

Колонка главного редактора



Более полугода прошло с того момента, как я получил свой долгожданный Neo FreeRunner. Однако меньше месяца мне потребовалось для того, чтобы вдоволь «наиграться» с устрой-

ством, установив на него несколько существовавших на то время дистрибутивов и опробовав их в действии. С тех пор Open Source-смартфон неподвижно лежал, дожидаясь того момента, когда я созрею для написания обзора как самого устройства, так и актуальных версий прошивок.

И вот ситуация повернулась неожиданной для кого-то стороной (см. новости в этом выпуске «Open Source»), но вполне логичной для действующих пользователей FreeRunner. Конечно же, сразу начались публичные «похороны» с выкрикиванием громких фраз о том, что это «очередное доказательное несостоятельности Open Source». Отнюдь не так. Все участники грандиозной инициативы Openmoko получили бесценный опыт и, несмотря ни на что, доказали миру возможность создания таких интересных и уникальных устройств, как FreeRunner.

Главная проблема коренится вовсе не в Open Source, а правильной организации рабочего процесса и грамотном (целевом) маркетинге. Что важно, даже после решения этой проблемы не получится новый iPhone, но подождите... Зачем? Ведь массовых и ориентированных на «обычных пользователей» смартфонов и без того хватает.

Главный редактор
Дмитрий Шурупов
(osa@samag.ru)

«Open Source»

электронное приложение к журналу

«Системный администратор»

№45, 11 июня 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 статьи ответственность несет автор. Все права на опубликованные материалы защищены.

Новости мира Open Source

Canonical запускает Android-приложения в Ubuntu Linux

На мероприятии Ubuntu Developer Summit в Барселоне (Испания) компания Canonical, занимающаяся коммерческой поддержкой Linux-дистрибутива Ubuntu, продемонстрировала прототип среды исполнения приложений мобильной среды Android в Ubuntu Linux.

Цель очередной инициативы компании Canonical – привести на desktop-растущий мир сторонних приложений, доступных для продолжающей набирать популярность мобильной Linux/Java-платформы Android. Несмотря на то, что Android использует в своей основе Linux-ядро, все приложения работают в особом Java-окружении, так что запуск таких программ в обычной Linux-системе даже с установленной Java Runtime Environment невозможен.

В Canonical намерены добавить то недостающее звено в виде прослойки между Linux-ядром и Android-приложением, и первые результаты в виде рабочего прототипа уже были продемонстрированы на барселонском мероприятии. Разработчикам удалось скомпилировать окружение запуска Android-приложений с родной для Ubuntu версией базовой библиотеки libc и запустить в нем приложение на обычном Linux-ядре. Для того, чтобы финальная версия Android-окружения нормально функционировала, разработчики планируют включить в Linux-ядро патчи, добавляющие в него присутствующую в Android систему межпроцессного взаимодействия Binder.

Исходный код наработок, проведенных усилиями Canonical для запуска Android-приложений в среде Ubuntu Linux, будет скоро опубликован.

Стартовал проект Qt Mobility

Подразделение Qt Software, ранее известное как компания Trolltech, а ныне являющееся частью Nokia, объявило о запуске нового проекта, получившего название Qt Mobility.

Цель Qt Mobility – разработать для Qt, популярного Open Source-фреймворка для создания графического пользовательского интерфейса, новые интерфейсы прикладного программирования (API), которые обеспечат функции, необходимые для мобильного применения.

Сейчас в Nokia заняты созданием «мобильных» API для многих функциональных. В первую очередь будут созданы та-

кие базовые вещи, как Service Framework и Bearer Management API, а затем – контакты (Contacts API), обмен сообщениями (Messaging) и другие средства, которые представляют интерес для конечного пользователя. К слову, в публичном репозитории Qt уже появились первые разработки Qt Mobility – Service Framework.

SourceForge приобретает онлайн-платформу Ohloh

Компания SourceForge, известная своими веб-ресурсами для онлайн-сообществ и электронной коммерции, объявила о достижении договоренности по покупке Ohloh.

Ohloh – созданный бывшими менеджерами корпорации Microsoft сайт, предоставляющий веб-сервисы и онлайн-платформу, предназначенную для разработки программного обеспечения с открытым кодом усилиями сообщества. Благодаря автоматизированному сбору данных из различных репозиторий (CVS, Subversion, Git) на Ohloh публикуется статистика о различных Open Source-проектах, их лицензиях и т.п.

SourceForge с покупкой Ohloh намерена интегрировать возможности этой платформы в свои ресурсы, улучшив таким образом онлайн-среду для Open Source-разработок. Команда разработчиков Ohloh присоединится к штату сотрудников SourceForge и продолжит развитие своих технологий.

Финансовые подробности сделки SourceForge и Ohloh не разглашаются.

Вышел офисный пакет KOffice 2.0.0

После более трех лет разработок команда KOffice объявила о выпуске финальной версии крупного обновления к офисному пакету для KDE – KOffice 2.0.0.

Главным новшеством в KOffice 2.0.0 стало завершение работ по портированию этого офисного набора на библиотеки фреймворка Qt 4 и графической среды KDE 4. Несмотря на это, разработчики не позиционируют релиз как решение для конечных пользователей и не рекомендуют включать его в состав Linux-дистрибутивов в качестве офиса по умолчанию. Причины довольно просты: несмотря на то, что это финальный релиз KOffice 2.0.0, во-первых, пока еще он может содержать заметные проблемы, а во-вторых, в нем еще не реализованы все те возможности, которые были в KOffice 1.6.0. «Полноценным» и готовым к массовому

использованию продуктом он станет к релизу 2.1 или 2.2.

Заявляется, что KOffice 2 стал намного более гибким решением, чем когда-либо был KOffice 1. Одним из революционных новшеств данного релиза называются Flake Shapes («наслаивающиеся формы»), благодаря которым все компоненты KOffice получили возможность работать с любыми «формами». Ими могут быть как простые элементы вроде квадратов и кругов, так и более сложные – диаграммы или музыкальные рейтинги. Благодаря этому в текстовом процессоре (KWord) теперь можно вставлять растровые изображения, в растровом графическом редакторе (Krita) – векторные изображения, а в векторном графическом редакторе (Karbon) – диаграммы.

Кроме того, в KOffice 2.0.0 был переработан внешний пользовательский интерфейс, который лучше подходит для современных широких дисплеев, и появилась родная поддержка формата документов OASIS OpenDocument Format (ODF).

Появилась рекламная сеть свободного ПО

Организация свободного программного обеспечения (Free Software Foundation, FSF) выпустила пресс-релиз, в котором «поприветствовала» запуск новой рекламной сети Ad Bard Network.

Ad Bard – это рекламная сеть для технологических сайтов, предназначенная для продвижения продуктов и сервисов, связанных со свободным ПО. Ее создала компания Tag1 Consulting для тематических сайтов с целью помочь им взаимовыгодно взаимодействовать с компаниями, продающими продукты и услуги, нацеленные на аудиторию поклонников и пользователей FLOSS. Заявляется, что Ad Bard должна решить проблему обычных рекламных сетей, допускающих рекламу проприетарных продуктов на сайтах, владельцы и посетители которых этого не приемлют.

«У сообщества свободного программного обеспечения теперь есть своя альтернатива рекламным сетям, продвигающим проприетарное ПО, – заявил Питер Браун (Peter Brown) из FSF. – Это серьезная победа для многих сайтов, работающих с сообществом. Мы желаем успехов AdBard и всем сайтам, сотрудничающим с этой сетью. Также мы надеемся, что дан-

ный шаг побудит другие рекламные сети принять подобную политику».

Опенмоко больше не занимается Open Source-смартфонами

Шон Мосс-Пультц (Sean Moss-Pultz) объявил в почтовой рассылке Openmoко, что компания больше не будет заниматься Open Source-смартфонами и оставляет их будущее на усмотрение энтузиастов из сообщества.

Как стало известно из его письма, 25 мая значительная часть сотрудников Openmoко была уволена, поскольку компания свернула разработку смартфонов. Несмотря на это, многие из них продолжают трудиться над обновленной версией аппаратного обеспечения Neo FreeRunner, будучи искренними приверженцами свободного ПО. Дальнейшая судьба проекта полностью в руках сообщества, которое вправе развивать его по своему усмотрению.

