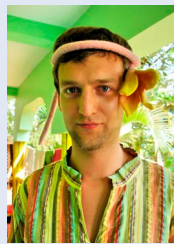


Колонка главного редактора



Минувший июнь запомнился не прекращающимся ни на минуту шумом вокруг мобильных устройств, усиленным на этот раз, помимо прочего, выпуском первого устройства с webOS

(Palm Pre), открытием некоторых компонентов webOS и анонсом очередного Android-смартфона от HTC (Hero). Другим заметным событием стала череда крупных релизов известных Open Source-проектов. Все они пришлись на самый конец месяца, а подробности о некоторых из них смотрите в новостях.

Тем временем, к финальной стадии подошел конкурс на лучший видеоролик про Linux, организованный весной этого года компанией Novell. Жюри отобрало 14 роликов, которые лучше других отвечают на главный вопрос – «Что вы делаете с Linux?». Теперь пользователям предлагается выбрать победителей путем голосования на странице <http://howdoyoulinux.com/#watchvideos>. Среди финалистов – сразу несколько роликов от русскоязычных энтузиастов. Причем любопытно, что в двух из них заметно пересекаются, как сюжеты, так и названия: «Living with Linux» и «Life with Linux». Не удержусь упомянуть и том, что приложил значительные усилия для создания первого из них. Окончательные результаты будут объявлены 1 августа, но уже сейчас можно с уверенностью сказать, что наши соотечественники выступили достойно.

Главный редактор
Дмитрий Шурупов
(osa@samag.ru)

«Open Source»

электронное приложение к журналу

«Системный администратор»

№46, 3 июля 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

Для улучшения удобства Ubuntu Linux собирают недостатки системы

Разработчики популярного Linux-дистрибутива Ubuntu объявили о запуске инициативы One Hundred Paper Cuts («сотня порезов от бумаги»). Ее цель – собрать те мелочи, которые мешают пользователям удобно работать в системе, и, основываясь на этих данных, улучшить ощущения от использования Ubuntu Linux.

Как заявляется на странице One Hundred Paper Cuts, этим «проектом руководит команда Canonical Design and User Experience с целью улучшения ощущений от использования Ubuntu за счет определения 100 небольших проблем, которые испытывают пользователи при работе с системой, и их устранения».

В качестве примеров таких проблем приводятся следующие: возможность задавать стандартное в системе имя файла печати в PDF вместо стандартного и «технического» «~/output.pdf»; улучшение работы переключения между рабочими пространствами через тачпад; увеличение ширины всплывающих уведомлений (notify-osd); упрощение терминологии в пунктах меню для извлечения компакт-дисков и других носителей (понятия «unmount» и «eject» затруднительны для неопытных пользователей).

Проект нацелен собрать 100 таких незначительных проблем в юзабилити системы и устранить их к грядущему релизу Ubuntu Linux 9.10, который состоится в октябре этого года.

Nokia и Intel объявили о запуске проекта ConnMan

Компании Intel и Nokia объявили о совместном запуске Open Source-проекта ConnMan – программного обеспечения для управления сетевыми подключениями на мобильных устройствах.

ConnMan предоставляет демона для управления сетевыми подключениями и является модульной системой, которую легко расширять с помощью плагинов. Распространяется под лицензией GNU GPLv2.

ConnMan уже вошел в состав бета-версии нового релиза (2.0) мобильной Linux-системы Moblin, ориентированной на использование на таких ультрапортативных устройствах, как MID и нетбуки, и изначально разработанной Intel, а теперь управляемой некоммерческой организацией Linux Foundation. Кроме того, этот менеджер сетевых подключений будет

официально поддерживаться операционной системой Maemo, разрабатываемой Nokia на базе Debian GNU/Linux для своих интернет-планшетов.

Официальный анонс ConnMan от компаний Intel и Nokia, которые намерены поддерживать и развивать этот проект на благо всему мобильному Open Source-сообществу, ознаменовался запуском сайта connman.net, призванного объединить разработчиков программы.

Анонсирован проект новой реализации libc – libposix

Хенрик Алмейда (Henrique Almeida) анонсировал в почтовых рассылках для разработчиков Ubuntu Linux свой проект по созданию новой реализации базовой библиотеки libc – libposix.

По словам автора, у проекта несколько целей. Первая из них – создать с нуля библиотеку, строго соответствующую POSIX 2008, которая унифицирует реализацию libc на всех UNIX-платформах. Другие цели «не менее важны, чем первая», и определены так: чистый и удобный в изучении код, простая поддержка кодовой базы, дружелюбное сообщество разработчиков и пользователей.

Судя по сайту libposix, публичная работа над библиотекой началась около месяца назад, а первый коммит в репозитории был зафиксирован 31 мая, но ориентированный на широкую аудиторию анонс проекта состоялся только теперь. При этом автор называет Ubuntu Linux основной средой разработки, поэтому в первую очередь он обратился за помощью именно к разработчикам Ubuntu.

Palm опубликовала пакеты Open Source-компонентов webOS

Компания Palm Inc. запустила сайт своих Open Source-разработок – opensource.palm.com – и опубликовала на нем пакеты Open Source-компонентов своей мобильной операционной системы webOS, используемой в новом смартфоне Palm Pre.

На странице Open Source-пакетов Palm представлены архивы с оригинальными версиями компонентов с открытым исходным кодом, задействованных в webOS, и патчи, которые были добавлены разработчиками Palm для их использования в webOS.

Среди них, помимо ядра Linux, можно найти, например, библиотеку Purple, используемую IM-клиентом Pidgin, компактный набор базовых консольных UNIX-

средств BusyBox, звуковую подсистему ALSA, мультимедийный фреймворк GStreamer с плагинами, а также такие системные компоненты, как binutils, sysvinit и udev. Для некоторых пакетов выложены только оригинальные архивы, что означает их включение в webOS без дополнительных патчей.

Представлен новый смартфон HTC с Android – HTC Hero

24 июня на пресс-конференции в Лондоне компания HTC представила свой очередной смартфон, работающий под управлением мобильной Linux/Java-системы Android, – HTC Hero.

Смартфон отличается от предыдущих устройств с Android наличием нового интерфейса HTC Sense, благодаря которому «персонализация достигла небывалого уровня» за счет еще большей ориентированности на виджеты вместо традиционных меню. Новые возможности создания профилей доступны с помощью Scenes. Фотоальбом получил интеграцию с онлайн-сервисами Flickr и Facebook, а поиск – с Twitter.

Технические характеристики новинки:

- ☑ габариты: 112 x 56,2 x 14,35 мм; вес: 135 граммов;
- ☑ поддержка HSPA/WCDMA 900/2100 МГц, GSM/GPRS/EDGE 850/900/1800/1900 МГц;
- ☑ 3,2-дюймовый TFT-LCD тачскрин с поддержкой разрешения 320x480 (HVGA);
- ☑ процессор Qualcomm MSM7200A с тактовой частотой 528 МГц;
- ☑ 288 Мб ОЗУ (RAM) и 512 Мб ПЗУ (ROM);
- ☑ 5-мегапиксельная камера с автофокусировкой;
- ☑ слот для карты памяти microSD (SD 2.0);
- ☑ GPS, Bluetooth 2.0 с EDR и A2DP, Wi-Fi;
- ☑ ExtUSB, стандартный аудиовыход (3,5 мм);
- ☑ трекболл с клавишей нажатия (Enter);
- ☑ литий-ионный аккумулятор емкостью 1350 мАч.

