс»



**Максим Владимирович Смирнов**



Ведущий программист отдела разработки компании «Битрикс»

**Александр К.** – сторонник Unix-way. Молодой, но перспективный член дружной команды экспертов.



**Валентин Сеницын** Поддерживает проект Slackware Reiser4, интересуется настольными Linux-технологиями и рад помочь Вам разобраться с ними.



## Grub-Grub

**В** Я пытался клонировать свой системный жесткий диск с Linux – SCSI-диск, если это важно знать. У меня есть новый диск, размеченный как системный: на swap-раздел и корневой (ext3). Для /home у меня отведен отдельный жесткий диск.

Я загрузился с KNOPPIX, смонтировал оба диска и скопировал всю информацию с помощью конвейера из *dump* и *restore*. Теперь у меня есть надежный дубликат. После всего этого нужно было сделать новый диск загрузочным, и тут-то я и застрял.

Старый диск – **/dev/sda (sda1: swap, sda2: ext3) : grub : hd0**.

Новый диск – **/dev/sdb (sdb1: swap, sdb2: ext3) : grub : hd1**.

Когда я вынимаю старый диск и пытаюсь загрузиться, я получаю “Grub boot disk error” или что-то подобное. Чего я только ни перепробовал, но похоже на то, что я установил Grub туда, откуда он не загружается. То есть когда я загружаюсь с **/dev/sda** и устанавливаю Grub на **hd1**, а затем вынимаю **/dev/sda**, **hd1** становится **hd0** и Grub не может найти диск. Если все именно так, то что же делать?

### С форума LXF

**О** Grub нумерует диски в том порядке, в каком они определены в BIOS. Когда Вы удаляете первый диск, оставшиеся диски перенумеровываются, и **hd1** становится **hd0**, как Вы правильно заметили.

Есть два пути решения этой проблемы. Простейший способ потребует LiveCD с включенным в него Grub, например, KNOPPIX или минимальный установочный диск Gentoo (который значительно меньше, чем KNOPPIX). После отключения старого диска, загрузитесь с LiveCD. Если у Вас KNOPPIX, напечатайте **knoprix 2** в параметрах загрузки, чтобы загрузиться в командную строку от имени суперпользователя. Теперь установите Grub на жесткий диск, как Вы делали это ранее:

```
Grub-install /dev/sda
```

Если **Grub-install** выдаст ошибки, сделайте установку вручную через собственную консоль загрузчика:

```
Grub
root (hd0,0)
```

```
setup (hd0)
quit
```

Предполагается, что Вы уже отредактировали **/etc/fstab** и **/boot/Grub/menu.lst** и заменили **/dev/sdb** на **/dev/sda**.

Другой способ – воспользоваться функцией смены порядка жестких дисков, доступной в некоторых версиях BIOS. Пока на новом диске нету Grub, компьютер будет загружаться со второго диска, но когда Вы запустите Grub для установки на новый **hd0**, загрузка будет происходить с первого диска. После этого верните прежний порядок загрузки. Этот метод работает далеко не на каждом компьютере, поэтому к нему стоит прибегать только в случае невозможности загрузки с LiveCD. **НБ**

## Защита Apache'й

**В** Не могли бы вы рассказать о генерации SSL-сертификата и применении его к Apache на серверах под управлением Red Hat Enterprise Linux 4 и Fedora Core 4?

### Джимми [Jimmy]

**О** Настройка безопасных соединений в Apache на серверах под RHEL4 и FC4 – одна из тех полезных вещей, которым можно научиться, администрируя Apache. Абсолютное большинство коммерческих

web-сайтов с публичным доступом используют сертификаты, подписанные специальным ключом для обеспечения наивысшего уровня безопасности.

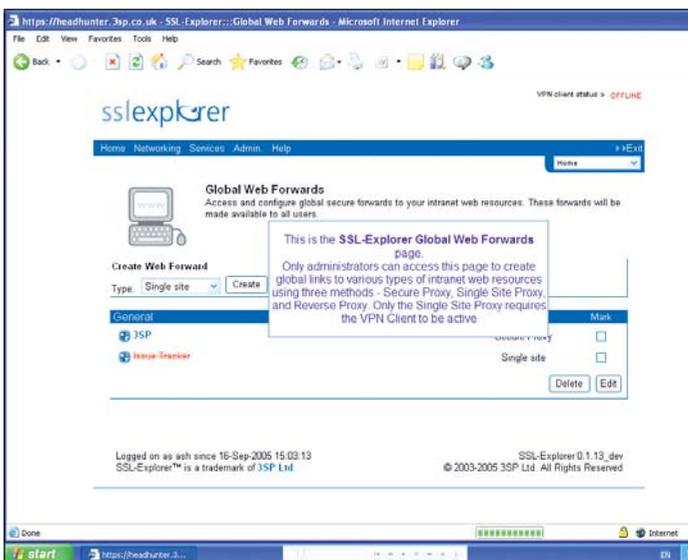
Вы можете создать такой ключ с помощью *OpenSSH* ([www.openssl.org](http://www.openssl.org)), который, как я полагаю, установлен у Вас как стандартный системный компонент. Для начала создайте личный (приватный) ключ. Вы можете защитить его паролем, но я бы не советовал это делать, поскольку при каждом перезапуске web-сервера Вам придется вводить этот пароль, а сервер будет простаивать.

Ознакомьтесь со справочными материалами по пакету **openssl**, чтобы создать сертификат от имени суперпользователя. Перед заменой старого сертификата перенесите его в безопасное место:

```
mv /etc/httpd/conf/ssl.* /server* /root/
```

После этого смените текущий каталог на **/etc/pki/tls/certs** (FC4) или **/usr/share/ssl/certs/** (RHEL4) и запустите **make testcert**. Вас спросят о пароле, который мы потом удалим. Заполните остальные поля. Самое важное из них – Common Name [], куда Вы должны вписать имя домена, для которого Вам понадобился сертификат.

После генерации ключа все файлы попадут по местам. Вы должны убедиться,



SSL-Explorer запускается на Linux, но доступен из любой ОС через обычный браузер (даже Internet Explorer).

что конфигурационный файл *Apache mod\_ssl* (`/etc/httpd/conf.d/ssl.conf`) содержит верную информацию – параметры `SSLCertificateFile` и `SSLCertificateKeyFile` должны указывать соответственно на сертификат и ключ.

Теперь удалите пароль, если Вы хотите, чтобы *Apache* мог свободно запускаться и перезапускаться без какого-либо ручного вмешательства:

```
cd /etc/httpd/conf/ssl.key/
openssl rsa -in server.key -out server.nopassphrase.key
mv server.key server.key.orig
mv server.nopassphrase.key server.key
chkconfig httpd on
```

Это все, что нужно для обеспечения обработки HTTPS-запросов на FC4 и RHEL4 для документов из DocumentRoot. Естественно, для каждого TLS/SSL-сайта у Вас должен быть отдельный IP-адрес. **КК**

### Продаться в SSL

В Я вообще-то администратор Windows, но умею работать и с Linux, и меня уже тошнит от стоимости и обслуживания Windows. Ищу решение для замены нашего VPN-сервера на Windows на Linux-сервер.

Я надеялся, что буду использовать открытый SSL VPN, которым можно управлять через браузер, но ничего подобного мне найти не удалось. Может быть, вы расскажете, что сейчас самое крутое на рынке Linux VPN? Я знаю, что *FreeSwan* популярен, но это IPsec. *OpenVPN* выглядит отлично, но он не запускается через браузер. Я рассматривал и коммерческие продукты (*SmoothWall*), но я хочу сделать все сам (и к тому же сэкономить).

Мой коллега, работающий с Unix больше 20 лет, хочет использовать SSH для VPN, но я слышал, что это потребует немалых ресурсов процессора и снизит производительность.

**А. Дж. Берч [A. J. Burch]**

В данной области преобладают коммерческие продукты, и, неудивительно, т.к. подобные решения рассчитаны прежде всего на корпоративных пользователей. Но есть и открытый проект, *SSL Explorer*. Скорее всего он вам и подойдет – это SSL VPN, с доступом из любого стандартного web-браузера.

*SSL Explorer* доступен на <http://sourceforge.net/projects/sslexplorer>. С точки зрения экономии это прекрасное решение. Но если Вы собираетесь обеспечивать доступ к коммерческой сети, возникает проблема безопасности и финансовой ответственности в случае неправильной настройки. Если у Вас мало опыта, разумно будет выбрать коммерческое решение вроде *SSL-Explorer Xtra* (\$490 за лицензию для пользователей числом до пяти), включающее дополнительные программы и, самое главное, техническую поддержку.

Любая форма шифрования каналов отнимает ресурсы. Это относится и к SSH, и к SSL, так что позаботьтесь, чтобы Ваш сервер вынес дополнительную нагрузку. Преимущество SSL в том, что использование сертификатов дает гарантию соединения с нужным сервером, и Ваши данные не утекут в неизвестном направлении. **НБ**

### Вечный RAID?

В Недавно я настраивал файл-сервер на базе SUSE Enterprise Server 9. В системе три жестких диска: один на 80 Гб и два по 120 Гб. На первом установлена система. Два других собраны в RAID 1 и являются хранилищем пользовательских данных. RAID реализован программно, файловая система – ReiserFS. Все работает отлично, и, надеюсь, будет работать так еще долгое время.

Однако на каком-то этапе один из зеркалированных дисков даст сбой, и потребуются его замена. Будет ли в *LXF* статья о замене «разбитого зеркала» и восстанов-



лении данных с другого диска? И каким методом или утилитой можно справиться о здоровье RAID? Информации о достоинствах и об организации RAID хватает, а вот об обслуживании и восстановлении пишут очень мало.

**Найджел Куэйл [Nigel Quayle]**

О Подробная статья о RAID – идея хорошая. А пока краткий обзор. Есть несколько способов проверки состояния RAID-массива. Следующий код выдает краткую информацию о массиве:

```
cat /proc/mdstat
mdadm --detail /dev/md*
```

Программа *mdadm* также запускается и в режиме демона (фоновом). Вы должны отредактировать `/etc/mdadm.conf` и протестировать все в командной строке, после чего можно ставить *mdadm* на автозагрузку (Yast > System > System Services). Если возникнут какие-либо проблемы, Вам придет уведомление по электронной почте.

