

## Колонка главного редактора



Чтение очередных новостей и статей на тему готовности GNU/Linux для «обычных пользователей» давно вызывает улыбку. Одна из ключевых проблем людей, пытающихся сделать те или иные громкие заявления по этому поводу, заключается в крайне размытом представлении о том, кто же они такие — «обычные пользователи». Один считает, что на эту роль отлично подходит его сын, другой приводит в пример жену, а третий — бабушку. И вот насколько же в действительности «обычен» каждый из называемых представителей? Ведь у всех них заметно различаются не только интересы, но и жизнедеятельность в целом.

Причем это лишь первая ступень некорректного «преобразования», называемая в философии индукцией: автор обобщает свой частный случай успешной (или безуспешной) работы с Linux, основываясь на каком-то близком ему примере «обычного пользователя». Потом непременно случается обратный эффект (дедукция), и сторонний «обычный пользователь» понимает, что в его случае это правило уже не работает. В результате общая «погрешность» складывается и возрастает до уровня, пренебрегать которым вдвойне недопустимо.

Вот и получается, что оба мнения не только имеют право на жизнь, но и легко обосновываются. С другой стороны, они не с большей сложностью опровергаются. И прийти к единой позиции банально невозможно.

Главный редактор  
Дмитрий Шурупов  
(osa@samag.ru)

**«Open Source»**  
электронное приложение к журналу  
«Системный администратор»  
№50, 30 сентября 2009 г.

### РЕДАКЦИЯ

**Исполнительный директор**

Владимир Положевец

**Главный редактор**

Дмитрий Шурупов

**Верстка и оформление**

Владимир Лукин

**Сайт электронного приложения:**

<http://osa.samag.ru>

За содержание статей ответственность несет автор. Все права на опубликованные материалы защищены.

## Новости мира Open Source

### Debian переходит на событийную систему загрузки (upstart)

В почтовой рассылке devel-announce популярного дистрибутива Debian GNU/Linux объявлено о переходе проекта на новую параллельную систему загрузки сервисов (инициализации ОС), которая основана на событиях. Разработчики Debian GNU/Linux давно осознают проблему медленной загрузки системы при использовании классического (последовательного) подхода. Для ее решения они намерены заменить существующий /sbin/init более современной реализацией, поддерживающей события Linux-ядра. Такой событийный подход позволяет значительно быстрее запускать системные службы в параллельном режиме и с учетом их зависимостей от других сервисов.

Анализ существующих в других Linux-дистрибутивах систем загрузки привел разработчиков Debian к выбору в пользу upstart, которая используется в Ubuntu Linux и Fedora. Переход Debian GNU/Linux на upstart в качестве init-системы состоится уже в ближайшем релизе — 6.0 (Squeeze). Его выпуск запланирован на весну 2010 года.

### В Linux 2.6.31 улучшена интерактивность десктопов

9 сентября вышла новая версия сердца операционной системы GNU/Linux — Linux ядро 2.6.31. Благодаря усилиям компании Intel в Linux 2.6.31 появилась поддержка новой версии протокола USB — 3.0. Другим значительным новшеством стали усовершенствования в работе ядра на настольных ПК, позволившие повысить отзывчивость (интерактивность) работы X-сервера вдвое при высоких нагрузках на оперативную память или swap. Кроме того, появилась поддержка Kernel Mode Setting (KMS) для видеокарт ATI Radeon (на данный момент — R1XX, R2XX, R3XX, R4XX, R5XX, а также уже ведутся работы над R6XX и R7XX).

Из прочих заметных изменений можно выделить: поддержку стандарта IEEE Std 802.15.4 (Low-Rate Wireless Personal Area Networks), NFS 4.1 в NFS-клиенте, инструмент тестирования Gcov, средство отладки Kmemcheck, средство определения возможных утечек памяти Kmemleak, новый бэкэнд для уведомления об изменениях в файловой системе Fnotify. Заметные улучшения представлены в файловой системе btrfs — в частности, у нее появилось автоматическое определение SSD.

### Представлен первый Android-смартфон Motorola — CLIQ

10 сентября был анонсирован первый в мире смартфон компании Motorola, работающий под управлением мобильной платформы Android. Устройство получило название CLIQ.



Одна из ключевых особенностей новинки — «инновационное решение, управляющее коммуникациями и интегрирующее их», — MOTOBLUR. Эта надстройка к Android обеспечивает интеграцию смартфона с такими онлайн-сервисами, как Facebook, Twitter и MySpace. Технические характеристики Motorola CLIQ таковы:

- ✓ габариты: 114 x 58 x 15,6, вес: 163 г, цвет: титан и белый;
- ✓ 3,1-дюймовый тачскрин с поддержкой разрешения 320 x 480;
- ✓ quad-band GSM/GPRS/EDGE 850/900/1800/1900 МГц, WCDMA 900/1700/2100, HSDPA;
- ✓ процессор Qualcomm MSM7201A с тактовой частотой 528 МГц;
- ✓ память: 256 МБ RAM и 512 МБ ROM, слот для microSD-карты (до 32 Гб);
- ✓ 5-мегапиксельная камера с автофокусом;
- ✓ Bluetooth 2.0+EDR, Wi-Fi (802.11 b/g);
- ✓ microUSB 2.0, стандартный аудиоджек (3,5 мм);
- ✓ AGPS;
- ✓ литий-полимерный аккумулятор на 1420 мАч (заявлено 13,5 дней в режиме ожидания);
- ✓ операционная система — Android 1.5 с MOTOBLUR.

Продажи в США начнутся позже этой осенью оператором T-Mobile. Цена пока неизвестна.

### Состоялся первый релиз Haiku — R1/Alpha 1

Проект операционной системы с открытым кодом Haiku представил свой первый официальный релиз — R1/Alpha 1. Haiku — это проект по созданию операционной системы с открытым кодом, совместимой с BeOS. Он был запущен еще в 2001 го-

ду и изначально назывался OpenBeOS. Развитие проекта поддерживается некоммерческой американской организацией Haiku, созданной в 2003 году. Авторы Haiku хотят создать систему для персональных компьютеров, которая будет быстрой, эффективной, простой в использовании и изучении, функциональной. Исходный код распространяется под лицензией MIT. Цель первого официального релиза – подготовить стабильную сборку Haiku для разработчиков, доступ к которой получит более широкая аудитория из энтузиастов, заинтересованных в тестировании и отладке.

### Microsoft создала свою Open Source-организацию

Корпорация Microsoft объявила о создании организации CodePlex Foundation, цель которой – стать связующим звеном между Open Source-разработчиками и коммерческими компаниями. Среди задач новой организации CodePlex Foundation – обмен кодом и налаживание понимания между Open Source-сообществами и софтверными компаниями, повышение активности участия компаний в Open Source-проектах, пополнение ряда существующих Open Source-организаций новым форумом, в котором будут принимать активное участие и компании, и представители сообщества. Несмотря на схожесть названия новой организации с хостингом проектов Microsoft CodePlex, заявляется, что это независимые инициативы, объединенные общей миссией. Единственным спонсором CodePlex Foundation на данный момент является Microsoft, однако в организации заинтересованы в поиске новых партнеров среди компаний-разработчиков программного обеспечения.

### В тестовую версию Firefox добавлена поддержка WebGL

В последние тестовые сборки веб-браузера с открытым кодом Firefox вошла реализация поддержки нового стандарта WebGL для использования трехмерной графики (OpenGL). Стандарт WebGL еще очень молод и находится только в стадии становления и обсуждения, однако это не помешало инженерам Mozilla уже реализовать поддержку его промежуточной версии в Firefox. Рабочая группа WebGL появилась в этом году, когда представители Mozilla объединили усилия с организацией Khronos Group, стоящей за OpenGL. В рамках WebGL ожидается разработка низкоуровневых JavaScript API, которые обеспечат поддержку OpenGL ES 2.0 в браузере.

Желающие протестировать поддержку WebGL в Firefox уже могут скачать по-

следние ночные сборки браузера и включить опцию «webgl.enabled\_for\_all\_sites» в конфиге («about:config»).

### GNOME 2.28 – новая версия графической рабочей среды

24 сентября был официально представлен очередной релиз популярной графической рабочей среды GNOME – 2.28. GNOME 2.28 стал первым релизом среды, в который вошел новый модуль GNOME Bluetooth, помогающий пользователям управлять Bluetooth-устройствами. В нем предусмотрена интеграция с PulseAudio для беспроводных наушников и с NetworkManager для использования мобильных устройств в качестве беспроводной точки доступа в Интернет. В клиенте обмена сообщениями Empathy проведена заметная работа над контакт-листом, где теперь стало удобнее переносить контакты и устанавливать свой статус, а в диалоге бесед появились новые темы. Кроме того, добавлен полноэкранный режим для аудио- и видеочатов, а также возможность разделения своего рабочего стола с другими пользователями (из списка контактов) через Vino.

Как и ожидалось, веб-браузер Epiphany перешел на использование движка WebKit (Safari, Chrome) вместо Gecko (Firefox). В мультимедийном плеере Media Player была улучшена работа с DVD-меню. У приложения Cheese для веб-камер обновился интерфейс, появился режим Burst (для создания нескольких изображений за раз). PDF-просмотрщик Evince научился редактировать/сохранять текстовые аннотации, а также был портирован на Microsoft Windows. Кроме того, у GNOME Power Manager появилась поддержка ноутбуков с несколькими аккумуляторами, текстовый редактор Gedit был портирован на Mac OS X, в Pango улучшили рендеринг текста (с помощью движка OpenType).