Поставки HTC Hero в Европу начнутся в июле этого года, в Россию и Азию – в августе, а в Северную Америку – до конца этого года. Официальная цена смартфона пока не называется.

Обновились популярные IDE: Eclipse 3.5 Galileo и NetBeans 6.7

В конце июня с разницей в несколько дней состоялись новые релизы двух популярнейших интегрированных сред разработки (IDE) с открытым исходным кодом: Eclipse 3.5 под кодовым названием

«Galileo» и NetBeans 6.7.

Релиз Eclipse Galileo назван самым крупным за всю историю этой IDE: в него входят 33 проекта и более 24 миллионов строк кода. В его создании участвовали более 380 человек из 44 различных организаций. Три главных направления, которыми руководствовались авторы при создании Eclipse 3.5: улучшение применения Eclipse в корпоративном секторе, инновации в области технологии моделирования с помощью Eclipse, расширение возможностей технологии запуска приложений EclipseRT. Среди новых возможностей, представленных в Eclipse Galileo, выделяются: обновленная поддержка 32- и 64-разрядных сборок интерфейса Mac OS X Cocoa; новый инструмент анализа памяти (Memory Analyzer) для анализа потребления памяти Java-приложениями; PDT (PHP Development Tools) 2.1 с поддержкой PHP 5.3; поддержка WikiText в Mylyn для редактирования и парсинга wiki-разметки; новая XSL-утилита для редактирования и отладки XSL; обновление Eclipse Equinox для поддержки промежуточной версии спецификации OSGi Release 4 (4.2); Xtext – новый проект Eclipse, предназначенный для создания предметно-ориентированных языков (Domain Specific Languages, DSL); фреймворк Connected Data Objects (CDO) для распределенных EMF-моделей, ориентированный на масштабируемость, транзакционность и персистентность.

Одним из главных новшеств в NetBeans IDE 6.7 стала интеграция с хостингом для Open Source-приложений Project Kenai, анонсированным компанией Sun в сентябре прошлого года. Из других изменений выделяются: в Maven – просмотрщик графа зависимостей библиотек, поддержка Java EE, Web Services, в PHP – поддержка PHPUnit, автодополнение SQL-кода в редакторе, в GlassFish – автодополнение для Glassfish v3 Prelude. Кроме того, в релизе NetBeans IDE 6.7 появились поддержка Groovy and Grails 1.1 из коробки и автоматическое дополнение для Groovy and Grails, удаленная отладка для Ruby and Rails, инструменты профилирования DLight для проектов C++, интегрированная поддержка популярных библиотек и утилит Qt, экспорт данных результатов профилирования в форматы CSV, HTML и XML, поддержка Java Card Platform 3.0 и включение в состав Java Platform Micro Edition Software Development Kit 3.0.

Вышел веб-браузер Firefox 3.5

Объявлен выпуск финальной версии популярнейшего веб-браузера с открытым

исходным кодом – Firefox 3.5.

Релиз браузера Firefox 3.5 основывается на новой, более быстрой, версии движка рендеринга веб-страниц Gecko 1.9.1, работа над которым велась в течение последнего года. Среди ключевых новшеств в Firefox 3.5 отмечаются следующие: переход на JavaScript-движок TraceMonkey, обеспечивающий значительно более высокую производительность JS-скриптов; режим конфиденциальной работы в сети (Private Browsing Mode); поддержка мультимедийных элементов, встроенных в веб-страницы в соответствии со спецификацией HTML5 (теги audio и video); поддержка геолокации (Location Aware Browsing); поддержка «родного» JSON (native JSON); поддержка скачиваемых шрифтов, селекторов JavaScript-запросов, трансформаций SVG.

Виртуальная машина VirtualBox обновилась до версии 3.0.0

30 июня состоялся релиз новой версии виртуальной машины с открытым исходным кодом, разрабатываемой при поддержке компании Sun Microsystems, – VirtualBox 3.0.0.

Три главных новшества в VirtualBox 3.0.0: поддержка SMP для гостевых машин (до 32 виртуальных CPU, только VT-x и AMD-V); экспериментальная поддержка использования в Windows-гостях приложений с Direct3D 8/9; поддержка OpenGL 2.0 для гостей Windows, Linux и Solaris.

Из прочих, менее значительных, изменений можно выделить поддержку RDP-клиента Windows 7, использование аккаунтов Sun Online в диалоге регистрации, поддержку интерфейсов IDisplay и IGuest в веб-сервисах, появление в графическом интерфейсе минипанели для переключения в полноэкранный режим, обновление внешнего вида диалогов настроек (в GUI), исправления для корректной компиляции модуля VirtualBox к Linux-ядру 2.6.31.

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

Обзор дистрибутива InfraLinux.org 9.04

Обзор одного из самых популярных дистрибутивов постепенно превращается в рутину: каждые полгода кто-то из авторов «Open Source» пишет, что же изменилось в дистрибутиве по сравнению с предыдущей версией. Хотелось бы немного исправить эту ситуацию и сделать обзор дистрибутива, родственного Ubuntu, но имеющего некоторые изменения по сравнению с оригиналом. На этот раз предметом рассмотрения стал InfraLinux.org (ILo) от небезызвестной компании «Инфра-Ресурс». Сразу скажу, что дистрибутив оставил хорошие впечатления: сказались как изменения в Ubuntu, так и небольшие доработки в ILo (в статье затрагиваются новшества обеих систем).

Скачать установочный образ в любом варианте можно на официальном сайте дистрибутива: <http://infralinux.org>. Там же можно найти всю интересующую информацию, а также посмотреть скриншоты. В данном обзоре использован LiveCD для архитектуры x86: <http://releases.infralinux.org/infralinux-releases/jaunty/infralinux-9.04-desktop-i386.iso>.

Обновление компонентов

Своевременное обновление компонентов и включение в дистрибутив свежих версий ПО – это важная задача. В данном случае все версии ПО перекочевали из Ubuntu в InfraLinux (за исключением OpenOffice.org):

- ☒ Linux-ядро 2.6.28;
- ☒ X Server 1.6;
- ☒ GNOME 2.26;
- ☒ GCC 4.3.3;
- ☒ glibc 2.9;
- ☒ Python 2.6.1;
- ☒ OpenOffice.org Pro 3.1.

Установка

Изменения в InfraLinux прослеживаются с самого начала, то есть от загрузочного меню. После вставки и загрузки с диска перед нами открывается меню установки дистрибутива, которое, кстати говоря, не предоставляет возможности выбора языка сразу (по-прежнему можно открыть диалог выбора языка), зато по умолчанию уже выбран русский язык. Компания «Инфра» не забыла и про логотипы, которые заменили стандартные лого Ubuntu. Режимы загрузки те же, что в Ubuntu: запуск в LiveCD, установка, проверка целостности диска, проверка оперативной

памяти и загрузка с жесткого диска. Лично я решил опробовать загрузку без установщика, после чего был встречен ошибкой X Server – в общем, ознакомиться с новым рабочим столом перед инсталляцией не получилось.