В случае сбоя одного из дисков, RAID 1 будет работать на оставшемся исправном диске. Для замены неисправного диска нужно сначала удалить его из массива:

```
mdadm /dev/mdX --fail /dev/hdYn --
remove /dev/hdYn
```

где **mdX** и **hdX** – массив и раздел соответственно. После этого отключите питание, замените диск, создайте на новом диске нужные разделы, а затем добавьте их в массив:

```
mdadm /dev/mdX --add /dev/hdYn
```

и RAID-массив будет перестраиваться автоматически; в это время Вы ощутите некоторое падение производительности. По любой из двух команд для проверки массива можно выяснить, закончена ли его перестройка.

Вместо *mdadm* можно воспользоваться пакетом утилит *raidtools*, но *mdadm* мне нравится больше – он новее и логичнее. Для слежения за состоянием самих дисков можно установить *smartmontools*. **НБ**

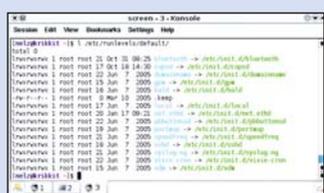


## КРАТКАЯ СПРАВКА: ССЫЛКИ

Решение проблемы с игрой Argonium требует создания символической ссылки, поскольку программа ожидает найти файл в своей директории, а он находится совершенно в другом месте. Можно, конечно, просто скопировать файл, однако это создает некоторые неудобства: расходуется дисковое пространство, и если файл обновится, его копию тоже придется обновлять. Файловые системы в Linux имеют возможность доступа к одному и тому же файлу из нескольких мест, путем ссылок. Есть два типа ссылок – жесткие и символические. Технически, каждый файл – это жесткая ссылка от имени к содержимому (данным на диске); но

обычно, под термином «жесткая ссылка» подразумевают дополнительную ссылку, т.е. один файл может иметь несколько имен. Однако применение жестких ссылок ограничено: они должны указывать на файл в той же файловой системе, и не допускаются жесткие ссылки на директории.

Символические ссылки гораздо более гибки, и вы можете увидеть их в любой Linux-системе, просмотрев содержимое директории командой `ls` с ключом `-l`. Классический пример – директория `/usr/src`, где `/usr/src/linux` является символической ссылкой на директорию с исходными кодами текущего ядра. То есть можно иметь исходные коды не одного ядра, а несколь-



### Символических ссылок полно в директории /etc.

ких, а приложения находят нужное через `/usr/src/linux`. Множество ссылок находится в `/etc/rc.d`.

Символические ссылки создают так:

```
ln -s /путь/к/файлу /путь/к/ссылке
```

Оба пути могут быть относительными.

# ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ: BLUETOOTH

**FAQ Bluetooth [в переводе – синий зуб] звучит как нечто связанное с зубными врачами, но, наверно, это все-таки про компьютеры?**  
Верно. Bluetooth – это беспроводный протокол для персональных сетей (PANs).

**FAQ Но беспроводных протоколов и так хватает: 802.11 а, в, г и прочие. Нужен ли еще один?**  
Bluetooth от них отличается. Он создан в конце 90-х годов для ближней связи недорогих устройств с низким энергопотреблением. Его можно рассматривать как

замену IrDA. Поскольку Bluetooth использует радиоканал, он не требует, чтобы устройства были в зоне прямой видимости, как при инфракрасной связи, и не так подвержен помехам.

**FAQ Значит, это не сетевая штука?**  
Хотя с помощью Bluetooth и можно связать два компьютера, но он не предназначен для этого. Bluetooth задуман для связи мобильных телефонов, КПК, смартфонов с компьютером или друг с другом. Он является удобной заменой соединительных кабелей: не нужно мучиться с согласованием разъемов.

**FAQ Как это работает с Linux?**  
Ядро Linux содержит поддержку стандартных Bluetooth-устройств. Кроме того, понадобятся утилиты из пакета **bluez-utils**. Это имя одинаково для многих дистрибутивов, но в некоторых пакет называется по-другому. Воспользуйтесь вашим пакетным менеджером для установки пакета, заодно будут разрешены зависимости вроде **bluez-libs**.

**FAQ И нужно установить все утилиты bluez?**  
Нет, нужны не все – некоторые даже могут конфликтовать. Например, пакет **bluez-kernel** годится только для старых ядер.

После версии 2.6 его функции встроены в ядро.

**FAQ А вдруг кто-то захочет прочитать мою информацию через Bluetooth-устройства, на свой КПК или ноутбук?**  
Устройства должны «спариться» перед обменом информацией. Чтобы устройство А соединилось с устройством Б, устройство Б должно быть в режиме поиска (то есть открыто для соединений), и пользователь должен ввести PIN устройства Б в устройство А. Каждое Bluetooth-устройство имеет четырехзначный PIN. Для беспроводных гарнитур это обычно 0000, однако у любого устройства, хранящего данные, PIN можно поменять. Так что для обмена данными активны должны быть оба устройства.

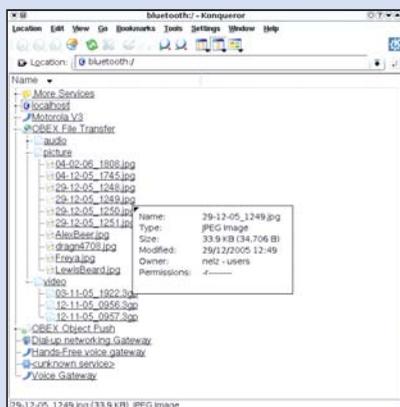
**FAQ Что можно делать, когда мой компьютер подключен к другому устройству?**  
Вы можете пересылать данные, напри-



Смартфон Motorola A910 – одно из Bluetooth-устройств, поддерживающих Linux.

мер, фотографии и мелодии. Пользователи KDE могут просмотреть список доступных Bluetooth-устройств через *Konqueror*, введя `bluetooth://` в адресную строку. *KMobileTools* (<http://kmobiletools.berlios.de>) синхронизирует ваш телефон с компьютером и его адресной книгой; можно даже делать звонки с компьютера.

**FAQ Ну а почему все-таки Bluetooth?**  
В честь короля Дании Гаральда Синезубого, который славился умением вызывать на разговор друг с другом самые разные группы людей. Да, у него и вправду были синие зубы.



Konqueror просматривает содержимое мобильного телефона.

## << Argonium, на старт!

**В** Пытался запустить игру *Argonium* на Ubuntu 5.10 с диска *LXF73*, но запуск почему-то не получается. Я распаковал игру и запустил в ее каталоге `./Argonium`, а она выдала следующие сообщения:

```
couldn't exec config.cfg
/dev/dsp: Broken pipe
LoadLibrary(".\refresh.so")
ref_gl version: GL 1.0
./libGL.so: cannot open shared object file: No such file or directory
Segmentation fault
```

Я в Linux новичок, и совершенно не понимаю, что происходит. Искал `config.cfg`, но не нашел. Может, проблема как раз в этом, но пожалуйста, помогите мне!  
**Джеймс Кено [James Cuenod]**

**О** Здесь сообщения о нескольких ошибках, и не все они критичны. Например, первая, про `config.cfg`, ни на что не влияет. Конфигурационный файл не найден, потому что Вы ни разу не запускали игру и не меняли настроек. Когда Вы сделаете это, игра создаст конфигурационный файл в Вашей домашней директории (`~/.Argonium/data`). Следующее сообщение, про `/dev/dsp`, посерьезнее: оно означает, что звуковых эффектов не будет. `/dev/dsp` – это файл звуковой карты для OSS, старой звуковой подсистемы, пока еще используемой многими программами. Новая подсистема ALSA может эмулировать OSS. Для этого загрузите нужный модуль ядра:

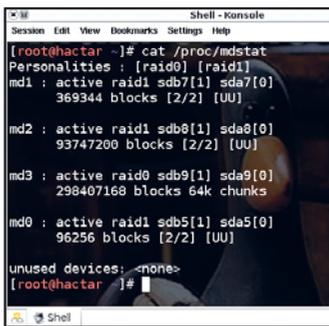
```
sudo modprobe snd_pcm_oss
Чтобы модуль загружался автоматически, добавьте его в /etc/modules:
echo snd_pcm_oss >> /etc/modules
```

Следующая ошибка и того хуже. *Argonium* пытается загрузить библиотеку

`libGL.so` из текущей директории, а на самом она находится в `/usr/lib`. Ситуацию исправит символическая ссылка (см. врезку):

```
ln -s /usr/lib/libGL.so.1 libGL.so
```

После этого игра должна запускаться безо всяких ошибок. Обратите внимание, что Вам нужна видеокарта с аппаратным 3D-ускорением и настроенным видеодрайвером.



`cat /proc/mdstat` – для просмотра статуса массива, `mdadm` – для постоянного мониторинга.

ром. Для видеокарт NVIDIA драйверы можно взять на сайте [www.nvidia.com/object/unix.html](http://www.nvidia.com/object/unix.html). **HS**

## Blueyonder бузит

**В** Успешно установил SUSE 10.0 с рабочим окружением KDE, но не могу заставить подключаться через USB-порт к Интернету модем Motorola SB4200 от моего провайдера Blueyonder.

Все форумы советуют подключаться через NIC-соединение, но может, вы располагаете информацией, как заставить SUSE работать с моим модемом?

**Роберт Холдкрофт [Robert Holdcroft]**

**О** Есть множество причин, по которым люди советуют выбирать Ethernet-модемы. Вот главные три:

- Ethernet быстрее
- Ethernet гораздо проще в настройке
- Ethernet быстрее



**Argonium** нелегко заставить работать, но труды окупаются.

Разница в скорости просто огромна. Я не пробовал Ваш модем, но на моем разница между USB- и Ethernet-подключениями ошеломляет. Еще бы, ведь Ethernet создан для сетевых соединений, а USB – всего лишь универсальная шина для низкоскоростных устройств.

