



№68, 15 сентября 2010

**Обзор простого видеоредактора
для Linux – OpenShot**

**Нестандартные GUI-тулкиты:
знакомство с Ultimate++**

**Право выбора, или Открытые форматы:
MNG и APNG**

**QLandKarte GT как замена OziExplorer
в GNU/Linux**

Колонка главного редактора



Видные представители корпоративного мира играют с проектами свободного программного обеспечения исключительно по своим правилам. И несмотря на то,

что эти правила навряд ли официально зафиксированы в тоннах макулатуры, составлены они не менее профессионально и ловко, чем договоры тех же компаний. Настолько ловко, что некоторым людям всерьез начинает казаться, что вон те бизнес-ребята абсолютно искренне заинтересованы в продвижении FLOSS-проектов, возвращении кода в upstream и добровольных пожертвованиях только потому, что это «правильно», «красиво», «честно» и т.п. Благо, циничная современность научила искать в подобных проявлениях и другие мотивы – куда более практические и, как ни крути, жизненные. Тогда всё вроде бы встает на свои места...

Но не всё. Потому что далеко не каждый в момент осознания реальных бизнес-задач перестает кричать, что пресловутые «ребята» поступили «неправильно» и «некрасиво» по отношению к сообществу. И вот тогда хочется предложить этим людям на мгновение представить себя в роли руководителя такой компании. Руководителя, который действительно любит FLOSS и понимает его плюсы, но вынужден смотреть на спорную ситуацию честно. Фанатизм или результат? К сожалению, компромиссы не всегда выгодны и не для всех возможны.

Главный редактор
Дмитрий Шурупов
(osa@samag.ru)

«Open Source»

электронное приложение к журналу

«Системный администратор»

№68, 15 сентября 2010 г.

РЕДАКЦИЯ

Исполнительный директор

Владимир Положевец

Главный редактор

Дмитрий Шурупов

Верстка и оформление

Владимир Лукин

Иллюстрация на обложке

Анна Герцова

Сайт электронного приложения:

<http://osa.samag.ru>

За содержание статей ответственность несет автор.

Все права на опубликованные материалы защищены.

Новости мира Open Source

WeTab и Samsung Galaxy Tab – планшеты на базе Android

2 сентября прошла совместная пресс-конференция компаний Intel и 4tuitoo, стоящей за разработкой интернет-планшета WeTab. На ней было объявлено о том, что устройство работает под управлением мобильной Linux-системы MeeGo. WeTab – планшет из Германии, о котором стало известно еще в апреле. Его продажи в Германии начнутся в этом месяце.

Ранее сообщалось, что в качестве операционной системы используется некая WeTab OS, «основанная на Open Source-технологиях и ядре Linux», а также обладающая поддержкой запуска приложений для Android. Как выяснилось позже, в ее основе был дистрибутив Ubuntu Linux. Теперь же объявлено о том, что разработчики решили отказаться от Ubuntu в пользу MeeGo, поскольку это позволит им увеличить время работы от аккумулятора и общую производительность.

Основные технические характеристики WeTab: 11,6-дюймовый тачскрин с поддержкой multitouch и разрешения 1366x768; процессор Atom N450 Pineview-M с тактовой частотой 1,66 ГГц; 16 или 32 Гб памяти для хранения данных; Wi-Fi 802.11a/b/g/n, Bluetooth 2.1 + EDR.

Тем временем, Samsung представила общественности свой планшет, работающий под управлением Android, – Samsung Galaxy Tab. Он был впервые продемонстрирован на проходящей в Берлине выставке IFA 2010. Продукт позиционируется как достойная альтернатива Apple iPad с операционной системой Android 2.2 на борту, хотя пока точно неизвестна ни дата начала его продаж, ни цена.

Основные характеристики Samsung Galaxy Tab: 7-дюймовый емкостной тачскрин с поддержкой разрешения 1024x600 (WSVGA); процессор ARM Cortex A8 с тактовой частотой 1 ГГц; 4 Гб оперативной памяти; 3,2-мегапиксельная камера с автофокусировкой, LED-вспышкой, возможностью записи видео в разрешении 720x480; поддержка мобильной связи: 2.5G (GSM/GPRS/EDGE v12), 3G (HSPA 5,76 мбит/с, HSDPA 7,2 мбит/с); Wi-Fi 802.11 b/g/n, Bluetooth 3.0.

Выход планшета Samsung Galaxy Tab на европейские рынки запланирован на середину сентября, а США и Азии – в конце года. Предполагаемая стоимость модели Samsung Galaxy Tab с 16 Гб на российском рынке – от 32 тысяч рублей (по оценке Эльдара Муртазина).

Редактор Bepin переименовали в Mozilla Skywriter

Проект фреймворка для совместного редактирования программного кода Bepin, создаваемый в недрах Mozilla, получил новое название: Mozilla Skywriter. Кроме того, его разработка была переведена на GitHub.

Вообще, авторы задаются целью передать в названии проекта идею «написания кода в облаках». По их мнению, Skywriter отражает ее лучше, чем Bepin. Тем временем, сам проект приближается к релизу версии 1.0, и его серверную часть решено переписать на node.js – событийного фреймворка с открытым кодом, предназначенного для запуска JavaScript-кода на стороне сервера. Таким образом, разработчики намерены подготовить десктоп-версию Skywriter на базе XULRunner и новую серверную версию, основанную на node.js.

Перевод официального репозитория Skywriter на GitHub связан с тем, что многие разработчики высказывали желание работать на этом сервисе и даже создавали его неофициальные зеркала. Теперь все будут работать через централизованный репозиторий кода на GitHub.

Microsoft передала 25 тысяч долларов Mercurial

Проект CodePlex.com, созданный и поддерживаемый корпорацией Microsoft, передал 25 тысяч USD на развитие популярной Open Source-системы управления версиями Mercurial. Этот шаг стал ответом на опубликованный еще в апреле призыв лидера Open Source-проекта Мэтта Мэколлы (Matt Mackall) помочь ему в финансировании полного рабочего дня для развития Mercurial. К спонсированию приглашались все компании, которые считают себя «серьезным пользователем Mercurial». И вот Microsoft оказалась одним из таких серьезных пользователей, а причина тому – ее хостинг для Open Source-проектов CodePlex.

Сообщается, что большинство проектов на CodePlex используют для управления версиями своего программного обеспечения продукт Microsoft – Team Foundation Server (TFS). Но несмотря на это, более 1500 проектов работают с Mercurial. Кроме то-

го, это популярный выбор для тех, кто переходит на CodePlex с других хостинговых платформ. Все это делает Mercurial «важной возможностью CodePlex».

Вышла редакция дистрибутива Linux Mint на базе Debian

7 сентября проект Linux-дистрибутива Mint, основанного на Ubuntu, представил редакцию своей системы на базе Debian GNU/Linux – Linux Mint Debian (LMDE). Первый релиз LMDE получил версию 201009. Авторы называют выпуск нового дистрибутива важным шагом для всего проекта. Заявляется, что благодаря использованию в качестве базы Debian GNU/Linux, новый дистрибутив (LMDE) будет работать быстрее, а интерфейс станет более отзывчивым (по сравнению с основными редакциями Linux Mint). В LMDE представлены все те же возможности, что и в обычном Linux Mint, – изменилась лишь база, где Debian пришел на смену Ubuntu.

Таким образом, в составе Linux Mint Debian помимо стандартных пакетов из репозитория Debian (обеспечена совместимость с «testing» и «squeeze») можно найти разработки проекта Linux Mint, среди которых – новый инсталлятор и менеджер программного обеспечения (Software Manager). При сравнении LMDE с основными редакциями Linux Mint отмечается, что LMDE будет менее стабильным и, возможно, иногда менее «дружелюбным», зато обеспечит лучшую скорость работы и более оперативное обновление пакетов ПО.

Пока LMDE распространяется только в 32-битной версией с рабочей средой GNOME. Вопрос появления других редакций LMDE еще пока открыт. Дистрибутив LMDE называется «экспериментом».

Oracle и NetApp урегулировали разбирательство по ZFS

Три года назад компания NetApp (Network Appliance) подавала иск на Sun Microsystems из-за нарушения ее патентов в файловой системе ZFS. Новая владелица ZFS, корпорация Oracle, урегулировала эти споры.

В том же 2007 году, уже в октябре, Sun ответила на действия NetApp взаимным иском, утверждая, что претензии NetApp не обоснованы, и обвинив NetApp в нарушении патентов Sun.

«До кучи», в апреле 2008 года, Sun подала еще один иск против NetApp, утверждая, что в программном обеспечении Onano SANscreen и NAS Insight используются запатентованные Sun технологии.

И вот теперь NetApp и Oracle пришли к компромиссу, отказавшись от взаимных претензий. Условия, по которым было проведено урегулирование, не разглашаются. Так или иначе, все «сторонние» пользователи ZFS теперь могут не беспокоиться за свои продукты.

Broadcom выпустила Open Source-драйвер для ядра Linux

Компания Broadcom представила первый релиз нового драйвера для поддержки своего беспроводного оборудования в Linux. Его исходный код открыт, и в нем используются «родные» средства ядра Linux.

Новый Linux-драйвер от Broadcom использует фреймворк SoftMAC, он уже попал в ветку staging разработки ядра Linux. И ожидается, что скоро этот драйвер войдет в состав ядра. Актуальность этих событий сложно переоценить, поскольку поддержка беспроводного сетевого оборудования Broadcom всегда была проблемой для пользователей Linux из-за наличия проприетарного блоба.

На данный момент, в новом драйвере Broadcom обеспечена поддержка BCM4313, BCM43224 и BCM43225. Вероятно, в дальнейшем спектр поддерживаемого оборудования будет расширен. По информации разработчика из Canonial, новый драйвер Broadcom может быть включен в состав ближайшего релиза Ubuntu Linux – 10.10 «Maverick Meerkat».

Apple перестала возвращать код в проект GCC

Как стало известно из почтовой рассылки GCC (GNU Compiler Collection), компания Apple изменила свою политику касательно разработок для этого популярного проекта свободных компиляторов.