Вместе с выпуском GNOME 2.28 обновились и ключевые библиотеки, используемые в разработках проекта. В GTK+ 2.18 были проведены значительные улучшения в диалоге выбора файлов, а в Glib компонент GNIO стал частью GIO. Следующим релизом GNOME станет версия 2.30, финальный выпуск которой ожидается 31 марта 2010 года.

### Мобильная система Moblin обновилась до версии 2.0

24 сентября вышла новая версия мобильного Linux-дистрибутива, предназначенного для использования на нетбуках, – Moblin 2.0. Система Moblin была разработана компанией Intel как мобильная платформа для нетбуков, использующих процессор Intel Atom. Сейчас ее разработкой

руководит некоммерческая организация Linux Foundation при поддержке Intel.

Среди изменений в новой версии Moblin – 2.0 – можно выделить систему Fastboot, обеспечивающую быструю загрузку Moblin, новый менеджер сетевых подключений ConnMan, обновленный пользовательский интерфейс на базе Clutter, новое средство создания образов Moblin и установки – Moblin Image Creator (MIC2). Установка приложений в системе теперь может осуществляться с помощью утилиты Moblin Application Installer и каталога программного обеспечения Moblin Garage. Одновременно с выпуском Moblin 2.0 анонсированы планы по следующей версии системы – 2.1. Релиз запланирован на четвертый квартал этого года, а среди ожидаемых новшеств – поддержка дополнительных разрешений дисплеев неттопов, улучшения в возможностях обмена сообщениями и локализациях, поддержка передачи данных в 3G-сетях.

### Объявлен конкурс «Лучший свободный проект России»

Журнал Linux Format при поддержке компаний Intel, PingWin Software, «ГНУ/Линуксцентр», Wimaxstore объявил о начале конкурса «Лучший свободный проект России». Основная цель конкурса – стимулирование российских разработчиков свободного ПО за счет поддержки наиболее успешных и востребованных проектов, а также составление современной карты свободных проектов, разрабатываемых на территории Российской Федерации и ее ближайших соседей. Призовой фонд конкурса составляет 500 000 рублей.

В конкурсе могут принимать участие проекты, существующие не менее трех месяцев, разрабатываемые компаниями или частными лицами, постоянно проживающими в России или СНГ и распространяющиеся по лицензии, одобренной Free Software Foundation. Зарегистрировать проект в конкурсе могут как сами разработчики, так и пользователи программного продукта, желающие поддерживать свой любимый проект. Заявки принимаются до 15 ноября включительно. Итоги конкурса будут подведены 15 декабря 2009 года. Победители будут определены решением компетентного жюри, в состав которого войдут представители компаний-организаторов, а также эксперты в области свободного ПО и GNU/Linux. Более подробная информация о конкурсе – на <http://linuxformat.ru/foss-contest>.

Дмитрий Шурупов,  
по материалам [www.nixp.ru](http://www.nixp.ru)  
([osa@samag.ru](mailto:osa@samag.ru))

# Обзор графической среды KDE 4.3

## Релизы KDE 4.3 и 4.3.1

В начале августа вышла очередная версия KDE – 4.3, а 1 сентября – обновление 4.3.1. На этот раз релиз нацелен на отладку и стабилизацию кода, законченность функциональных возможностей, предоставляемых пользователю. В качестве подтверждения серьезности намерений разработчиков достаточно привести число исправленных за полгода ошибок, которое превысило 63000. За этот же период было реализовано порядка 2000 пожеланий пользователей. Более того, по начальному плану разработки KDE именно этот релиз должен был стать рекомендуемым для «обычных пользователей».

Изменения затронули все основные компоненты KDE:

- ☑ Plasma (оболочка рабочего стола);
- ☑ виджеты;
- ☑ KWin (оконный менеджер KDE);
- ☑ KRunner (система запуска и выполнения команд);
- ☑ приложения KDE.

Промежуточное обновление KDE 4.3.1 практически не добавляет функций, однако исправляет достаточно большое количество ошибок, некоторые из них известны уже более 2-3 лет (например, [http://bugs.kde.org/show\\_bug.cgi?id=138725](http://bugs.kde.org/show_bug.cgi?id=138725)). Более подробно о выходе KDE 4.3.1 можно прочитать на русскоязычной странице анонса – <http://kde.ru/content/vypushchen-kde-431>.

## Plasma

Не случайно первым в списке компонентов, подвергшихся изменениям в KDE 4.3, стоит Plasma. Из выпуска в выпуск разработчики радуют нас огромными улучшениями в этом направлении. Последний релиз не стал исключением. Нововведения затрагивают все: от панелей до управления виртуальными рабочими столами. Что не менее важно, выражение «Plasma упала», которое некогда использовалось в качестве аргумента против нового KDE, уже почти вышло из употребления.

Первое, что бросается в глаза при взгляде на обновленный рабочий стол, – новая тема оформления Air, которая выглядит гораздо «легче» прошлой – Охуген (она по-прежнему доступна пользователям в настройках). И не только выглядит – по моим наблюдениям, после продолжительной работы в KDE 4.3.1 можно ощутить весомый прирост производительности.

Здесь же стоит отметить возможность полной настройки под себя клавиатурных сочетаний (ранее были доступны не все возможности), что тоже увеличивает скорость работы в среде.

В очередной раз изменения и улучшения коснулись панелей KDE – теперь в них появился разделитель для виджетов, размер которого может изменяться вручную или автоматически. Они приобрели возможность растягиваться очень широко или очень узко – маленький размер панели делает ее более элегантной. При достаточной ширине значки в системной лентке выстраиваются в два и более рядов. Кстати, если говорить о системной лентке, то здесь используется новый протокол, разработанный совместно с FreeDesktop.org, – более подробно о новом лентке можно прочитать в заметке на <http://knotes.ru/2009/05/new-tray-in-action>. Если кратко перечислить его достоинства, то это группировка значков и возможность скрытия этих групп, автоматическое скрытие неиспользуемых значков, более привлекательные всплывающие подсказки. Что касается уведомлений, то теперь они формально поделены на две группы: уведомления приложений и задачи. К первой группе относятся, например, сообщения Kopete и KTorrent, а ко второй – операции копирования, перемещения и удаления. Особо выделяются и выполненные задания – на-

пример, завершенные задачи с возможностью открыть место назначения копирования. Таким образом, работать с уведомлениями стало гораздо приятнее: можно сворачивать или закрывать отдельные уведомления либо их группы, либо всё сразу. Следить за выполнением заданий стало проще за счет анимации и показа количества заданий, которые, кстати, можно приостанавливать. Особенно приятна эта новость на фоне саммита, посвященного разработке свободных окружений и прошедшего в этом году на Канарских островах (речь о Gran Canaria Desktop Summit 2009, где встретились сообщества KDE и GNOME. – Прим. ред.). Там было решено объединить усилия по разработке уведомлений в приложениях (GTK+ и Qt) и подготовке стандарта в рамках D-Bus для этих уведомлений (<http://www.kdedevelopers.org/node/4028>).

Теперь – о других улучшениях в Plasma. Значок «Кешью» в правом верхнем углу теперь может менять свое положение и занимать любое свободное место на краях экранов. Основное назначение этого значка – управление базовыми пользовательскими функциями Plasma: комнаты, виджеты, панель виджетов. На рабочем столе теперь могут быть обои нестандартных видов. Помимо обычных черепицы, фона, изображения или слайдшоу можно установить:

- ☑ обои в стиле вируса, которые «поедают» экран;
- ☑ погодные обои, меняющие изображение в соответствии с текущей погодой;
- ☑ фрактал Мандельброта, обои реального времени;



Куб в KWin с новой темой Air, прозрачностью, тонкими панелями



- ☑ глобус, который требует наличия Marble (о нем – см. ниже) и обладает частью его возможностей (например, автоматический поворот Земли в свет за линией освещённости).

Для многих пользователей KDE был важен вопрос, можно ли ставить разные обои на разные виртуальные рабочие столы. И вот начиная с версии KDE 4.3 это можно сделать, нажав на «Кешью → Отдалить → Настроить Plasma» и поставив галку напротив «Отдельная комната для каждого рабочего стола». В результате пользователь получает комнаты/рабочие столы, где каждый из них – со своими обоями и набором виджетов.

## Виджеты

С выпуском KDE 4.3 не только произошли изменения в старых виджетах, но и появились новые. Если говорить о ранее известных виджетах, то изменения можно найти в следующих:

- ☑ виджет ведения заметок получил возможность использования «обогащенного» текста (rich text);
- ☑ виджет микроблоггинга стал поддерживать identi.ca;
- ☑ Pastebin был полностью переделан;
- ☑ виджет времени и даты научился показывать праздничные дни (среди них есть и российские: от Куликовской битвы до Дня ракетных войск) и проговаривать время;
- ☑ «просмотр папки» теперь умеет заглядывать внутрь структуры каталогов: при наведении курсора на директорию открывается всплывающее окно с на-

ходящимися в ней каталогами и файлами (и так – вплоть до конца ветки файловой системы);

- ☑ в Lancelot добавили очистку истории последних файлов, возможность настраивать клавиши.