Если на Вашем компьютере нет встроенной сетевой карты, можно купить PCI-вариант Ethernet-контроллера меньше чем за 10 долларов и жить без забот о настройке подключения в SUSE. Возможно, понадобится зарегистрировать MAC-адрес (уникальный аппаратный идентификатор) в Blueyounder – фирма использует его для авторизации. Чтобы узнать MAC-адрес, запустите Yast и перейдите в Hardware > Hardware Info. Выберите свою сетевую карту, а затем Resources > Hwaddr. Обратитесь в службу поддержки Вашего провайдера и сообщите этот номер.

Соединение через USB устанавливается драйвером CDCether, включенным в стандартное ядро SUSE 10.0, а значит, модем обязан «просто работать». Распознает ли SUSE модем, когда Вы его подключаете? Если распознает, но не создает для него сетевого интерфейса, сделайте это сами через секцию Network Device в Yast: выберите USB и Automatic Address Setup. **НБ**

### Семейные раздоры

Покупка вашего журнала стала моим первым шагом в мир Open Source. Когда я услышал об идее свободного ПО, она мне очень понравилась, однако я чувствовал, что польза от нее возможна только при высоком уровне квалификации. Компьютеры стоят недешево, и я не хотел бы угрожать своей.

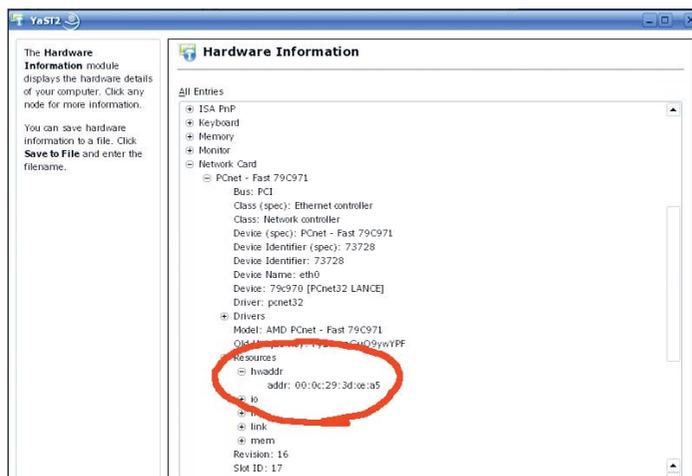
У меня есть доступ к эмулятору PC на Mac G4, и я решил, что ничего не потеряю,

установив на него Slackware. Система работала просто отлично. Конечно, я немного растерялся, когда машина сказала мне darkstar login, но обратившись на форумы, нашел ответы на свои вопросы. Пока все было неплохо.

Вдохновившись первой победой, я отважился на следующий шаг: установить систему на ноутбук моей жены, Advent 7081 CELM350, в дополнение к Windows, и надеялся, что потом мы оба от Windows откажемся. Установка не удалась, и я запаниковал. По-моему, проблема связана с нехваткой в устанавливаемой системе каких-то драйверов или файлов. Что теперь делать, не знаю. Ко всему прочему, теперь при попытке загрузки я получаю сообщение 'OS отсутствует'.

На наш брак это повлияло далеко не лучшим образом. Вам знакомы подобные ситуации? То немногое, что я знаю о дистрибутивах и сообщество, произвело на меня такое хорошее впечатление, что бросать идею не хочется.

**Алекс**



**Yast** покажет вам MAC-адрес сетевой карты, который вы должны сообщить провайдеру.

О Невозможность загрузки компьютера из-за неудачной установки – случай необычный. Загрузчик, как правило, прописывается в главную загрузочную запись в самом конце установки, а до этого момента перезагрузка будет запускать прежнюю ОС. Разумно предположить, что сбой произошел именно при установке загрузчика. Для исправления ошибки, загрузитесь с установочного диска Windows в режиме восстановления (rescue) и введите команду **fixmbr** (это для Windows XP; если у вас Windows 98, то команда другая – **fdisk /mbr**).

При установке Linux на ноутбуки трудности бывают, поскольку в ноутбуках часто используется нестандартное оборудование, а эмуляторы, напротив, предусматривают в основном стандартные компоненты. Самый безопасный вариант – использовать дистрибутив с LiveCD: его можно запустить просто с компакт-диска, без всякой установки, и проверить поддержку оборудования. Вам подойдут PCLinuxOS ([www.pclinuxos.com](http://www.pclinuxos.com)), SimplyMEPIS ([www.mepis.org](http://www.mepis.org)), Kubuntu ([www.kubuntu.org](http://www.kubuntu.org)) или Knoppix ([www.knoppix.com](http://www.knoppix.com)), это полноценные дистрибутивы, работающие прямо с CD. Убедившись, что ваше оборудование поддерживается, Вы сможете опять попробовать Slackware. **НБ**

### Начинаем!

В Я заинтересован в переходе с Windows на Linux. Мне нужно создавать web-сайты, редактировать фотографии, использовать MS Office, электронную почту, Интернет и играть в авиатренажеры. Знаю, что имеются аналоги MS Office и штуки вроде VMWare для запуска Windows-программ, и это хорошо, поскольку я могу запустить некоторые Windows-программы, сохранить их на внешние накопители, а затем перенести в Linux. Я не программист, и ищу что-нибудь простое в установке и использовании. Моя конфигурация:



**Чтобы разобраться в системных требованиях вашей машины, попробуйте один из LiveCD, хотя бы SimplyMEPIS с диска LX79.**

**Материнская плата** Asus K8V SE Deluxe  
**Процессор** AMD 64 3000  
**ОЗУ** 1 Гб  
**Видеокарта** BFG 6600 OC GeForce 128 МБ

Я хочу, чтобы ОС использовала мой 64-разрядный процессор и обеспечивала мои нужды без необходимости запуска Windows (ну разве что в VMWare). Какую ОС вы мне посоветуете? В прошлый раз, когда я пробовал Linux, там не было поддержки USB.

Поддерживает ли VMWare игры? Я сыт по горло Windows, но хочу иметь возможность просматривать все web-сайты и использовать все мое оборудование.

**Дэнниэл [Daniel A Day]**

О Есть несколько 64-разрядных дистрибутивов, подходящих неопытному пользователю. Назову следующие (порядок перечисления не связан с приоритетом): Mandriva ([www.mandriva.com](http://www.mandriva.com)), SUSE ([www.suse.com](http://www.suse.com)) и Ubuntu ([www.ubuntu.com](http://www.ubuntu.com)), и они имеют LiveCD-варианты, то есть не нуждающиеся в установке и работающие прямо с компакт-диска – работают они немного медленнее, чем обычные, но зато позволяют оценить систему перед установкой (см. в этом номере обзор LiveCD).

Если из Windows-программ Вам нужен только MS Office, недорогую и тяжеловесную программу VMWare устанавливать не обязательно, достаточно будет эмулятора вроде CrossOver Office ([www.codeweavers.com](http://www.codeweavers.com)); он умеет работать под Linux не только с MS Office, но и со многими другими приложениями Windows.

Еще лучше использовать OpenOffice.org, он есть во всех популярных дистрибутивах и как минимум не хуже, а кое в чем и превосходит MS Office. Другие Ваши запросы также перекрываются ПО Linux более чем адекватно.

А вот играть в VMWare вряд ли получится – это одна из причин, по которым поль-



« зователи все еще держат Windows на своих жестких дисках. **НБ**

### Потерял память

В Недавно установил SUSE 10.0 на свой ноутбук Toshiba L10. Получилась просто конфетка, но система не распознала мой USB-диск, то есть брелок LG 1 GB. Я его вставил и переустановил SUSE, все заработало нормально.

После установки SUSE "Мой компьютер" отображал мой жесткий диск **hda2** и брелок LG 1GB. А после перезагрузки стал показывать CD-привод, жесткий диск **hda2** и другой жесткий диск, **sda1**.

**Адриан Симмондс [Adrian Simmonds]**

О Так как у Вас ноутбук, в нем скорее всего только один жесткий диск, а увиденный Вами **sda** как раз ваш USB-брелок. IDE-диски обозначаются как **hda**, **hdb** и так далее. Флэш-диск и другие USB-накопители определяются как SCSI-устройства и обозначаются **sda**, **sdb** и т.д. Номер обозначает номер раздела, т.е. **sda1** – первый раздел на первом SCSI-диске, на Вашем USB-брелке он же и единственный.

**НБ**

### Зависания повсюду

В У меня на сервере установлен Red Hat Enterprise Linux ES 4. В качестве FTP-сервера используется *Vsftp*. FTP работает вроде нормально, но я заметил, что когда соединяюсь с сервером с помощью ADSL или через брандмауэр, то не могу прочитать

содержимое директорий и закачать файлы. Никакой ошибки нет – все просто виснет, независимо от FTP-клиента. У меня установлен *Iptables*, и я думаю, что это связано с ним: когда я его отключаю, FTP работает нормально.

**Чэд Родинис [Chad Rodinis]**

О Проблема связана с модулями *Iptables*, работающими на сервере. Вы должны включить два вспомогательных модуля для *Iptables*: **ip\_nat\_ftp** и **ip\_conntrack\_ftp**:

```
modprobe ip_nat_ftp
modprobe ip_conntrack_ftp
```

Теперь *lsmod* должен выдавать что-то наподобие

Module	Size	Used by
ip_nat_ftp	4913	0
iptable_nat	23037	1
ip_nat_ftp		
ip_conntrack_ftp	72689	1
ip_nat_ftp		
ipt_LOG	6465	1

USB-брелок – удобная вещь для переноса больших объемов данных, но иногда с ними бывают осложнения.



ip_nat_ftp	1857	1
ip_conntrack	40565	4
ip_nat_ftp,iptable_nat,ip_conntrack_ftp,ipt_state		
iptable_filter	2753	1
ip_tables	16705	4
iptable_nat,ipt_LOG, ipt_state, iptable_filter		

Обратите внимание, что модули не будут загружены при перезапуске *Iptables* – придется снова запустить *modprobe*. Чтобы избежать этого, отредактируйте **/etc/sysconfig/iptables-config**, добавив в него следующие записи:

```
IPTABLES_MODULES="ip_nat_ftp ip_conntrack_ftp"
```

Теперь при перезапуске *Iptables* вы увидите вот что:

```
Loading additional iptables modules: ip_nat_ftp ip_conntrack [ OK ]
```

Это должно решить все проблемы, возникающие при использовании FTP через брандмауэр. **КК**

### 1-2-VNC

В Я произвел минимальную установку CentOS 4 на домашний компьютер, который планирую приспособить под электронную почту, DNS, web-хостинг и т.д. (в Linux я новичок). Мне понравились графические утилиты *system-config-tools* от Red Hat. Я почитал про VNC и установил vnc-сервер из RPM, однако запустить его не сумел. Не могли бы вы мне помочь?