Продажи актуальной версии смартфона Neo FreeRunner не прекращаются – выпущенных устройств осталось еще много, и в интересах Openmoко Inc продать и их. Сама же компания намерена заняться созданием «устройства принципиально другого типа» – для этого и было проведено сокращение (с целью оставить лишь тех сотрудников, которые будут задействованы в новом проекте). Подробности обещаются в ближайшие месяцы.

Вышел Linux-дистрибутив Fedora 11

9 июня было объявлено о выходе новой версии популярного бесплатного Linux-дистрибутива – Fedora 11 (под кодовым названием «Leonidas»).

Среди ключевых новшеств в Fedora 11 отмечают: автоматическая установка шрифтов, требующихся для просмотра и редактирования документов, и привязок приложений к типам данных (MIME types) с помощью PackageKit; унификация управления уровнем громкости звука через PulseAudio; улучшения в поддержке KMS (kernel modsetting) для видеокарт ATI, Intel и NVIDIA; включение отпечатков пальца в процедуру создания пользовательских аккаунтов, интеграция такой (fingerprint) аутентификации во всю систему; расширение Presto к пакетному менеджеру yum, позволяющее при обновлениях па-

кетов скачивать лишь необходимый минимум изменившихся файлов, а не пакеты целиком.

Из других интересных изменений, представленных в релизе Fedora Linux 11, можно выделить переход на ext4 в качестве файловой системы по умолчанию, включение в состав дистрибутива пакета MinGW для кросскомпиляции программ для возможности их запуска в Windows, уменьшение времени загрузки системы до 20 секунд, интеграция PolicyKit с демоном печати CUPS, обновление RPM до версии 4.7, использование TigerVNC в качестве VNC-клиента по умолчанию.

Среди пакетов программного обеспечения в составе дистрибутива Fedora 11: Linux-ядро 2.6.29.4, X-сервер Xorg 1.6, графические среды KDE 4.2, GNOME 2.26, Xfce 4.6, веб-браузер Firefox 3.5, набор компиляторов GCC 4.4, язык программирования Python 2.6, интегрированная среда разработки NetBeans 6.5.

Российские FLOSS-проекты удостоились наград

Целый ряд Open Source-разработок российских авторов получили награды на прошедшем в июне конкурсе Les Trophées du Libre 2009.

Les Trophées du Libre – один из крупнейших в мире Open Source-конкурсов, основная цель которого – популяризировать инновационные проекты программного обеспечения, распространяемого по Open Source-лицензиям. Мероприятие организовывается с 2003 года французскими энтузиастами, чем и объясняется соответствующее название.

В завершившемся 6 июня этого года, уже пятом по счету, конкурсе в качестве жюри выступали около 30 IT-экспертов, журналистов и менеджеров. Среди финалистов этого года оказались 1 украинский и 3 российских проекта: UniConvertor занял первое место в категории Media, MathGL победил в категории Sciences, Inquisitor стал третьим в категории Security, а GSQL оказался на 4-й строчке рубрики Professional. Полный список финалистов доступен на http://www.trophees-du-libre.org/index.php?option=com_content&task=blogcategory&id=24&Itemid=69.

Дмитрий Шурупов,
по материалам www.nixp.ru
(osa@samag.ru)

Обзор аудиопроигрывателя «Неформал»

Введение: коллекции и плейлисты

В мультимедийных проигрывателях нынче распространена мода на коллекции. Причем везде: от компьютера – до мобильного телефона. А ведь по сути коллекция музыки в виде базы данных мало чем отличается от представления музыки файловой системой – разве что потребуются отбросить поиск по жанрам и тому подобные вещи. Так ли удобны коллекции, что их стоит повсеместно использовать? Обычно коллекция подразумевает наличие тегов в файлах. Нет тегов – файлы представлены где-то на задворках коллекции, в виде безымянного мусора. Теги в другой кодировке – то же самое. Да, мне возразят, что не так уж сложно перевести все теги в UTF-8, но лично у меня, например, руки не всегда доходят...

Другое повторение существующего – списки воспроизведения или, как говорят, «плейлисты». Как правило, файлы уже лежат в каталоге в определенном порядке. Зачем создавать еще какой-то виртуальный список воспроизведения? Чтобы переставить песни местами. Вы часто этим занимаетесь? Многие люди – и я в том числе – загружают альбом на воспроизведение дедовским способом: перетаскиванием из файлового менеджера в окно проигрывателя. Почему? Потому что, если в проигрывателе есть какое-то средство выбора файлов, то оно зачастую далеко от удобства для перемещения по файловой системе.

«Неформальный» ответ

Исходя из перечисленных выше соображений и родился аудиопроигрыватель под названием «Неформал»/Neformal (<http://neformal.sourceforge.net>), графический интерфейс которого создан с помощью Qt. Суть работы программы проста: у вас есть файловый менеджер, именуемый в руководстве «файловым приказчиком». Вы дважды щелкаете по звуковому файлу (или нажимаете <Enter>), «Неформал» воспроизводит его, а затем автоматически переходит к сле-

дующему звуковому файлу – и так до конца каталога. В результате, основное рабочее пространство «Неформала» занимает этот самый файловый приказчик. По ощущению он подобен Krusader или Total Commander – работает даже Backspace для подъема на каталог уровнем выше. Файлы можно удалять, переименовывать, перетаскивать во внешние программы. Справа от панели файлов есть список с закладками. Любой каталог можно добавить в эти закладки, чтобы потом быстро перейти в него.

Теги песен отображаются внизу главного окна. Кодировку можно выбрать в меню «Вид → Кодировка». Там же находится еще один пункт меню – «Предыдущая кодировка». Его удобно использовать, чтобы переключать по кругу две кодировки – например, UTF-8 и Windows-1251. Выбранная кодировка влияет и на отображение текстовых файлов: «Неформал» умеет их открывать и показывать. Это сделано для возможности просмотра файлов с текстами песен и прочими сведениями. Текст текущей песни может быть получен также из Сети – для этого предусмотрен пункт меню «Функции → Текст песни». Тексты качаются со свободного ресурса lyricwiki.org. Русских там мало (кстати, хороший повод добавить!), зато англо-

язычных очень много, причем и редких песен в том числе.

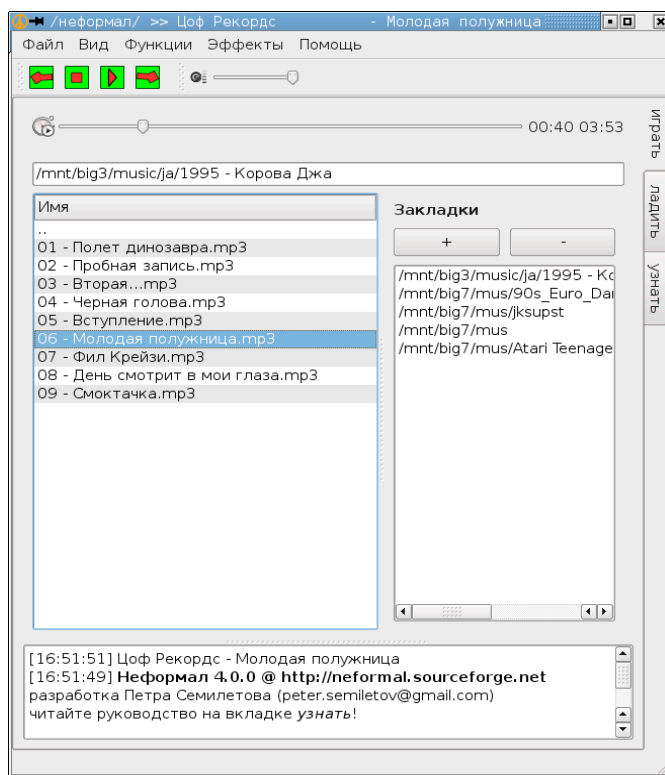
Кроме текстов «Неформал» способен показывать картинки во встроенном просмотрщике. Полезно для обложек, фотографий и тому подобного. В «смотрелке» действуют клавиши <PageUp/Down> (вперед/назад по каталогу), <Home/End> (первая/последняя картинка), <[-> и <]> (переворот, без изменения данных файла). В последней версии (релиз 4.1.0 вышел 2 июня. – Прим. ред.) она научилась и масштабировать изображения.