Разработкой GCC руководит некоммерческая организация Free Software Foundation (FSF). Выяснилось, что Apple не намерена тратить усилия на включение в состав основной ветки разработки GCC своих патчей, реализующих поддержку Objective-C 2.0. Крис Лэттнер (Chris Lattner), старший архитектор группы специалистов по компиляторам в Apple, сообщил, что в компании «больше нет внутреннего процесса для передачи кода организации FSF». Высказывается предположение, что такая политика связана с нежеланием Apple принимать последнюю версию «главной» лицензии FSF на свободное программное обеспечение, под которой распространяется и GCC, – GNU GPLv3.

Представлен NPF – новый межсетевой экран для NetBSD

В почтовой рассылке NetBSD был представлен новый межсетевой экран для этой операционной системы – NPF. Заявляется, что он ориентирован на высокую производительность на многопроцессорных системах и простую расширяемость. NPF создан литовцем Миндаугасом Расюкевичиусом (Mindaugas Rasiukevicius), а его разработку спонсирует организация NetBSD Foundation. Он стал третьим фильтром пакетов (packet filter) для NetBSD после IP Filter и PF. Одна из его уникальных характеристик – использование интерпретатора байт-кода в движке разбора пакетов.

Ожидается, что к концу января 2011 года NPF оснастят всеми возможностями, предоставляемыми другими фильтрами пакетов для NetBSD, а также документацией и примерами конфигурации. Единственное, чего будет не хватать, – поддержки IPv6. Разработчик ищет помощников и готов предоставить им техническую поддержку по реализации IPv6 в NPF.

Linux-дистрибутив sidux переродился как aptosid

Проект дистрибутива sidux, основанного на нестабильной ветке Debian GNU/Linux, снова жив – теперь под названием aptosid.

Дистрибутиву sidux пришлось сменить название из-за конфликта с организацией sidux e.V., обладающей правами на торговую марку «sidux» на территории Европы. Новый дистрибутив – aptosid – предоставит всем пользователям sidux возможность простой миграции. Aptosid по-прежнему основан на ветке Debian GNU/Linux «sid».

Первым релизом aptosid стала версия 2010-02 с кодовым названием «Κηφές». В ней представлено ядро Linux 2.6.35 и рабочая среда KDE 4.4.5. Подробности можно найти на новом сайте проекта: <http://aptosid.com>.

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

Обзор простого видеоредактора для Linux – OpenShot

В августе 2008 года Джонатан Томас (Jonathan Thomas) начал работу над новым свободным и простым видеоредактором. Проект получил название OpenShot (<http://www.openshotvideo.com>). Как сказал сам разработчик в одном из интервью (<http://www.gnomejournal.org/article/90/interview-with-jonathan-thomas-creator-of-the-openshot-video-editor>), OpenShot создается им в свободное от основной работы (разработка проприетарного программного обеспечения в одной из компаний в Техасе) время.

Openshot – это свободный редактор нелинейного видеомонтажа для операционной системы GNU/Linux. Программа разработана с использованием Python, GTK+ и фреймворка Media Lovin' Toolkit (MLT, <http://www.mltframework.org>), ориентированного на телевизионное вещание. К слову, при изучении инструментов разработки автор делал выбор между Mono и Python, но остановился на последнем в силу его «впечатляющей поддержки биндингов мультимедийных библиотек, таких как GStreamer и MLT». В свою очередь, GTK+ был выбран, поскольку это стандартный инструментарий GNOME – требовалось, чтобы OpenShot выглядел как «родное» приложение этой среды.

Основные возможности

Благодаря встроенной поддержке FFmpeg видеоредактор может работать с большим количеством форматов аудио, видео и графики. Среди них – ogg, avi, dv, flv, mov, matroska, mp2, mp3, mp4 и другие. С полным списком поддерживаемых форматов и кодеков можно ознакомиться в меню: «Изменить → Параметры → AV Форматы». В правой части окна будут показаны установленные аудио-, видеокодеки и поддерживаемые форматы. Для обновления информации о последних имеется специальная кнопка сверху окна – «Перезагрузить кодеки». Дополнительную информацию по поддерживаемым форматам и кодекам можно получить из встроенной в программу справки в пункте 23 – List of Formats & Codecs.

Представлены возможности для творческой работы с видео: более 40 видов переходов и около 40 эффектов. Кроме того, OpenShot интегрирован со средой GNOME с поддержкой drag & drop (например, перетаскивание рисунка или видеофайла из каталога или рабочего стола в окно программы на место эскизов, а также для перемещения объектов как внутри одной временной шкалы, так и с одной на другую). Имеется поддержка изменения размера и обрезки клипов на шкале времени, работы с несколькими треками одновременно, перекодирования видео (с использованием кодеков FFmpeg).

Установка

Основная страница проекта на Launchpad (я пользуюсь Trisquel GNU/Linux 3.5) находится здесь: <https://launchpad.net/openshot>. Но мне потребовалась экспериментальная ветка, потому что статус поддержки OpenShot для Ubuntu 9.10, на котором

основан Trisquel, носит экспериментальный характер. На странице <https://launchpad.net/~jonoomph/+archive/openshot-edge> в разделе Adding this PPA to your system я выбрал Karmic 9.10, после чего появились ссылки на соответствующий репозиторий с OpenShot. Перед обновлением менеджера пакетов необходимо добавить специальный ключ, который удостоверяет пакеты с данного репозитория PPA:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com \
--recv-keys EDFBD1F9
```

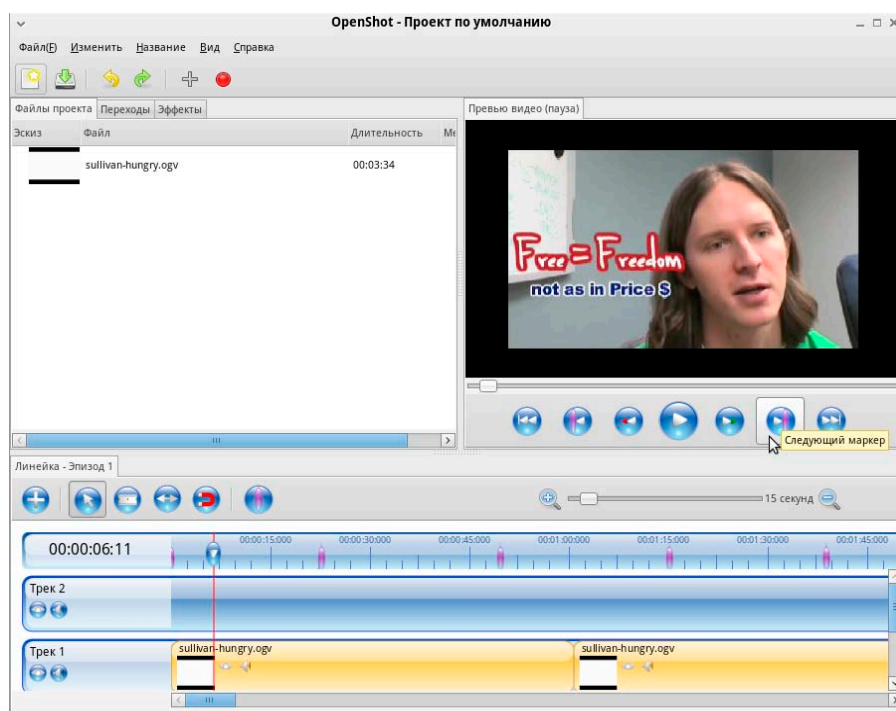
Теперь осталось добавить указанные выше ссылки на репозитории в «Источники приложений» Synaptic («Настройки → Репозитории → Другое ПО → Добавить»), а затем обновить менеджер пакетов. Собственно, данный репозиторий включает два пакета: Openshot версии 1.1.3 и Python-MLT версии 1.0.3, а также их исходные коды. Пользователям Ubuntu 10.04 «Lucid Lynx» совершать эти действия не потребуется – все нужные пакеты есть в основном репозитории.

Установка программы осуществляется традиционно – например, через Synaptic или командную строку:

```
sudo apt-get install openshot
```

Общий объем загружаемых пакетов с зависимостями для только что установленной операционной системы Trisquel 3.5 составил ~ 25,2 Мб. При этом пакет libstdc1.2.debian-alsa был помечен для удаления, а libstdc1.2.debian-pulseaudio – для установки. После инсталляции в разделе «Аудио и видео» главного меню появился пункт Open Shot Video Editor.

Что касается других дистрибутивов, то OpenShot доступен по умолчанию в операционной системе AV Linux (<http://www.bandshed.net/AVLinux.html>), а также в виде RPM-пакета для Fedora 11/12 и в виде архива с исходным кодом на официальном



Интерфейс OpenShot

ной странице проекта. Кроме того, в репозитории SlackBuilds есть пакет (с версией 1.1.1) для Slackware 13. На момент написания статьи мне больше неизвестны репозитории, включающие сборку данного видеоредактора. (Впрочем, это не повод отчаиваться. Например, и для Gentoo Linux в ряде «оверлеев» представлен ebuild: <http://gpo.zugaina.org/media-video/openshot>. — Прим. ред.)

Интерфейс

Интерфейс программы отличается продуманностью и эргономикой. Окно видеоредактора разделено на три основные части: слева — область с файлами проекта; справа — область предварительного просмотра подготовленного видео; внизу — временная линейка для размещения треков, снабженная кнопками для таких инструментов, как «Добавить дорожку», «Указатель» (используется, например, для смещения файлов влево/вправо относительно трека), «Бритва» (обрезает файл в указанном месте), «Изменение размера», «Прикрепление» и «Добавить маркер» (см. ниже), а правее — инструмент масштабирования «шкалы времени» на треках (по умолчанию задан шаг в 15 секунд).

Работаем с OpenShot

Рассмотрим ряд возможностей видеоредактора OpenShot на практике. Для этого понадобится хотя бы один видеофайл. Я загрузил [sullivan-hungry.ogv](http://www.artfiles.org/gnu.org/ftp/pub/video), в котором Джон Салливан (John Sullivan) из Free Software Foundation рассказывает о свободном ПО, копилефте и т.п. — он доступен в специальном архиве по адресу <http://www.artfiles.org/gnu.org/ftp/pub/video>. После запуска OpenShot откройте в нем видеофайл — кстати, заодно можно убедиться в работающем drag & drop.

Переместим видеофайл из области файлов проекта на Трек 1 внизу окна и воспользуемся инструментом Бритва, чтобы «разрезать» видео. Для этого нажмем кнопку «Бритва» (обратите внимание на всплывающую подсказку), опустим курсор на Трек 1 — он принимает форму бритвы — и наведем на нужное место в видеофайле. Щелчок левой кнопкой мыши разделит файл на две части.

Работу со следующим инструментом — «Изменение размера» — можно оптимизировать. Для этого нажмем кнопку «Изменение размера» и переместим курсор приблизительно к середине видеофрагмента. Курсор примет форму стрелки, направленной влево или вправо (в зависимости от позиции курсора). Если стрелка направлена влево, то при клике левый край видеофрагмента будет перемещен в указанную (текущим курсором) позицию. Аналогично можно и дальше перемещать курсор по видеофайлу, изменяя размеры без перетаскивания краев файла вручную.

Еще одним инструментом, повышающим удобство работы с OpenShot, являются маркеры. Кнопка «Добавить маркер» (на ней изображен розовый овал, напоминающий лепесток), активирующая этот инструмент, находится на панели инструментов «Линейки» внизу окна программы. Маркеры добавляются путем перемещения текущей позиции по шкале времени и нажатия на эту кнопку. На их месте появляются «розовые овалы». После установки маркеров можно осуществить воспроизведение видео, перемещаясь от одного маркера к другому. Это удобно, например, если планируется «обрезка» видео — можно проработать разные варианты. Для перехода по маркерам во время воспроизведения нужно воспользоваться соответствующими кнопками в окне предпросмотра.

Сохранение проекта и экспорт видео

Несколько слов о работе с проектами в OpenShot. Они сохраняются в специальный формат .osp («Файл → Сохранить про-

ект/Сохранить проект как...»). После того, как работа над проектом завершена, можно экспортировать подготовленное видео: «Файл → Экспортировать видео» — и появится окно с одноименным названием. В нем выбираем имя файла и директорию для экспорта. Есть возможность выбора профиля (автоматическая установка параметров, оптимизированная под устройство, сервис и т.д.):

- ☒ Blu-Ray/AVCHD (AVCHD-диски);
- ☒ DVD (DVD-NTSC/PAL);
- ☒ веб (Flickr, Metacafe, Picasa, YouTube, Vimeo);
- ☒ все форматы (AVI, FLV, MOV, MP4, MPEG и OGG);
- ☒ устройство (AppleTV, Nokia nHD и Xbox 360).

Затем выбираем профиль видео (зависит от выбранного профиля проекта) и качество (низкое/среднее/высокое). Указанные параметры экспорта можно детализировать, получив более широкие возможности настройки. Для этого необходимо перейти на вкладку «Дополнительно» окна экспорта видео. Когда все настройки экспорта заданы, для начала операции нажимаем кнопку «Экспортировать видео».

Недостатки

При работе с эффектами (добавление эффекта для файла ogv, а затем воспроизведение видео) у меня периодически происходили падения программы. После того, как впервые столкнулся с этой проблемой в Trisquel, решил проверить, будет ли возникать подобная ситуация в Ubuntu 10.04 (где пакет не носит статус «экспериментального»). Проблема по-прежнему проявлялась и там. К слову, на странице проекта на Launchpad есть раздел с известными ошибками. Если на странице <https://bugs.launchpad.net/openshot/+filebug> поискать по «effect», можно получить список всех ошибок, так или иначе связанных с эффектами. Например, есть ошибка #526061 — падение при использовании эффекта «Nervous». Итак, я хотел добавить информацию об обнаруженной проблеме и для этого запустил видеоредактор из консоли (с целью сохранить вывод о работе программы), но по какой-то причине ошибка больше не повторялась. В общем, на мой взгляд, это говорит о недостаточной стабильности приложения, что подтверждает и беглый просмотр всех ошибок, о которых сообщили пользователи.

К недостаткам отнесу и скромные возможности работы с аудио. Все доступные опции — это импорт аудиофайлов в проект, добавление на трек, возможность отдельного воспроизведения аудио и видео (например, во время работы с проектом — см. кнопки на каждом из треков), управление затуханием звука в начале/конце воспроизведения и временным интервалом, на протяжении которого оно будет происходить.

Заключение

OpenShot порадовал интерфейсом и удобством работы. Кроме того, его отличает поддержка большого количества аудио-, видео- и графических форматов. Все это позволило включить его, например, в репозиторий Ubuntu 10.04. К сожалению, приложение ещё недостаточно стабильно — на Launchpad на момент написания статьи можно было увидеть более 300 известных ошибок. Кроме того, видеоредактор пока не в состоянии конкурировать по функциональности, например, с тем же Kdenlive (Его обзор можно найти в «Open Source» №048 от 17.08.2009. — Прим. ред.). В то же время продолжается работа над расширением функциональности OpenShot и авторы обещают добавлять разнообразные улучшения.

Игорь Штомпель
(keepercoder@gmail.com)

Нестандартные GUI-тулкиты.

Часть 1: Знакомство с Ultimate++

При создании GUI-приложений для Linux выбор тулкита играет первоочередную роль. Именно от него будет зависеть не только трудоемкость разработки, но переносимость программы, ее внешний вид и возможность дальнейшего развития. Для определенности предположим, что программа пишется на C++, т.к. рассмотрение привязок для различных языков – это отдельная обширная тема. По сути выбор сводится к тройке наиболее известных и популярных тулкитов: Qt, GTK+ и wxWidgets. Для небольших «легких» программ иногда используется FLTK. К сожалению, каждый из этих «трех китов» имеет свои специфические недостатки, поэтому альтернативные нестандартные GUI-тулкиты продолжают вызывать интерес. Нестандартные решения могут заинтересовать разработчиков либо выдающимся внешним видом виджетов, либо оригинальными приемами программирования, которые экономят время на разработку и поддержку программ. Чтобы быть конкурентоспособными, они должны быть кроссплатформенными, поддерживать Unicode и иметь интегрированную RAD-среду разработки или хотя бы визуальный редактор форм.

В этой статье рассмотрен тулkit Ultimate++. Внешний вид его виджетов ничем не примечателен (используются «родные» виджеты для каждой платформы), однако используемые приемы программирования кардинально отличаются от всех других тулкитов. Все компоненты U++ очень специфичны и в буквальном смысле не похожи ни на что другое. Это относится и к синтаксису, в котором иногда трудно узнать C++ как таковой (хотя это абсолютно стандартный C++, без каких-либо дополнений, как в Qt), и к откровенно странной IDE. Не ждите от U++ интуитивной понятности – тривиальные вещи нужно, зачастую, долго «раскапывать» в документации и на форумах. «Барьер вхождения» в U++ высок, но вложенные усилия затем быстро окупаются.

Обзор U++

Ultimate++ (<http://www.ultimatepp.org>), известный как U++ и UPP, позиционируется не столько как GUI-тулkit, сколько как целостная инфраструктура (framework) для быстрой разработки кроссплатформенных приложений на C++. U++ распространяется по свободной лицензии BSD и является бесплатным как для открытых, так и для коммерческих разработок. Код GUI-приложений, написанный с использованием U++ чрезвычайно лаконичен, и напоминает код «сверхвысокого уровня», характерный для скриптовых языков. Это достигается повсеместным использованием современных возможностей C++ во внутренней структуре U++. На официальном сайте имеется впечатляющее сравнение идентичных программ написанных на U++, Qt, wxWidgets и Java ([http://www.ultimatepp.org/www\\$suppweb\\$comparison\\$en-us.html](http://www.ultimatepp.org/www$suppweb$comparison$en-us.html)). Во всех случаях код на U++ намного короче и понятнее, хотя и выглядит необычно.

Инфраструктура U++ спроектирована так, что в идеале программисту не нужны

ни сторонние библиотеки, ни даже STL и Boost – все, начиная от контейнеров и заканчивая работой с XML и SQL, делается силами самой U++. Этим «глобальным изобретением велосипеда» U++ напоминает Qt. U++ включает несколько компонентов:

☑ Набор стандартных библиотек:

- **Core** – строки, даты и время, библиотека контейнеров NTL (аналог STL с несколько иным поведением и, как утверждается, более быстрая), обратные вызовы, многопоточность, работа с XML и т.д.
- **Draw** – канва для растровой графики.
- **CtrlCore** – ядро GUI-приложений, обеспечивающее отображение окон, передачу событий, обработку логических координат и т.п.
- **CtrlLib** – библиотека виджетов.
- **RichText** – сложный форматированный текст с импортом/экспортом в RTF и HTML.
- **SQL** – функции для работы с SQL-запросами и интерфейсы к СУБД SQLite3, MySQL, PostgreSQL, Microsoft SQL, Oracle.
- **Esc** – встраиваемый интерпретатор специфического скриптового языка.
- **Web** – работа с сетью и интернетом.

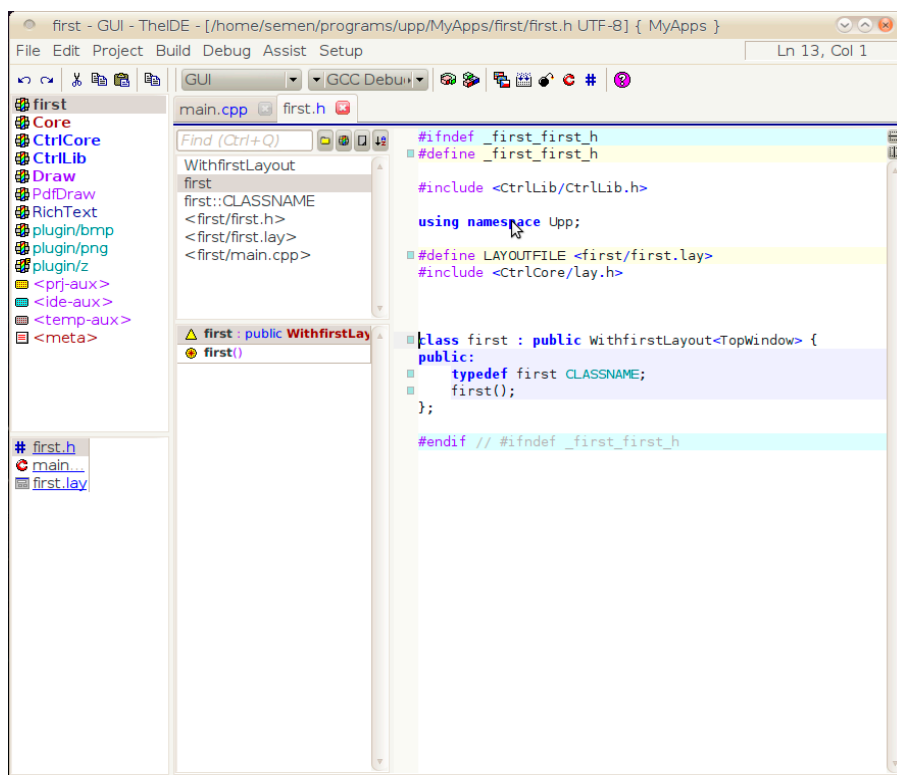
☑ Система управления компиляцией BLITZ-build.

☑ Полноценная интегрированная среда разработки TheIDE с визуальным редактором компоновки виджетов.

☑ Система документирования Topic++.

☑ Система анализа, автодополнения и оптимизации кода Assist++.

Все эти компоненты рассчитаны на совместное использование и по отдельности работать не будут. Кроме того, орга-



Главное окно TheIDE

низация кода в виде «пакетов» в U++ настолько специфична, что разработка с помощью U++ автоматически подразумевает использование исключительно «родной» IDE. Учитывая специфичность TheIDE, это можно считать достаточно серьезным недостатком.

U++ работает в Linux и Windows и в настоящее время портируется для MacOS X. Программы, написанные на одной платформе, без изменений компилируются на другой.

Установка

U++ не входит в состав основных дистрибутивов Linux, однако на официальном сайте есть DEB-пакеты для Ubuntu, RPM – для Fedora, Mandriva и OpenSUSE, а также ebuild для Gentoo. Кроме того, U++ можно собрать из исходных кодов на любой системе, но большое число зависимостей может сделать этот процесс довольно сложным. В Ubuntu последнюю версию U++ можно установить из ppa-репозитория следующими командами:

```
$ sudo add-apt-repository ppa:dolik-rce/upp-nightly
$ sudo apt-get update
$ sudo apt-get install upp
```

При этом будет установлено около 60 новых пакетов объемом более 50 Мб.

Пакеты, гнезда, коллекции

Перед тем, как рассматривать специфическую для U++ среду разработки TheIDE, необходимо разобраться с понятиями пакетов и коллекций. Обычно разработчики на C++ мыслят в категориях библиотек и их заголовочных файлов, однако в U++ используется идея пакетов, которая будет привычной для программистов, работавших с Delphi или Lazarus. Если «подключить» пакет к проекту, то IDE автоматически установит необходимые пути и флаги для компилятора. Каждый проект в свою очередь является пакетом, который можно при определенных условиях подключать к другим. Технически пакет – это просто отдельная директория, содержащая исходные коды и файл-описание с расширением upr (это очень похоже на пакеты в Python или Tcl). Файлы .upr обновляются автоматически самой IDE и по сути являются аналогами Make-файлов с описанием зависимостей и флагов компиляции. Несколько пакетов объединяются в «гнездо» (nest), а гнезда – в коллекции (assemblies), которые позволяют сортировать их по категориям.

Все программы, созданные пользователем, помещаются в общее глобальное дерево пакетов, а все файлы располагаются в соответствующих поддиректориях. Эта особенность может показаться неудобной, однако автоматическое поддержание порядка в дереве проектов, на мой взгляд, перевешивает непривычность такого подхода.

Корень дерева пакетов выбирается пользователем при первом запуске TheIDE. По умолчанию это ~/upr. При первом запуске TheIDE копирует файлы всего дерева пакетов U++ в эту директорию. В ее поддиректориях в дальнейшем будут располагаться все создаваемые пользователем программы и переместить их в какое-то другое место нельзя.

TheIDE

При каждом запуске TheIDE открывается окно выбора главного пакета проекта. Если мы хотим создать новую программу, то выбираем коллекцию MyApps (пока пустую) и щелкаем New Package. В диалоге создания пакета нужно задать его имя и выбрать шаблон (пустой проект, консольная программа, программа с графическим интерфейсом и т.п.). При выборе шаблона справа отображается содержимое генерируемых файлов.

После выбора главного пакета (или создания нового) мы оказываемся в главном окне TheIDE. Интерфейс этой среды достаточно необычен (см. **рисунок**). Слева расположена панель, где отображаются используемые в данном проекте пакеты (вверху) и исходные файлы (внизу). Правее расположена панель навигации. В ее верхней части отображаются компоненты файла, пакета либо целого гнезда, а в нижней – функции, классы и определения. Еще правее расположено окно текстового редактора.

Нажав <Ctrl> + <F5> можно скомпилировать проект и запустить программу. При первой компиляции будут собраны все подключенные к проекту стандартные пакеты U++, что занимает некоторое время. Последующие компиляции происходят намного быстрее. Нужно сказать, что система сборки BLITZ-build оправдывает свое название: компиляция и компоновка происходят на удивление быстро.

Структура меню TheIDE, мягко говоря, непривычна. Пункт Help напрочь отсутствует. Вместо него есть только кнопка на панели инструментов. В меню File нет пункта New – вместо этого для создания нового файла нужно выбрать Edit file и ввести несуществующее имя. Зато в меню File есть пункты для разделения окна редактора, закладок и скрытия панели уведомлений, которые, казалось бы, должны находиться в отсутствующем меню View. В интерфейсе имеются и другие странности. Например, добавить к проекту новый файл можно, исключительно выбрав Special → Insert any file(s) в контекстном меню списка файлов в левой панели. Понять логику работы панели навигации мне так и не удалось: что именно отображает верхняя часть панели и по какому принципу, осталось загадкой. Иными словами, интуитивность – явно не главное для разработчиков TheIDE, и с этим нужно смириться.

При использовании темных системных тем GTK+ белый текст в меню TheIDE отображается на светлом фоне и не читается. Чтобы избавиться от этого неприятного явления, нужно открыть диалог Setup → Environment → IDE и в поле GUI Theme установить любую понравившуюся тему вместо Host platform.

Первая программа

Перейдем к реальному примеру. Напишем простую бесполезную программу, которая выполняет выбранное арифметическое действие с двумя числами. Запустим TheIDE и создадим новый пакет demo1 в коллекции MyApps. В качестве шаблона выберем Basic CtrlLib Application. Будет создан единственный исходный файл main.cpp с таким кодом:

```
#include <CtrlLib/CtrlLib.h>
GUI_APP_MAIN{
}
```

Это и есть минимальная программа на U++, которая пока не делает ничего. Макрос GUI_APP_MAIN заменяет функцию main и отвечает за инициализацию библиотеки. Первое, что нужно сделать, – добавить главное окно:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

class MyWindow: public TopWindow {
};

GUI_APP_MAIN {
    MyWindow w;
    w.Run();
}
```

Все классы U++ находятся в пространстве имен Upp, поэтому лучше сразу же его задействовать.

Мы наследуем класс `MyWindow` от класса окна `TopWindow`, создаем его экземпляр и входим в цикл обработки событий методом `Run`. Скомпилировав программу, можно увидеть пустое окно, которое умеет только закрываться. Добавим кнопки сворачивания-разворачивания, возможность менять размер и установим начальные размеры окна, вызывая нужные методы в конструкторе класса:

```
class MyWindow: public TopWindow {
public:
    MyWindow(){
        Title("Hello world!").MinimizeBox().MaximizeBox()
        .Sizeable().SetRect(0, 0, 300, 300);
    }
};
```

Такое «нанизывание» вызываемых методов характерно для `U++` и делает код очень компактным.

Добавим в окно нужные виджеты: два текстовых поля для операндов и подписи к ним, выпадающий список для арифметических операций с подписью, кнопку «Вычислить» и надпись для вывода результата:

```
class MyWindow: public TopWindow {
public:

    EditDouble val1, val2;           // Поля ввода для операндов
    Label l1,l2;                    // Подписи для полей ввода
    DropList operation;             // Выпадающий список операций
    Label l3;                        // Подпись для списка
    Button compute;                 // Кнопка «вычислить»
    Label result;                   // Надпись для результата
```

Обращают на себя внимание несколько моментов. Во-первых, в `U++` существуют особые типы полей ввода для вещественных и целых чисел. С использованием готового виджета `EditDouble` можно забыть о неправильном вводе числа пользователем. Во-вторых, все виджеты создаются статически как обычные члены класса и нет нужды явно указывать их иерархию. Читательность и простота кода от этого сильно выигрывают. Сравните:

```
QLabel* l1 = new QLabel(parent);    //Qt
Label l1;                           //U++
```

Следующий этап – разместить виджеты в окне. В `U++` нет «резиновой» компоновки, реализуемой в `GTK+` или `Qt` с помощью «сайзеров» (sizers). Вместо нее применяются так называемые логические координаты, отсчитываемые от краев родительского виджета. Для каждого виджета определена функция `LeftPos(a,size)`, задающая расстояние от левого края родителя до виджета и ширину виджета. Аналогично работают `RightPos`, `TopPos` и `BottomPos`. Функция `HSizePos(a,b)` задает расстояния от левого и правого края родителя до виджета и дает виджету менять ширину вместе с родителем. Так же работает `VSizePos`. Забегая вперед, скажу, что в `TheIDE` есть удобный визуальный редактор компоновок, который будет описан в следующей части

статьи. Пока же укажем всю геометрию вручную. Делается это в конструкторе с помощью функции `Add(widget)`, которая отображает переданный ей виджет согласно его логическим координатам. Полностью программа теперь выглядит так:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

class MyWindow: public TopWindow {
public:

    EditDouble val1, val2;           // Поля ввода для операндов
    Label l1,l2;                    // Подписи для полей ввода
    DropList operation;             // Выпадающий список операций
    Label l3;                        // Подпись для списка
    Button compute;                 // Кнопка «вычислить»
    Label result;                   // Надпись для результата