А вот новые виджеты рабочего стола:

- ☑ **Bubble Monitor** – забавный системный монитор с пузырьками;
- ☑ **Magnifique** – самая обычная лупа;
- ☑ **Media Player** – проигрыватель мультимедиа, который у меня не захотел работать (точнее, он не стал показывать видео, хотя со звуком все нормально);
- ☑ **openDesktop** – виджет интеграции с одноименным сервисом, который показывает информацию о вас и вашем местоположении, ваших друзей и другое;
- ☑ **Remember The Milk Plasmoid** – виджет для интеграции с популярным онлайн-сервисом, позволяющий создавать и удалять задачи, ставить отметки, однако мне почему-то не удалось осуществить вход;
- ☑ **Преобразование величин** – конвертирование единиц измерения.

## KWin и KRunner

Оконный менеджер KDE получил прирост производительности и новые эффекты «Лист» и Slide Back. Повышение скорости заметно невооруженным глазом: эффекты работают быстро и плавно, без задержек. С эффектом для переключения окон «Куб» стало приятно работать: он плавно отъезжает назад, а его прокрутка не вызывает нареканий. Была реализована

поддержка SVG-тем для рабочего стола и всплывающих подсказок.

Прежде чем говорить об изменениях KRunner, стоит рассказать о нем самом. KRunner – это система выполнения команд и запуска приложений. Она доступна несколькими способами: через <Alt>+<F2>, через Kickoff (стандартное меню управления приложениями) и в Lancelot. KRunner построен на основе подключаемых модулей и может быть настроен по своему усмотрению. Одно из изменений в интерфейсе запуска команд – справка по возможностям, которая доступна по нажатию на знак вопроса рядом с кнопкой закрытия. Среди возможностей:

- ☑ запуск приложений;
- ☑ поиск по посещенным страницам в Интернете, открывавшимся документам и файлам, закладкам браузера, в Nepomuk, в письмах и контактах;
- ☑ калькулятор;
- ☑ конвертер единиц;
- ☑ открытие сеансов Kate, Konsole, Konqueror, веб-адресов;
- ☑ выход из системы.

Достаточно мощное приложение поиска и запуска – особенно благодаря новому способу отображения результатов, пришедшему на смену старому и беспорядочному. Во-первых, новый способ подразумевает использование значков для результатов, подписей длиной в строку и в случае необходимости описание найденного элемента (например, «последние документы»). Найденные варианты также получили дополнительные параметры – к примеру, запуск от имени другого пользователя.

## Приложения KDE

Перейдем к перечислению нововведений в приложениях. Если говорить о «Параметрах системы», то был возвращен режим отображения в стиле KControl (древовидный), а в обычный режим во всплывающих подсказках добавили показ внутренних пунктов. Был изменен диалог работы с ошибками программ, который помог разработчикам повысить эффективность подготовки финального выпуска KDE 4.3: в него внедрили оценку сформированного отчета, чтобы повысить качество собираемого материала, и добавили подсказки по тому, как повысить это качество. В KDE включен PolicyKit, который позволяет более гибко управлять привилегиями. Klrpper научился более «интеллектуально» обрабатывать полученные из буфера данные, предугадывая действия и позволяя настроить их. Менеджер архивов Ark получил поддержку LZMA и улучшения



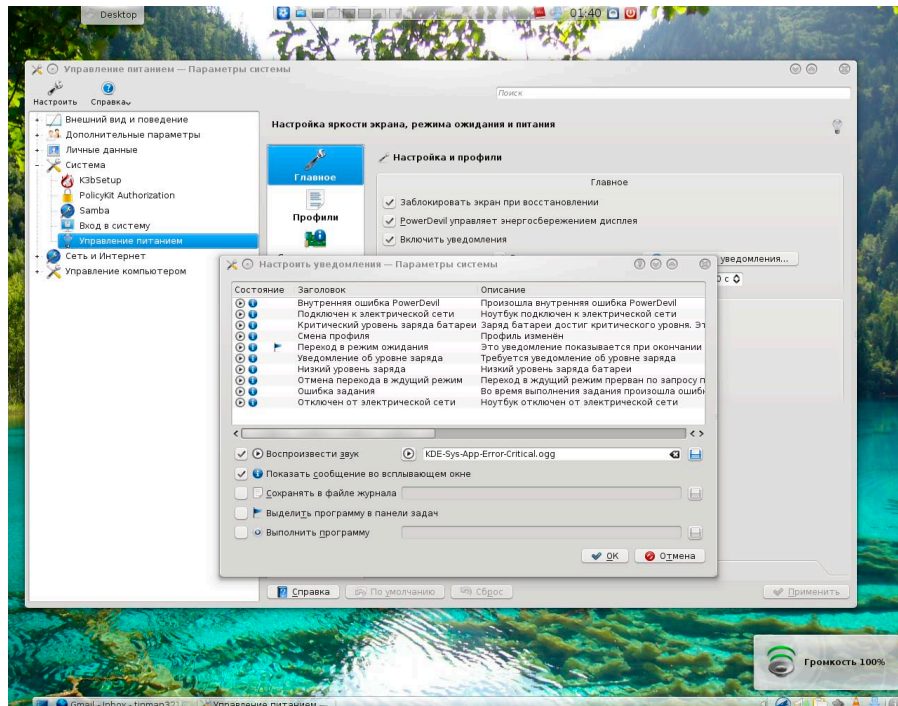
Виджеты рабочего стола

для других форматов, KPGP теперь интегрирован с Solid для получения данных об активном соединении и переведен на новый системный лоток.

Dolphin, файловый менеджер KDE, впервые получил возможность сортировки каталогов и поддержку служб Zeroconf (чтобы убедиться в этом, можно ввести в адресной строке «network:/»). Если внутри директории есть графические файлы, они будут показываться в виде небольших эскизов на значке каталога. Кроме того, теперь можно быстро просмотреть видеофайл благодаря интеграции с Phonon. Появились настройки корзины: резервируемое место и пути. Там же, в настройках Dolphin, стало возможным конфигурировать контекстное меню файлового менеджера, переключая дополнительные контекстные возможности (вроде быстрого прожига ISO-образа).

Отдельно стоит упомянуть Marble 0.8, новый выпуск которого состоялся одновременно с выходом KDE 4.3. Marble — это глобус KDE, включающий разнообразные карты, поддержку StreetMaps и другое. В него была внедрена поддержка английской системы мер, Wikipedia и изображений с Flickr, добавлена карта Луны и некоторые другие исторические карты. Более подробно почитать об изменениях можно на <http://knotes.ru/2009/08/marble-visual-changelog>.

Приложения из набора PIM (менеджера персональной информации) тоже подверглись некоторым изменениям. KOrganizer получил синхронизацию с Google Calendar благодаря новой возможности Akonadi. Представлены небольшие изменения в списке контактов Kopete,



Настройки системы в стиле kcontrol, всплывающее окно от нового трея

поддержка drag'n'drop, экспорта и увеличение настроек для KAlarm. С выходом KDE 4.3.1 в KMail было закрыто большое количество ошибок, добавлена поддержка показа в теле сообщения вложенных изображений.

Для некоторых игр из состава KDE Games помимо использования общей, египетской, темы добавлены уровни. Кроме того, появилась новая игра — KTron, а в Vovo усовершенствовали интеллект.

Хотя мой обзор новшеств в KDE 4.3/4.3.1 на этом закончен, я перечислил далеко не все, что было сделано сообществом KDE и сторонними разработчиками за последнее время.

## Выводы

Релиз KDE 4.3 — это еще один большой скачок вперед в плане создания лучшей свободной рабочей среды. Как видно из обзора, изменения охватили практически все сферы: от работы «под капотом» до композитных эффектов. KDE продолжает развиваться, а мы продолжаем пользоваться и следить за событиями. Ответить для себя на вопрос «Готов ли KDE 4 для рабочего стола?», поднятый в обзоре KDE 4.2 (см. «Open Source» 039 от 20.02.2009), по-прежнему можете лишь вы сами, но лично я уже ответил на него положительно.

Никита Лялин  
([tinman321@gmail.com](mailto:tinman321@gmail.com))

## Ускорение Ubuntu Linux. Часть 1

В любой операционной системе есть настройки, изменение которых приводит к ее ускорению. Настройки по умолчанию обычно всегда консервативны, так как их назначение — работать на большинстве компьютеров. Если понимать, что приобретаешь и что теряешь, изменяя те или иные настройки, можно добиться существенного ускорения работы системы и приложений. В статье вы не найдете тестов, которые количественно указывают, насколько быстрее стали те или иные операции или приложения. В ней приведены эмпирические советы, которые можно с осторожностью приме-

нять, наблюдая за изменениями и последствиями в вашей системе.

В материале речь идет об «обычной» версии Ubuntu Linux, поэтому рассматриваются графическая среда GNOME и другие стандартные приложения, входящие в поставку дистрибутива по умолчанию.