Возвращаясь к теме тегов, отмечу, что в «Неформале» используется очень гибкий поход к их чтению. Дело в том, что некоторые MP3-файлы хранят в себе теги сразу двух форматов: id3v1 и id3v2. Сведения в этих тегах иногда повторяются (причем в id3v1 они короче из-за ограничений формата). Иногда бывает, что id3v2 есть, но пустой или заполнен мусором, а иногда – все наоборот. Чтобы пользователь мог получить доступ к тегам обоих форматов, в «Неформале» есть переключатель «Предпочитать id3v1 перед id3v2» – находится он на вкладке главного окна с надписью «Ладить». Если включить эту опцию, «Неформал» будет сначала пытаться прочесть тег формата id3v1. Если тег есть и прочитан, «Неформал» покажет данные из него, а в противном случае «Неформал» будет читать данные из id3v2. Если же галочка на переключателе не стоит, то «Неформал», напротив, сначала читает данные из id3v2, а затем, в случае такой необходимости, из id3v1. Увидеть одновременно сведения из тегов обеих форматов можно с помощью пункта меню «Функции → Сведения о файле».

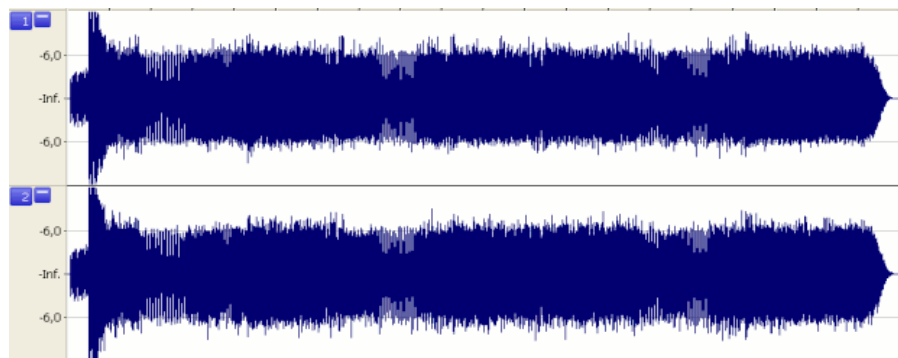
Главное окно «Неформала» состоит из трех вкладок:

- ☒ Играть (там, где файловый приказчик).
- ☒ Ладить (настройки программы).
- ☒ Узнать (руководство).

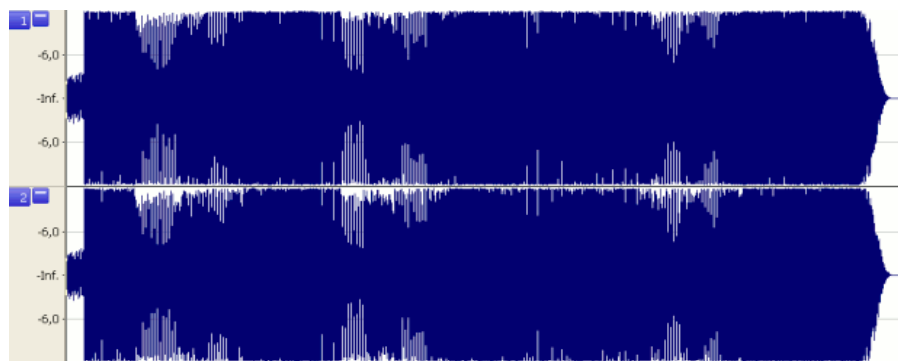
Есть также две инструментальные панели: одна с кнопками управления, другая – с регулятором громкости. Панели можно перемещать и парковать по сторонам окна. Иконки на кнопках внешне некрасивы, зато векторные. Громкость советую менять не программно, а аппаратно (регулятором на колонках). Впрочем, это касается не только «Неформала», а любых проигрывателей, поскольку любое цифро-



Общий вид «Неформала»



Волновая форма «Smells Like Teen Spirit» 1991 года



Волновая форма «Smells Like Teen Spirit» 2002 года

вое изменение громкости приводит к исчезновению (в той или иной мере) тихой части звукового сигнала (аналоговое изменение громкости в этом плане больше «щадит» звук).

В меню «Функции → Сведения о файле» кроется большая сила. Если в файле есть метаданные, «Неформал» показывает все, которые способен прочесть. Для WAV и MP3 «Неформал» задействует собственные функции чтения таких данных, а за остальные форматы отвечает используемый движок (об этом – ниже). Если файл в формате WAV, PCM 16 bit, то подсчитывается его RMS. Не вдаваясь в подробности, скажу, что по RMS можно судить об общей громкости песни. Чем ближе RMS к нулю (в цифровом звуке отсчет идет ниже нуля), тем громче песня. Например, в начале 90-х рок-музыку старались сводить так, чтобы RMS был на уровне примерно -18 дБ. Сейчас планка поднялась – не редкость и -10 дБ. Исчезает динамика, музыка становится чересчур плотной и нагружающей слух. Функция вычисления RMS привычна в волновых редакторах под Windows, а в Linux её по загадочной причине обходят стороной, хотя звукоинженерам и диджеям она необходима.

Волновые формы можно увидеть на изображениях: в оригинальной песне Nirvana «Smells Like Teen Spirit» (1991 год) RMS равен -18 дБ. А в ее ремастеренной редакции, выпущенной на сборнике Nirvana в 2002 году (тот диск, где была опубликована «You Know You're Right»), RMS уже около -12 дБ.

Громче? Да, но и менее внятное звучание – смазанный, тархтящий звук. Послушайте какие-нибудь старые альбомы и новые. В старых – невысокий RMS, хорошая динамика, поэтому уши не устают. После этого попробуйте прослушать подряд несколько современных альбомов. Речь, впрочем, не идет о джазе или симфонической музыке – её продолжают сводить с низким RMS.

Использование Phonon

В качестве звукового движка «Неформал» использует Phonon. Phonon – это, грубо говоря, не отдельный движок, а программная прослойка между настоящим движком (вроде Xine, Gstreamer, Direct Show) и программой. Он также служит движком для других проигрывателей (например, Amarok 2).

Поддержка форматов зависит от «настоящего движка, который в обиходе называется бэкэндом. Например, в Mandriva по умолчанию установлен backend, основанный на GStreamer, а в Ubuntu – на Xine. И поддержка форматов зависит от установленных к GStreamer или Xine плагинов.

В случае же Direct Show под Windows – от фильтров-кодеков. Допустим, для поддержки Ogg Vorbis и FLAC надо скачать кодек с <http://xiph.org/dshow> и установить его.

В отличие от некоторых Phonon-проигрывателей, которые воспроизводят звук на устройство, заданное в свойствах Phonon в настройках KDE, «Неформал» позволяет выбрать такое устройство

прямо на вкладке «Ладить», в списке «Выводить звук на устройство».

Если ваша звуковая карта способна аппаратно воспроизводить звук с частотой оцифровки 44.1 кГц, обратитесь к руководству «Неформала» за сведениями, как добиться вывода звука именно с такой частотой. Ведь по умолчанию звуковая подсистема ALSA переоцифровывает все звуковые данные в частоту 48 кГц, что искажает звучание тех записей, которые оцифрованы в 44.1, а это почти все MP3-файлы и все аудиодиски.

Бытует мнение, что если проигрыватель сделан на движке Phonon, то к нему нельзя приделать эквалайзер. Это мнение ошибочно. Phonon поддерживает эффекты обработки звука. Другое дело, что набор эффектов на разных бэкэндах различается. Так, в GStreamer есть отличный эквалайзер на 10 полос, а вот в Xine эквалайзера нет, зато присутствуют другие эффекты.

На вкладке «Ладить» вы можете добавлять доступные эффекты в цепочку обработки звука. Цепочку можно сохранить под именем – тогда она появится в виде предварительной настройки (пресета) в меню «Эффекты». Подробно это описано в документации. Отмечу, что для редактирования параметров эффектов используется виджет от самого Phonon, и все параметры представлены как подписанные числовые значения. Это далеко не красивые интерфейсы VST-плагинов, зато всё работает. В будущем, вероятно, в «Неформале» будет свой виджет редактирования параметров – более эргономичный. Поддержка эффектов для Win32-сборки отключена по причине сырости бэкэнда.