**М. Томпсон [M. Thompson]**

О Установка *Vncserver* на CentOS/RHEL очень проста, но требует установки дополнительных пакетов. Так как у Вас минимальная установка, то понадобятся еще **vnc-server**, **xorg-x11**, **gnome-session** и **gnome-terminal**. Пакеты **xorg-x11** и **gnome-session** имеют много зависимостей, и если их устанавливать через *Yum*, да еще с медленным Интернетом, хватит времени выпить кофе.

Можно также избежать установки Gnome-пакетов, используя стандартный оконный менеджер *TWM*. Он находится в пакете **xorg-x11-twm**. Если Вы выберете этот вариант, то **~/vnc/xstartup** остается без изменений.

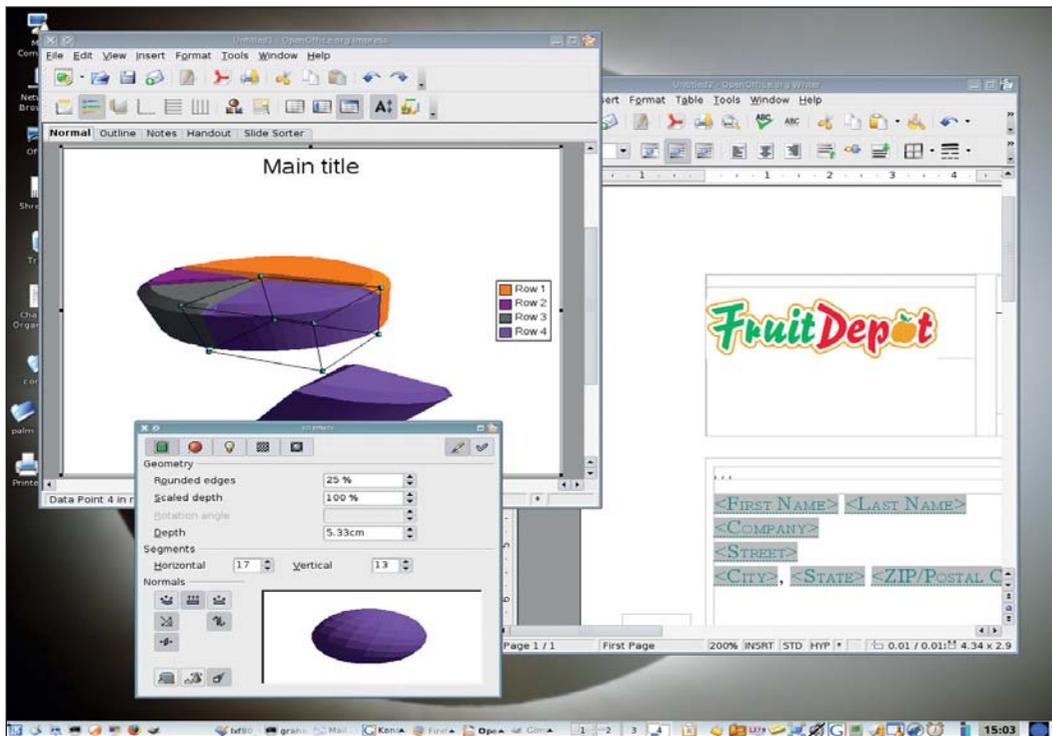
В случае с Gnome, воспользуйтесь следующим:

```
#!/bin/sh
vncconfig -iconic &
gnome-terminal &
gnome-session &
```

Сохранив этот *xstartup*, сделайте его выполняемым. Наконец, запустите VNC-сервер командой *vncserver*. Вас спросят о пароле для подключения к серверу. Если вы не укажете номер дисплея для VNC-сервера, он будет работать на первом доступном.

Теперь можно подключиться к серверу с помощью VNC-viewer, указав в качестве параметра IP-адрес сервера и номер дисплея (например, **192.168.1.10:1**).

Чтобы заставить *Vncserver* запускаться автоматически при старте системы, воспользуйтесь командой *chkconfig*. И еще, конфигурационный файл для VNC-серве-



Можно запустить MS Office и под Linux, но зачем? Есть же OpenOffice.org (на рисунке).

ра в Centos/RHEL находится в `/etc/sysconfig`. В этом файле прописаны пользователи, которые могут подключаться к серверу. **КК**

### Чистый экран

**В** Начал потихоньку осваивать Python по статьям в вашем журнале. Возник такой вопрос: Есть ли в Python оператор или функция очистки экрана, наподобие CLS в Бейсике?

**Александр**

**О**ператора CLS, конечно, нет – все-таки, Python – это не Бейсик, но способов очистить экран существует предостаточно. Самый простой (но не самый правильный) – вызвать команду оболочки `clear`, которая, собственно, и выполняет всю работу:

```
import os
os.system('clear');
```

Более корректный вариант – воспользоваться функциями стандартного модуля `curses`:

```
import curses
curses.initscr(); curses.reset_shell_mode();
curses.doupdate()
```

**BE**

### Вредный администратор

**В**Здравствуйте! Никак не могу сообщить, как правильно настроить маршрутизацию. Дело в том, что администратор сети не говорит, что конкретно необходимо сделать. Он посоветовал использовать команду «route» и перечислил следующие реквизиты:

**ip:** 212.220.210.2  
**Маска подсети:** 255.255.255.192  
**Шлюз:** 212.220.210.62  
**DNS1:** 212.220.202.158  
**DNS2:** 212.220.204.2  
**DNS3:** 81.91.52.190

Можно ли, зная эти данные настроить доступ к Интернету, и каким образом?

Администратор сети помогать не хочет, больше помощи просить не у кого.

**Сергей Иванов**

**О** Да, приведенных Вами данных вполне достаточно. Чтобы настроить доступ в сеть, нужно:

- 1) Активировать сетевой интерфейс
 

```
ifconfig <имя_интерфейса> 212.220.210.2 netmask 255.255.255.192
```
- 2) Прописать маршрут по умолчанию
 

```
route add default gw 212.220.210.62
```
- 3) Добавить DNS-сервера в файл `/etc/resolv.conf` (от имени root):
 

```
#nano /etc/resolv.conf -> вызов редактора
nameserver 212.220.202.158
nameserver 212.220.204.2
```

**BE**

## ★ ВОПРОС МЕСЯЦА!

# LINUX

ФОРМАТ

Редакция вместе с компанией Bitrix выбрали самый интересный вопрос месяца и поздравляют победителя, **Александра**, с заслуженной наградой – сертификатом на 1000 рублей в LinuxCenter.ru



### Рекламация

**В**Здравствуйте! Попробовал установить FreeBSD 6.0 с диска журнала. Все делаю согласно инструкции, но когда дело доходит уже до конкретного создания разделов выдает следующую ошибку:

```
Couldn't make filesystems properly.
Aborting.
```

```
Installation completed with some errors.
```

You may wish to scroll through the debugging messages on VTY1 with the scroll-lock feature. You can also choose "No" at the next prompt and go back into the installations menu to retry whichever operations have failed.

Создать разделы вручную тоже не получается. Что посоветуете?

**Александр**

**О**тветить на этот вопрос мы попросили штатного специалиста по FreeBSD в команде LinuxFormat Russia Сергея Супрунова:

К сожалению, на тестовой машине получить подобную ошибку не удалось, а предоставленной Вами информации недостаточно (некоторые отладочные данные можно получить на второй консоли, нажав [Alt+F2], либо прокрутив вверх первую, используя [Scroll Lock]). Попробую сделать несколько предположений:

1) FreeBSD некогда любила выдавать ошибки геометрии. В версии 6.0 ситуация

заметно улучшилась, но вам, похоже, особенно «повезло». Если в процессе установки было предупреждение типа «WARNING: A geometry of 310101/6/63 for ad0 is incorrect. Using a more likely geometry», то можно попробовать задать «правильную» геометрию (ту, которая установлена в BIOS) – `sysinstall`, несмотря на предложение всё исправить автоматически, не всегда с этим хорошо справляется. Для ручной корректировки геометрии в окне `FDISK` следует нажать [G].

Кстати, в BIOS рекомендуется использовать режим LBA. Правда, изменение геометрии уже работающего диска может привести к невозможности запустить уже установленную Windows XP (надеюсь, вы не забыли создать резервную копию ценных данных, приступая к установке FreeBSD?).

2) Если на одном контроллере с винчестером установлен, например, CD-привод, то можно попробовать разнести диск и CD на разные контроллеры: иногда BIOS выставляет «несовместимые» режимы контроллера (например, PIO, в то время как `sysinstall` пытается работать с диском в предположении, что включён DMA). Кстати, можно попробовать принудительно изменить режим работы контроллера в BIOS.

3) Ещё одна рекомендация – попробовать отключить режим DMA во время установки. Например, это можно сделать следующим образом. При загрузке с CD

перейти в режим командной строки загрузки (нажав [6] в момент появления представленного на рисунке окна) и ввести следующие команды:

```
set hw.ata.ata_dma=0
set hw.ata.atapi_dma=0
boot
```

Учитывая, что инсталляция выполняется, по сути, в однозадачном режиме, отключение DMA не должно сильно замедлить установку.