    MyWindow(){
        Title("Hello world!").MinimizeBox().MaximizeBox()
        .Sizeable().SetRect(0, 0, 300, 230);

        Add( l1.SetLabel("Первое число").
            LeftPosZ(10,STDSize).TopPosZ(10,30) );
        Add( val1.HSizePosZ(90, 10).TopPosZ(15, 20) );

        Add( l2.SetLabel("Второе число").
            LeftPosZ(10,STDSize).TopPosZ(40,30) );
        Add( val2.HSizePosZ(90, 10).TopPosZ(45, 20) );

        Add( l3.SetLabel("Операция").
            LeftPosZ(10,STDSize).TopPosZ(70,30) );
        Add( operation.HSizePosZ(90, 10).TopPosZ(75, 20) );

        Add( compute.SetLabel("Посчитать!").
            HSizePosZ(10, 10).TopPosZ(105, 20) );

        Add( result.SetLabel("Результат:").
            HSizePosZ(10, 10).TopPosZ(135, 20) );
    }
};

GUI_APP_MAIN {
    MyWindow w;
    w.Run();
}
```

Легко заметить, что кириллица поддерживается без всяких проблем. Выпадающий список пока пуст. Заполним его:

```
operation.Add("+").Add("-").Add("*").Add("/") .SetIndex(0);
```

Функция `SetIndex` устанавливает выбранной первую операцию «+», т.к. по умолчанию ни один элемент не выбран. Нужно сказать, что элементами выпадающего списка могут быть не только строки, а любые пары «ключ-значение». В данном случае используется простейший вариант, когда задаются только значения, а ключи генерируются автоматически.

На этом создание интерфейса можно считать законченным. В следующей части статьи я опишу реализацию логики нашей программы. Именно там в полной мере проявится «магия» `U++`.

Семен Есилевский
(yesint4@yahoo.com)

Право выбора, или Открытые форматы. Часть 1: MNG и APNG

Много можно говорить о том, что открытые форматы – альтернатива частным, коммерческим и так далее, но любая альтернатива «работает» лишь в случае, когда люди о ней осведомлены. Вот был бы человеку предложен выбор, а человек волен решать, что ему больше по вкусу... Од-

нако в современном мире технологий корпорации навязывают свои правила игры. Правила эти просты: никакого выбора или ограниченный выбор (по вкусу производителя). Например, программа-диктофон в вашем мобильном телефоне умеет записывать в MP3, однако не дает возможности выбрать битрейт,

а по умолчанию пишет в низком. Более того, она могла бы давать записывать в WAV, т.е. без сжатия, но в этом случае – о ужас! – телефон уже залез бы в коммерческую нишу тех «отдельных», карманных диктофонов, которые обладают чудесной способностью записывать в WAV.

Посмотрите: развитие коммерческого ПО (да и железа) фактически остановилось. Увеличиваются только размеры: растут гигабайты дополнительных клипартов, кистей, шаблонов, звуков. Пара броских, однако никому не нужных новых функций, да возросшие системные требования. Зачастую выходит и так: новый продукт выпускается изначально урезанным, а потом к нему в качестве «новинок» добавляют то, чего не было, однако давно могло быть. Раньше, если выпускалась коммерческая программа, то, как правило, был только один её вариант, сразу со всеми функциями. Сейчас рынок программ подобен рынку мобильных телефонов: есть множество вариантов одной версии, функции которой раскиданы по этим вариантам. И так, функции при таком подходе стоят денег. Поддержка форматов – тоже. Разница в цене от десятка до сотен долларов.

На первый взгляд картина может показаться черно-белой – эдакое противостояние корпораций любым открытым форматам. Мол, корпорации просто не обращают на них внимание и стараются выжать доллар из каждого байта своих закрытых и запатентованных технологий. Но давайте попробуем разобраться, почему открытые форматы, альтернативные существующим, столь медленно продвигаются в общество. На конкретных примерах.

MNG – подрезанный на корню

Хотел указать в заголовке «убитый в зародыше», но рука не поднялась. Ведь MNG жив, но кто о нем знает, кто использует?..

Было светлое, революционное время, когда на каждом углу больших городов уличные проповедники кричали, что формат GIF – это зло, ибо там используется LZW-сжатие, защищенное патентом. К тому времени уже существовала некоторая ему замена – PNG, который не поддерживал анимацию, но мог хранить данные как в формате индексированного цвета, с использованием палитр, так и полноцветные изображения. PNG был поднят как знамя и раскручен – во всяком случае на сайтах, относящихся к свободному ПО и подобным, баннеры вы встретите чаще всего именно в формате PNG. Повторюсь, PNG не поддерживал анимацию, однако у него был брат – MNG. MNG мог содержать в себе кадры двух форматов: PNG и JNG. Более того, MNG был оснащен функциями спрайтовой анимации, включая в себя команды по копированию и перемещению встроенных объектов (с учетом альфа-канала), зацикливание и многие другие полезные для анимации вещи.

Проповедники спросили у мира: «Вы хотите анимацию?». И сами же ответили: «Получите MNG!». В то время MNG поддерживался в таких браузерах, как Mozilla и, кажется, Konqueror. Казалось бы, кто, как ни набирающая популярность Mozilla, может послужить для раскрутки формата? Но вдруг происходит любопытная вещь. Вернее, не совсем вдруг, а незадолго до истечения в США срока действия патента на LZW – в 2003 году (в большинстве других стран этот срок истек годом позже).

В феврале 2003 года в недрах разработки Mozilla поднимается вопрос об удалении поддержки MNG. Подробности изложены в историческом документе на https://bugzilla.mozilla.org/show_bug.cgi?id=195280, а я опишу дело вкратце. Представитель IBM (T. Rowley) предложил удалить поддержку MNG, поскольку Mozilla с ней занимала на пару сотен килобайт больше, анимированные GIF и Flash покрывали возможности MNG, MNG не входит в число форматов, рекомендуемых W3C, MNG вообще мало кто использует. Сразу в обсуждении возникли возра-

жения – одни возражения! MNG оказался популярен на Востоке. Библиотеку MNG можно было использовать вместо библиотеки поддержки PNG, т.к. первая фактически «держит» оба формата, и удалось бы сэкономить пресловутую сотню-другую килобайтов. Разработчик MNG (Gerard Juyn), описав интересные возможности в новой, грядущей версии MNG, библиотеку поддержки которой он хотел выпустить в ближайшее время, написал на той же странице обсуждения, что он чувствует себя так, будто три года его жизни спускают в унитаз, но если это пойдет на благо проекта (Mozilla), то он ничего не может сделать. Также сообщил, что говорил с разработчиками IE и они вроде бы хотели включить поддержку MNG в свой продукт в ближайшем будущем. Что до W3C и их рекомендаций, то W3C в то время была занята продвижением «своего» формата, SVG – в некотором роде конкурента MNG.

Март 2003 года. Возражения против удаления MNG продолжают. Но создатель темы обсуждения отвечает: новая версия libmng увеличивает декодер MNG в Mozilla на 33 424 байта под Linux и 25 056 байтов под Win32. Создается впечатление, что речь идет о тех достопамятных временах, когда «640 килобайт должно быть достаточно для каждого». Однако разработчик MNG соглашается на всё – лишь бы оставить поддержку MNG в Mozilla. Он может изменить API MNG нужным образом, если это поможет; готов серьезно поработать над уменьшением размера кода библиотеки, но какой в этом смысл, если поддержка MNG будет выкинута? И далее он спрашивает: «Как вообще будет принято это решение: демократическим голосованием или просто решением одной личности (или небольшой группы)? Покамест я насчитал 7 голосов против удаления, 2 – за, 1 – воздержавшийся». Ответом ему, в тот же день, служит патч для удаления.

Май 2003 года. Всё еще обсуждают. Люди всё еще против удаления MNG. Один программист (Glenn Randers-Pehrson) сообщает, что провел некоторую работу по уменьшению размера libmng. Через несколько дней ответом будет жесткое: «Patch checked in. libmng removed from the tree». И так, патч удаления MNG из дерева исходников Mozilla был применен 3 июня 2003 года. Снова люди возмущаются, но их утешают: Ну что вы, еще несколько версий у вас будет поддержка MNG, версии без MNG появятся немного позже, не так всё страшно. И утихло, успокоилось... Только почти 4 года спустя в той ветке обсуждений появилась умная запись: ведь во время удаления MNG из Mozilla, PNG и MNG были по сути на одном уровне популярности – малоизвестном. За четыре года PNG «вышел в свет», и то же могло произойти с MNG.