Вместо заключения

Исходник «Неформала» весит около 50 Кб и распространяется под лицензией GNU GPLv3. Для его сборки требуется одна лишь Qt и Phonon (если Qt не собрана с Phonon). На сайте можно скачать также готовую сборку под Windows, а еще бинарные файлы для Mandriva и Ubuntu. Оформлять эти бинарники в пакеты у разработчика нет ни времени, ни умения.

В планах на будущее – возможно, взаимодействие с сетевыми сервисами вроде Last.fm и Jamendo.

Петр Семилетов
(tea@list.ru)

Обзор Linux-игры Caster

В предыдущих выпусках «Open Source» (№№41 и 42) я рассказывал о родных играх для операционной системы GNU/Linux: Yo Frankie!, Savage 2 и World of Goo. Помимо того, что все они работают в Linux как «родные» приложения, у них есть и другая общая особенность – использование свободных компонентов и библиотек. Эта статья не станет исключением и осветит еще один игровой проект, доступный под GNU/Linux и использующий открытые и свободные библиотеки. Встречайте – Caster.

Caster

12 мая этого года в блоге разработчика игры Майка Смита (Mike D. Smith) появилась небольшая запись (<http://elecorn.com/blog/2009/05/caster-released-on-linux>) примерно следующего содержания: «Frank Earl только что закончил сборку и упаковку Caster для Linux! Это особенно интересно тем, что Caster изначально разрабатывался под Linux, а позже был перенесен на Windows, Mac и iPhone. Наслаждайтесь, пингины!». Среди задействованных в игре компонентов – SDL (Simple DirectMedia Layer), GLEW (OpenGL Extension Wrangler), libvorbis и libvorbisfile, libogg.

Демо-версия игры доступна бесплатно на сайте (<http://elecorn.com/caster3d>) – там можно скачать ее версию под любую из поддерживаемых платформ (Linux, Windows, Mac, iPod). На том же сайте представлена и платная версия. Стоимость одной копии под одну платформу составляет чуть менее 5 долларов США, что на момент покупки обошлось мне в 164 рубля. К слову о платной версии: не могу однозначно сказать, стоит ли ее покупать. С одной стороны, игра довольно интересна, а цена – небольшая, но с дру-

гой – полная версия весьма коротка (демонстрационный вариант содержит почти половину уровней). Дополнительным бонусом для покупки является тот факт, что разработчик обещает бесплатное обновление с продолжением истории.

Установка и требования

Запуск бинарного Linux-файла игры (caster-linux.bin) открывает графическую утилиту установки в стиле Windows: нужно прочесть лицензионное соглашение (и согласиться с ним), выбрать каталог установки и проинсталлировать игру. В результате создается .desktop-файл, после чего игру можно запустить из меню «Игры» (если используется GNOME или KDE).

Несмотря на то, что игра собрана под архитектуру x86, запуск в 64-битных системах возможен – для этого необходимо лишь установить соответствующие библиотеки (речь о ia32-libs. – Прим. ред.). Требования для игры не являются заоблачными: процессор с тактовой частотой от 1 ГГц, оперативная память – от 64 Мб, поддержка видеодрайвером 3D-ускорения и OpenGL 1.1. Сам я ее тестировал на Dell Vostro 500, и заметных проблем не возникло – за исключением того, что иногда, вне зависимости от настроек, игра на некоторое время начинала тормозить (с чем это связано, мне выяснить так и не удалось).

Геймплей

Управление в игре тривиально: стандартные клавиши движения (WASD), «пробел» для прыжка, <shift> для ускорения, <tab> для того, чтобы убрать все панели с экрана и <F5> для снятия снимка экрана. Мышью управляется вид, переключение оружия и атака.

В игре нужно отбиваться от самых различных монстров, которые в общем назы-

ваются Flanx. Они заселились в этих местах и отравили землю, из-за чего погибли все деревья и зелень в округе. От этого страдают местные жители, которые призвали на помощь одного из Caster Clan. И вот наш подопечный, юный воин клана, должен очистить весь регион от разнообразных созданий.

Жанр игры определен как шутер от третьего лица, но это не тот шутер, которыми в последнее время балуют крупные издатели. Этот шутер скорее похож на аркаду, и действительно – все очень просто и невероятно увлекает. Управляя главным героем, можно выбирать между шестью типами оружия, которые открываются по мере прохождения игры. Первое и основное оружие – импульсное, – пожалуй, самое эффективное и универсальное, а также самое быстрое. Два других созданы для модификации ландшафта: с ними можно поднимать или опускать местность по своему усмотрению. Впрочем, после того как другое оружие будет максимально «развито» (об этом см. ниже), им тоже можно воздействовать на ландшафт. Еще одно оружие создает барьер вокруг персонажа, при котором снаряды летают по орбите и при столкновении с врагом поражают его. И кроме того, ученик из клана Caster имеет самонаводящееся оружие и оружие, которое временно замедляет Flanx. Некоторое оружие наносит больший урон при удерживании кнопки атаки на мыши.

В игре также имеется RPG-составляющая, а это значит, что зарабатываемые в миссиях очки можно использовать в магазине между миссиями для усовершенствования навыков и оружия. Помимо того, что можно увеличивать мощь имеющегося оружия, можно развивать щит, который напрямую влияет на жизнеспособность нашего персонажа, и радиус сбора (расстояние, на котором примагничиваются «души», увеличивающие количество очков).



И главное, без чего Caster не был бы таким интересным — это «стремительное перемещение» и «суперпрыжок». Если с прыжком все просто и понятно: можно выпрыгнуть из зоны поражения врагами, перепрыгнуть препятствие и так далее, то с перемещением все интереснее. Оно позволяет убежать от врагов на безопасное расстояние, либо бегать с большой скоростью вокруг неподвижного Flanx, но избегая его атак. После того как навык будет достаточно развит, мож-



но бегать по поверхности воды или лавы и по отвесной стене, что заметно раздвигает «границы» игры. Так, например, в одной из миссий есть бронированный тип Flanx, которого «не берет» оружие. Таких врагов можно заманить в лаву, просто бегая по её поверхности — AI соперников достаточно прямолинеен. После таких «пробежек» вся остальная игра кажется очень медленной.

Для прохождения миссии дается несколько попыток (в зависимости от сложности игры), после чего нужно будет начинать уровень заново. Если миссия пройде-

на без повреждений, то в качестве бонуса добавляются очки. Уровни сами по себе небольшие: их можно не спеша обойти в поисках сфер энергии, которые принесут дополнительные очки, а может, и новых врагов. Цели игры просты: помимо тотального освобождения земель от Flanx, от миссии к миссии нужно будет оживать деревья (их, кстати, можно оживать в любой миссии, просто выстрелив в них — после этого выпадает энергетическая сфера на 10 очков), собирать энергию и патрулировать окрестности. Уровни отличаются друг от друга: в одних играешь как

в узком коридоре, в других — приходится плавать, в третьих — бегать по островам среди лавы, а в четвертых — крошечная тьма. Все это достаточно интересно, а когда игра заканчивается, не хочется останавливаться.

Заключение

Игра доставляет много приятных впечатлений: это и скорость, и затягивающий аркадный геймплей, и желание пройти игру по максимуму, и эпически короткая битва с боссом в конце. В то же время обеща-

ние бесплатного продолжения от разработчика не дает покоя и заставляет вновь и вновь посещать его блог. Судя по записи от 3 июня, в которой Майк анонсирует звуковую дорожку из второй части, работа над ней не заброшена, и в скором времени она увидит свет. Стоимость игры приблизительно равна стоимости регионального DVD-издания, а сама она может подарить несколько часов приятного времяпрепровождения.

Никита Лялин
(tinman321@gmail.com)

Вопросы и ответы #002

На вопросы читателей отвечают специалисты украинской компании «Линукс Саппорт» (<http://lsupport.net>). Мы ждем ваших новых вопросов по e-mail: osa-question@samag.ru!

Как установить несколько программных пакетов в Debian/Ubuntu?