После перезагрузки мною был выбран режим установки, и тогда запустился уже знакомый инсталлятор Ubuntu, в котором вновь предварительно задан русский язык. Новая возможность – предлагаемая раскладка клавиатуры – пришлась очень кстати, поскольку больше нет необходимости останавливаться на этом шаге, а достаточно просто нажать Forward. Впрочем, тут и неприятная особенность: несмотря на то что все кругом до неприличия на русском языке, две кнопки так и остались Back и Forward. Помимо них не переведены на русский названия городов в выборе часового пояса из предыдущего шага, и об этой настройке расскажу подробнее.

Те, кто постоянно устанавливают Ubuntu с «нуля», могли проследить развитие пресловутой карты часовых поясов/городов. Сначала это была карта с возможностью подсветить нужный город, потом она приобрела функцию масштабирования, которая со временем тоже немного изменялась. Но лично меня в последний раз эта возможность скорее заставляла ругаться, чем помогала в установке. И вот теперь мы получили новую карту часовых поясов, разлинованную на 24 поло-

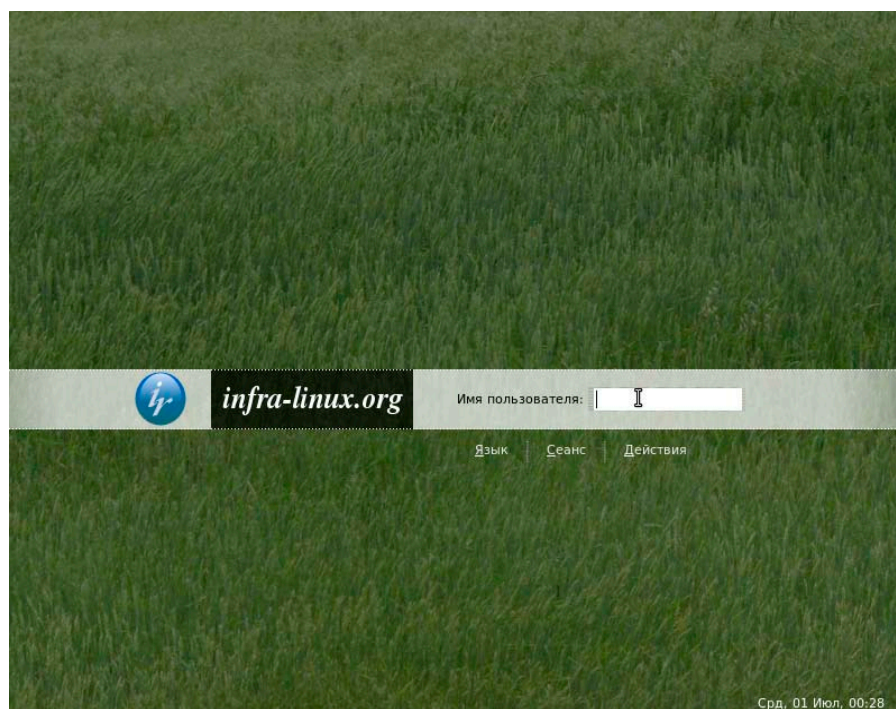
сы. С одной стороны, удобно, что нужный часовой пояс подсвечивается, но с другой – все вновь сводится к выбору города, который на маленькой карте стало не так-то легко найти. Кроме того, жаль, что пояса никак не взаимодействуют с Гринвичем: будет удобнее, если хотя бы во всплывающих подсказках покажут отклонение от нулевого часового пояса. Думаю, эту возможность ещё не раз переделают.

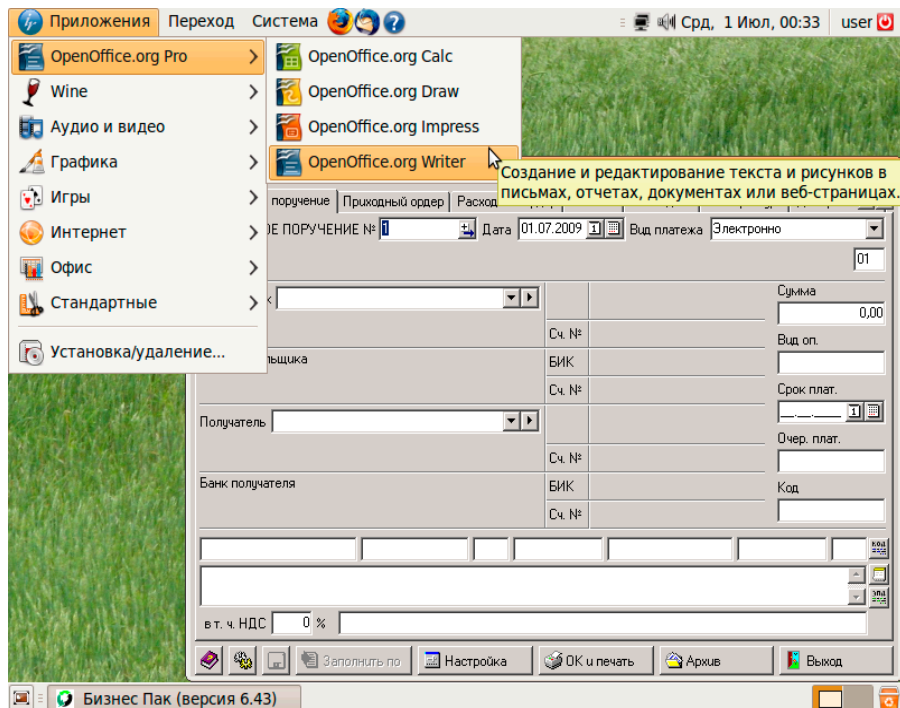
Следующий шаг – разметка диска с привычными вариантами. Так как на используемом жестком диске уже была установлена Ubuntu 8.04.2, установщик нашел её и выдал предупреждение, что в режиме автоматической разметки при использовании всего диска операционная система будет стерта – к слову, предупреждение было на английском языке. Если же разбивать жесткий диск вручную, то можно выбрать новую версию классической файловой системы Linux – Ext4, она также доступна для установки на корневой раздел и /boot. Перейдя к следующему шагу, можно обнаружить, что после ввода реквизитов для пользователя есть настройка для автоматического входа в систему, хотя лично я все же предпочитаю постоянный ввод пароля.

На этом все предварительные настройки заканчиваются, и можно переходить к самой установке, которая проходит весьма быстро. Последний шаг – перезагрузка и извлечение диска из привода.

Интерфейс

Первое, что заметно при начальной загрузке дистрибутива, – измененный логотип загрузки, аналогичный тому, что можно увидеть в загрузочном меню дис-





ка. Второе – это специальная тема для GDM, в которой используются изображения и текст, говорящие о том, что запущен дистрибутив InfraLinux. Несмотря на то что тема оформления практически стандартная, изменены картинки на рабочем столе и по умолчанию установлена та, что находится на фоне темы GDM. Кстати говоря, обоим из InfraLinux.org доступны в официальном репозитории, о котором речь пойдет немного позже. Следующее, что бросается в глаза помимо измененного внешнего оформления, – это изначально установленная вся доступная локализация. На мой взгляд, это огромный плюс для русскоязычного неопытного пользователя: русскоязычный интерфейс и документация не отпугнут новичка и избавят его от необходимости прикладывать дополнительные усилия по настройке.