Какова же его судьба после 2003? Сначала – еще пара слов о Firefox/Mozilla. Вплоть до 2007 года энтузиастами выпускался патч Mngzilla с поддержкой MNG/JNG. Я искал дополнение поддержки MNG для современного Firefox, однако не нашел. Из известных мне браузеров MNG «из коробки» поддерживается только в Konqueror. Вроде бы MNG поддерживается в каких-то вариантах мобильного браузера NetFront. Я проверил в своем телефоне – увы (хорошая страничка для проверки находится на <http://pmt.sourceforge.net/opussum>). Плагин поддержки MNG есть в библиотеке Qt, а учитывая её растущий вес (читайте двояко: в мегабайтах и в распространении), можно предположить, что MNG в любой момент может стать востребованным по крайней мере на мобильных платформах, но... Сама библиотека libMNG в 2007 году перешла к новому ведущему, и с тех пор нет никаких обновлений. MNG не имеет своего зарегистрированного MIME-типа, традиционно (для галочки) поддерживается многими просмотрщиками изображений и используется в темах для телефонов Sony Ericsson.

Вспомним еще раз одну из причин, по которой из Mozilla убрали поддержку MNG. MNG своими возможностями «залезал» на территорию Flash и SVG, «дублировал» их – стало быть, являлся возможным конкурентом. А вот PNG ни с кем особо не конкурировал – это не боевой, а «мирный» формат. PNG был нужен тому же SVG для хранения растровых данных. PNG не покушался на вотчину GIF и Flash в плане анимации. Наконец, PNG не прирос корнями и 8-битной графике 80-х/начала 90-х. Тем не менее, PNG не потеснил GIF в области баннеров, и уж тем более не пересекся с Flash. PNG развивается в своих нишах – кнопки, полноцветные обои, изображения в свободных играх.

APNG – пришедший на смену

Любопытно, что происходило в недрах разработки Mozilla спустя год после удаления поддержки MNG. Если почитать обсуждение, относящееся к удалению, то мы увидим, что был нанесен удар по такой части интерфейса, как *throbber*'ы – анимированные элементы с логотипами или разными отображениями хода выполнения операции, т.е. всякие вращения в кружках и тому подобное. Ведь оказалось, что в темах к Mozilla многие *throbber*'ы были сделаны именно на MNG. Итак, для нужд Mozilla в области поддержки *throbber*'ов (не подходил GIF, а MNG убрали) в 2004 году изобретают новый формат – APNG (<http://animatedpng.com>). Новый, но на основе старого доброго PNG. С тем же расшире-

нием и MIME-типом. Задумывалось так: если браузер или другая программа не поддерживает анимационные навороты APNG, то отображает его как статичное изображение. Так собственно и происходит.

Поддержка APNG была встроена в Firefox в 2007 году. Группа разработчиков PNG не принимает APNG, поэтому основная на сегодня библиотека поддержки PNG – libPNG – не поддерживает APNG. Ведь PNG задумывался как формат статичных изображений и, по мнению разработчиков PNG, анимационные надстройки над этим форматом просто разрушают его и усложняют жизнь программистам.

Поддержка APNG сейчас, однако, не столь велика, как у MNG, если считать разнообразные программы. Но браузеров, поддерживающих APNG, больше, чем в случае MNG. Я проверял APNG в Opera, Firefox, Chrome, Konqueror, Firefox – из них анимация была только в Opera и Firefox. Сейчас APNG – больше «внутренний» формат Firefox, я не видел реального использования APNG на случайных сайтах. На страницах с искусственными примерами – да, наблюдал, впечатляет. То же, что умеет и MNG, но меньше. Можно сказать, что функционально APNG – это урезанный MNG.

Петр Семилетов
(tea@list.ru)

QLandKarte GT как замена OziExplorer в GNU/Linux

И для кого не секрет, что, наверное, самой популярной средой для работы с GPS-навигаторами и вообще картографической информацией с целью навигации под Windows является OziExplorer. Большой плюс программы – её простота (осо-

бенно если сравнивать с различными GIS). Популярность Ozi стала причиной появления в интернете огромного количества геоинформации в его формате, в частности: путевые точки (Waypoints), треки (Tracks) и файлы привязок растровых карт (Map-файлы).

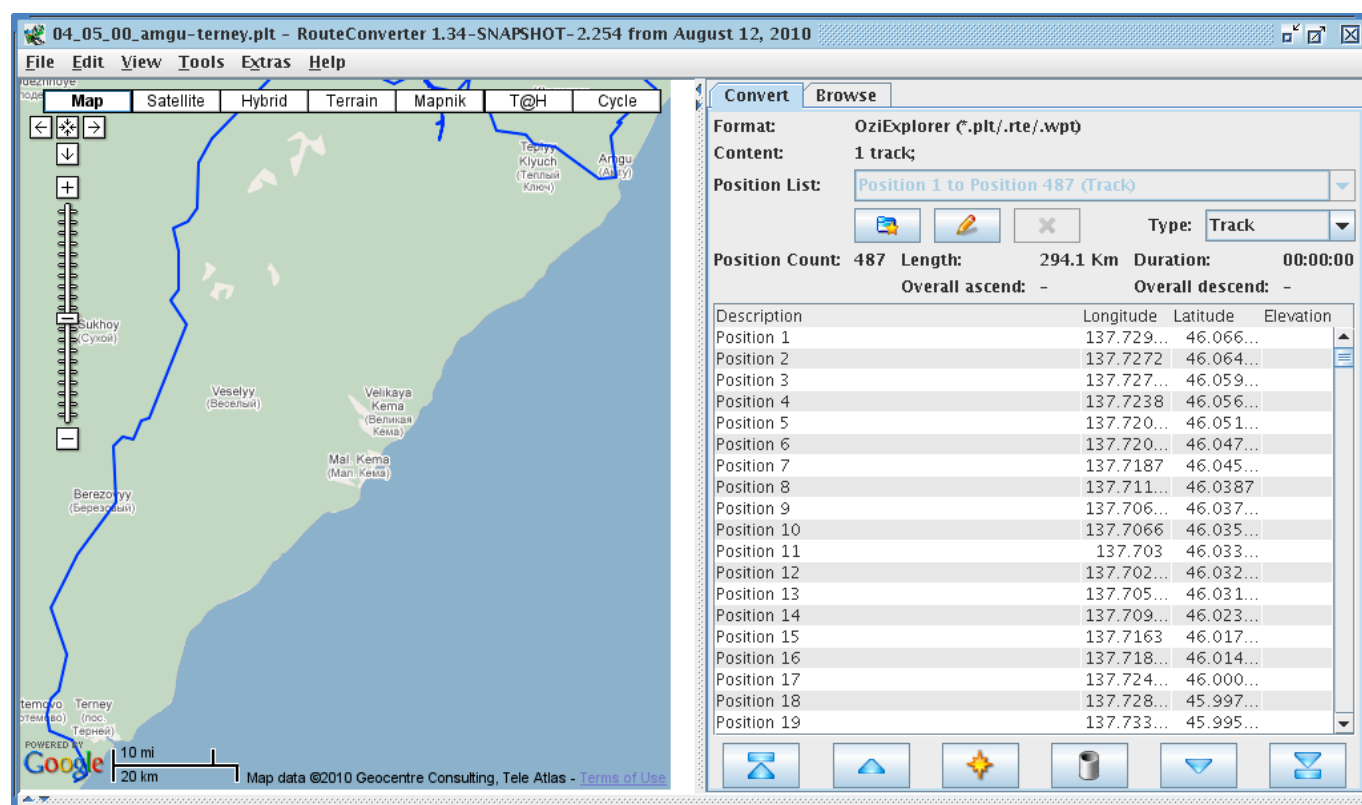


Рисунок 1. RouteConverter с загруженным треком

Естественным желанием становится использовать как-то всю эту прорву данных и в GNU/Linux. Но на пути становится первая преграда: OziExplorer существует только в версии для Windows. Выручить тут могут два фактора:

- ☑ Все версии Ozi прекрасно запускаются под Wine (но для легального и полноценного использования нужно купить лицензию, которая стоит около 90 USD).
- ☑ Несмотря на закрытость самой программы, авторы сделали свои форматы открытыми (простые текстовые файлы) и даже дали описание в справочном руководстве, за что им огромное спасибо (посмотреть описание можно, например, на <http://www.realbiker.ru/OziExplorer/fileformats.shtml>).

Логично, что первый вариант, хоть и имеет право на жизнь, но не является достаточно изящным решением. Поэтому я предпринял попытку найти готовое программное обеспечение, которое может работать с форматами OziExplorer. Сделаю одну оговорку: к сожалению, вопрос непосредственного сопряжения с устройствами GPS рассматривать не буду ввиду отсутствия подобных устройств в зоне моей доступности.

Работа с картами, атласами, геоданными и GPS

По совокупности характеристик, таких как простота и функциональность, в данную категорию попадает, пожалуй, только одна программа – QLandKarte GT (<http://www.qlandkarte.org>). Есть ещё версия «М», которая предназначена для различных карманных устройств. По сути эта программа – прямой аналог OziExplorer. Она позволяет производить привязку растровых карт (например, сканированных атласов или других источников), в качестве основного формата которых используется GeoTiff. Кроме того, поддерживаются векторные карты Garmin в формате tdb (формат img требует дополнительных действий от пользователя, почитать об этом на английском языке можно на <http://sourceforge.net/apps/mediawiki/qlandkartegt/index.php?title=FAQ>). Вместе с тем,

она позволяет загружать геоданные (точки, треки) из файлов на диске (набор форматов ограничен, но главное, что есть GPX (<http://ru.wikipedia.org/wiki/GPX>), подробнее – ниже). Эти же данные она может экспортировать обратно в файлы, а также читать из GPS-устройства и загружать в него. Ещё одной важной и нужной особенностью программы является возможность создавать атласы, в которых указываются для разных масштабов различные растровые карты. Подробно на описании этой программы останавливаться не буду, так как информация есть в интернете (хотя и не очень много). Основной список источников, которые следует прочитать перед началом работы, приведены в конце статьи.

Во многих дистрибутивах данную программу можно найти в репозиториях. Например, в случае Arch Linux она есть в AUR в виде двух пакетов: qlandkartegt и qlandkartegt-svn. Также стоит обратить внимание на комплект драйверов для работы QLandKarte с GPS Garmin (там же в AUR): garmindev. Установка не должна занять много времени и сил. В общем, всё вроде хорошо – вот только ни привязанные в OziExplorer карты, ни геоданные в его формате QLandKarte читать не умеет. Когда я это обнаружил, чуть было не расстроился, но, поискав в интернете, нашел решения нужных мне вопросов.

Карты

На сайте разработчиков QLandKarte GT есть небольшая сводка о том, где брать карты, у меня в блоге представлена краткая выборка [1]. Одним из пунктов идет конвертация карт OziExplorer в формат GeoTiff при помощи скрипта на Python – ozi2geotiff.py. Сам скрипт можно достать на <http://sourceforge.net/projects/ozi2geotiff>, но разработчики не выкладывают пакеты и доступ возможен только через SVN. Для пользователей Arch Linux я поместил в AUR правила для сборки пакета для этого скрипта, так что для них сложностей быть не должно (пакет называется ozi2geotiff-svn, там же лежит и скрипт ozi2gpx, речь о котором пойдет ниже).

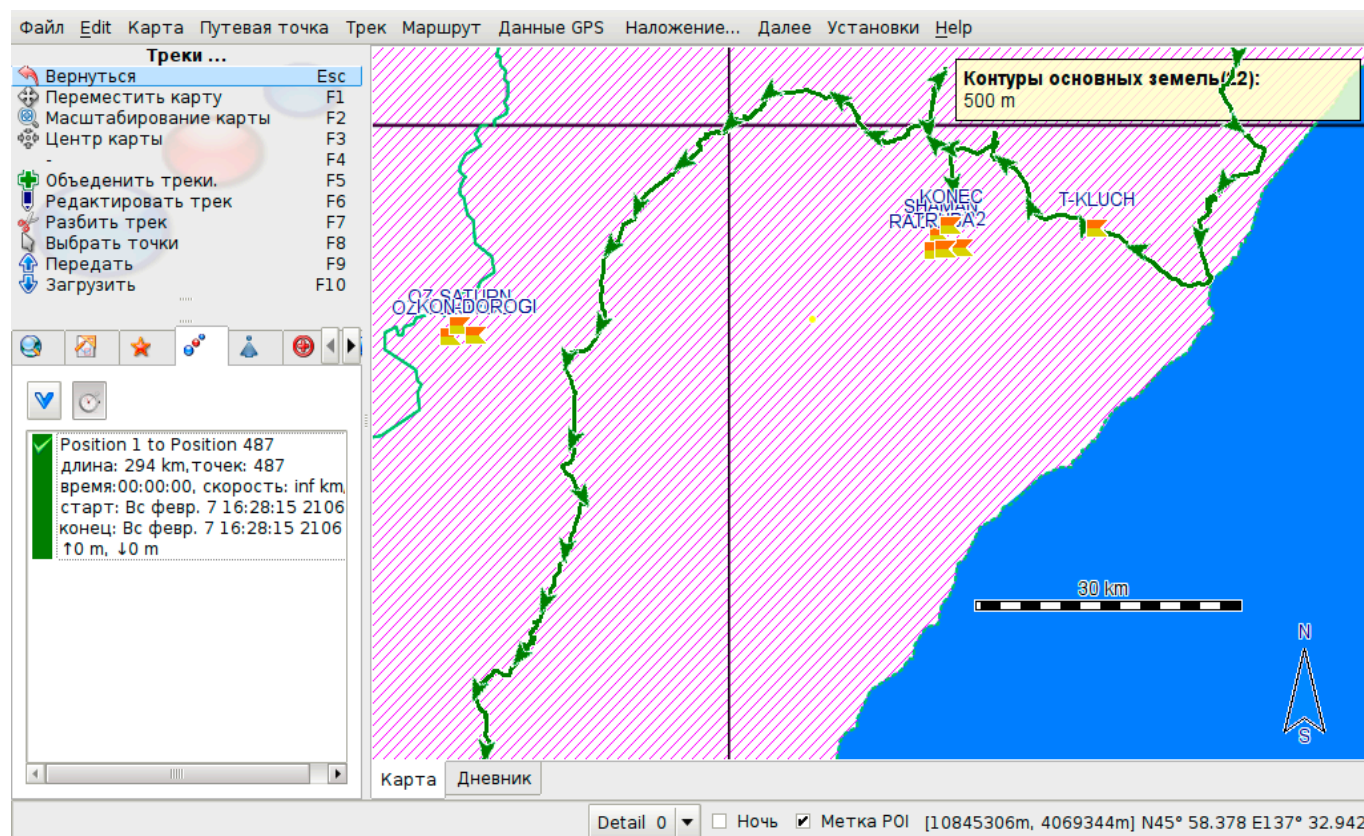


Рисунок 2. Загруженные векторные карты Garmin и трек

После сборки и установки пакета, конвертер вызывается командой:

```
$ ozi2geotiff <params>
```

Остальные могут скачать скрипт таким образом:

```
$ wget -O ozi2geotiff.py http://ozi2geotiff.svn.sourceforge.net/ \
viewvc/ozi2geotiff/trunk/ozi2geotiff.py?revision=HEAD
```

Перед запуском не забудьте проверить, что у вас установлены все необходимые пакеты:

- ☒ python;
- ☒ libgeotiff;
- ☒ gdal;
- ☒ imagemagick.

Запуск осуществляется так:

```
$ python ozi2geotiff.py <params>
```

Если не указывать параметры – будет выведен экран справки. Использовать скрипт достаточно просто:

```
$ ozi2geotiff -i karta.map --in-img=karta.gif -o karta.tiff
```

Можно ещё добавить параметр -v для более подробного вывода. Что же за параметры мы использовали?

- ☒ **-i karta.map** – задаем файл привязки, обязательный параметр;
- ☒ **--in-img=karta.gif** – задаем растр, к которому применяется привязка (формат GIF здесь только для примера), параметр не обязательный, если путь к растровой карте в map-файле задан корректно (в этом случае он берется оттуда);
- ☒ **-o karta.tiff** – карта в формате GeoTiff, её мы уже сможем загрузить в QLandKarte GT и работать.

Справедливости ради стоит отметить, что существует ещё один скрипт для подобного преобразования, написанный на Ruby. Вот только работать у меня он отказался (и даже при всей своей нелюбви к Python, я не стал разбираться в причинах; для меня главное – конечный результат). Скачать этот скрипт можно на странице <http://sites.google.com/site/osmbelarus/Home/files>. Хочется отметить, что скрипт ozi2geotiff.py не является финальным продуктом и может содержать ошибки, но автор – русскоговорящий, обо всех найденных недочетах можно смело писать ему на почту.

Посмотрим, что можно сделать из карт для Garmin в формате img (для Приморского края, к примеру, они доступны на <http://john.bdk.com.ru/gpsmaps/pkmap2/main.htm>). Напрямую эти файлы загрузить не получится – нужно собрать их в альбом (файл с расширением .tdb). Для этого нам поможет программа mkgmap (<http://www.mkgmap.org.uk/page/main>). В случае Arch Linux она имеется в AUR: mkgmap-svn (для работы нужна Java).

Единственный нюанс – файлы нужно будет переименовать в вид #####.img, где ##### – 8 цифр, внутреннее представление имени файла. Как получить эту цифру правильно, я не понял, поэтому сделал в лоб:

```
$ strings karta.img | grep 'RGN'
```

Записей будет несколько, но смысл понятен. Затем все эти файлы собираем в одной директории и выполняем команду:

```
$ mkgmap --input-file=00000001.img ... \
--input-file=0000000n.img --tdbfile
```

В результате у нас дополнительно появятся два файла osmmap.img и osmmap.tdb. Загружая карту, выбираем osmmap.tdb, затем будет предложено выбрать .img-файл, выбираем osmmap.img, все нужные загрузятся автоматически.

Естественно, всё это вручную делать утомительно, поэтому я подготовил скрипт img2tdb:

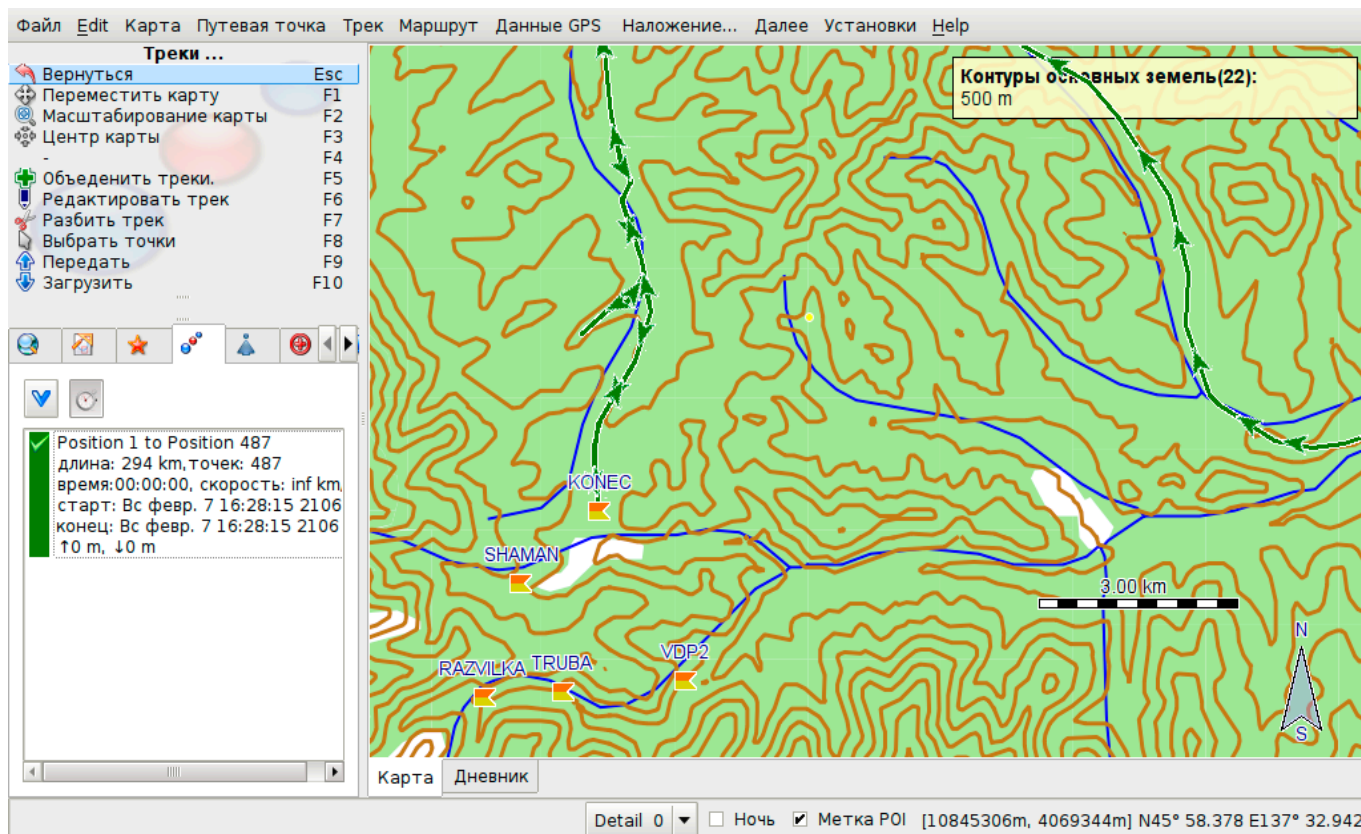


Рисунок 3. Загруженные векторные карты Garmin и трек с деталями