### Отключение Compiz

Compiz — композитный менеджер окон для X Window, использующий OpenGL для предоставления графических эффектов, доступных в различных графических средах. Если у вас установлены драйверы к видеокарте, то скорее всего Compiz бу-

дет (по умолчанию) включен. Однако через некоторое время умильные вздохи по поводу «сгоравших» окон проходят и появляется мысль упростить Compiz или отключить его совсем. Если хочется упростить эффекты, нужно установить пакет compizconfig-settings-manager через Synaptic или скопировать в терминале `sudo aptitude install compizconfig-settings-manager`. Через меню «Система → Параметры → Менеджер настройки Compiz Config» вы получите доступ к возможностям Compiz и сможете упростить эффекты по вкусу. Если желаете просто отключить Compiz, это можно сделать через пункт «Меню → Система → Параметры → Внешний вид», где во вкладке «Внешний вид» выставить «Без эффектов». У каждого будут свои доводы за и против Compiz. Лично я отключил его, обнаружив, что моя видеокарта nVidia с включенным Compiz

# Электронное приложение «Open Source»

в режиме «Производительность по запросу» постоянно работает на полную нагрузку и не собирается сбавлять обороты вентилятора. Включать или не включать – решать вам.

## Уменьшение задержек GTK

Многие знают, что в Ubuntu рабочей средой является GNOME, в котором важную роль играет GTK+. Изменяя общие настройки GTK+, можно влиять на все приложения, использующие эти библиотеки. Я предлагаю модификацию следующих параметров, отвечающих за задержки (паузы):

- ☑ **gtk-menu-popup-delay** – минимальное время в миллисекундах, которое указатель должен оставаться на пункте меню перед появлением подменю. По умолчанию установлено значение 225.
- ☑ **gtk-menu-popdown-delay** – время в миллисекундах перед скрытием подменю, когда указатель перемещается на подменю. По умолчанию – 1000.
- ☑ **gtk-menu-bar-popup-delay** – задержка в миллисекундах перед появлением подменю после панели меню. По умолчанию – 0.

Подробнее обо всех этих параметрах можно прочитать на <http://library.gnome.org/devel/gtk/2.15/GtkSettings.html>.

Запустите «Меню → Стандартные → Текстовый редактор» и вставьте следующие строки:

```
gtk-menu-popup-delay = 0
gtk-menu-popdown-delay = 0
gtk-menu-bar-popup-delay = 0
```

Сохраните в своем домашнем каталоге под названием .gtkrc-2.0 и обратите внимание на первый символ в имени файла – это должна быть точка. Таким образом, создается скрытый файл, содержащий параметры для GTK+. Теперь можно запустить терминал и проверить себя, введя команду:

```
cat ~/.gtkrc-2.0
```

или в файловом менеджере (Nautilus) нажать <Ctrl> + <H>, после чего найти скрытый файл .gtkrc-2.0 в домашнем каталоге. Теперь можно завершить сеанс GNOME и зайти снова – быстрое переключение в различных меню должно быть вам наградой.

Минусы: возможно, придется отключить Compiz, который многими вещами занимается сам, а следовательно перекрывает некоторые параметры. Как отключить Compiz – см. выше.

## Утонченная настройка GNOME

В GNOME настройки хранятся по аналогии с реестром MS Windows – с той лишь разницей, что «реестр» GNOME – это XML-файлы. Доступ к «реестру» можно получить через редактор конфигураций gconf-editor или через вызов команд типа gconftool-2. Самый легкий способ внести требуемые изменения – это запустить «Меню → Стандартные → Терминал» и копировать/вставлять строки, начинающиеся с gconftool-2. Права суперпользователя не нужны, так как мы правим свои настройки, а не системные.

Вот и сами команды:

```
gconftool-2 --type bool -j
--set /apps/metacity/general/reduced_resources true
```

Metacity (оконный менеджер GNOME) будет стараться избегать анимации и уменьшит потребление ресурсов. Единственное, что визуально изменится, – при перемещении окон не будет отображаться их содержимое. Предварительно узнать зна-

чение параметра и/или убедиться, что изменения произошли, можно запустив в терминале команду:

```
gconftool-2 --get /apps/metacity/general/reduced_resources
```

Команда:

```
gconftool-2 --type bool -j
--set /apps/panel/global/enable_animations false
```

выключает анимацию включения/выключения панелей. Предварительно узнать значение параметра и/или убедиться, что изменения произошли, можно командой:

```
gconftool-2 --get /apps/panel/global/enable_animations
```

Команда:

```
gconftool-2 --type bool -j
--set /desktop/gnome/interface/accessibility false
```

отключает «вспомогательные технологии» клавиатуры и мыши, специальные технологии при входе в систему для людей с ограниченными возможностями. Включить обратно можно в «Меню → Система → Параметры → Вспомогательные технологии». Предварительно узнать значение параметра и/или убедиться, что изменения произошли, можно командой:

```
gconftool-2 --get /desktop/gnome/interface/accessibility
```

Команда:

```
gconftool-2 --type string --set /apps/panel/global/
panel_animation_speed panel-speed-fast
```

устанавливает «быструю» скорость анимации панелей. Предварительно узнать значение параметра и/или убедиться, что изменения произошли, можно командой:

```
gconftool-2 --get /apps/panel/global/panel_animation_speed
```

Еще одна команда:

```
gconftool-2 --type integer -j
--set /apps/metacity/general/auto_raise_delay 100
```

Если у вас установлен режим, в котором при наведении мыши на окно открытого приложения последнее выдвигается на передний план, то параметр auto\_raise\_delay контролирует задержку между наведением мыши и выдвиганием окна на передний план. Предварительно узнать значение параметра и/или убедиться, что изменения произошли, можно командой:

```
gconftool-2 --get /apps/metacity/general/auto_raise_delay
```

## Ускорение запуска программ Qt

В мире \*nix традиционно считается, что GNOME – это GTK+, а KDE – это Qt, но пользователю ничто не мешает запускать программы GTK+ в KDE, а Qt-приложения – в GNOME. Более того, авторы GNOME и KDE многое сделали для того, чтобы внешний вид «неродных» программ выглядел так же, как и «родных». Поэтому стоит уделить внимание тому, как ускорить запуск программ, написанных с использованием Qt, в GNOME.

Но сначала нужно уяснить, что программа, написанная на Qt, может не зависеть от KDE, а может дополнительно использовать возможности KDE. Вот такие зависящие от KDE



программы при своем старте должны получить «минимальный набор KDE».

Как узнать, какие программы зависят от KDE? В Ubuntu программы, которые устанавливаются по умолчанию, чаще всего используют GTK+ или Python. Qt-приложения вы скорее всего инсталлировали уже самостоятельно. Если помните, что это были за программы, можно найти их в Synaptic и убедиться, прочитав описание к программе и заглянув в ее зависимости. Кроме того, Synaptic есть пункт меню «Файл → История», который напоминает хронологию событий. Если в терминале вызвать команду:

```
sudo aptitude search '~i!~nlib(~Dqt|~Dkde)'
```

то она выведет пакеты, которые установлены (~i), не являются библиотекой (!~nlib), зависят от Qt (~Dqt) или от KDE (~Dkde). Эти программы скорее всего выиграют от данного метода.

Заранее запуская «минимальный набор KDE», мы получим ускорение в старте многих Qt-программ. Для этого в «Меню → Система → Параметры → Запускаемые приложения» добавим /usr/bin/kdeinit под названием FastQt. После перезагрузки компьютера Qt-программы будут быстрее стартовать, так как не нужно дополнительно запускать «минимальный набор KDE».

**Минусы:** добавление лишней записи в «Запускаемые приложения» замедляет время старта компьютера. Выигрыш в скорости запуска Qt-приложений достигается за счет удерживаемых в памяти библиотек KDE, запущенных заранее. Если у вас достаточно оперативной памяти, то это стоящая сделка. Если оперативной памяти мало, скорее всего лучше не применять данный совет и использовать в Ubuntu только GTK-программы, избегая установки Qt/KDE-приложений (часто их названия начинаются с буквы «k»).

## Ускорение запуска программ с помощью preload

Preload – работающий в фоновом режиме демон, который собирает информацию о наиболее часто используемых программах, кэширует их и используемые ими библиотеки, что приводит к повышению скорости загрузки программ. Нужно просто установить preload в терминале (или через Synaptic) командой:

```
sudo aptitude install preload
```

Настройки preload по умолчанию подходят большинству, поэтому изменять что-либо в файле /etc/preload.conf необязательно.

**Минусы:** если у вас действительно мало ОЗУ, то возможно, не стоит устанавливать preload: пусть тогда лучше ваш компьютер работает в режиме «запустил программу – поработал – закрыл программу и освободил ОЗУ».

## Корректное преобразование имени компьютера

При установке Ubuntu на свой компьютер на одном из этапов вы задавали имя компьютера. Многие пользователи не имеют в локальной сети своего доменного имени, а домашние пользователи, не имеющие сети, вряд ли занимались созданием своего DNS-сервера. В то же время многим программам нужно перейти от имени компьютера к IP-адресу – это называется прямым преобразованием DNS-имени в IP-адрес. Если преобразование не удастся, возникнут ненужные временные задержки. Проверьте, так ли это, вызвав терминал и вставив в него команду nslookup `hostname` (обратите внимание на кавычки в команде). Если имя вашего компьютера не удалось корректно преобразовать в IP-адрес, следует сделать следующее:

В терминале скопируйте:

```
sudo gedit /etc/hosts
```

вы вызовете текстовый редактор с правами суперпользователя, которые дадут вам возможность изменить системный файл.

Найдите строку:

```
127.0.0.1 localhost
```

Она связывает зарезервированное слово localhost с адресом 127.0.0.1. Кратко говоря, localhost и 127.0.0.1 – это и есть ваш компьютер.