Бывают такие ситуации, когда в репозитории используемого дистрибутива нужная вам программа имеет старую версию, либо версию, которая содержит ошибки, из-за чего отдельные функции программы выполняет неверно. Если вы уверены в том, что программа настроена правильно и это действительно ошибка, первое, что стоит сделать — заглянуть на ее официальный сайт и проверить наличие новой версии. Часто для работы программы требуется устанавливать дополнительные компоненты (deb-пакеты). Если программа состоит из одного пакета, ее можно установить привычным «щелчком мыши», а вот для установки нескольких придется воспользоваться терминалом. Утилита Gdebi, которая используется по умолчанию в дистрибутиве Debian для установ-

ки deb-пакетов, не позволяет одновременно выбрать и устанавливать более одного deb-пакета.

Чаще всего достаточно просто скопировать все необходимые deb-пакеты в один каталог (к примеру, /home/user) и выполнить несколько команд.

1. Перейти в каталог, в котором находятся пакеты:

```
cd /home/user/deb
```

2. Установить все пакеты из текущего каталога (потребуется права администратора):

```
sudo dpkg -i *.deb
```

Как добавить раскладку клавиатуры и настроить переключение раскладок в KDE4?

Настройки переключения раскладок в KDE4 можно выполнить в центре управления KDE (Системные параметры) — «Страна/регион и язык → Раскладка клавиатуры».

Как поменять способ переключения раскладки клавиатуры не перезагружая X.org?

Для изменения раскладок клавиатуры «на лету» вы можете использовать команду setxkbmap с нужными вам параметрами.

К примеру, для того, чтобы использовать три раскладки клавиатуры (английская, русская, украинская), можно использовать следующую команду:

```
setxkbmap -model pc104 -layout us,ua,ru -j  
-variant ,winkeys,winkeys
```

Какие фреймворчные оконные менеджеры есть в Linux?

- ✓ awesome (<http://awesome.naquadah.org>);
- ✓ dwm (<http://dwm.suckless.org>);
- ✓ ion (<http://modeemi.fi/~tuomov/ion>);
- ✓ xmonad (<http://xmonad.org>).

X-сервер (X.org) не перезагружается при нажатии комбинации клавиш Ctrl+Alt+Backspace. Что делать?

Скорее всего, комбинация Ctrl+Alt+Backspace отключена в конфигурационном файле xorg.conf. Чтобы это исправить, откройте его на редактирование:

```
sudo nano /etc/X11/xorg.conf
```


Найдите следующую секцию:

```
Section "ServerFlags"
    Option      "DontZap"      "false"
EndSection
```

Напишите true вместо false и перезагрузите X-сервер (или компьютер, если вам так привычнее).

Как я могу обратиться за помощью и советом в список почтовой рассылки, конференцию jabber, к товарищу в irc, если у меня не загружается X.org (GNOME, KDE) (на примере Ubuntu)?

Такие ситуации часто бывают после неудачного обновления или неправильной настройки графического сервера X.org. Вы можете использовать следующие программы, непосредственно в консольном режиме:

- ✓ **IRC** – irssi (<http://irssi.org>), weechat (<http://weechat.flashtux.org>).
- ✓ **ICQ** – finch (консольная часть pidgin, <http://www.pidgin.im>), centerim (<http://www.centerim.org>).
- ✓ **Jabber** – finch, mcabber (<http://mcabber.com>).
- ✓ **E-mail** – alpine (<http://www.washington.edu/alpine>), mutt (<http://www.mutt.org>).

Как настроить консольный почтовый клиент alpine для работы с Gmail?

Настройка почтового клиента Alpine – не сложное занятие, т.к. вся конфигурация производится непосредственно в самой программе. Проведем настройку alpine для работы с Gmail по протоколу IMAP для пользователя username@gmail.com. Сразу стоит оговориться, что IMAP должен быть включен в настройках аккаунта Gmail.

Для этого выполните следующие действия:

- ✓ Войдите в свой аккаунт Gmail.
- ✓ Нажмите на ссылку «Настройки» в верхней части любой страницы Gmail.
- ✓ Перейдите на вкладку «Пересылка и POP/IMAP».
- ✓ Выберите команду «Включить IMAP».

Запустите в консоли (терминале) почтовый клиент командой alpine. В меню Setup → Config заполняем поля:

```
Personal Name = username@gmail.com
User Domain = gmail.com
SMTP Server (for sending) =
smtp.gmail.com:587/tls/novalidate-cert
Inbox Path = imap.gmail.com/ssl/
novalidate-cert/user=username@gmail.com
```

На этом первичная настройка почтового клиента alpine завершена, программа готова к работе.

Обзор проектов свободного «железа»

*А кое-кто поверил второпях –
Поверил без оглядки, бестолково.
Но разве это жизнь – когда в цепях?
Но разве это выбор – если скован?*

В. Высоцкий. «Приговоренные к жизни»

Свободное и открытое аппаратное обеспечение

Часто, говоря о свободе пользователей или организаций в области информационных технологий, подразумевают свободное программное обеспечение. Но, как известно, информационные технологии основываются не только на программном обеспечении (software), но и на аппаратных составляющих (hardware). Получается, что вопрос о свободе «железа» остается в тени.

Для начала необходимо определить, что же понимается под свободным «железом». Итак, свободное «железо» (Free Hardware) – это аппаратный компонент или их упорядоченное множество, объединенное определенными принципами функционирования и взаимодействия, который(е) защищен свободными лицензиями, а также программируются с использованием свободного программного обеспечения. Открытое «железо» (Open Hardware) – это аппаратный компонент или их упорядоченное множество, объединенное определенными принципами функционирования и взаимодействия, которые необязательно защищены свободными лицензиями и могут программироваться с использованием несвобод-

ного программного обеспечения. Из этих определений следует важный вывод: если аппаратные компоненты закрыты или программируются с помощью закрытого ПО, то они не могут называться свободными, но могут – открытыми.

Стоит отметить, что в действительности с терминологией в этой области все довольно сложно. Устоявшихся определений Open и Free Hardware на данный момент не существует. При этом на Wikipedia понятие открытого аппаратного обеспечения весьма поверхностно (хотя и идеологически корректно) определяется как «аппаратное обеспечение, разработанное в том же стиле, что и свободное и открытое ПО». Там же разница между «открытым» и «свободным» в контексте аппаратного обеспечения игнорируется и эти понятия считаются идентичными. – **Прим. ред.**

Если обратиться к истории, то важно вспомнить, что в 1999 году Дамьян Ламперт (Damjan Lampert) основал проект OpenCores (<http://www.opencores.org/?do=faq>) как основу для формирования открытого сообщества для развития и распространения IP core на базе VHDL/Verilog. IP core – это основной логический блок или блок данных, используемый при создании интегральных схем: FPGA (Field-Programmable Gate Array, программируемая логическая интегральная схема) или ASIC (Application Specific Integrated Circuit, интегральная схема, ориентированная на решение конкретной задачи). Проект OpenCores существует и по сей

день (<http://www.opencores.org>), и в его рамках создано большое количество различных продуктов, например, в таких категориях, как коммуникационные контроллеры (10Gb Ethernet MAC, FireWire), сопроцессоры (FPU, Floating Point Unit), криптографические базовые блоки (128/192 AES, SHA cores), процессоры (1-bit Microprocessor, AVR Core, MicroRISC II и множество других).

Сегодня практически не существует единого сообщества в области свободного/открытого «железа». Когда появляются различные проекты, вокруг них образуется сообщество разработчиков. Но надежду на исправление ситуации дает появившаяся концепция (<http://antipastohw.blogspot.com/2009/03/introducing-open-source-hardware.html>), предполагающая создание Центрального банка открытого «железа» (Open Source Hardware Central Bank). Основная идея заключается в том, чтобы форсировать решение проблемы вывода открытого устройства в производство. Например, если у разработчика устройства есть определенное количество покупателей, готовых купить продукт, банк выделит необходимые средства. Такая модель позволит уменьшить итоговую цену устройства, а самое главное – даст возможность существования перспективным открытым проектам. Немаловажная деталь – управление банком должно быть открытым, т.е. финансовая политика и руководство определяются сообществом.

В прошлом выпуске «Open Source» (№044 от 22.05.2009) была рассмотрена платформа Arduino, которая имеет открытый аппаратный дизайн и свободную среду разработки. В этот раз я приведу краткий обзор других подобных проектов.

Единственное – сразу отметить, что рассматриваемые ниже проекты не имеют полностью свободных аппаратных компонентов, но развиваются, как правило, с использованием свободного программного обеспечения и имеют открытый аппаратный дизайн.