Если обратиться к приложениям, то можно заметить, что под OpenOffice.org Pro выделен целый пункт меню. На CD-версии изначально доступны четыре компонента: Writer, Calc, Draw и Impress – этого вполне должно хватить для полноценной работы. Также можно расширить состав приложений, просто доустановив нужные пакеты. Скорость работы OOo Pro меня приятно удивила: на моей хост-машине (Arch Linux и официальная сборка OpenOffice.org) с 2 Гб оперативной памяти офисный пакет запускается медленнее, чем на виртуальной с 350 Мб памяти. Если кому-то интересно, чем отличается Pro-сборка от обычной (vanilla), смотрите в официальном анонсе: <http://i-rs.ru/O-kompanii/Novosti/OpenOffice.org-pro-3.1>.

Другим «горячим» нововведением Ubuntu стала система уведомлений. Эту

идею долго обсуждали на форумах и в блогах, и вот она стала явью. Суть заключается в том, что на экране создается очередь уведомлений (как это реализовано в KDE 4.2) и при любом системном событии на экран всплывает небольшое сообщение. Довольно удобно наблюдать, что происходит в системе, следить за сообщениями в Pidgin, подключением к сети и так далее. В общем, всё то же самое, что доступно любому пользователю KDE с версии 4.2.

Приложения

Помимо того что в меню «Приложения» доступен OpenOffice Pro, есть и некоторые другие изменения. Среди них, например, измененная стартовая страница в Firefox с логотипом InfraLinux и поиском от Google, а также включена русская локаль по умолчанию. Во время работы в системе совершенно случайно был обнаружен консольный файловый менеджер Midnight Commander (mc), которого иногда очень не хватает в стандартной поставке Ubuntu.

Среди утилит, доступных пользователям Ubuntu 9.04, стоит отметить программу чистки системы и приложение для создания загрузочного USB. Обе доступны из меню «Система → Администрирование». Первая создана для того, чтобы нерадивые или неопытные пользователи не копили на жестком диске ненужные пакеты. Программа создания загрузочного USB-диска позволяет, используя доступный ISO-образ, сделать установку ещё одной копии Ubuntu/InfraLinux.org на USB-накопитель. Здесь доступен минимумстроек, однако есть главное – включить

возможность сохранения данных сессии на этом же накопителе, для чего на нем создается специальный раздел.

Репозиторий

Пора вернуться к обсуждению репозитория InfraLinux.org. Точнее, нескольких репозиторий, которые настраиваются так же, как и все остальные: либо правкой файла `/etc/apt/sources.list`, либо через программу «Источники приложений» в меню «Система → Администрирование». Помимо уже упомянутого пакета с обоями в репозитории можно найти метапакет `infraLinux-desktop` для полной установки на систему без графики или на Ubuntu, а также пакеты с документацией, настройками, темой для GDM и прочее. Там же, в репозитории, находятся готовые пакеты с Prism-приложениями от Google, а именно: Calendar, Gmail, Reader, Maps, Groups, Docs. С их помощью можно достаточно легко и просто получить доступ к сервисам от Google, снизив нагрузку на процессор, но задействовав интернет-канал. При всём этом приложение будет выглядеть так, как будто его запустили на этом же компьютере – никаких адресных строк, панелей навигации, закладок и прочих привычных атрибутов веб-браузеров.

Из офисных приложений помимо упомянутого OpenOffice Pro доступны два других: «Бизнес Пак» и «Мой склад». Первое использует для своей работы WINE и служит для формирования счетов, платежных поручений и прочего, а второе выполняет как приложение Prism. Программы могут пригодиться организациям и предпринимателям, в то время как рядовому пользователю они вряд ли понадобятся.

Заключение

Как уже было написано во введении, дистрибутив произвел на меня положительное впечатление. В этом заслуга полной российской локализации из коробки, фирменного графического оформления, высокой скорости загрузки и работы как самой ОС, так и OpenOffice.org, Prism-приложений, легкого доступа к программам «Бизнес Пак» и «Мой склад». Часть этих достоинств принадлежит Ubuntu, часть – компании «Инфра-Ресурс», однако в целом получился неплохой продукт для конечного пользователя. Если учесть, что сама компания предлагает коммерческую поддержку организациям, то у дистрибутива вполне может быть светлое будущее.

Никита Лялин
(tinman321@gmail.com)

Научная графика силами Open Source.

Часть 1: Введение

Словосочетание «научная графика» часто ассоциируется с чем-то запредельно сложным и оторванным от жизни. На самом деле практически каждый из нас неоднократно с ней сталкивался. Научная графика, в том смысле, который принят в сфере ИТ, — это любые графики и диаграммы, визуально отображающие некие числовые данные (графики функций в учебнике математики, кривые падения ВВП в период кризиса, карты температуры земной поверхности в прогнозах погоды и т.п.). Учитывая повсеместную распространенность таких изображений, неудивительно, что существует множество программ, как свободных, так и коммерческих, предназначенных для их создания. Приходится с сожалением констатировать, что позиции Open Source в области научной графики не так прочны, как хотелось бы. Ни одна из открытых разработок не дотягивает до уровня коммерческих продуктов, признанных стандартами де-факто. Однако ситуация не безнадежна. В нескольких частях этой статьи будут рассмотрены наиболее распространенные свободные программы для научной графики в сравнении с коммерческими «китами». Я сознательно не буду касаться специфических областей (картографии, анимации физических процессов), а сосредоточусь на статических двух- и трехмерных графиках, которые наиболее часто встречаются в повседневной жизни школьников, студентов, преподавателей и ученых.

Почему электронные таблицы не подходят?

Большинство людей, которым необходимо построить график по набору чисел, не сговариваясь, используют для этого Microsoft Excel или, в лучшем случае, его свободные аналоги OpenOffice.org Calc или Gnumeric — в общем, программы, не предназначенные для научной графики, а следовательно, не такие удобные и функциональные, как специализированные средства, о которых и пойдет речь дальше. Электронные таблицы не лучший выбор для научной графики по трем причинам. Во-первых, полученная «картинка» больше подходит для красочной, но безвкусной бизнес-презентации, чем для строгого научного журнала или дипломной ра-

боты. Многие опции оформления графика или вообще не предусмотрены (как, например, разрывы осей), или реализованы очень неудобно, а оформление не соответствует принятым стандартам. Одна из самых неприятных особенностей — статический размер текста подписей и легенды. Текст не масштабируется при изменении размера графика и все текстовые метки приходится менять вручную.

Во-вторых, импорт исходных числовых данных и экспорт графика в нужный формат представляют серьезную проблему. Открыть текстовый файл с колонками чисел в OpenOffice.org Calc бывает проблематично. (Впрочем, при копировании и вставке данных из обычных текстовых редакторов в Calc он предлагает преобразовать их в столбцы, указав нужный разделитель, чего обычно бывает вполне достаточно. — Прим. ред.) Сохранить график из Excel в «издательском» формате ers штатными средствами невозможно.