Эту строку нужно дополнить до вида:

```
127.0.0.1 localhost Имя_Вашего_Компьютера
```

Если вы забыли имя вашего компьютера, то команда hostname в терминале напомнит вам.

Сохраните изменения в текстовом редакторе и проверьте снова, как происходит преобразование имени в IP-адрес с помощью той же команды:

```
nslookup `hostname`
```

**Минусы:** вы редактируете системный файл, с которым требуется осторожность. Желательно перед изменениями в системных файлах делать их резервную копию, чтобы иметь возможность откатить все модификации.

## Чистка и создание заново индексов SQLite

Многие программы в Ubuntu хранят данные не в текстовых файлах, а в базах данных. И часто для последнего используется система управления базами данных (СУБД) SQLite. Средствами самой SQLite можно провести чистку пустых записей и создать заново индексы файла базы данных. Лучший вариант – написать скрипт, содержащий вызовы SQLite, для его периодического запуска время от времени. Итак:

1. Установите sqlite3 командой:

```
sudo aptitude install sqlite3
```

2. Запустите файловый менеджер Nautilus и создайте каталог bin в домашней директории.

3. Перейдите в созданный bin и, будучи в нем, выберите в меню Nautilus: «Файл → Создать документ → Пустой файл». Дайте имя файлу – например, optimizer.sh.

4. Откройте этот файл в текстовом редакторе и вставьте в него следующие строки:

```
#!/bin/sh
# Скрипт для оптимизации данных в СУБД SQLite, используемых
# прикладными программами. Он выполняет переиндексацию и удаление
# из таблиц пустых записей
clear
echo "Старт SQLite VACUUM REINDEX"

# Оптимизация для Firefox:
find ~/.mozilla/ -name '*.sqlite' -print -exec sqlite3 {} \;
"VACUUM; REINDEX;" > /dev/null 2>&1 \;

# Оптимизация для Epiphany:
find ~/.gnome2/epiphany -name '*.sqlite' -print -exec sqlite3 {} \;
"VACUUM; REINDEX;" > /dev/null 2>&1 \;

# Оптимизация для Liferea
sqlite3 ~/.liferea/liferea.db "VACUUM; REINDEX;" > /dev/null 2>&1
echo "Окончание SQLite VACUUM REINDEX"
```

5. Сохраните изменения в файле. Теперь в Nautilus на файле `optimizer.sh` щелкните правой кнопкой мыши и вызовите «Свойства», в закладке «Права» выставьте права «Выполнения» (это можно сделать и в терминале с помощью команды:

```
chmod +x ~/bin/optimizer.sh
```

6. Теперь можно периодически вызывать этот скрипт (`optimizer.sh`), когда за-

крыты программы, файлы баз данных которых вы оптимизируете. В данном случае это браузеры Firefox и Eiphanu, RSS-агрегатор Liferea.

Скрипт `optimizer.sh` оптимизирует файлы базы данных SQLite, находящиеся в домашнем каталоге пользователя. Если у вас в системе зарегистрированы несколько пользователей, то каждый из них должен вызывать `optimizer.sh` для оптимизации своих файлов SQLite.

На этом первая часть статьи заканчивается, а ее продолжение выйдет в следующем номере «Open Source» (051). В нем будут рассмотрены возможности ускорения работы Eiphanu и системного меню, создание кэшей, оптимизация подкачки, быстрый старт OpenOffice.org. — **Прим. ред.**

Василий Алексеенко  
([vasilisc@rambler.ru](mailto:vasilisc@rambler.ru))

## Свободный паскаль. Часть 1: Free Pascal и GNU Pascal

Это первая часть статьи, где речь пойдет о свободных реализациях языка Pascal. Я не буду касаться синтаксиса и приемов программирования на нем (за исключением новой для Pascal концепции шаблонов) — об этом написано огромное количество книг и статей. Все основное внимание уделено обзору компиляторов, утилит и IDE из проектов GNU Pascal (<http://www.gnu-pascal.de/gpc/h-index.html>), Free Pascal (<http://www.freepascal.org>) и Lazarus (<http://www.lazarus.freepascal.org>).

### Введение

Ни для кого не секрет, что Pascal — очень популярный язык программирования на просторах бывшего СНГ. Он был создан Никлаусом Виртом в 70-х годах прошлого века как язык, предназначенный для обучения студентов структурному программированию. С тех пор Pascal прочно закрепился как учебный и благодаря этому становился первым языком как для большинства будущих профессиональных программистов, так и для любителей.

Постепенно Pascal развивался в полноценный современный объектно ориентированный язык Object Pascal, попутно дав начало другим (среди них — Ada, Modula и Oberon).

Последний всплеск интереса к Pascal пришелся на 2000-2005 годы и был связан с огромной популярностью Delphi ([http://en.wikipedia.org/wiki/CodeGear\\_Delphi](http://en.wikipedia.org/wiki/CodeGear_Delphi)) — интегрированной среды разработки для Windows, базирующейся на Object Pascal. Начиная с Delphi 7 сам язык претерпел некоторые изменения и также стал формально называться Delphi. Однако в свободных реализациях его обычно продолжают называть Object Pascal (этим названием я буду пользоваться в статье). В последние годы среда Delphi утратила былую популярность, и Pascal опять «припомнили» его имидж «несерьезного» языка. Этому поспособствовало и наличие своеобразной обратной связи: язык не очень часто используется в серьезных проектах именно потому, что мало серьезных проектов реализованы на нем. Есть, конечно, и объективные причины: в отличие от C++

в Pascal до последнего времени отсутствовали средства обобщенного программирования. Однако после появления в последних версиях Object Pascal обобщенных классов он по своим возможностям приблизился к таким «серьезным» языкам, как C++ или Java. Тем не менее часто можно услышать мнение, что Pascal — мертвый язык, который практически перестал развиваться и использоваться. Лучшим подтверждением того, что это не так, являются его свободные реализации.

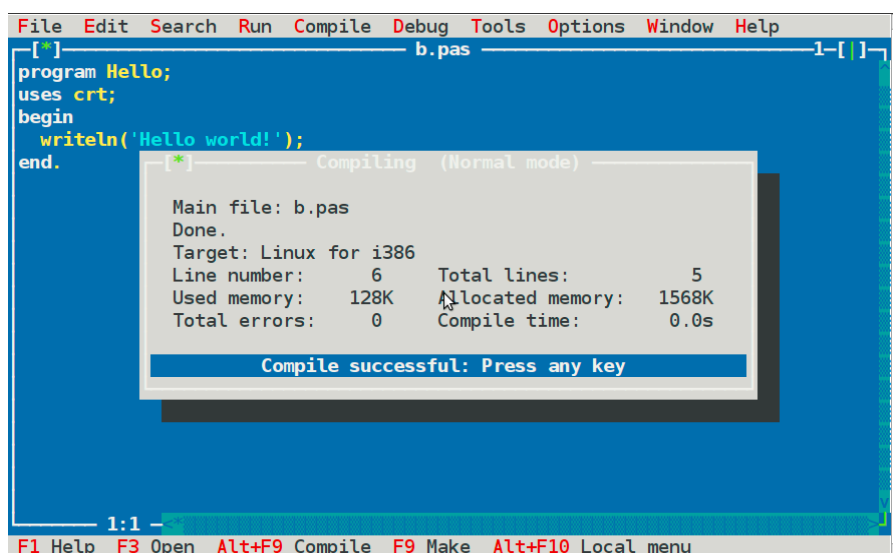
Строго говоря, «свободный паскаль» включает в себя два взаимосвязанных проекта: Free Pascal и Lazarus. Еще отдельно от них существует компилятор GNU Pascal, входящий в GCC, но об этом — см. ниже. Free Pascal в свою очередь включает консольный компилятор `fps`, полноценную текстовую IDE, богатую библиотеку модулей и набор утилит. Lazarus — это свободный кроссплатформенный клон среды разработки Delphi (о нем речь пойдет во второй части статьи). Free Pascal работает на разных платформах, включая Linux, \*BSD, Windows и Mac OS X. Free Pascal не является частью GCC, однако в Linux компилятор `fps` использует компоновщик `ld` (GNU linker, входящий в GCC).

### Установка

Free Pascal присутствует в репозиториях всех основных Linux-дистрибутивов, и его установка не вызывает сложностей. Однако если вы планируете установить и Lazarus, нужно быть готовым к тому, что версия `fps` из репозитория может не совпадать с той, которая требуется для Lazarus (подробнее об этом — во второй части статьи).

### Компилятор fps

Диалект Паскаля, который лежит в основе компилятора `fps`, практически совпадает со стандартом Object Pascal, который использовался в Delphi 7 (он же «язык Delphi»). Компилятор может работать в различных режимах совместимости, эмулируя поведение компиляторов Turbo Pascal 7.0 (опция `-Mtp`), Delphi 7 (`-Mdelphi`)



Текстовая IDE fp — реинкарнация Turbo Pascal 7.0



или Macintosh Pascal (-Mmacpas). Имеется также собственный диалект Free Pascal (очень похожий на Turbo Pascal – опция -Mfpc) и комбинированный режим Free Pascal + Object Pascal (-Mobjfpc), который используется чаще всего. Компилятор может генерировать как 32-, так и 64-битный код.