4) Иногда помогает такой «трюк» – после создания слайса (и сохранения изменений) завершить процесс установки и выйти из `sysinstall`. Затем запустить установку снова и воспользоваться уже созданным слайсом, продолжив с разбиения его на разделы. В этом случае `sysinstall` при втором подходе считывает уже обновлённую разметку диска, соответствующую тому, что «получилось», а не тому, что «хотели сделать».

Принимая во внимание трудозатраты, проверять эти рекомендации лучше в порядке 4-3-2-1. Надеюсь, хотя бы одна из них поможет решить проблему.

Ну и ещё одно замечание – если слайс для FreeBSD подготавливался в сторонней программе, то нужно помнить, что FreeBSD должна устанавливаться на первичный раздел. Это ограничение главным образом связано с загрузчиком, который не умеет анализировать дополнительные разделы, но и в процессе инсталляции, думаю, могут возникнуть проблемы.

### СПРАШИВАЙТЕ ПРАВИЛЬНО!

- Пожалуйста, не забывайте сообщать все необходимые данные о вашей системе. «У меня не работает X» нам мало что скажет, если мы не знаем, какую версию X вы имеете в виду и на какой аппаратной конфигурации пытаетесь ее запустить.
- Опишите свою проблему с максимальной точностью. Причитания типа «Оно не работает» или «У меня ошибка» вряд ли дадут нам возможность помочь Вам. Каким именно образом нечто не работает? Чего вы от него хотели? Какое поступает сообщение об ошибке?
- Помните, пожалуйста, что люди, работающие в нашем журнале, НЕ являются авторами или разработчиками Linux или Вашего конкретного пакета (дистрибутива). Иногда нужная Вам информация имеется на соответствующем сайте. Ознакомьтесь с документацией!

Мы стараемся ответить на все вопросы. Если вы не нашли ответа на свой, посмотрите другие ответы – может быть, там разбирается проблема, аналогичная вашей. Все вопросы, к сожалению, поместить невозможно.

# Диск Linux Format

В этом месяце на нашем диске не одна, не две, а аж три операционных системы!

## ДИСТРО-НОСТАЛГИЯ



Майк Сондерс готов к спору, какой процессор круче: 6502 или Z80. А как насчёт x86 супротив 68k?

Я просмотрел последние 80 выпусков *LXF*, вспоминая дистрибутивы, рассмотренные в журнале или попавшие на наши диски. Многие из них — казалось, способные потрясти мир — довольно быстро канули в Лету. Мне особо нравится рекламный слоган Xpresso Linux: «Предназначен для улучшения жизни всего

человечества с помощью видимых и невидимых средств». Ни за что не догадаешься, что несколько парней решили вот этак сменить бренд Red Hat 6.0!

«XPRESSO LINUX  
ОБЕЩАЛ УЛУЧШИТЬ  
ЖИЗНЬ ВСЕГО  
ЧЕЛОВЕЧЕСТВА.»

AlphaNet, Storm, Icepack, Definite, WinLinux... все отправились в большую коробку для компакт-дисков где-то на небесах: их потенциала хватило, чтобы выделиться на рынке дистрибутивов Linux, но сейчас они принадлежат эпохе Linux-бума в начале нашего века. Очевидно, дистрибутивы, анонсировавшие больше, чем могли реально предложить — например, способность полностью заменить Windows — пострадали больше всего, тогда как Red Hat и SUSE просто укреплялись на рынках серверов и технических рабочих станций, избегая мифотворчества.

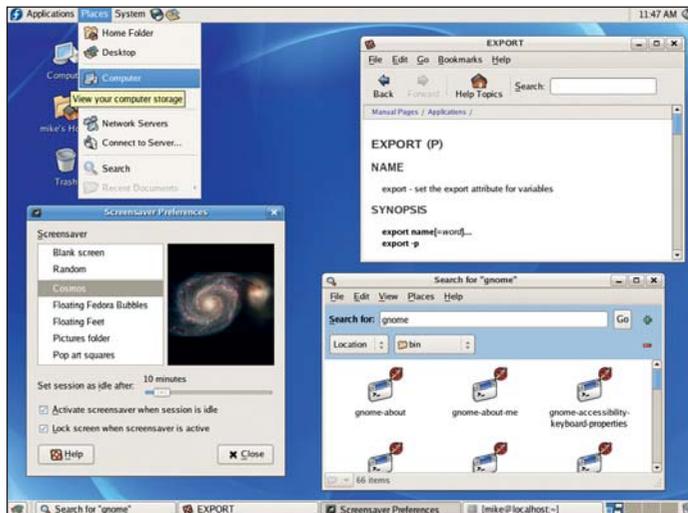
Были и проекты, привлекавшие всеобщее внимание, но так и не создавшие ничего достойного. Помните UserLinux? Попытка сделать полностью бесплатный дистрибутив для нужд бизнеса сразу же погрязла в мелочных дебатах. Ubuntu перехватила инициативу, и UserLinux теперь можно считать покойником. Досадно, но такова жизнь в мире Linux — лучшие дистрибутивы занимают верхние строчки хит-парадов, оставляя другим роль исторических диких.

[mike\\_saunders@futurenet.co.uk](mailto:mike_saunders@futurenet.co.uk)

## РАБОЧИЙ СТОЛ GNOME 2.14

Точно по расписанию, Gnome 2.14 вышел через шесть месяцев после 2.12, и он не разочаровал постоянных пользователей Gnome. В обзоре *LXF79* мы уже говорили, что этот дистрибутив привнёс на рабочий стол желанные и долгожданные возможности: графические эффекты Cairo, утилиты *Sabayon* и *Pessulus*, а также *NetworkManager*, упрощающий работу с сетевыми соединениями. Была продолжена работа над улучшением производительности и снижением потребления памяти Gnome, и результаты видны уже сейчас (это особенно заметно в Gnome Terminal, всегда работавшем чрезвычайно медленно). Теперь Gnome стал шустрее и легче, так что если раньше вы находили его медлительным, попробуйте новый релиз.

Что не изменилось, так это процедура установки. Вы получаете по архиву на каждое, хотя бы и мелкое, приложение, а не один общий пакет, как это сделано в KDE. В результате при компиляции из исходного кода вам не обойтись без помощи системы сборки *Garnome*. Если вы не знакомы со структурой Gnome или с тем, как собирать программы из исходных текстов, вам лучше подождать, пока ваш дистрибутив



Довольно изящные нововведения Gnome 2.14: новый хранитель экрана, просмотр map-страниц в окне справки и поиск в Nautilus.

не выпустит бинарные пакеты для этого релиза Gnome. А ещё лучше — найдите опытного пользователя Linux и попросите его помочь!

В разделе Desktop на первой стороне DVD вы найдёте исходный код Gnome 2.14, который можно скопировать на жёсткий диск и скомпилировать. Распакуйте файл **garnome-2.14.x.tar.gz** и обратитесь к файлу README: в частности, к части, опи-

сывающей **GARCHIVEDIR**, который позволит вам собрать Gnome из архивов, расположенных на вашем диске, а не копировать архивы из сети.

Файл README объясняет, как использовать *Garnome*, чтобы полностью собрать рабочий стол Gnome 2.14, и, если вы будете действовать точно по его инструкциям, проблем у вас не возникнет.



## ДИСТРИБУТИВЫ ДВУСТОРОННИЙ DVD НА ДИСКЕ

Как вы уже поняли, в этом месяце наш диск — просто сокровищница программного обеспечения, объёмом более 8 ГБ; хорошо, что есть на свете двусторонние диски! Благодаря этому мы смогли разместить в этом выпуске не только Fedora Core 5, но и Live-дистрибутивы, обзоры которых имеются в этом номере. Неважно, новичок вы в Linux или ветеран — на нашем DVD вы найдёте, во что вонзить зубы!

Первая сторона содержит самый полный дистрибутив Fedora — загрузитесь и установите. На второй стороне находятся Live-дистрибутивы, и они, как и в прошлом месяце, загрузочные; мы лезли из кожи вон, чтобы влихнуть их на диск как можно больше.

А для тех дистрибутивов, которые мы не смогли вставить в загрузчик, мы включили ISO-образы, их можно записать на компакт-диск.

В результате вы можете:

- загрузить свой ПК с первой стороны и установить Fedora.
- загрузиться со второй стороны и посмотреть на подборку Live-дистрибутивов.

Здесь мы не хотели бы вдаваться в детали Live-дистрибутивов, их обзор приведен на стр. 20. Информация об использовании диска хранится в файле **index.html** (он присутствует на обеих сторонах DVD).

## ДИСТРИБУТИВЫ FEDORA CORE 5 НА ДИСКЕ

В прошлом месяце Энди Хадсон (Andy Hudson) в своём обзоре наградил Fedora девятью очками из десяти возможных. Новый релиз дистрибутива, спонсируемого Red Hat и поддерживаемого сообществом, принёс изобилие новых возможностей, например, AIGLX, Mono, NetworkManager и систему обновления Pup.

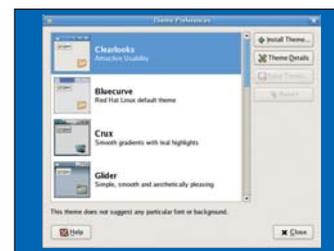
Все эти технологии построены на базе ядра 2.6.15 и X.org 7.0, и улучшения коснулись скорости загрузки и общей производительности. Fedora стала выглядеть по-настоящему зрелым и отлаженным дис-

трибутивом. Хотя этот дистрибутив считается отладочным для линейки продуктов Red Hat Enterprise, он великолепно подходит для повседневных нужд.

Минимальные системные требования для Fedora Core 5 – процессор Pentium со 128 МБ ОЗУ, но этого достаточно только для текстового режима (т.е. для серверного варианта). Варианту системы с рабочим столом потребуется процессор не ниже Pentium II 400 МГц и 256 МБ ОЗУ, а уж для комфортного использования мы рекомендуем процессор с частотой 1 ГГц и 512 МБ памяти. По части использования дискового пространства, вы можете уместить серверную установку в 500

МБ, но для полноценного рабочего стола потребуется по крайней мере 5 Гб. Утилита разбивки диска в инсталляторе позволит вам выделить необходимое пространство под Linux-раздел, а также создать раздел подкачки (виртуальная память – обычно 256–512 МБ).