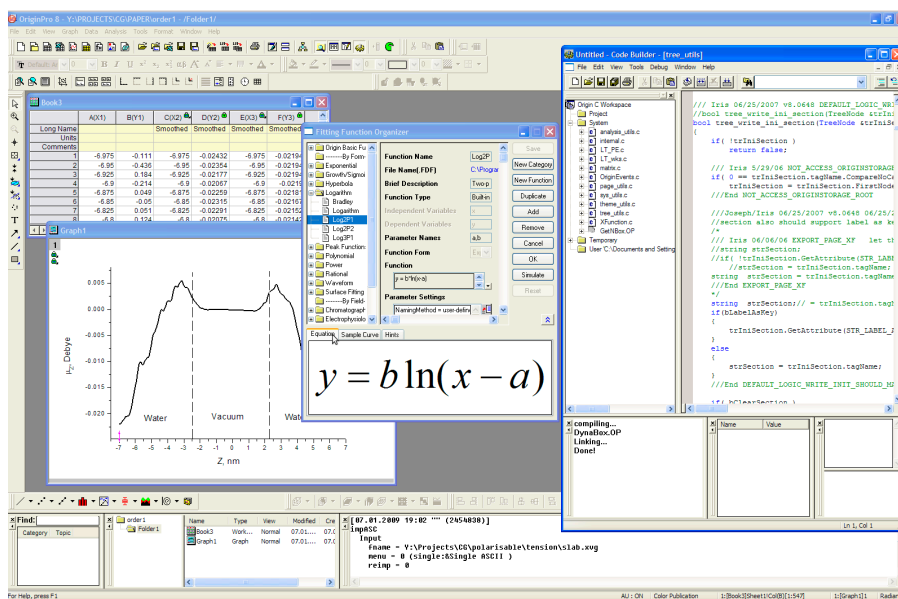
В-третьих, электронные таблицы практически невозможно использовать для сложной обработки данных (сглаживания, преобразования Фурье, интегрирования, частотной фильтрации). Возможностей электронных таблиц с лихвой хватит для школьника или нетребовательного студента, но в остальных случаях необходимо использовать специальные инструменты.

Немного теории

Числовые данные в подавляющем большинстве случаев хранятся в виде обычного текстового файла, где каждая колонка соответствует значениям некоего параметра (очень большие объемы данных обычно хранят в бинарном виде, но для нужд визуализации их, как правило, преобразуют в текстовую форму после соответствующей обработки и фильтрации). Набор колонок, содержащий независимую и зависимую величины, называют набором данных (data set). Каждый набор данных (а их могут быть в файле десятки) можно отобразить на графике разными способами.

На внешний вид графика влияют: его тип (точки, символы, линии, сглаженная кривая, столбчатая диаграмма, круговая диаграмма-«пирог»); параметры координатных осей (интервал значений, линейный, логарифмический, обратный масштаб, разрывы в местах, где нет данных); формат засечек и чисел на осях; вид линии координатной сетки; подписи осей; формат легенды и многое другое. Несколько графиков зачастую нужно группировать сложным образом (таблицы из графиков, графики с одной общей осью, вставки «график в графике»).

Кроме того, очень часто нужно выполнять математические операции с наборами данных. Они могут быть как простыми (например, вычесть один столбец из другого, чтобы отобразить разницу между ними, или масштабировать данные так, чтобы максимальное значение равнялось 100), так и очень сложными (построить по данным логарифмическую гистограмму или график преобразования Фурье, аппроксимировать точки прямой линией или сложной аналитической функцией со мно-



Встречайте монстра: Origin 8.0 с импортированным набором данных, построенным по ним графиком, библиотекой аналитических функций и встроенной IDE для языка Origin C

жеством параметров). Поэтому программы научной графики, как правило, являются и программами для анализа данных с возможностью написания полноценных программ или хотя бы коротких скриптов на встроенном языке.

Коммерческий стандарт

Сегодня в мире стандартом де-факто для научной графики является Origin (<http://www.originlab.com>) – это сложная универсальная программа для анализа и отображения данных, работающая только в Windows. В США Origin стоит от 500 (академическая лицензия) до 1500 USD (коммерческая). Студенческие лицензии стоимостью 50 USD действуют, к сожалению, только в США и Канаде. Официальной информации о ценах в странах СНГ нет, но вряд ли они существенно ниже. Кроме ожидаемой функциональности по чтению данных в различных форматах, их анализа и отображения в виде графиков и диаграмм десятков разных типов, Origin предоставляет своего рода комплексную среду разработки. Последние версии Origin – это сложнейшие монстры со встроенным C-образным компилируемым языком программирования, полновесной IDE (!) с возможностью создания диалогов, встраиваемых в интерфейс, скриптовым языком, модулем доступа к базам данных с поддержкой SQL, модулем обработки изображений, интеграцией с Microsoft Office и т.д.

Интерфейс программы до версии 8.0 был очень хорошо отшлифован и стал привычен для многочисленной армии пользователей. Однако начиная с восьмой версии он сильно усложнился. На-

метилась тенденция к превращению специализированного инструмента в своего рода «швейцарский нож», перегруженный ненужной функциональностью. Однако основная концепция – полностью интерактивное форматирование графиков по принципу WYSIWYG – остается реализованной практически идеально.

Нужно признать, что полноценного свободного аналога Origin на сегодняшний день не существует. До недавнего времени не существовало даже ничего похожего на него по базовой функциональности и удобству WYSIWYG-интерфейса. Сейчас такие программы появились (QtiPlot – <http://soft.proindependent.com/qtiplot.html>; LabPlot – <http://labplot.sourceforge.net>; SciGraphica – <http://scigraphica.sourceforge.net>; SciDAVis – <http://scidavis.sourceforge.net>) и уже могут тягаться с гигантом в большинстве практических задач. О них пойдет речь впереди. Однако и до их появления мир Linux не был обделен в плане научной графики. Начнем с главных старожилов – gnuplot и grace, которые не отличаются красотой и интуитивностью интерфейса, но при удивительно маленьком размере могут быть разумной альтернативой раздутому Origin и во многих ситуациях оказываются гораздо практичнее.

Gnuplot

Первый по старшинству – gnuplot (<http://www.gnuplot.info>). Этот ветеран существует и активно развивается с 1986 года. Графического интерфейса как такового у него нет – за исключением окна, в котором выводится сам график. Все управление осуществляется из командной строки. Gnuplot позволяет строить двух- и трех-

мерные графики аналитических функций и графики по данным, считанным из текстовых файлов. Документация очень подробна и снабжена множеством примеров. Программа присутствует в репозиториях всех распространенных дистрибутивов. Gnuplot используется как «движок» для построения графиков в других программах (таких, например, как пакет компьютерной алгебры wxMaxima и математический пакет Octave). Существует также несколько GUI-надстроек для gnuplot – xgfe (<http://www.uni-koeln.de/rrzk/software/grafik/visualization/xgfe/xgfe.html>), unignuplot (<http://sourceforge.net/projects/unicalculus>) и qgfe (<http://www.xm1math.net/qgfe>). Практическая польза от этих надстроек сомнительна, поскольку пользователь все равно должен знать синтаксис gnuplot, а диалоговые окна не включают и половины всех доступных команд и опций.

Интересно, что gnuplot распространяется не по лицензии GPL, а по особой свободной лицензии, придуманной его авторами. Эта лицензия идентична по духу обычной GPL, однако запрещает распространять модифицированный исходный код. Вместо этого модификации должны распространяться как патчи к официальному релизу. Для конечного пользователя эти тонкости совершенно не важны, но для разработчиков могут создавать некоторые неудобства.

