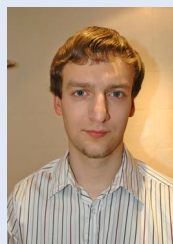


Колонка главного редактора



Как я в какой-то момент выяснил, люди считают, что писать данную колонку – элементарное занятие. Казалось бы, близкий к абсолютному полет фантазии, который в результате

обычно производит впечатление чего-то легкого и само собой разумеющегося – что тут может вызывать трудности? Особенно для человека, которому нравится писать, нравится делиться своими мыслями. На деле же оказывается, что данный скромный компонент издания редко появляется в голове сам по себе и тем более в нужный момент. И кажутся необъятными те усилия, что каждый раз требуются для создания чего-либо вменяемого для того, чтобы читатель просто окинул взглядом в очередной выпуск – пусть и в первую очередь.

С другой стороны, это всегда интересное испытание, способное поддерживать редактора в нужной форме. Ведь эта маленькая колонка требует квинтэссенции размышлений на произвольные темы (или констатации каких-либо сведений, или полезных идей, или недавнего опыта... – неважно), выраженной в удобоваримой форме, такой, которая вообще-то не очень подразумевает подобную плотность наполнения. Насколько это удачно получается у меня, я до сих пор не оценил, поскольку не видел ни одного комментария на сей счет. Может, они появятся теперь?

Главный редактор
Дмитрий Шурупов
(osa@samag.ru)

«Open Source»

электронное приложение к журналу

«Системный администратор»

№26, 27 мая 2008 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

Qt 4.4: теперь с WebKit

6 мая норвежская компания Trolltech объявила о выходе финальной версии своего ключевого продукта – фреймворка для разработки межплатформенных приложений – Qt 4.4. Библиотека Qt служит основой для KDE, популярной графической среды для GNU/Linux и других UNIX-подобных операционных систем. Главный акцент в этом релизе Qt разработчики сделали на работу с Web и мобильными устройствами. Среди ключевых новшеств в Qt 4.4 отмечается интеграция с Open Source-движком WebKit, лежащим в основе веб-браузера Safari, появление поддержки ОС Windows CE и мультимедийного фреймворка Phonon (он используется в KDE 4.0), улучшенная поддержка XML (в том числе, XQuery и XPath), появление API для многопоточных приложений (Concurrency Framework).

Qt 4.4 распространяется в двух версиях: под лицензией GNU GPL (версий 2 и 3) для Open Source-программистов и под коммерческой лицензией для проприетарных разработок.

Началось бета-тестирование OpenOffice.org 3.0

Проект популярного офисного пакета с открытым кодом OpenOffice.org объявил о выпуске бета-версии своего третьего крупного релиза. Среди ключевых изменений в OpenOffice.org 3.0 отмечается появление нового центра запуска (Start Centre), обновление коллекции иконок и элемента управления масштабированием в панели состояния. Среди основных функциональных изменений: в Calc отмечается новый компонент для расчетов, поддержка совместной работы с электронными таблицами, увеличение максимального числа столбцов до 1024; в Writer улучшена работа с заметками, появилось отображение нескольких страниц при редактировании; множество обновлений в Chart; улучшенная функция кадрирования (crop) в Draw и Impress.

Кроме того, в OpenOffice.org 3.0 обещают поддержку грядущего стандарта OpenDocument Format (ODF) 1.2, а также возможность открытия документов, созданных в MS Office 2007 или MS Office 2008 для Mac OS X.

Также сообщается, что OpenOffice.org 3.0 станет первым релизом, который будет работать в Mac OS X без X11 с интерфейсом других приложений Aqua. Вместе с тем появится частичная поддержка VBA (Visual Basic for Applications) для этой плат-

формы, будет обеспечена интеграция с различными API системы, достигнута лучшая доступность открытого офиса из других приложений Mac OS X.

Skype проиграла дело против сторонников лицензии GPL

Харальд Вельте (Harald Welte) объявил о том, что Skype проиграла в судебном разбирательстве против энтузиастов, защищающих лицензию GNU GPL. Вельте возглавляет проект gpl-violations.org, специализирующийся на поиске нарушений лицензии GNU GPL, под которой распространяется свободное программное обеспечение (СПО). Еще в прошлом году Мюнхенский районный суд (Германия) признал Skype виновной в нарушении GPL по делу, инициированному gpl-violations.org. Однако недавно состоялось еще одно слушание, в котором представители Skype попытались апеллировать к судье, утверждая, что лицензия GPL неправомерна в принципе. Но суд отдал предпочтение сторонникам свободного программного обеспечения, отклонив апелляцию Skype.