Инсталлятор Fedora (Anaconda) чрезвычайно прост в использовании, и если у вас имеется хоть какой-то опыт работы с Linux, затруднений не будет. Ниже мы окинули беглым взглядом процедуру установки; новичкам в Linux есть смысл прочесть полное руководство по установке, расположенное в [docs/index.html](#) (первая страница DVD). Распечатайте его или откройте



Fedora 5 щеголяет новой темой – заголовки окон стали ярче и обзавелись гладкими мульками.

его копию на другом ПК, чтобы видеть его в процессе установки.

## УСТАНОВКА FEDORA



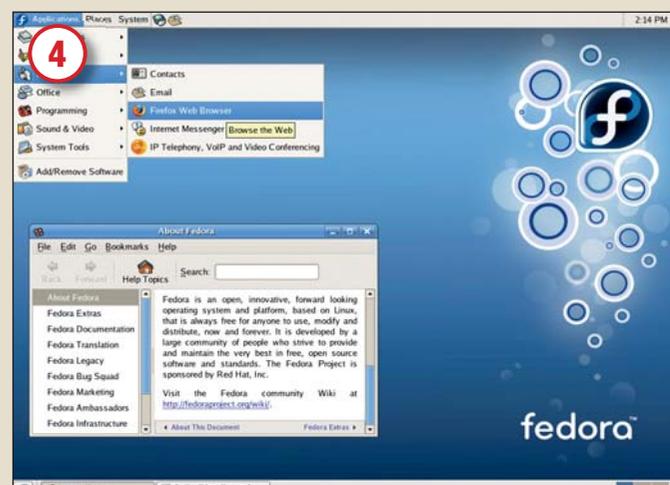
Загрузите свой ПК со стороны 1 DVD – увидите этот экран; для запуска установки просто нажмите Enter. (Возможно, вам потребуется настроить свой BIOS на загрузку с DVD-диска). Если во время загрузки возникла проблема, перезапустите компьютер и нажмите на этом экране кнопки F2 и F3 – вы увидите дополнительные опции.



После установки загрузчика Grub и настройки сети вам предлагают выбрать устанавливаемое программное обеспечение. Можете оставить все как есть, либо выбрать опцию Customize Now, для получения более полного списка пакетов. После этого программы скопируются на ваш диск, и система перезагрузится в финальный экран настройки. Прочтите текст и нажмите кнопку Forward.



Экран Media Check (здесь можно проверить целостность скачанных дисков) можно пропустить, нажав Tab и затем Enter. Затем запустится инсталлятор: нажмите кнопку Next и следуйте инструкциям. Пользоваться ими так же легко, как и инструментом разбивки дисков (на рисунке), позволяющим выделить место на диске для Linux.



Итак, вы в блистательном новом рабочем столе Fedora Core 5! Постоянные пользователи Fedora сразу же оценят новую тему и логотип. Отсюда вы можете приступить к исследованию установленных программ, а также выйти в сеть и скачать дополнительные пакеты. Если у вас возникли какие-то проблемы или вопросы, обращайтесь на форумы [www.fedoraforum.org](http://www.fedoraforum.org)



# СОДЕРЖИМОЕ DVD

## DVD ПЕРВАЯ СТОРОНА

### Журнал

<b>Article index</b>	Список статей LXF.
<b>HotPicks</b>	Программы из раздела HotPicks.
<b>iFolder</b>	Дополнительные файлы.
<b>OOo</b>	Basic. Дополнительный код к руководству.

### Рабочий стол

<b>Gnome</b>	Рабочий стол.
<b>Kalva</b>	Утилита видео записи KDE.
<b>Kino</b>	Редактор видео.
<b>Monotheka</b>	Управление коллекциями видео.
<b>Wine</b>	Запуск Windows-приложений под Linux.

### Разработка

<b>Cream</b>	Улучшения для Vim.
<b>Kompile</b>	Утилита KDE для сборки программ из исходных кодов.
<b>MonoDevelop</b>	Среда разработки для Mono.

### Дистрибутивы

<b>Fedora Core 5</b>	Операционная система, спонсируемая Red Hat.
----------------------	---------------------------------------------

### Игры

<b>DOSBox</b>	Эмулятор DOS.
<b>GPC-Slots2</b>	Казино, работающее в текстовом режиме.
<b>HighMoon</b>	Червяки в космосе!
<b>Vultures</b>	Интерфейс для NetHack.
<b>Zatacka</b>	Многопользовательская игра на базе Line.

### Графика

<b>Gimp</b>	рафический редактор.
<b>Inkscape</b>	Векторный графический редактор.
<b>LittleCMS</b>	Цветовая палитра.
<b>Scribus</b>	Издательское приложение для рабочего стола.
<b>Xara Xtreme</b>	Векторный графический редактор.

### Интернет

<b>CheckGmail</b>	Проверка почты на Gmail.
<b>Lightning</b>	Календарь для Thunderbird.
<b>SkipStone</b>	Лёгкий web-браузер.
<b>Sylpheed-Claws</b>	Почтовый клиент.
<b>Thunderbird</b>	Почтовый клиент.

### Офис

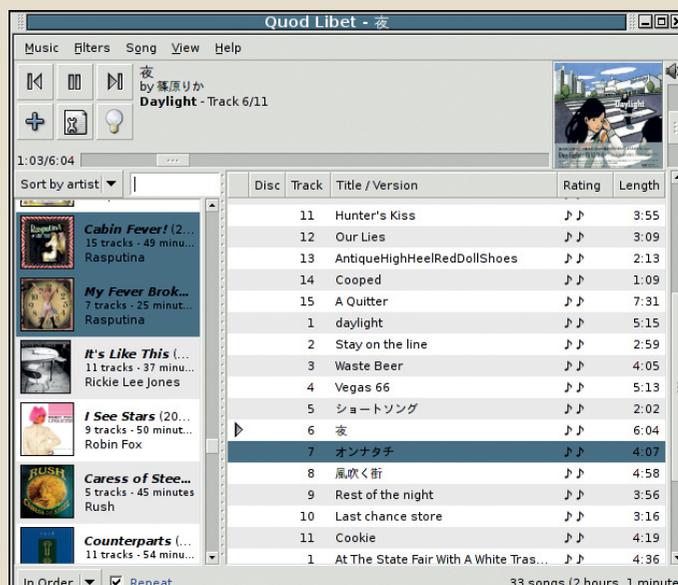
<b>AbiWord</b>	Текстовый процессор.
<b>Faces</b>	Управление проектами.
<b>KPDFTool</b>	Работа с PDF файлами.
<b>OpenOffice.org</b>	Офисный пакет.

### Звук

<b>Audacity</b>	Редактор звука.
<b>IDJC</b>	Клиент shoutcast.
<b>Quod Libet</b>	Проигрыватель музыки.

### Система

<b>Amanda</b>	Система резервного копирования.
---------------	---------------------------------



Quod Libet: проигрыватель и менеджер музыки с GTK-интерфейсом.

### Bootchart

<b>DM-Crypt</b>	Отображение зашифрованных устройств.
<b>EncFS</b>	Шифрованная файловая система.
<b>Fuse</b>	Файловая система в пространстве пользователя.
<b>InitNG</b>	Скрипты для старта системы.
<b>Qemu</b>	Эмулятор ПК.
<b>Sussen</b>	Сканер уязвимостей.

### Главное

<b>Allegro</b>	Библиотека для написания игр.
<b>Avifile</b>	Чтение и запись AVI файлов.
<b>CheckInstall</b>	Создание бинарных пакетов.
<b>CSV</b>	Индекс содержимого диска.
<b>GLib</b>	Низкоуровневая библиотека.
<b>Glibc</b>	Библиотека GNU C.
<b>Gtk</b>	Инструментарий разработчика пользовательского интерфейса.
<b>Guile</b>	Встраиваемая реализация Scheme.
<b>Jigdo</b>	Утилита распространения дистрибутивов.
<b>Kernel</b>	Последнее ядро Linux.
<b>Libsigc</b>	Система обратных вызовов с проверкой типов.
<b>LibXML</b>	Анализатор XML для C и инструментарий.
<b>Ncurses</b>	Инструментарий для создания окон в консольных приложениях.
<b>Rawrite</b>	Запись образов дисков на дискеты.
<b>SDL</b>	Мультимедийная библиотека.
<b>Smart Boot Manager</b>	Системо-независимый менеджер загрузки.
<b>WXWidgets</b>	Среда для создания графического интерфейса на C++.

## DVD ВТОРАЯ СТОРОНА

### Дистрибутивы

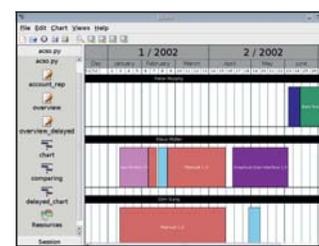
<b>Damn Small Linux</b>	Лёгкий Live-дистрибутив.
<b>Games Knoppix</b>	Live-дистрибутив, содержащий множество игр.
<b>Looking Glass 3D Live</b>	Шикарная трёхмерная демо-версия.
<b>Morphix</b>	Live-дистрибутив на основе Debian.
<b>SUSE Linux 10 0 Live</b>	Популярный дистрибутив для настольной системы.
<b>Ubuntu 5 10</b>	Live-версия Ubuntu.

### Справка

<b>LDP</b>	Проект документации Linux.
------------	----------------------------



HighMoon. Стреляйте аккуратно, чтоб не попасть в самого себя ...



Faces - «гибкое, автоматическое и прочая, и прочая» приложение.

## Линуксцентр представляет авторизованные курсы Mandriva Linux.

Курсы не привязаны к конкретному дистрибутиву и сертифицированы международной организацией LPI - Linux Professional Institute ([www.lpi.org](http://www.lpi.org)).