Подробности о работе с gnuplot и возможностях этой программы читайте в следующем выпуске «Open Source».

Семен Есилевский
(yesint4@yahoo.com)

Обзор проекта свободного BIOS – Coreboot

*Пусть будет теплою стена
И мягкой – скамейка...
Дверям закрытым – грош цена,
Замку цена – копейка!*

Б. Окуджава. «Песенка об открытой двери»

Coreboot – это проект по созданию свободной прошивки (микропрограммы) BIOS. Он был начат в 1999 году и являлся частью исследования в области кластеризации в лаборатории Cluster Research Lab, являющейся частью Advanced Computing Laboratory, входящей в Лос-Аламосскую национальную лабораторию (Los Alamos National Laboratory) США. Тогда проект получил

название LinuxBIOS. Его возглавил Рональд Минних (Ronald G. Minnich) (<http://archive.fosdem.org/2007/schedule/speakers/ronald+g+minnich>), а основной целью значилось создание специализированного BIOS для суперкомпьютеров.

С течением времени проект привлек внимание Организации свободного программного обеспечения (Free Software Foundation, FSF). Вот что сказал по этому поводу Вард Вандеведж (Ward Vandeweghe), старший системный администратор FSF, в интервью журналу «Системный администратор» в феврале 2006 года: «Наша организация познакомилась с LinuxBIOS достаточно давно, так как этот проект

развивался уже более пяти лет. Мы даже начали кампанию за свободный BIOS год назад» (<http://www.fsf.org/campaigns/free-bios.html>). Сейчас разработка свободного BIOS поддерживается FSF и входит в перечень его приоритетных проектов (<http://www.fsf.org/campaigns/priority.html>). Все разработки проекта coreboot (выпущены две версии, а третья находится в альфа-стадии разработки) доступны под лицензией GNU GPL. Официальный сайт проекта – <http://www.coreboot.org>.

Возможности и особенности

Одна из главных особенностей coreboot заключается в том, что после быстрой инициализации материнской платы и периферии (около 3 секунд) управление передается на так называемую полезную нагрузку (payload). Последняя отвечает за работу с загрузчиками, операцион-

ными системами или даже играми (<http://www.coreboot.org/Payloads>).

С помощью coreboot можно запускать такие операционные системы, как GNU/Linux, FreeBSD, OpenBSD, NetBSD, OpenSolaris и Windows:

- ✓ Ядро Linux может использоваться в payload напрямую – оно встроено в микросхему ПЗУ, где находится coreboot.
- ✓ FreeBSD загружается через coreboot благодаря ADLO (<http://www.coreboot.org/ADLO>) – дополнительной прослойке на основе 16-битного PC-BIOS из Open Source-эмулятора Bochs.
- ✓ OpenBSD тоже загружается через coreboot с использованием ADLO, хотя этот процесс еще недостаточно протестирован.
- ✓ Для загрузки NetBSD используется SeaBIOS (<http://www.coreboot.org/SeaBIOS>) – реализация BIOS, ранее известная как LegasyBIOS, позволяющая прочитать файловую систему coreboot и извлечь различные опции для использования жестких дисков и payload.
- ✓ Для загрузки Windows также используется SeaBIOS (протестировано на версиях Windows XP, Vista и 7 Beta).
- ✓ OpenSolaris задействует для загрузки GRUB2.

Среди поддерживаемых загрузчиков можно выделить:

- ✓ **Official GRUB 2** – это официальная версия GNU GRUB 2, которая может быть использована в payload для загрузки операционной системы с жесткого диска.
- ✓ **GRUB2** – это модульный загрузчик, основанный на старой версии GRUB2 и имеющий поддержку мультизагрузки. Он был разработан Патриком Джорджи (Patrick Georgi) и представлен на Google Summer of Code 2007.
- ✓ **FILO** – простой загрузчик с поддержкой файловых систем ext2, fat, jfs, minix, reiserfs, xfs и iso9660. Среди его возможностей: загрузка с жесткого диска IDE или SATA, CD-ROM и системной памяти (ROM); поддержка образов ELF и zImage/bzImage; загрузка с raw-устройств; полная поддержка ELF Boot Proposal. В FILO применяется libpayload – небольшая библиотека, лицензированная под BSD и предназначенная для использования в качестве основы для coreboot payload. Вместо него рекомендуется использовать GRUB2.
- ✓ **Etherboot** – предназначен для загрузки с использованием сети. Он является

заменой для проприетарных PXE ROM и содержит большое количество дополнительных функций (DNS, HTTP, iSCSI и другое). Устаревшие версии Etherboot включали части загрузчика FILO, которые обеспечили поддержку загрузки с использованием SATA и USB. Новая версия GPXE пока не поддерживается, поскольку требуются некоторые изменения в коде с целью обеспечения совместимости с coreboot.

- ✓ **Open Firmware (OFW)** (http://www.openfirmware.info/Open_Firmware) – представленная в 2006 году компанией Firmworks разработка, распространяющаяся под лицензией BSD. Она была реализована в рамках стандарта на прошивку (firmware) загрузчика (инициализация, конфигурация) IEEE1275-1994. Обладает поддержкой архитектур x86, PowerPC и ARM – впрочем, разработчики заявляют, что при необходимости к этому списку можно добавить и другие архитектуры, такие как SPARC и MIPS. Версия Open Firmware для x86 используется на ноутбуках OLPC «ХО», разрабатываемых некоммерческой организацией One Laptop Per Child (<http://olpc.com>).
- ✓ **OpenBIOS** (http://www.openfirmware.info/Welcome_to_OpenBIOS) – разработанный в рамках стандарта IEEE1275-1994 загрузчик, который представляет собой наиболее свободную реализацию из всех существующих Open Firmware. Сфера его применения весьма широка. Например, он используется в качестве загрузочного диска (boot ROM) для эмулятора QEMU для PPC, PPC64 и Sparc32. С помощью OpenBIOS/Sparc32 можно загрузить Linux, NetBSD и OpenBSD. OpenBIOS/Sparc64 существует, но пока работает нестабильно. OpenBIOS/PPC позволяет загрузить Linux. Coreboot использует OpenBIOS в качестве payload на платформе x86.

Поддержка оборудования

На сегодняшний день поддержка оборудования представлена 203 материнскими платами и определенным набором чипсетов. Уровень поддержки различных компонентов материнских плат разделяется на несколько категорий:

- ✓ платы, работа над которыми продолжается или временно прекращена;
- ✓ платы, работа над которыми в основном завершена, но остались нерешенными некоторые вопросы;
- ✓ платы, которые поддерживаются полностью.

Например, coreboot 3 поддерживает такие материнские платы, как AMD DB800 и Norwich, AMP TinyGX, Artec Group DBE61 и DBE 62; GYGABYTE GA-M57SLI-S4, Jetway J7F2, VIA EPIA-CN, а coreboot 2 совместим с такими материнскими платами, как Abit BE6-II V2.0, ASUS A8N-E, A8N-SLI, A8N5X, A8V-E SE и A8V-E Deluxe; GYGABYTE GA-6BXC, GA-2761GXDК и GA-M57SLI-S4, MS-6119, MS-6147 и MSI MS-6178.