Возможности оптимизации в fpc достаточно скромны по сравнению с компиляторами GCC. Тем не менее доступны четыре уровня оптимизации по скорости (от -O- до -O3, где режим -O- означает отсутствие оптимизации и аналогичен -O0 в GCC) и оптимизация по размеру кода. Оптимизация для конкретной процессорной архитектуры ограничивается различными видами процессоров Pentium и наборами инструкций SSE, SSE2 и SSE3. Доступны также разворачивание циклов (loop unrolling) и некоторые другие оптимизации (fpc -i показывает полный список).

Скорость компиляции в сравнении с компиляторами C и C++ из GCC – очень большая. Компилятор способен «пережевать» мегабайты кода за считанные секунды. В Linux эти показатели серьезно портит компоновщик ld, который печально известен своей медлительностью. Именно поэтому в следующие версии Free Pascal планируют включить собственный линкер для Linux. Скорость компиляции объясняется тем, что все компиляторы паскаля традиционно являются однопроходными, тогда как компиляторы C и C++ могут проходить по коду несколько раз. Минусом такого подхода является меньшая способность к оптимизации – программы на Pascal, как правило, работают немного медленнее аналогов на C/C++. Эта разница редко превышает несколько процентов и в большинстве задач незаметна, но критические вычислительные ядра все же лучше не писать на Pascal. В то же время компилятор fpc использует меньше памяти, чем gcc и особенно g++ (<http://shootout.alioth.debian.org>).

## IDE fp

Текстовая IDE fp, входящая в Free Pascal, является практически точным клоном знаменитого Turbo Pascal 7.0 (см. **рисунок**).

Можно смело утверждать, что это эталон для «легких» текстовых IDE по удобству использования. Несмотря на свою схожесть с «древним» прототипом, fp – вполне современная среда разработки, не унаследовавшая практически никаких ограничений системы MS-DOS. В fp нет понятия «проекта», как в полновесных IDE, поэтому использовать ее для больших программ со сложным деревом исходных файлов и многочисленными зависимостями нецелесообразно (как правило, этого и не требуется – для таких целей используется Lazarus). В fp сохраняется список последних открытых файлов, и можно настраивать пути к используемым модулям и библиотекам, поэтому для небольших программ ее возможностей вполне хватает. Для написания простых консольных программ и для программистов «старой школы» такая IDE – настоящая находка. Для начального обучения программированию она также подходит как нельзя лучше, поскольку прекрасно работает даже на очень слабых машинах (которых и сейчас много в школах и вузах), и не ошеломляет новичка огромным количеством малопонятных функций. Хорошо интегрированный отладчик и понятные сообщения об ошибках также способствуют эффективности обучения. Практически полная совместимость с Turbo Pascal 7.0 достигается и на уровне модулей, так что в учебном процессе можно использовать многие проверенные временем примеры и программы.

Как интерфейс Turbo Pascal 7.0 был построен на библиотеке Turbo Vision от Borland, так и интерфейс fp построен на свободном аналоге – Free Vision ([http://wiki.freepascal.org/Free\\_Vision](http://wiki.freepascal.org/Free_Vision)). По сути, это «текстовый GUI» с окнами, кнопками, списками, диалогами и текстовым редактором, API которого полностью совместим с Turbo Vision. К сожалению, документации по Free Vision

на официальном сайте нет, из чего можно сделать вывод о том, что разработчики Free Pascal рассматривают эту библиотеку исключительно как служебную.

Ложкой дегтя является и то, что fp не поддерживает кириллицу в Linux (в Windows этой проблемы не возникает). Информации по этому вопросу в Сети крайне мало, но создается впечатление, что это принципиальное ограничение Free Vision, обойти которое невозможно.

## Стандартная библиотека и утилиты

Библиотека модулей для Free Pascal состоит из стандартной библиотеки времени выполнения (RTL) и библиотеки свободных компонентов (FCL). RTL содержит модули совместимости с Turbo Pascal и Delphi, кроссплатформенные модули для работы с мышью и клавиатурой в консольном режиме, модули для платформо-зависимых функций и базовые модули для работы с математикой, датами и т.п. В FCL входят модули для различных структур данных (списков, хешей и т.п.), некоторые алгоритмы шифрования, модули для потокового ввода-вывода. FCL служит базой для построения библиотеки визуальных компонентов LCL в Lazarus. Кроме базовой библиотеки на официальном сайте доступны и многочисленные модули, написанные пользователями. Среди них – интерфейсы к базам данных (dbase, ODBC, PostgreSQL), регулярные выражения, интерфейсы к архиваторам, разнообразные модули для работы с графикой в различных форматах, для работы с HTML, сетью и Интернетом. Нужно заметить, что на практике чаще используется не базовая FCL, а LCL, в которой гораздо больше современных модулей (например, для работы с MySQL). О ней пойдет речь во второй части статьи.

Free Pascal включает также набор утилит (<http://www.freepascal.org/tools/tools.var>). Большая их часть – служебные программы, применяемые разработчиками самого Free Pascal. Внимания заслуживают немногие из них:

- ☑ **h2pas** – конвертирует заголовочные файлы C в модули Паскаля. Используется для создания «обертки» вокруг библиотек, написанных на C.
- ☑ **fpcmake** – утилита для создания Make-файлов для сложных программ и модулей. По принципу работы напоминает Automake.
- ☑ **ptop** – форматирует исходный код на паскале с соблюдением отступов и других правил. Аналог утилиты indent для C.

## Шаблоны классов в Object Pascal

Как я уже упоминал, одним из серьезных недостатков Pascal долгое время являлось отсутствие средств обобщенного программирования – шаблонов. Любой алгоритм на паскале был жестко «привязан» к определенному типу данных, и создать обобщенные алгоритмы и структуры данных наподобие библиотеки STL в C++ было невозможно. Проблему можно было в некоторых случаях обойти, используя полиморфные указатели на классы, связанные наследованием, но это не решало ее в принципе. Сейчас в Free Pascal существуют шаблонные классы (шаблонов функций пока нет). Шаблоны реализованы как своеобразные макросы, которые просто подставляют нужный тип данных в обобщенный код на этапе компиляции. Насколько мне известно, такая реализация не позволяет писать «шаблонные метапрограммы», по сути, расширяющие возможности самого компилятора, как это можно сделать на C++ ([http://en.wikipedia.org/wiki/Template\\_metaprogramming](http://en.wikipedia.org/wiki/Template_metaprogramming)). Шаблоны в Pascal выполняют только свою первичную функцию – позволяют не писать один и тот же код для разных типов данных.

Вот простейший пример, иллюстрирующий использование шаблонов:

```
program Hello _template;

type
// Объявляем обобщенный (generic) класс
generic Tadder< _T>=Class(TObject)
// Вместо _T будет подставлен нужный тип данных
// после специализации шаблона
procedure add(item1, item2: _T);
end;

// Необходимо явно специализировать шаблоны для всех
// типов данных, которые мы хотим использовать
Tint _adder = specialize Tadder<integer>;
Tstr _adder = specialize Tadder<string>;

// Тело метода шаблонного класса. В отличие от C++
// повторять объявление «generic» не нужно.
procedure Tadder.add(item1, item2: _T);
begin
writeln('Sum of elements: ',item1+item2);
end;

var adder1: Tint _adder;
    adder2: Tstr _adder;
begin
    adder1.add(1,2);
    adder2.add('Hello', ' world!');
end.
```

Как видно из этого примера, шаблон должен быть специализирован для конкретного типа данных до его использования путем создания нового типа. Не очень удобно по сравнению с C++, но все же намного лучше, чем ничего. Чтобы скомпилировать приведенную программу, нужно явно указать компилятору, что используется диалект «Object Pascal»:

```
fpc -Mobjfpc filename.pas
```

## GNU Pascal

В GCC входит свой свободный компилятор паскаля – gpc, который не имеет отношения к Free Pascal. Он существенно уступает в популярности fpc. Самые свежие материалы по gpc в Се-ти датируются 2006 годом. Создается впечатление, что проект в настоящее время не развивается.

Gpc доступен для всех платформ, где работает GCC – в том числе Linux, \*BSD, Windows и Mac OS X. В Ubuntu gpc входит в состав устаревшей gcc 4.1, поэтому вместе с ним устанавливаются и соответствующие базовые библиотеки gcc этой версии. Компилятор базируется на стандарте языка Turbo Pascal 7.0 с добавлениями из Delphi и Mac Pascal. Полный список особенностей языка доступен на официальном сайте (<http://www.gnu-pascal.de/gpc/h-index.html>). Среди этих особенностей – поддержка нескольких расширений языка, таких как динамические массивы со схемами (shemata). Поддержки шаблонов в gpc нет, так как они появились в языке уже после прекращения активности в этом проекте. На официальном сайте имеются ссылки на несколько IDE, поддерживающих gpc, однако все эти проекты заброшены не позже 2005 года. Степень совместимости gpc и fpc не ясна. В старых версиях fpc существовал режим совместимости с gpc, включаемый опцией -Mgpc, но в текущей версии его нет. Это косвенно подтверждает тот факт, что GNU Pascal, по всей видимости, заброшен и использовать его стоит только в тех случаях, когда Free Pascal по каким-то специфическим причинам не подходит.