### ★ LNX70: Введение в Linux.

Предназначен для пользователей, которые хотят приобрести навыки использования Linux. В рамках курса рассматриваются вопросы миграции на Linux и даются базовые знания для пользователей.

*Цель курса: Обучить пользователя оптимальной работе с Linux.  
Требования к соискателям: Базовая компьютерная подготовка.*

### ★ LPI101: Администрирование Linux для начинающих.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования Linux. Рассматриваются следующие вопросы: конфигурирование системы, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

*Цель курса: Подготовить системного администратора Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI101.*

*Требования к соискателям: Знания по Linux в объеме курса LNX70.*

### ★ LPI201: Администрирование Linux для специалистов.

Курс предназначен для системных администраторов, которые хотят повысить свою квалификацию в области управления сложными Linux-системами. Рассматриваются следующие вопросы: настройка ядра Linux, восстановление системы, восстановление файловой системы, настройка RAID, сопровождение системы, автоматизация задач администрирования.

*Цель курса: Подготовить системного администратора к управлению сложными Linux-системами в соответствии с требованиями к соискателям на сдачу экзамена LPI201.*

*Требования к соискателям: Знания по Linux в объеме курса LPI101 или LPI102.*

### ★ LPI102: Сетевое администрирование Linux.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования сетей на базе Linux. Рассматриваются следующие вопросы: конфигурирование периферийных устройств, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

*Цель курса: Подготовить системного администратора к управлению сетями на базе Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI102.*

*Требования к соискателям: Знания по Linux в объеме курса LPI101.*

### ★ LPI202: Сетевое администрирование Linux для специалистов.

Курс предназначен для опытных системных администраторов, которые хотят усовершенствовать навыки управления сетевыми сервисами на базе Linux. Рассматриваются следующие вопросы: углубленное конфигурирование сети, настройка служб электронной почты и новостей, расширенная настройка DNS, обслуживание интернет-сервера, настройка DHCP, NIS, LDAP и PAM, управление безопасностью сетевых служб, контроль за трафиком.

*Цель курса: Подготовить системного администратора к управлению сложными сетевыми сервисами в соответствии с требованиями к соискателям на сдачу экзамена LPI202.*

*Требования к соискателям: Знания по Linux в объеме курса LPI101.*

Узнайте подробности и закажите курсы на сайтах - [www.linuxcenter.ru](http://www.linuxcenter.ru) и [www.unixedu.ru](http://www.unixedu.ru)



## Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

### 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, следует заглянуть именно туда.

### ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли 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, 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 бинарный 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 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** – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;  
**имя\_программы-1.0.1.ppc.Suse9.rpm** – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;  
**имя\_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

### Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: [disks@linuxformat.ru](mailto:disks@linuxformat.ru)

LINUX ФОРМАТ В ГИГАБАЙТЕ DVD

6 LIVE DISTROS

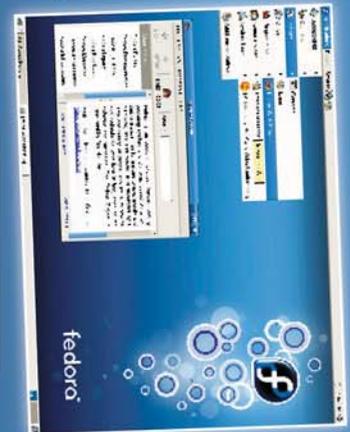
LINUX  
ФОРМАТ

Попробовать Linux еще никогда не было так просто: 6 дистрибутивов и никакой инсталляции!

Настоящие ПО, игры, 3D-эффекты и многое другое!!!



- SUSE Live
- Ubuntu Live
- Morphix
- Games Knoppix
- Looking Glass 3D Live CD
- Damn Small Linux



OpenOffice.org 2.0.2  
 Mono, Xep и Zetlik  
 Ашвид 2.5.0 – менеджер пакетов системы разнородного копирования  
 Svalen – движок видео системы в безопасности  
 Графическое ПО – по материалам Libre Graphics Meeting  
 Lighting – календарь для Thunderbird!  
 КОЕ 3.5.1

Ф е д о р а С о г е т и

5

Свободный дистрибутив, спонсируемый Red Hat – еще быстрее, еще полнее, еще универсальнее!

Сторона 1

Рабочий стол  
Gnome -- популярный рабочий стол  
Kaiya -- программа для записи видео  
Kino -- видеоредактор  
Mopotheka -- менеджер видеокolleкции  
Wine -- реализация Win32 API

Разработка  
Creat -- настроенный Vtm  
Kompile -- интерфейс для сборки программ  
MonoDeveloper -- среда разработки для Mono

Дистрибутивы  
Fedora Core 5

Игры  
DOSBox -- эмулятор DOS  
Highmoon -- космические дуэли  
Vultures -- интерфейс к NetHack  
Zatachka -- многопользовательская "змейка"  
grcslots2 -- казино

Графика  
GIMP -- растровый графический редактор  
Inkscape -- векторный графический редактор  
LittleCMS -- движок цветокоррекции  
Scribus -- программа верстки  
XaraXtreme -- векторный редактор

Интернет  
CheckGmail -- утилита для проверки почты  
Lightning -- календарь для Thunderbird  
Skipstone -- веб-браузер  
Sylphed-Claws -- продвинутый почтовый клиент  
Thunderbird -- популярный почтовый клиент

Сторона 2

LiveCD:  
Damn Small Linux  
Games Knoppix  
Looking Glass LiveCD  
Morphix  
SUSE LiveCD  
Ubuntu LiveCD

Офис  
AbiWord -- текстовый процессор  
Faces -- утилита для менеджмента проектов  
KPDFTool -- интерфейс для GhostView  
OpenOffice.org -- офисный пакет

Звук  
Audacity -- аудиоредактор  
IDJC -- shoutcast-клиент  
QuodLibet -- аудиоплеер

Система  
Amanda -- система создания резервных копий  
Bootchart -- анализатор загрузки  
EncFS -- файловая система с поддержкой шифрования  
FUSE -- файловая система в пользовательском пространстве  
InitNG -- новая система инициализации  
QEMU -- эмулятор ПК  
Suse -- сканер безопасности

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru  
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принять на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Кардидер», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8296. Лицензия МПТР России ВАР № 77-68



Поставляется вместе с журналом LINUXFORMAT номер 10(80) Июнь 2006

СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала 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 – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы 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, 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. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

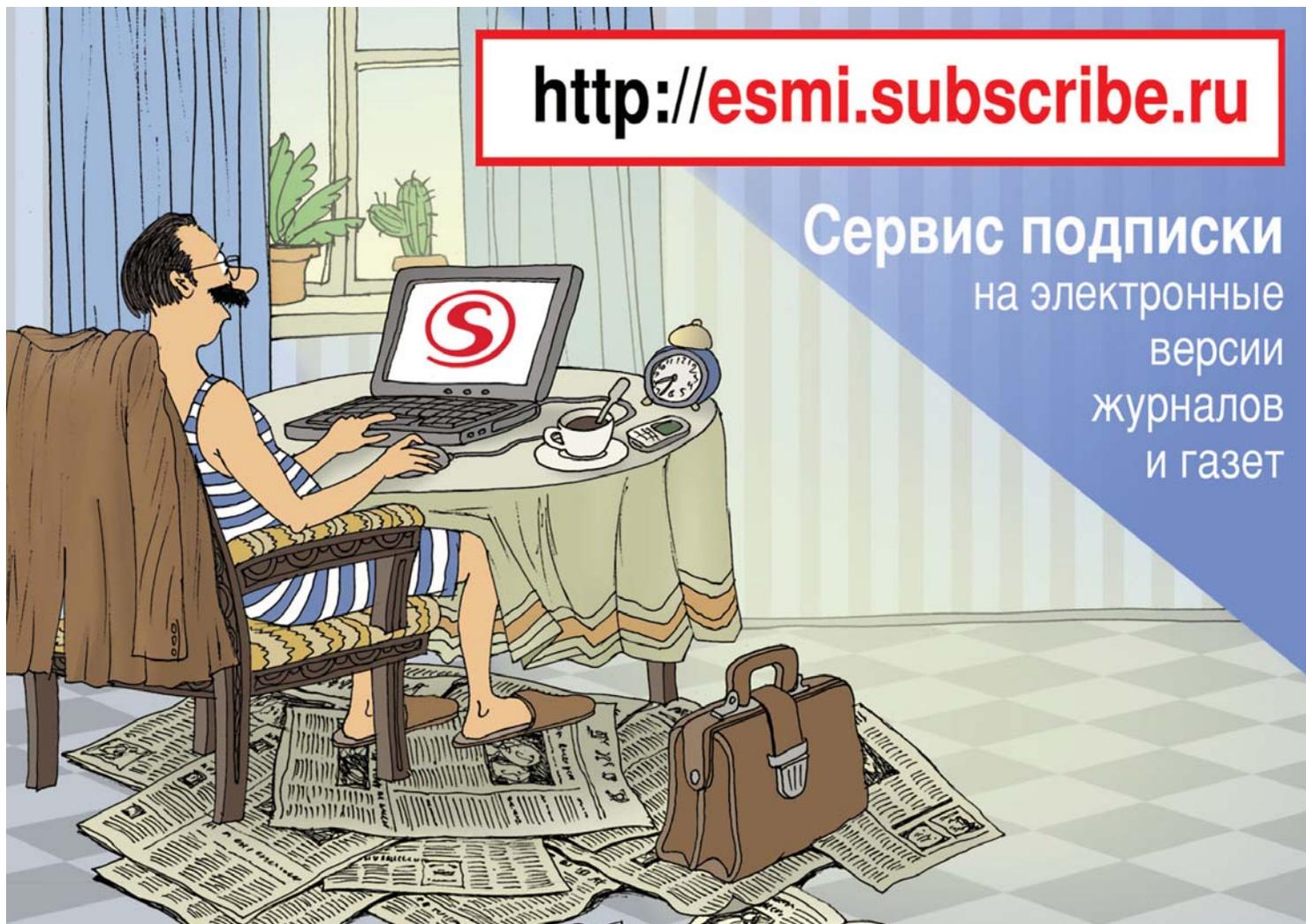
А что если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не придется использовать Linux для записи дисков, подойдет любая операционная система, способная распознать привод записи дисков (см. выше).