Свободный BIOS – Coreboot

Coreboot – это проект по созданию свободной прошивки (микропрограммы) BIOS, ранее известный как LinuxBIOS. Он был начат в 1999 году в национальной лаборатории Лос-Аламоса. Сегодня разработка свободного BIOS поддерживается Организацией свободного программного обеспечения (Free Software Foundation) и входит в перечень его приоритетных проектов (<http://www.fsf.org/campaigns/priority.html>). Все разработки проекта coreboot доступны под лицензией GNU GPL. Сайт проекта – http://www.coreboot.org/Welcome_to_coreboot.

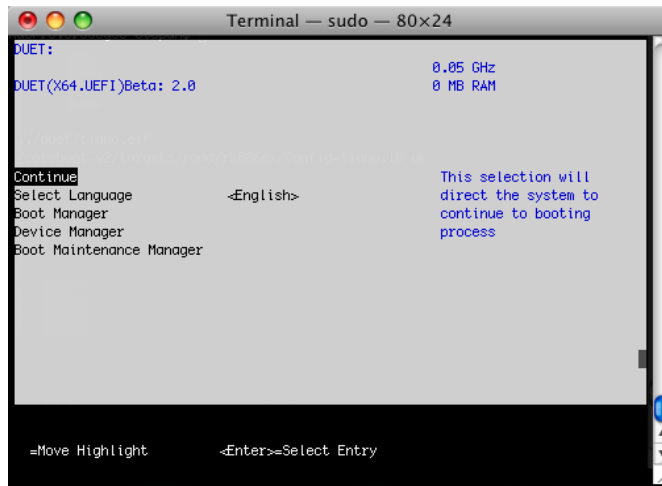
Определенным достижением разработчиков является то, что свободная микропрограмма позволяет начать загрузку операционной системы через 3 секунды после включения компьютера. Существенный недостаток – ограниченное количество поддерживаемого оборудования (на данный момент это 203 материнские платы). Более подробно о Coreboot читайте в следующем выпуске «Open Source».

Open Graphics Project

9 апреля 2009 года проект Open Graphics Project (<http://wiki.opengraphics.org/tiki-index.php>) в лице Тимоти Норманд Миллера (Timothy Normand Miller) через сайт Организации свободного программного обеспечения обратился с призывом к энтузиастам принять участие в развитии проекта (<http://www.fsf.org/blogs/community/help-the-open-graphics-project>). Первоначально в Open Graphics Project трудились над созданием свободной видеокарты – OGD1 (<http://wiki.opengraphics.org/tiki-index.php?page=OGD1>). Разработав рабочий вариант карты, сообщество решило пойти дальше и создать компьютерную систему, которая будет полностью документирована, вплоть до транзисторов. В опубликованном обращении призывались все желающие принять участие в осуществлении этих амбициозных планов.

Openmoko и Neo FreeRunner

О компании OpenMoko и ее проекте Neo FreeRunner уже публиковался матери-



Coreboot в действии (взято с официального сайта проекта)

ал в «Open Source» (№036 от 09.12.2008). За это время в проекте произошли существенные изменения. В начале июня стало известно, что работа над смартфоном Neo FreeRunner прекращена (<http://article.gmane.org/gmane.comp.handhelds.openmoko.community/43641>). 25 мая компания провела сокращения сотрудников, а 2 июня руководитель Openmoko Шон Мосс-Пульц (Sean Moss-Pultz) сообщил, что компания больше не будет сама разрабатывать Open Source-смартфоны.

В свою очередь бывшие сотрудники Openmoko уже начали реорганизацию процесса разработки Neo FreeRunner. Компания передаст им всю необходимую информацию и ресурсы, включая Wiki, Git и т.д., и будет выступать спонсором разработки в области поддержания серверной инфраструктуры, оказания корпоративной помощи там, где она необходима.

Также стоит отметить, что группа мобильного мультимедиа вещания (Mobile Multimedia Broadcasting, MMB) из Канадского исследовательского центра в области коммуникаций (Communication Research Centre Canada, CRC) разработала первое устройство, основанное на свободном смартфоне Neo FreeRunner – Openmokast (<http://www.openmokast.org>).

Основные изменения в «железе» по сравнению с продуктом Openmoko заключаются в интеграции ресивера USB DAB/DMB (приемник потокового вещания), подключенного к внутреннему USB-приемнику, питающемуся от аккумулятора. Ресивер необходим для приема различных радиосигналов (DAB audio (musicam), DAB+ audio (AAC+ v2), MOT Slideshow, Journaline и других). Произведенные новшества были лицензированы под Creative Commons Attribution-ShareAlike 3.0 License.

Свободная камера Elphel

Проект компании Elphel Inc был начат в 2001 году. Сегодня он представляет со-

бой высокопроизводительную камеру, аппаратный дизайн и программное обеспечение которой открыты под лицензией GNU GPL. Камера Elphel имеет интуитивно понятный графический интерфейс и отличается гибкими возможностями настройки. Например, Google активно использует свободные камеры в своих проектах (см. <http://www3.elphel.com>).

В устройствах используется микропроцессор Axis ETRAX, который оптимизирован для запуска GNU/Linux-

систем. Его заметная часть лицензирована под GNU GPL. Высокая скорость обработки информации достигается за счет использования гибко программируемой Xilinx FPGA, которая разрабатывается на языке описания аппаратных средств – Verilog. Более подробно свободная камера будет рассмотрена в одном из следующих выпусков «Open Source».

Свободное радио – USRP2

USRP (Universal Software Radio Peripheral, универсальная программируемая радиопериферия) – платформа, которую выпускает Ettus Research LLC (<http://www.ettus.com>). Продукты семейства USRP позволяют создавать различные устройства в области радиотехнологий (базовые станции GSM, FM-приемники и другие), аппаратная часть которых выпускается Ettus Research LLC, а программируется с использованием, например, GNU Radio. Все схемы и техническая документация открыты и находятся в свободном доступе (<http://www.ettus.com/download> и <http://gnuradio.org/trac/wiki/USRP2>).

Кроме того, на основе USRP создан программный проект openBTS, разработчиком которого является Kestrel Signal Processing (<http://www.kestrelsp.com>). OpenBTS представляет собой стек сетевых протоколов GSM с сетевым интерфейсом SIP. Стек обеспечивает GSM-интерфейс (Um), созданный в соответствии со стандартом GSM, и использует Asterisk PBX для обработки вызовов. Таким образом, стандарт openBTS предполагает использование модифицированного радиоприемника USRP и Asterisk PBX. По утверждению разработчиков, сопряжение интерфейса GSM и VoIP могло бы стать основой нового типа сотовых сетей, которые требуют гораздо меньших затрат (1/10) на развертывание и обслуживание. Как указано на официальном сайте проекта (<http://openbts.sourceforge.net>), раз-

вертывание ряда пилотных систем запланировано на осень 2009 года. Для сборки и запуска openBTS на сайте GNU Radio доступна специальная инструкция (<http://gnuradio.org/trac/wiki/OpenBTS/BuildingAndRunning>).

Заключение

В представленном материале приведена попытка определить, что же такое свободное и открытое «железо». Исходя из обзор

ра существующих проектов, можно сделать вывод, что свободное «железо» находится на начальном этапе своего развития, а рассмотренные проекты представляют множество открытого «железа». Многие из них уже достигли высокого уровня развития, а некоторые (например, openBTS) имеют потенциал, который только начинает реализовываться.

Напоследок отмечу, что проекты, построенные на отличных от проприетар-

ных принципах, подвержены воздействию ряда обстоятельств, которые могут негативно сказаться на существовании самой разработки. Как, например, произошло со свободным смартфоном Neo FreeRunner. Надеюсь, сообщество покажет себя с лучшей стороны в сложившейся ситуации.

Игорь Штомпель
(keepercoder@gmail.com)

Беседы о Qt. 5: Phonon

Пролог

Однажды я шел по рынку и увидел людей, похожих на индусов. Они продавали «лекарство от всего». Не знаю точно, что это было за снадобье, но полагаю, что Phonon на него частично похож.