## Выводы

Free Pascal является очень удачной кроссплатформенной реализацией Pascal, включающей компилятор, библиотеку модулей и текстовую IDE. Free Pascal очень хорошо документирован. Кроме исчерпывающей официальной документации к компилятору и IDE имеется обширное wiki (объединенное с wiki по Lazarus) на нескольких языках, включая русский и украинский (<http://wiki.freepascal.org>). Free Pascal хорошо приспособлен для обучения программированию (единственной серьезной проблемой здесь является отсутствие поддержки кириллицы в IDE в Linux). Однако полностью возможности Free Pascal раскрываются, если использовать его вместе со средой визуальной разработки Lazarus, о которой пойдет речь в следующей части статьи.

Семен Есиплевский  
([yesint4@yahoo.com](mailto:yesint4@yahoo.com))

## Cinelerra: прошлое, настоящее, будущее. Часть 2

Первую часть обзора программы нелинейного видеомонтажа Cinelerra читайте в предыдущем выпуске нашего издания – «Open Source» 049 (04.09.2009). – **Прим. ред.**

### Возможности и недостатки

Помимо всего прочего, Cinelerra оснащена внушительным списком эффектов реального времени для видео и звука. Здесь, как говорится, полный фарш – причем параметры эффектов, как уже отмечалось, могут быть автоматизированы. Но тогда возникает вопрос: если все хорошо, то зачем разработчики CV-версии взялись за создание новой, другой программы видеомонтажа – Lumiera (подробнее о ней – см. ниже)? Можно дать несколько ответов. Одно дело – возможность, а другое – способ её воплощения. Поместите

на монтажный стол в Cinelerra фрагмент видео и затем переместите его в произвольное место на дорожке. Конечно же, работает. Между тем такая возможность появилась относительно недавно – до этого свободного перемещения не было: надо было вырезать фрагмент и ставить его где-то, предварительно поместив на дорожку нужное количество пустого места (звучит странно, но так и есть, в виде функции Edit → Paste silence, оно же <Shift>+<Space>). Насколько я знаю, даже сейчас внутренне дело обстоит именно так: при перемещении фрагмента между ним и соседним слева объектом добавляется пустота – только теперь это происходит автоматически.

Если наложить эффект на фрагмент и попробовать опять переместить фрагмент, то окажется, что фрагмент пере-

местился, но вот эффект остался, где и был. Это вызвано тем, что все эффекты в Cinelerra накладываются на дорожку, а не на фрагмент. Так что вам придется брать мышью эффект на дорожке и перетаскивать его на перемещенный фрагмент. Благо при этом хоть остаются целыми ключевые кадры автоматизации, поэтому если вы в эффекте изменяли параметры при включенном создании ключевых кадров (кнопка Generate keyframes by tweaking), то, когда вы переместите эффект, все изменения параметров переместятся вместе с эффектом. Получается, эффекты в Cinelerra можно трактовать как почти что равноценные объекты наряду с фрагментами видео или звука. Такой подход снижает продуктивность работы, скорость монтирования.

Другая важная вещь – работа с так называемым футажом (footage) – видео и звуком, использующимися в проекте. Cinelerra не оснащена средствами предварительного просмотра в окне импорта файла. Кроме того, она не поддержива-

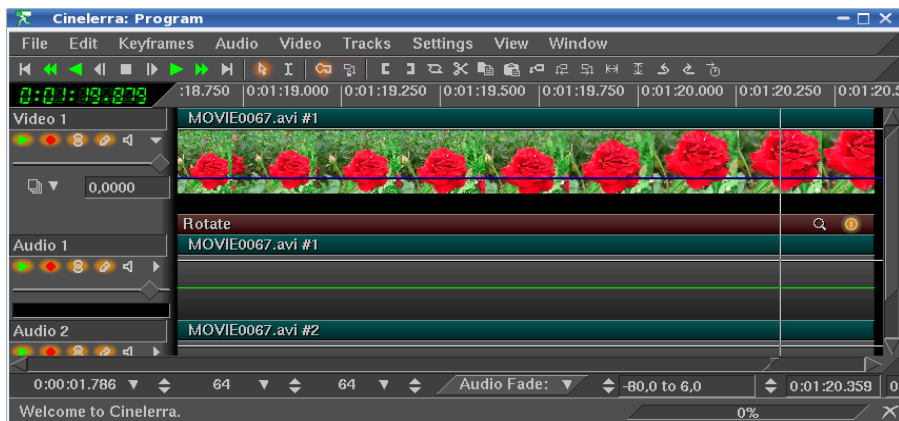
ет drag'n'drop, из-за чего в окне Cinelerra нельзя перетащить что-либо извне (например, из файлового менеджера). Теперь представьте, что у вас каталог, где несколько тысяч файлов, имена которых состоят из «временных штампов» (даты и времени съемки). Обычно так и делают: берут громадный исходный файл с отснятым видео и отдельной программой автоматически разрезают его на сцены (по сцене в каждом файле). Как выбрать нужный файл для вставки в проект? Очевидно, что надо либо импортировать все файлы и просматривать их по одному в окне Viewer внутри Cinelerra, либо импортировать в проект тот самый громадный исходный файл и разрезать его на сцены вручную уже в Cinelerra. Все импортированные в проект данные помещаются в раздел Media окна Resources, откуда также доступны эффекты, переходы и тому подобное. В Media нельзя делать вложенные виртуальные каталоги, то есть всё добавляется в общий список файлов футаж.

Чуть отвлекаясь от темы: время от времени в газетах пишут в разделе курьезов о людях, которые при жизни ходят едва не в лохмотьях, морят себя голодом, а когда помирают — оказываются тайными миллионерами: то у них золотые слитки под каждой паркетной, то банкноты за обоями утлой чердачной каморки.

Так вот, Cinelerra вызывает примерно такое же ощущение: потрясающие возможности скрыты в недоработанной, неудобной оболочке. Если с интерфейсом еще можно примириться, привыкнуть к нему или хотя бы принять его как данность, то ввод футаж и вывод материала в нужные форматы вызывают головную боль, которую не снять аспирином. Допустим, я вывожу из Cinelerra со следующими параметрами: контейнер AVI, кодек MPEG4 video. Программа рендерит — правда, по завершении считает, что рендеринг продолжается, но это ничего — файл-то уже готов. Однако оказывается, что в этом результирующем файле FOURCC (грубо говоря, сигнатура формата) выставлен от QuickTime-кодека MP4V. Популярные проигрыватели его воспринимают нормально, но вот, например, программа видеоредактирования AviDemux — уже нет. В итоге приходится прибегать к дополнительным манипуляциям для получения ожидаемого результата:

```
mencoder исходный_файл.avi -ovc copy -l
-oac copy -ffourcc DIVX -l
-o файл_результата.avi
```

Так мы принудительно выставляем FOURCC в более стандартный DIVX. О не-



Окно монтажного стола

правильном импорте видеофутажа в несжатом формате и несжатого звука из DV сообщалось выше, а ведь это важнейшие форматы, с которыми приходится работать при монтаже!

## Lumiera

Теперь немного обещанных подробностей о Lumiera (<http://lumiera.org>). Вообще разработчики CV-версии Cinelerra хотели переписать код программы и выдвинуть его как Cinelerra нового поколения — Cinelerra3. Но затем они отказались от этого решения и начали делать Lumiera. Довольно сложно понять, что происходит внутри проекта сейчас. Известно, что компилируются какие-то части Lumiera — статус git смотрите на <http://www.lumiera.org/gitweb?p=LUMIERA;a=summary> и <http://www.lumiera.org/gitweb>. Существует пробный графический интерфейс на GTK (вернее, на GTK-MM). Сама же архитектура Lumiera состоит из:

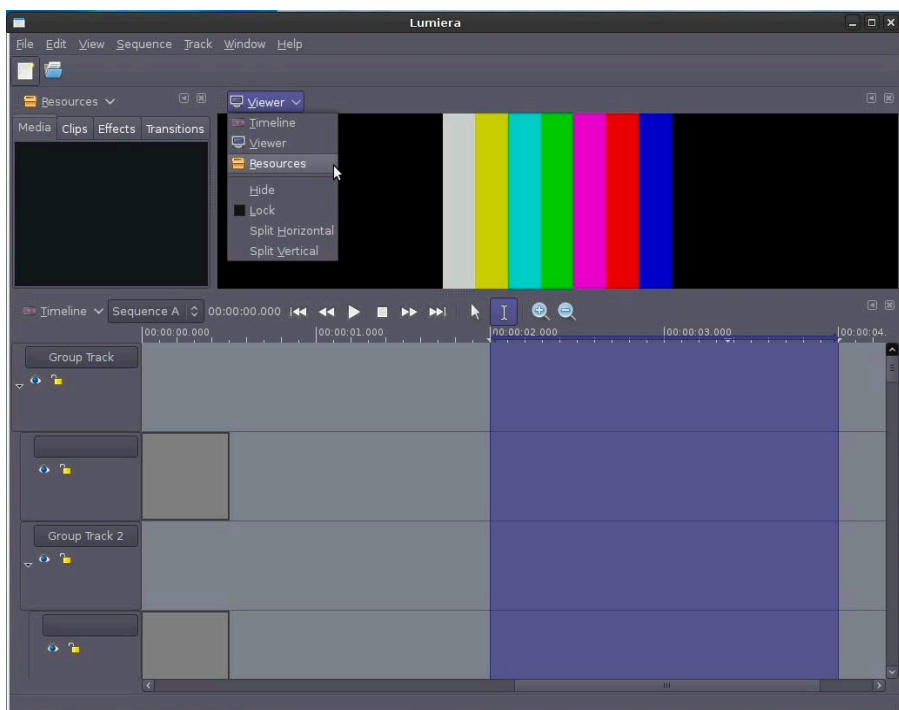
- ✓ бэкэнда, отвечающего за ввод-вывод;

- ✓ собственно монтажного движка;
- ✓ графического интерфейса.