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 его разработчика.

LXF

<http://esmi.subscribe.ru>

Сервис подписки  
на электронные  
версии  
журналов  
и газет



# СИСТЕМНЫЙ администратор

Клонировем Windows с помощью Symantec Ghost

Насколько неуязвима ваша беспроводная сеть?

Active Directory вместо рабочей группы

Настраиваем DSPAM – ваш личный спам-фильтр

Как спасти данные, если отказал жесткий диск

Модифицируем BIOS

Все ли возможности ClamAV вы используете?

Что важно знать об IP-телефонии

Админские сказки

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

## В «Системном администраторе» вы не прочтете о:

- котировках валют
- сплетнях
- погоде
- политике
- развлечениях



## В вашем распоряжении:

- опыт лучших IT-специалистов
- новые идеи и полезные советы
- самые эффективные решения в области системного и сетевого администрирования



**Подпишитесь сейчас!**

Роспечать – 20780, 81655  
Пресса России – 87836  
Online-подписка – [www.linuxcenter.ru](http://www.linuxcenter.ru)

Время подписки  
ограничено!

#### РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:

##### ГЛАВНЫЙ РЕДАКТОР

Валентин Синеицын [info@linuxformat.ru](mailto:info@linuxformat.ru)

##### Переводчики

Александр Бикмеев, Павел Гладков, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабуню, Павел Шер.

##### Литературные редакторы

Родион Водейко, Иван Мищенко, Елена Толстякова

##### Допечатная подготовка

Мария Пучкова

Родион Водейко

##### Креативный директор

Станислав Медведев

##### Технический директор

Денис Филиппов

##### Директор по рекламе

Денис Игнатов +7 812 965 7236 [advert@linuxformat.ru](mailto:advert@linuxformat.ru)

##### Заместитель генерального директора

Софья Виниченко

##### Генеральный директор

Павел Фролов

##### УЧРЕДИТЕЛИ

частные лица

##### ИЗДАТЕЛИ

Павел Фролов, Станислав Медведев

Отпечатано в типографии «Текст»

ООО «ПГК» «Текст»

188680, Ленинградская область,

Всеволожский район, Колтуши, д.32

Заказ \_\_\_\_\_

Пре-пресс: [drive-group](http://drive-group.ru)

#### РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:

Редактор Ник Вейч (Nick Veitch) [nick.veitch@futurenet.co.uk](mailto:nick.veitch@futurenet.co.uk)

Заместитель редактора Пол Хадсон (Paul Hudson) [paul.hudson@futurenet.co.uk](mailto:paul.hudson@futurenet.co.uk)

##### Художественный редактор

Джулиан Джефферсон (Julian Jefferson) [julian.jefferson@futurenet.co.uk](mailto:julian.jefferson@futurenet.co.uk)

Новостной редактор Майк Сондерс (Mike Saunders) [mike.saunders@futurenet.co.uk](mailto:mike.saunders@futurenet.co.uk)

##### Литературный редактор

Ребекка Смолли (Rebecca Smalley) [rebecca.smalley@futurenet.co.uk](mailto:rebecca.smalley@futurenet.co.uk)

##### Штатный автор

Грэм Моррисон (Graham Morrison) [graham.morrison@futurenet.co.uk](mailto:graham.morrison@futurenet.co.uk)

##### Ассистент по выпуску

Эндрю Грегори (Andrew Gregory) [andrew.gregory@futurenet.co.uk](mailto:andrew.gregory@futurenet.co.uk)

##### АВТОРЫ

Марк Бейн (Mark Bain), Ладислав Боднар (Ladislav Vodnar), Нейл Ботвик (Neil Botwick), Крис Браун (Chris Brown), Энди Ченелл (Andy Channelle),

Кингс Кобблер (Kings Cobbler), Алексей Федорчук, Евгений Балдин, Андрей

Боровой, Сергей Супрунов, Илья Шпаньков, Сергей Яремчук

##### ХУДОЖЕСТВЕННЫЙ ОТДЕЛ

Арт-ассистенты: Джон Блекшай (John Blackshaw), Симон Годдард (Simon Goddard), Дэнниел Винсент (Daniel Vincent)

Фотографии: Joby Sessions, Simon Lees, Superstock, Photodisc, Photolibary.com

Иллюстрации: Пол Бейтмен (Paul Bateman), Крис Винн (Chris Winn), Шейн Коллин (Shane Collinge)

Создание диска: Майк Сондерс (Mike Saunders)

#### КОНТАКТНАЯ ИНФОРМАЦИЯ

England: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel 01225 442244 [Email linuxformat@futurenet.co.uk](mailto:Email linuxformat@futurenet.co.uk)

Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37

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

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

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензией Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставяемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственности за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

За содержание рекламных материалов редакция ответственности не несет.

Linux-зарегистрированная торговая марка Линуса Торвальдса (Linus Torvalds). Название «GNU/Linux» заменяется на «Linux» в целях сокращения. Остальные торговые марки являются собственностью их законных владельцев.

Linux Format является торговой маркой Future Publishing Ltd (Future plc group company).

За информацией о журналах, издаваемых Future plc group company, обращайтесь

<http://www.futurepub.com>

© Linux Format 2005

© Future Publishing Ltd 2005

Future  
PUBLISHING LTD

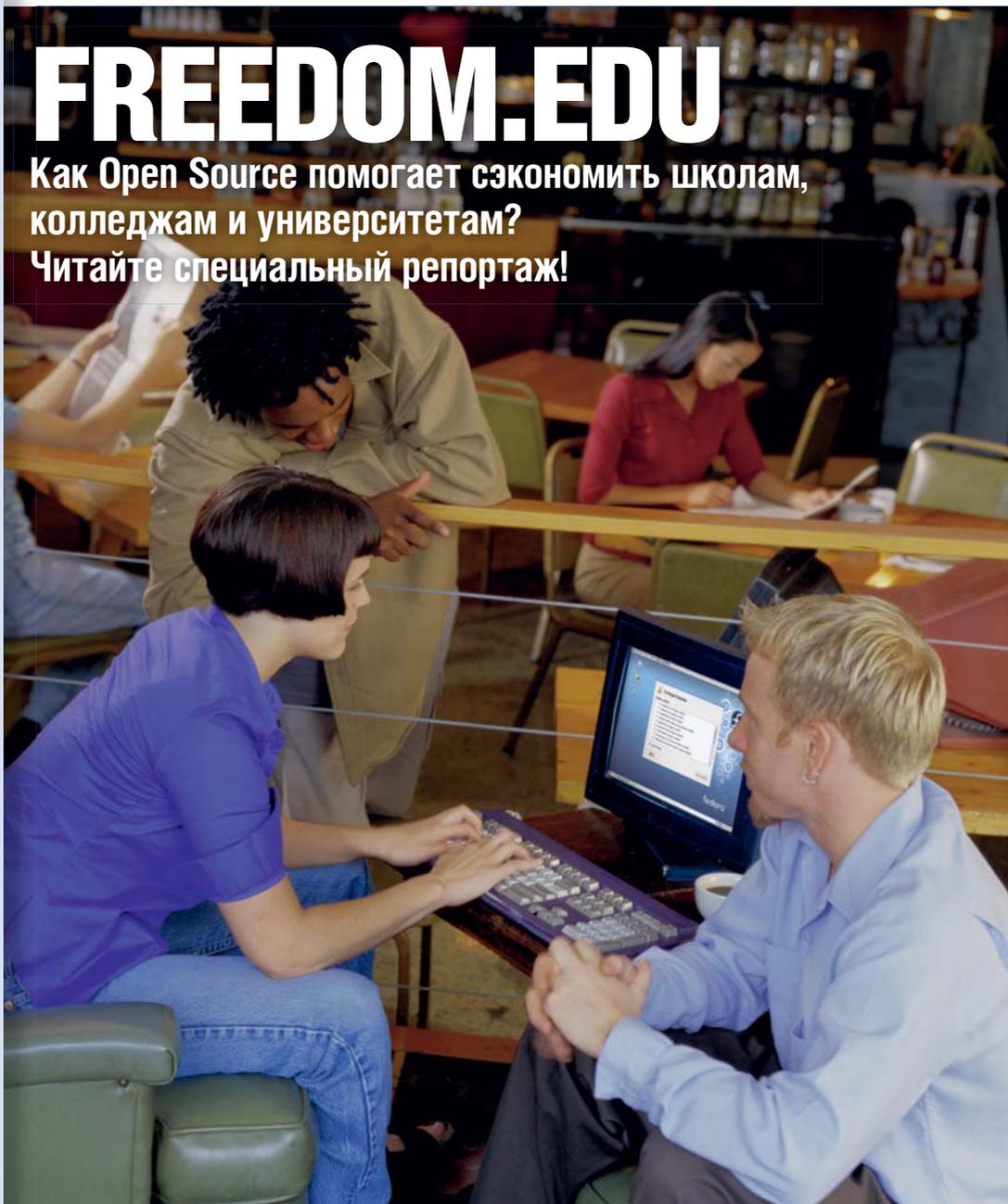
www.futurepub.com

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO

• SAN FRANCISCO •

# В следующем месяце

## LINUX FORMAT 7(81) ИЮЛЬ 2006



# FREEDOM.EDU

## Как Open Source помогает сэкономить школам, колледжам и университетам?

### Читайте специальный репортаж!

#### LXF ИНТЕРВЬЮ



## Грег Кроа-Хартман

Почему двоичные драйвера - зло

## Ubuntu 6.06 – Dapper Drake

Обновление всеми любимого дистрибутива

## Учебник Open-Xchange

Избавьтесь от Microsoft Exchange раз и навсегда!