Введение

Итак, Phonon — это мультимедийный движок, который пришел в Qt на смену QSound. QSound вообще использовался мало, потому что в UNIX-подобных системах он работает в редких условиях (должен быть запущен звуковой сервер NAS), а под Windows... Когда на Qt стали много программировать в Windows, тогда и появился Phonon. Сейчас QSound — это такой рудимент, сохраняемый скорее для совместимости. Вдруг какая-то программа его использует? Не нарушать же ее работу. И напоследок не умолчу о том, что QSound под Windows умеет воспроизводить только формат WAV.

Многие помнят, с каким шумом в Qt появился Phonon. Вообще говоря, корни его не в Qt, а в KDE. В Qt он попадает извне, как и WebKit. Но замечен Phonon стал во многом благодаря Qt. Слышались возгласы: вот, всего пара строк кода, и ваши мультимедийные файлы будут воспроизводиться на любой платформе! Лекарство от всего. Но одно дело — KDE, а другое — несколько более портируемая Qt. Любая универсальность имеет свои границы.

Задумка у Phonon превосходная! Существуют разнородные, грубо говоря, мультимедийные движки, которые предоставляют различные средства по воспроизведению и — некоторые — конвертированию и записи звуковых и видеоданных. Это GStreamer, Xine, а в Windows — Direct Show. Идея, которая лежит в основе Phonon, такова: мы получаем единый API, который через прослойку-бэкэнд работает уже с физическим движком. Допустим, есть бэкэнды для GStreamer, Xine, Direct Show, VLC и Mplayer. Нас не должно заботить, какой бэкэнд выбран пользователем, потому что мы не работаем с ним напрямую — все взаимодействие осуществляется с универсальными классами Phonon. Такое вот лекарство от всего.

Трудности с Phonon

Я не хочу повторять документацию и приводить некие азбучные примеры по тому, как через Phonon проиграть mp3-файл. Я хочу рассказать о трудностях, с которыми вы столкнетесь, решив использовать Phonon в своей программе. Рассказать, но не решать — решать придется вам своим трудом.

Трудность первая — сборка программ, использующих Phonon, в Linux. Обычно Phonon устанавливается в системе как часть KDE. Вернее, плагины-бэкэнды идут как плагины KDE, а сам Phonon выглядит как отдельная библиотека. Однако Phonon мо-

жет быть собран и «внутри» Qt, поскольку входит в его состав. При этом, такой Phonon будет искать бэкэнды в каталоге плагинов Qt, а не в каталоге плагинов KDE4. Такое положение дел совершенно естественно, когда в системе нет KDE4, однако есть Qt. Как правило, в системе «из коробки» или «из хранилища» Qt собрана без Phonon, а Phonon устанавливается отдельно и совершенно замечательно подхватывается системой сборки при обычном наборе команд:

```
qmake
make
make install
```

Иначе же сборщику пакета или пользователю придется потрудиться, создавая символические ссылки на Phonon и его бэкэнд, чтобы «исправить» пути.

Трудность вторая — сборка бэкэнда Direct Show в Windows под MinGW/GCC. Считается, что это невозможно, поскольку GCC плохо сочетается с некоторыми заголовками из SDK, или наоборот — они с ним (зависит от того, с какой точки зрения смотреть). Пикантности положению придает тот факт, что именно собранная MinGW-версия Qt и распространяется Nokia, причем без Phonon. Умельцы научились компилировать пресловутый бэкэнд и под MinGW, краткое описание процесса можно прочесть по ссылке: http://wiki.qtcentre.org/index.php?title=Building_the_Phonon_backend_plugin_on_Windows_using_MinGW.

Однако рецепт этот тоже нельзя считать универсальным. Внимание, тут мы входим в область плясок с бубнами, горлового пения и так далее. Опишу свой опыт. Смысл хитрости сводится к тому, что нужно использовать заголовки DirectX из пакета kdewin32. Берем пакет из SVN KDE:

```
svn checkout svn://anonsvn.kde.org/home/kde/trunk/kdesupport/ \
kdewin32
```

Распаковываем его, копируем файл comdef.h из kdewin32/include/mingw в kdewin32/include/directx. Потом в исходниках Qt меняем начало файла src/plugins/phonon/ds9/ds9.pro на вот такое:

```
DESTDIR = $$QT_BUILD_TREE/plugins/phonon_backend
QT += phonon
win32:!win32-g++:!win32-contains(QT_CONFIG,opengl):LIBS += \
+= opengl32.lib
win32:!win32-g++:!win32-contains(QT_CONFIG,opengl):LIBS += \
+= gdi32.lib
!win32-g++:!win32-msvc2005:DEFINES += _CRT_SECURE_NO_WARNINGS
!win32-g++:LIBS += strmiids.lib Dmoguiids.lib uuid.lib msdmo.lib
win32-g++:contains(QT_CONFIG,opengl):LIBS += -lopengl32
win32-g++:LIBS += -lgdi32 -lstrmiids -lDmoguiids -luuid \
-lmsdmo -lole32 -loleaut32
TARGET = phonon_ds9
DEFINES += PHONON_MAKE_QT_ONLY_BACKEND
PHONON_DS9_DIR = $$QT_SOURCE_TREE/src/3rdparty/phonon/ds9
INCLUDEPATH += ПУТЬ_K_kdewin32/include/directx
```

Внизу вместо ПУТЬ_K_kdewin32 подставьте свой настоящий путь. После этого якобы должно собираться по qmake/make, но не компилируется. Заголовки вроде бы исправленные, од-

нако все равно не те. Оказывается, не хватает описания нескольких структур из файла `dvdmedia.h`, а этот файл из состава официального SDK от Microsoft, как я уже написал, с MinGW «не дружит». Как быть? К счастью, у меня оказался установлен Perl для Windows – Strawberry Perl (strawberryperl.com). И в его состав входит – кто бы мог подумать? – заточенный нарочно для GCC заголовочный файл `dvdmedia.h`. Он находится в каталоге `strawberry/c/include`. Если у вас нет желания качать Strawberry Perl, возьмите один лишь `dvdmedia.h` с <http://www.onlinedisk.ru/file/136831>. Копируем `dvdmedia.h` в подкаталог `include` каталога MinGw. После этого компилируем Phonon и его бэкэнд. Теперь работает! Напомню, что для распространения собранной с Phonon программы надо будет скопировать в ее каталог файл `phonon4.dll`, а также создать подкаталог `phonon_backend` и поместить туда `phonon_ds94.dll` (берется из `plugins/phonon_backend` каталога Qt).

Другим, в какой-то степени более простым, способом использовать Phonon в программах для Windows является сборка Qt и ваших программ в Microsoft Visual Studio C++. Хотя в документации Qt написано, что надо предварительно установить еще и DirectX SDK, повозиться с установкой путей и запустить некий скрипт, в моем случае достаточно было тех заголовочных файлов, которые идут в составе MS Visual Studio 2008. Для сборки Qt в этой IDE надо сначала из сеанса, где установлены пути к Visual Studio (см. ярлык на командный процессор в подменю Visual Studio в «Пуске»), запустить программу `configure.exe` (в исходнике Qt), чтобы настроить исходник. Если вам не нужен WebKit, советую отключить его на уровне настройки – он компилируется очень долго. Также можно отключить OpenGL, если вам он не критичен. Иначе придется включать в дистрибутив программы. После настройки традиционный путь – это запуск `ntmake` и ожидание, пока соберутся библиотеки и примеры. Но для сборки лучше не вызывать `ntmake`, а сделать с помощью `qmake` проект Visual Studio:

```
qmake -t vcapp имя qt-проекта.pro
```

Тогда получится проект с расширением `vsproj`, который можно загрузить в Visual Studio и скомпилировать.

После этого вы можете собирать программы с Phonon. В Windows удобно иметь две версии Qt: одну – MinGW, а другую – самостоятельно собранную под Visual Studio. Разграничение между ними – через установку путей в `.bat`-файлах запуска командного процессора. У меня на это дело – два ярлыка запуска с разными переменными среды (для MinGW и для Visual Studio).

Но теперь возникает последняя проблема – распространение программ, собранных в Visual Studio. Для этого нужны файлы из состава Microsoft Visual C++: `msvcr80.dll` (число зависит от версии вашей Visual Studio), `msvcp80.dll` и другие. От версии к версии условия их видимости для программы меняются: раньше можно было просто скопировать в каталог программы, затем надо было указать их расположение через вызов функции `qApp->addLibraryPath()`, потом в определенных условиях это перестало работать – в общем, опять отдельная тема для разговора, в которую не будем углубляться (некоторые подробности можно найти в разделе *Deploying an Application on Windows* документации Qt).