```
#!/bin/sh

use()
{
  cat << _EOF_
  Use: $0 <out-dir> <1.img> ... [n.img]
  _EOF_
}

if [ -z "$2" -o "$1" == "--help" ]; then
  use
  exit
fi

OUT="$1"
IMG=""

shift

cwd=`pwd`
mkdir -p "$OUT"
cd "$OUT"

echo "Work dir: `pwd`"

while [ -n "$1" ]
do
  st=`echo "$1" | grep '^/'`
  if [ -z "$st" ]; then
    in_name="$cwd/$1"
  else
    in_name="$1"
  fi

  int_name=`strings "$in_name" | grep 'RGN' | \
    grep -v 'GARMIN' | head -1 | awk -F'RGN' '{print $1}'`
  out_name="$int_name.img"
  #

  echo "Copy '$in_name' -> '$out_name'"
  cp "$in_name" "$out_name"

  IMG="$IMG --input-file=$out_name"

  shift
done

echo "Run 'mkgmap' and prepare TDB map album"
mkgmap $IMG --tdbfile

echo "Done"
```

После сохранения скрипт надо сделать исполняемым:

```
$ chmod +x img2tdb
```

Пользоваться им просто:

```
$ img2tdb <out dir> <file1.img> ... [fileN.img]
```

Здесь первый параметр – директория, в которой будут сохраняться результаты работы, а дальше идут отдельные img-файлы, которые объединяются в альбом.

Можно указывать и маску:

```
$ img2tdb /tmp/map-out *.img
```

В дальнейшем планирую добавить этот скрипт в пакет ozi2geotiff.

Геоинформация: треки, точки, маршруты

Настала очередь геоинформации. Выше я писал, что QLandKarte GT поддерживает не так много форматов, среди которых нет форматов OziExplorer, но есть GPX, поэтому задачу ставим так: преобразовать форматы Ozi в GPX. В отличие от карт, набор инструментов тут побогаче – начнем с программы GPSTabel (<http://www.gpsbabel.org>), которая имеется в community-репозитории Arch Linux (похоже, единственная из этого обзора, которую не нужно собирать). Итак, GPSTabel – программа для закидки треков, путевых точек и т.п. на GPS-устройства, а также для получения этой информации с устройства и конвертации данных в различных форматах. Для нас же главное, что программа поддерживает форматы файлов OziExplorer и GPX. Синтаксис команды вновь тривиален:

```
$ gpsbabel -i ozi -f track.plt -o gpx -F track.gpx
$ gpsbabel -i ozi -f dots.wpt -o gpx -F dots.gpx
```

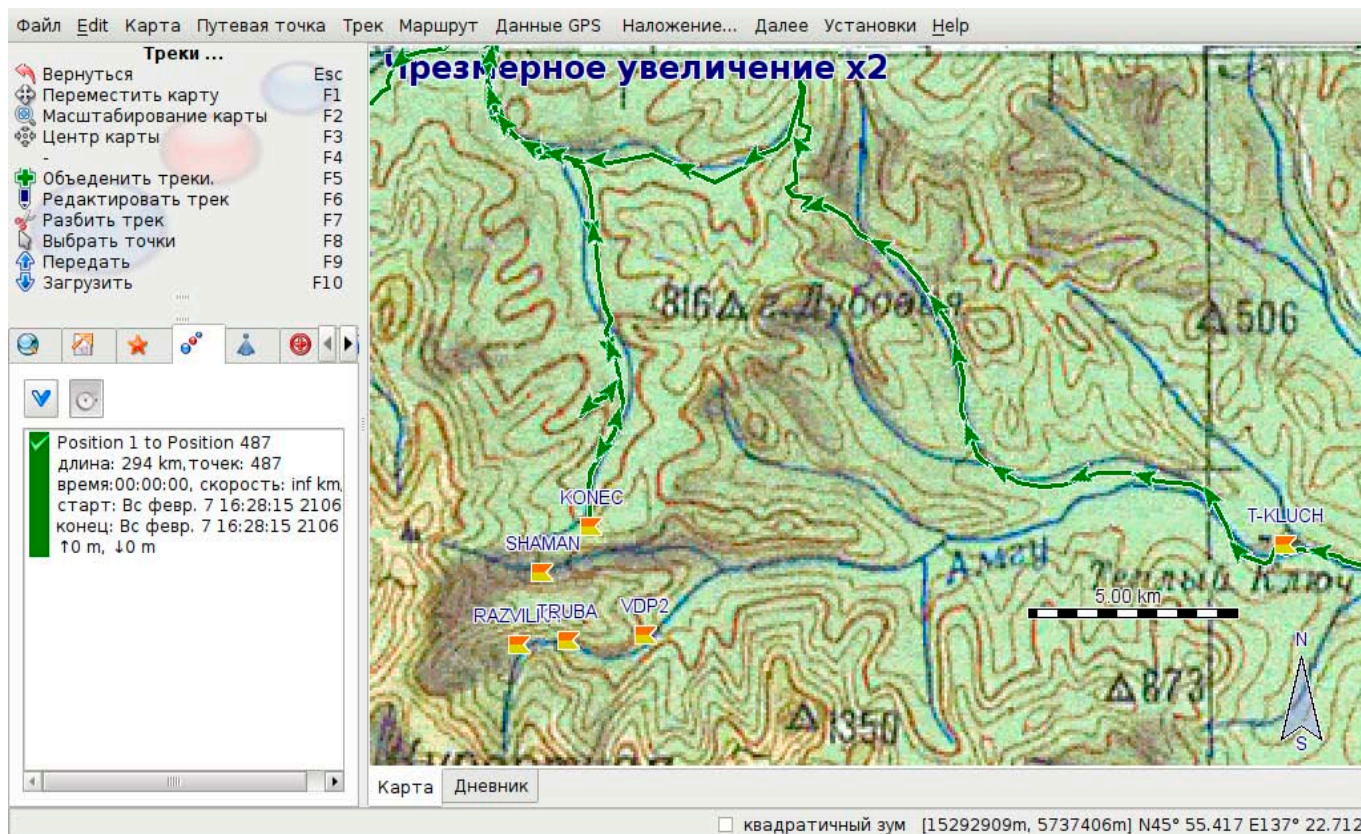


Рисунок 4. Загруженная растровая привязанная карта (GeoTiff) этого же района и трек

Небольшие пояснения:

- ☑ **-i** – формат входного файла;
- ☑ **-o** – формат выходного файла;
- ☑ **-f** – входной файл;
- ☑ **-F** – выходной файл.

Остальные опции доступны при вызове:

```
$ gpsbabel --help
```

Чтобы каждый раз не писать эти команды, я поместил простой скрипт `ози2gpx` в пакет `ози2geotiff`. Вызывается он так:

```
$ ози2geotiff <OziExplorer file: plt, wpt> <out.gpx>
```

Всё просто и лаконично. Для данной программы в AUR есть графическая оболочка под названием `gebabel` (<http://gebabel.sourceforge.net>). Так что, если командная строка вас пугает, можете воспользоваться её возможностями.

Вторая программа для тех же нужд называется `RouteConverter` (<http://www.routeconverter.de/en>), требует для своей работы Java, имеет графический и консольный интерфейсы. Пакет для Arch Linux я подготовил и выложил в AUR. У этой программы есть одна так называемая *killer feature*: она подключается к GoogleMaps и при загрузке трека или путевой точки рисует на карте оные и, соответственно, позиционирует положение на карте в нужное время – это может быть очень удобно для быстрого разбора и сортировки треков.

Как при помощи неё конвертировать? Тоже просто:

- ☑ File → Open... – указываем формат и открываем файл;
- ☑ File → Save as... – указываем формат и сохраняем файл.

Есть и вариант программы с интерфейсом командной стро-

ки, в пакете для Arch Linux они вызываются соответственно `routeconverter` и `routeconverter-cli`.

Что в итоге?

В результате, мы имеем базовый набор утилит для нормальной работы с картами под Linux, можем использовать привязки `OziExplorer`, его файлы с путевыми точками, треками с минимальными трудозатратами. На **рис. 2-4** представлены снимки экрана `QLandKarte GT` с загруженными данными для предстоящего похода.

1. Картографические приложения, а также визуализация – http://hatred.homelinux.net/wiki/zhurnal/2008-10-24_13.49_kartograficheskie_prilozhenija_a_tak_zhe_vizualizacija.
2. Работа с `QLandKarte GT` – <http://sites.google.com/site/osmbelarus/Home/manuals/qlandkarte-gt>. Это общая страница для размещения информации, но с 2008 года новых источников не появилось. Обзор интерфейса – <http://sites.google.com/site/osmbelarus/Home/manuals/qlandkarte-gt/qlandkarte-main-window>. Привязка растровых материалов – <http://sites.google.com/site/osmbelarus/Home/manuals/qlandkarte-gt/qlandkarte-make-geotiff>. Создание коллекций (атласов) – <http://sites.google.com/site/osmbelarus/Home/manuals/qlandkarte-gt/qlandkarte-make-collections>.
3. Подготовка карты для `qlandkarte` – http://takeoff.mipt.ru/wiki/index.php?title=%D0%9F%D0%BE%D0%B4%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%BA%D0%B0_%D0%BA%D0%B0%D1%80%D1%82%D1%8B_%D0%B4%D0%BB%D1%8F_qlandkarte.
4. Установка `QLandKarte GT` для работы с GPS Garmin EtrexVistaHCx – http://community.livejournal.com/ru_nix/11337.html.
5. Garmin GPSmap 60Cx: подключение и использование в Linux – <http://sovety.blogspot.com/2008/06/garmin-gpsmap-60cx-linux.html>.

Александр Дроздов
(hatred@inbox.ru)

«Open Source» приглашает к сотрудничеству!

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

Электронное приложение «Open Source» всегда открыто для сотрудничества с новыми авторами, с читателями и их конструктивными предложениями по улучшению издания, обогатившей 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;

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>. Свяжитесь с редакцией можно по электронной почте osa@samag.ru.

P.S. За статьи мы платим.

Подписные индексы:

20780*

+ диск с архивом статей

81655**

без диска

по каталогу агентства
«Роспечать»**88099***

+ диск с архивом статей

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» (МД-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