Стоит отметить поддержку ряда серверных материнских плат от таких производителей, как AMD, Dell, HP, IBM, Intel, MSI, Newisys, NVIDIA, Supermicro, Tyuan. Поддерживаются платы для ноутбуков OLPC (btest, rev_a) и ряда иных форм-факторов (Mini-ITX, Micro-ATX и других).

Что касается поддержки чипсетов, то она представлена для следующих вендоров:

- ✓ **северный мост** – AMD, IBM, Intel, Motorola, SiS, VIA;
- ✓ **южный мост** – AMD, Broadcom, Intel, NVIDIA, Ricoh, SiS, VIA, Winbond.

Вспомогательные средства

Помимо этого, в рамках проекта coreboot развивается ряд дополнительных утилит и библиотек:

- ✓ **Flashrom** – утилита для идентификации, чтения, записи, проверки и стирания flash-памяти чипов. Она используется для работы с флеш-образами BIOS, EFI, coreboot и различных прошивок. Поддерживает более 160 flash-чипов, 75 чипсетов, 100 материнских плат, а также 10 PCI-устройств, которые могут быть задействованы как внешние прогнаторы.
- ✓ **Superiotool** – запускаемая в пространстве пользователя утилита coreboot, ориентированная на определение контроллера ввода/вывода материнской платы (должна работать на большинстве UNIX-подобных систем, проверено на Linux и FreeBSD).
- ✓ **Nvramtool** – утилита для чтения/записи параметров coreboot и отображения информации из таблицы coreboot в CMOS/NVRAM. Раньше она была известна как lxbios и cmos_util.
- ✓ **Buildroom** – инструмент, позволяющий создавать образы coreboot с нуля.
- ✓ **Mkelfimage** – позволяет формировать образы ELF-загрузчика из ядра Linux.
- ✓ **Inteltool** – утилита, задачей которой является предоставление некоторой информации о процессоре/чипсете Intel в аппаратной конфигурации.
- ✓ **Msrtool** – еще одна небольшая утилита, которая создает дампы специфических MSR-регистров чипсета.

☑ **Ectool** – делает дампы памяти (RAM) контроллеров (Embedded/Environmental Controller, EC) ноутбуков.

Итоги

Coreboot поступательно развивается на протяжении десяти лет и за это вре-

мя успел заручиться поддержкой FSF, которая включила его в список приоритетных проектов. Растущая поддержка материнских плат и чипсетов и растущий интерес к свободному BIOS позволяют делать оптимистичные выводы в плане перспектив проекта. Потенциальные возмож-

ности, заложенные в проекте, и гибкость coreboot дают основания предполагать, что число его пользователей будет увеличиваться, а сам проект пришел «все-речь и надолго».

Игорь Штомпель
(keepercoder@gmail.com)

Беседы о Qt. Часть 6: QSettings и XML

Класс QSettings служит Qt-программистам верой и правдой, когда нужно сохранить или загрузить какие-нибудь настройки программы. Технические заботы по хранению настроек класс берет на себя. В Linux по умолчанию создается ini-подобный файл в каталоге \$HOME/.config/имя_программы, а в Windows записи помещаются в системный реестр. Такое поведение можно переопределить, вызвав конструктор с именем файла и указав формат: в случае QSettings::IniFormat и в Windows будет использован ini-подобный файл.

С точки зрения доступа к данным разницы нет никакой: для чтения и записи используются одни и те же функции вне зависимости от физического способа хранения данных. Пример чтения:

```
QString s = settings->value ("my string", QLatin1String("my default value")).toString();
```

Пример записи:

```
QString s;  
settings->setValue ("my string", s);
```

Кроме таких обыденных вещей, как хранение «единичных» переменных, QSettings предоставляет мощный механизм сохранения других данных, например состояния QSplitter (то есть размеров виджетов, вставленных в сплиттер), а также состояния вспомогательных виджетов главного окна, таких как инструментальных панелей.

У класса QMainWindow есть методы restoreState и saveState. Для сохранения достаточно вызвать при закрытии окна:

```
settings->setValue ("state", saveState());
```

Для восстановления можно поместить в конструктор следующее:

```
restoreState (settings->value ("state", QLatin1String("")).toByteArray());
```

В итоге будет восстановлено положение панелей инструментов, причем если они были не припаркованы, то это состояние тоже сохраняется и восстанавливается. Замечу, что для правильной работы этого механизма панелям управления надо сначала дать имена, а потом уже сохранять или восстанавливать. Например:

```
tb_my_toolbar = new QToolBar;  
tb_my_toolbar->setObjectName ("tb_my_toolbar");
```

К сплиттерам это не относится. Состояние сплиттера восстанавливается примерно так:

```
my_splitter->restoreState (settings->value (QLatin1String("splitterSizes")).toByteArray());
```

А сохраняется следующим образом:

```
settings->setValue ("splitterSizes", my_splitter->saveState());
```

Замечательное свойство QSettings сохранять и загружать данные любых типов объясняется тем, что записываются и возвращаются экземпляры не менее замечательного класса QVariant. В него можно поместить что угодно: от строки или числа до целой хэш-таблицы. Трудно вообразить для хранения пар формата «ключ=значение» что-либо удобнее, чем класс QSettings, однако, если возникнет надобность написать своё решение, это нетрудно сделать при наличии других классов из состава Qt. Всего-то и нужно, что прочитать текст из файла, разбить его на строки, разобрать пары из каждой строки и поместить их – для ускорения доступа – в хэш-таблицу.

Для чтения текста из файла можно использовать такую функцию:

```
QString qstring_load (const QString &fileName, QLatin1String charset)  
{  
    QFile file (fileName);  
  
    if (! file.open (QFile::ReadOnly | QFile::Text))  
        return QString();  
  
    QTextStream in(&file);  
    in.setCodec (charset.toAscii());  
  
    return in.readAll();  
}
```

Вот как читать текст из файла:

```
QString text = qstring_load ("/test/test.txt", "UTF-8");
```

Теперь – создание списка из полученного текста, предполагая, что в тексте каждая строка отделена от другой с помощью «\n»:

```
QStringList l = text.split ("\n");
```

Для краткости можно эти два действия объединить в одно:

```
QStringList l = qstring_load (fname).split ("\n");
```

Что случится, если возвращенная от qstring_load строка вдруг окажется пустой? Ничего – ведь это не null-строка как указатель, а экземпляр класса. Возвратится вполне готовый класс QString, и не его вина, что текста-то в нем может не быть, и делить на строки нечего. При попытке разобрать на строки такую «пустую» строку будет возвращен QStringList с нулем элементов.