Но есть универсальное решение. Включайте в установщик своей программы пакет Microsoft Visual C++ 2008 Redistributable Package (берется на сайте Microsoft) или дайте на него ссылку. Это чтобы не возиться с манифестами и тому подобным материалом. После установки пакета пользователем ваша программа запустится без всяких сложностей. Может оказаться и так, что у пользователя уже установлен этот пакет – сейчас многие про-

граммы содержат его вместе со своими установщиками и тоже предлагают при желании проинсталлировать.

Неясности с Phonon

От трудностей сборки перейдем к неясным местам в использовании Phonon. Phonon поддерживает те мультимедийные форматы, которые поддерживаются бэкэндом. В случае с Xine или Gstreamer это зависит от установленных плагинов Xine и Gstreamer, а для Direct Show – от установленных кодеков-фильтров Direct Show. Например, для того чтобы ваша программа под Windows «понимала» Ogg Vorbis и FLAC, надо поставить в систему кодек с <http://xiph.org/dshow>. Как программно узнать, поддерживается ли запущенным у пользователя бэкэндом некий файл? Вопрос простой, а ответ на него – не то чтобы сложный, но хлопотный.

Статичная функция `Phonon::BackendCapabilities::availableMimeTypes()` возвращает `QStringList`, наполненный названиями MIME-типов, которые поддерживаются бэкэндом. А уж как этим списком распорядиться – дело ваше. Опишу один из способов. Задача стоит так: узнать, какому MIME-типу соответствует файл, и посмотреть, входит ли этот MIME-тип в список поддерживаемых типов. Чтобы узнать первое, надо, во-первых, получить расширение из пути к файлу. Предлагаю для этого следующую, предусматривающую некоторые досадные неожиданности, функцию:

```
QString file_get_ext (const QString &file_name)
{
    if (file_name.isNull() || file_name.isEmpty())
        return QString();

    int i = file_name.lastIndexOf (".");
    if (i == -1)
        return QString();

    return file_name.mid (i + 1).toLower();
}
```

Если имя файла – пустое или же не содержит расширения, возвращается пустая строка, иначе – приведенное к нижнему регистру расширение, без точки.

Теперь понадобится таблица соответствий расширений файлов и MIME-типов. Есть относительно стандартная таблица от FreeDesktop – пакет с ней обычно называется `shared-mime-info` и находится в `/usr/share/mime` (смотрите файлы `globs*`, да и вообще полезно посмотреть содержимое этих каталогов). Эта таблица используется в GNOME и KDE. Дополнения к ней сохраняются в `$HOME/.local/share/mime`. Таблица большая, но отнюдь не полная. К тому же, в Windows её нет. Если вы пишете многоплатформенную программу, придется искать универсальное решение. А оно – на поверхности: надо создать собственную базу MIME-соответствий и включить её в свою программу как ресурс. Обычный `ini`-подобный файл, который загружаете в `hash`-таблицу, где ключами служат расширения файлов, а значениями – MIME-типы.

Это можно сделать с помощью следующего кода. Сначала напишем вспомогательную функцию, загружающую файл в экземпляр `QString`:

```
QString qstring_load (const QString &fileName,
                     const QString &charset)
{
    QFile file (fileName);

    if (! file.open (QFile::ReadOnly | QFile::Text))
        return QString();

    QTextStream in(&file);
    in.setCodec (charset.toAscii());

    return in.readAll();
}
```

Теперь – саму функцию чтения в хэш-таблицу:

```
QHash<QString, QString> hash_load_keyval (const QString &fname)
{
    QHash<QString, QString> result;

    if (! file_exists (fname))
        return result;

    QStringList l = qstring_load (fname, "UTF-8").split ("\n");

    foreach (QString s, l)
    {
        QStringList sl = s.split ("=");
        if (sl.size() > 1)
            result.insertMulti (sl[0], sl[1]);
    }

    return result;
}
```

И наконец – пример использования. Объявим хэш-таблицу mime_types и прочитаем в нее таблицу из ресурса:

```
QHash<QString, QString> mime_types = hash_load_keyval „
(“:res/mime.txt”);
```

В ресурсе mime.txt у нас строки вида:

```
aiff=audio/aiff
mod=audio/mod
aif=audio/x-aiff
aiff=audio/x-aiff
m4a=audio/x-m4a
wav=audio/x-basic
[и так далее]
```

Чтобы получить «известные» в этой таблице MIME-типы, надо обратиться к функции QHash::values() без параметра. Чтобы получить MIME-типы, соответствующие некоему расширению файла, надо вызывать её же, но с параметром:

```
//Получаем расширение файла
QString ext = file_get_ext (fname);
if (ext.isEmpty())
    return;

//Получаем список MIME-типов для расширения ext
QList<QString> mts = mime_types.values (ext);
if (mime_types.count() == 0)
    return;
```

Полученный список потребуется просмотреть в цикле и проверить, входит ли какой-либо элемент списка в список типов, поддерживаемых бэкэндом. Если да, то файл поддерживается на воспроизведение.

Вот еще одна тонкость, которая плохо документирована. Речь идет о выборе выходного звукового устройства. Иногда не хочется использовать общесистемные установки (они настраиваются в центре управления KDE, в разделе настроек Phonon).

Phonon умеет выдавать список доступных выходных устройств:

```
QList<Phonon::AudioOutputDevice> audioOutputDevices = „
Phonon::BackendCapabilities::availableAudioOutputDevices();
```

С ним мало что можно сделать полезного – разве что получить названия:

```
QStringList sl;
foreach (Phonon::AudioOutputDevice ad, audioOutputDevices)
    sl << ad.name();
```

чтобы потом наполнить ими выпадающий список и добавить к нему выбор устройства:

```
//Создаем комбо-бокс
QComboBox *cb_outdevs = new QComboBox (this);

//Наполняем его элементами списка
cb_outdevs->addItem (sl);

//Читаем из QSettings сохраненный в настройках индекс –
//номер устройства из списка, и делаем его текущим в комбо-боксе
cb_outdevs->setCurrentIndex (settings->value „
(output_dev”, 0).toInt());

//Связываем комбо-бокс со слотом переключения:
connect (cb_outdevs, SIGNAL(currentIndexChanged (int )),
        this, SLOT(cb_outdevs_currentIndexChanged (int )));
```

Вот и сам слот:

```
void MainWindow::cb_outdevs_currentIndexChanged (int index)
{
    //Пишем индекс устройства в настройки:
    settings->setValue (output_dev”, index);
    //Делаем устройство текущим, используя индекс
    audioOutput->setOutputDevice (Phonon::BackendCapabilities:: „
availableAudioOutputDevices() [index]);
}
```

Само же выходное устройство Phonon, с учетом возможности выбора устройства, создаем примерно так:

```
audioOutput = new Phonon::AudioOutput (this);
int idx = settings->value (output_dev”, 0).toInt();
audioOutput->setOutputDevice (Phonon::BackendCapabilities:: „
availableAudioOutputDevices() [idx]);
```

Подведу итог. От списка выходных устройств нужны две вещи: их имена и индексы. Имена отображаются пользователю, а индексы требуются, чтобы задавать нужное устройство вывода. Когда пользователь выбирает название устройства, записываем его индекс в настройки и затем задаем это устройство через функцию Phonon::AudioOutput::setOutputDevice(). То же происходит при загрузке программы сразу после создания экземпляра класса Phonon::AudioOutput.

Петр Семилетов
(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, обоснованной 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.

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 рабочим 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 национальным каталогам или по списку номенклатуры «АРЗИ»:

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

- ✓ **Казахстан** – по каталогу «Российская 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 номенклатуры «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 «Пошта Молдовей» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
по списку через ГУП «Почта Приднестровья» (МД-3300, г. Тирасполь, ул. Ленина, 17)
по прайс-листу через ООО агентство «Editil Periodice» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
- ✓ Подписка для **Украины**:
Киевский главпочтамт
Подписное агентство «KSS»
Телефон/факс (044)464-0220