И огромная страница (<http://www.pipapo.org/pipawiki/Lumiera/GuiBrainstorming>) отведена под мозговую штурм насчет этого самого интерфейса. Вообще все ссылки по теме можно найти на <http://www.pipapo.org/pipawiki/Lumiera/QuickStart>. Материалы частично устарели, в чем признаются сами разработчики, ибо еще прошлым летом они хотели взять за основу код Cinelerra, стали его переписывать, но затем остановились и передумали — стали писать с нуля. Всё это сопровождается встречами по IRC, обсуждениями прямо в Wiki проекта и так далее.

Весной сего года разработчики выложили свежий снимок экрана Lumiere. По нему мы более внятно, чем по Wiki, можем судить, что есть в наличии либо планируется:

- ✓ Проект состоит из последовательностей (sequences), которые в свою оче-



Внешний вид Lumiera



редь состоят из дорожек. Такой подход используется в Premiere Pro и позволяет вносить логическую ясность в монтируемый материал. Между последовательностями можно запросто переключаться при помощи списка.

- ☑ Есть групповые дорожки, способные хранить в себе дорожки с футажом, что также способствует удобному логическому представлению данных.
- ☑ Панель Viewer, наверное, заменила собой Compositor из Cinelerra (Viewer в самой Cinelerra служит для просмотра футаж, а не данных с монтажного стола), поскольку его (Compositor) следов уже не видно.
- ☑ Будут ли в списке с футажом виртуальные каталоги – непонятно (в списке пусто). Панели Resources, Viewer и Timeline можно отрывать, превращая в окна, тогда Lumiera станет похожа на многооконную Cinelerra.

Насколько воодушевляет этот скриншот? Мы видим только графический интерфейс. И никаких следов, что к нему подключены монтажный движок и бэкэнд ввода/вывода. набросок интер-

фейса оставляет впечатление умеренности, кивка в сторону существующих монтажных программ, удобства и привычности, однако и не отходит далеко от Cinelerra – любопытное и сильное решение. Что будет дальше с Lumiera? Поживем – увидим.

### Будущее Cinelerra

А что же с самой Cinelerra? Возможно, с выходом Lumiera угаснет CV-версия Cinelerra и останется лишь HV-вариант, который выпускается довольно редко. Если поправить поддержку форматов и хотя бы добавить предварительный просмотр импортируемого футаж – у программы долгая, счастливая жизнь, потому что Cinelerra, как ни крути, очень «сильна». Ну а если оставить всё как есть – будет некое практическое применение в идеальной для Cinelerra программно-аппаратной среде либо использование на уровне «попробовал» в других случаях.

С одной стороны, разработчик честно предупреждает: Cinelerra заточена для 64-битных систем, в остальных может быть нестабильна. С другой стороны, каким образом остановка воспроизведе-

ния на 32-битной системе может привести к зависанию программы? Что не так? Это же простейшее действие – поставить на воспроизведение монтируемый фильм. А Cinelerra может при этом повиснуть, когда остановишь воспроизведение. Есть, конечно, и третья сторона: почему бы мне самому, поскольку я более-менее разбираюсь в программировании, не найти в исходнике причину ошибки и не попробовать её устранить? Ведь в мире свободного ПО никто ничего никому не должен: глюки программы должно принимать либо пытаться их исправить, а автор программы не может сам проверить её работу на всех возможных конфигурациях компьютеров. Разумеется, разработчики Cinelerra осведомлены об ошибках. И надо думать, что когда-то у них дойдут руки до исправления... В общем, будет приятно пользоваться стабильной Cinelerra и попробовать Lumiera, которая, как только выйдет, сразу же обрадует своих создателей новыми сообщениями об ошибках, поступающими от пользователей.

Петр Семилетов  
([tea@list.ru](mailto:tea@list.ru))

## «Open Source» приглашает к сотрудничеству!

Электронное приложение «Open Source» всегда открыто для сотрудничества с новыми авторами, с читателями и их конструктивными предложениями по улучшению издания, обогатив критикой и любыми отзывами, с компаниями, занимающимися разработкой и продвижением программного обеспечения с открытым кодом. Приветствуются все энтузиасты, желающие опубликовать у нас свои статьи. Тематика нужных материалов очевидна из предназначения приложения, то есть FOSS (Free and

Open Source Software): теория и практическое применение; исторические сведения, анализ сегодняшнего положения, прогнозы на будущее и другие аспекты, связанные с открытым ПО.

Среди наиболее интересных на данный момент общих тем можно выделить:

- ☑ общие обзоры новых и/или интересных проектов Open Source и конкретных приложений, свежих версий дистрибутивов Linux, \*BSD и других систем;
- ☑ советы и рекомендации новичкам в GNU;

- ☑ истории успеха применения/распространения ПО с открытым кодом;
- ☑ философия и идеология Free Software;
- ☑ разработка приложений с применением средств Open Source.

Желательный объем статей: 6000 или 12000 символов (с пробелами). Примеры актуальных сейчас тем для статей публикуются на <http://osa.samag.ru/todo>. Но не стоит строго ограничиваться приведенными выше рамками!

Публичное обсуждение «Open Source» проводится на форуме сайта журнала «Системный администратор» по адресу: <http://osa.samag.ru/forum>. Связаться с редакцией можно по электронной почте [osa@samag.ru](mailto:osa@samag.ru).

P.S. За статьи мы платим.

## Подписные индексы:

**20780\***

+ диск с архивом статей  
2008 года

**81655\*\***

без диска

по каталогу агентства  
«Роспечать»

**88099\***

+ диск с архивом статей  
2008 года

**87836\*\***

без диска

по каталогу агентства  
«Пресса России»

\* Годовой  
\*\* Полугодовой  
\*\*\* Диск вкладывается  
в февральский  
номер журнала,  
распространяется только  
на территории России

# Подписка на журнал «Системный администратор»

## Российская Федерация

- ✓ Подписной индекс: годовой – **20780**,  
полугодовой – **81655**  
Каталог агентства «Роспечать»
- ✓ Подписной индекс: годовой – **88099**,  
полугодовой – **87836**  
Объединенный каталог «Пресса  
России»  
Адресный каталог «Подписка за ра-  
бочим столом»  
Адресный каталог «Библиотечный  
каталог»
- ✓ Альтернативные подписные агентства:  
агентство «Интер-Почта»  
(495) 500-00-60, курьерская доставка  
по Москве  
агентство «Вся Пресса»  
(495) 787-34-47  
агентство «Курьер-Пресссервис»  
агентство «ООО Урал-Пресс»  
(343) 375-62-74
- ✓ Подписка On-line  
<http://www.arzi.ru>  
<http://www.gazety.ru>  
<http://www.presscafe.ru>

## СНГ

В странах СНГ подписка принимается  
в почтовых отделениях по националь-  
ным каталогам или по списку номенкла-  
туры «АРЗИ»:

- ✓ **Азербайджан** – по объединенному  
каталогу российских изданий через  
предприятие по распространению пе-  
чати «Гасид» (370102, г. Баку, ул. Джа-  
вадхана, 21)

- ✓ **Казахстан** – по каталогу «Россий-  
ская пресса» через ОАО «Казпочта»  
и ЗАО «Евразия пресс»
- ✓ **Беларусь** – по каталогу изданий стран  
СНГ через РГО «Белпочта» (220050,  
г. Минск, пр-т Ф. Скорины, 10)
- ✓ **Узбекистан** – по каталогу «Davriy  
nashrlar», российские издания через  
агентство по распространению печати  
«Davriy nashrlar» (7000029, г. Ташкент,  
пл. Мустакиллик, 5/3, офис 33)
- ✓ **Армения** – по списку номенклатуры  
«АРЗИ» через ГЗАО «Армпечать»  
(375005, г. Ереван, пл. Сасунци Давида,  
д. 2) и ЗАО «Контакт-Мамул» (375002,  
г. Ереван, ул. Сарьяна, 22)
- ✓ **Грузия** – по списку номенклату-  
ры «АРЗИ» через АО «Сакпресса»  
(380019, г. Тбилиси, ул. Хошараульская,  
29) и АО «Мацне» (380060, г. Тбилиси,  
пр-т Гамсахурдия, 42)
- ✓ **Молдавия** – по каталогу через ГП «По-  
шта Молдовой» (МД-2012, г. Кишинев,  
бул. Штефан чел Маре, 134)  
по списку через ГУП «Почта Придне-  
стровья» (MD-3300, г. Тирасполь, ул.  
Ленина, 17)  
по прайс-листу через ООО агентство  
«Editil Periodice» (МД-2012, г. Киши-  
нев, бул. Штефан чел Маре, 134)
- ✓ Подписка для **Украины**:  
Киевский главпочтамт  
Подписное агентство «KSS»  
Телефон/факс (044)464-0220