Преобразование QStringList в хэш-таблицу делается просто. Вообразим, что мы хотим трактовать не только ключи (из прочитанного файла), но и сопоставленные им значения как строки. Такому положению вещей будет соответствовать хэш-таблица следующего вида:


```
QHash<QString, QString> my_table;
```

Напишем функцию, которая получает в качестве параметра имя файла, загружает из него «ini-данные» и возвращает их как готовую к использованию хэш-таблицу:

```
QHash<QString, QString> hash_load_keyval (const QString &fname)
{
    //Сюда будем записывать разобранные строки:
    QHash<QString, QString> result;

    //Читаем из файла в список:
    QStringList l = qstring_load (fname).split ("\n");

    //Проходим по всему списку:
    foreach (QString s, l)
    {
        //Делим текущую строку списка на два элемента,
        //разделенные знаком равенства:
        QStringList sl = s.split ("=");

        //Если элементов больше 1, всё правильно,
        //вставляем ключ и значение в хэш-таблицу
        if (sl.size() > 1)
            result.insertMulti (sl[0], sl[1]);
    }
    //Возвращаем заполненную таблицу:
    return result;
}
```

Обратите внимание на функцию QHash::insertMulti. Вместо неё можно было бы использовать просто insert, но в таком случае теряются все повторяющиеся значения (для ключа), кроме последнего.

Сохранение хэш-таблицы в файле выглядит примерно так: получаем список уникальных ключей QList<Key> QHash::uniqueKeys (), затем для каждого из них получаем список значений QList<T> QHash::values (const Key & key), формируем строки вида «ключ=значение» и записываем их в QStringList, после чего вызываем QStringList::join («\n»), чтобы получить готовую строку со всеми данными хэш-таблицы. Теперь эту строку можно сохранить в файл, причем хорошо бы отрезать последний символ строки (там находится лишний «\n»).

Теперь – об XML. К XML обращаются, как правило, в двух случаях: когда совсем неудобно хранить данные в формате «ключ=значение» или когда под рукой есть хороший XML-парсер и очень уж хочется его опробовать. По большому счету там, где можно обойтись ini-подобными файлами, нет нужды палить из пушки по воробьям. XML хорош там, где надо сохранять и загружать древовидные структуры данных. Qt дает целых три механизма работы с XML. Первый называется SAX и предоставляет способ работы с XML, основанный на событиях. То есть запускается парсер и на возникающие при разборе XML-данных события (например, открытие тега) вызываются заданные вами обработчики. Для работы методом SAX у Qt служит QDomSimpleReader и сопутствующие ему классы.

Другой механизм – QDomDocument. Подход тут другой: парсеру передается весь текст документа, который сразу разбирается, и вам остается «ходить» по уже разобранному дереву данных, доступному внутри QDomDocument. Остановлюсь на этом подробнее. Допустим, мы хотим написать простой парсер документов OpenDocument Text (ODT). Напомню, что ODT – это обычный ZIP-архив, наполненный файлами как с текстом/изображениями, так и служебного назначения (Подробнее о формате ODT см. в выпуске «Open Source» 011 от 30.08.2006. – Прим. ред.). Собственно текст находится в файле content.xml. Распаковка ZIP-архивов выходит за рамки этой статьи. Предположим, что мы уже прошли этап распаковки и можем передать парсеру содержимое файла content.xml. Напишем класс CODTXMLWalker, который будет извлекать весь текст из XML-дерева. В классе нам понадобятся следующие поля:

- ☑ **QString data** – сюда мы будем помещать параграфы текста, извлеченные из узлов дерева;
- ☑ **QDomDocument doc** – экземпляр парсера.

Важна функция void step (QDomNode node) – она послужит для итерационного прохождения по всем элементам дерева, на всех его уровнях. Итак, объявление класса:

```
class CODTXMLWalker: public QObject
{
    Q_OBJECT public:

    QString data;
    QDomDocument doc;

    void step (QDomNode node);
};
```

Воплощение функции step:

```
void CODTXMLWalker::step (QDomNode node)
{
    //Для всех детей узла node проходим в цикле:
    for (QDomNode n = node.firstChild(); ! n.isNull(); n = n.nextSibling())
    {
        //Преобразуем узел (текущее «дитя») в элемент:
        QDomElement e = n.toElement();

        //Если элемент ничего не содержит, переходим к следующему:
        if (e.isNull())
            continue;

        //Если имя узла равно «text:s», у нас возможен отступ:
        if (e.nodeName() == "text:s")
        {
            //Проверяем наличие отступа:
            QString indent = e.attribute ("text:c");

            //Если есть отступ, создаем строку из пробелов в количестве
            //значений отступа:
            if (! indent.isEmpty())
            {
                QString fillval;
                fillval = fillval.fill (' ', indent.toInt());
            }
            //И добавляем эту строку к data:
            data.append (fillval);
        }
        else
        {
            //Иначе, если такие-то имена узлов, то у нас параграф либо заголовок
            if (e.nodeName() == "text:p" || e.nodeName() == "text:h")
            {
                //Если текст в них не пуст, добавляем его к data
                if (! e.text().isEmpty())
                {
                    data.append (e.text());
                    data.append ("\n");
                }
            }
        }

        //Если у узла есть дети, делаем рекурсивный вызов step,
        //чтобы этих детей тоже обработать:
        if (e.hasChildNodes())
            step (n);
    }
}
```

Пример использования нашего парсера:

```
QString string_data = //Читаем сюда содержимое content.xml

//Создаем парсер:
CODTXMLWalker walker;

//Устанавливаем ему содержимое, что вызывает разбор одного:
walker->doc.setContent (string_data);

//Теперь проходим по всем элементам разобранного XML-файла:
walker->step (walker->doc.documentElement());

//Выводим извлеченный текст на консоль:
qDebug() << walker->data;
```

Конечно же, дерево объектов можно изменять (например, добавлять в него новые элементы), а потом получить готовый XML-файл в виде текстового объекта, вызвав функцию `QDomDocument::toString()`.

Наконец, в Qt существует третий способ работы с XML, который подразумевает использование классов `QXmlStreamReader` и `QXmlStreamWriter`. Оба «заточены» на работу с рекурсией, хотя сами классы обрабатывают данные последовательно. Что имеется в виду? Допустим, мы рекурсивно обходим какое-то дерево и по ходу дела пишем в XML-поток (это может быть файл, буфер в памяти и тому подобное) элементы. А при чтении мы последовательно получаем элементы, один за другим, «знакомясь» с ними по мере их вложенности. Чтобы проверить, является ли очередная разобранный составляющая потока началом элемента, достаточно вызвать функцию `isStartElement()`. Если она возвращает истину – мы находимся за открывающим тегом. Для проверки на закрытие тега есть парная функция – `isEndElement()`. В примере ниже показано чтение XML-файла и разбор его на элементы. В цикле разбора на консоль выводится название очередного элемента, а также список его атрибутов и их значений:

```
QFile file (filename);
if (!file.open (QIODevice::ReadOnly | QIODevice::Text))
return;
```

```
QXmlStreamReader xml (&file);

while (!xml.atEnd())
{
    xml.readNext();

    if (xml.isStartElement())
    {
        qDebug() << xml.name().toString();
        QXmlStreamAttributes attrs = xml.QXmlStreamReader::attributes();
        for (int i = 0; i < attrs.count(); i++)
            qDebug() << attrs[i].name().toString() << " = " << attrs[i].value().toString();
    }
}
```

Именно этот, третий, способ работы с XML кажется мне наиболее удобным при сохранении/загрузке каких-нибудь настроек или для построения интерфейса на основе XML. Ведь если в случае с `QDomDocument` вы после разбора XML должны пройти рекурсивно по всему дереву, то здесь можете что-либо делать прямо во время разбора. А `QDomDocument` более пригоден, когда требуется именно работа с элементами дерева, а не просто хранение и загрузка.

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