Причиной изначальных претензий к Skype, по которым компания и была признана виновной в 2007 году, послужил факт распространения телефонов, в программном обеспечении которых содержался лицензированный под GPL код из Linux, однако модифицированный код не был обнародован, чего требует лицензия на свободное ПО.

VIA открыла 16 тысяч строк кода под лицензией GPL

Компания VIA обнародовала более 16 тысяч строк кода из framebuffer-драйвера для Linux под лицензией GNU GPLv2.

Еще на мероприятии Linux Foundation Austin Summit от представителей VIA стало известно, что компания планирует начать Open Source-инициативу, подобную той, что недавно запустила AMD. И вот VIA объявила о выпуске значительного массива кода, относящегося к устройству framebuffer в Linux-ядре. Драйвер поддерживает VIA Unichrome CLE266, K400, K800, PM800, CN700, CX700, K8M890, P4M890, P4M900 и VX800, а ответственным за него назначен сотрудник компании Джозеф Чан (Joseph Chan).

Американские фондовые биржи выбирают Red Hat Linux

Американская компания Red Hat, ведущий мировой Linux-поставщик, объявила о том,

что NYSE Euronext, управляющая такими фондовыми биржами, как NYSE, Euronext, Liffe, Alternext и NYSE Arca Options, использует ее решения.

В истории успеха сообщается, что в NYSE Euronext работают с Red Hat Enterprise Linux и Red Hat Network для «поддержания надежности, высокой производительности и безопасности критически важной финансовой платформы». Благодаря решениям Red Hat в NYSE Euronext получили большую гибкость, свободу от поставщика, экономическую выгоду и возможность справляться с серьезными нагрузками, сохраняя высокий уровень производительности.

Нью-Йоркская фондовая биржа (NYSE), управляемая NYSE Euronext, — крупнейшая в мире и главная в США. По данным на 31 марта 2008 года, совокупная капитализация представленных в NYSE Euronext компаний, общее количество которых приближается к 4 тысячам, составляет 27,3 триллионов USD.

Fedora 9 — новый релиз бесплатного GNU/Linux-дистрибутива

Вышла новая версия популярного дистрибутива GNU/Linux, разрабатываемого Open Source-сообществом при поддержке компании Red Hat, — Fedora 9.

Среди ключевых изменений и обновлений в Fedora 9 разработчики отмечают: обновление инсталлятора системы Anaconda (с возможностями изменения размера разделов файловых систем ext2/3 и NTFS, создание зашифрованных файловых систем и установки на них дистрибутива); интеграцию фреймворка PackageKit для (независимого от дистрибутива) управления пакетами програм-

многo обеспечения — он пришел на смену Pirut, а его графический менеджер обновлений теперь используется вместо Pup; FreeIPA для управления учетными записями и правами доступа (это решение комбинирует службу каталогов Fedora Directory Server с FreeRADIUS, MIT Kerberos, NTP и DNS); поддержку новой файловой системы Ext4; Upstart — основанную на событиях замену демону /sbin/init.

Из новинок в области программного обеспечения в Fedora 9 можно найти следующее: Linux-ядро 2.6.25 (с автоматическими уведомлениями о падениях ядра на kerneloops.org); графические среды GNOME 2.22, KDE 4.0.3 и Xfce 4.4.2; менеджер сетевых подключений NetworkManager 0.7 с улучшенной поддержкой мобильных сетей (особенно отмечаются устройства GSM и CDMA); веб-браузер Firefox 3 Beta; офисный пакет OpenOffice.org 2.4; TeXLive взамен старого дистрибутива TeX; язык программирования Perl 5.10.0; полностью свободное окружение Java — OpenJDK 6.

Exherbo — новый Linux-дистрибутив от разработчика Gentoo

Брайан Остергаард (Bryan Ostergaard), бывший разработчик Gentoo, представил новый Linux-дистрибутив, получивший название Exherbo.

Работа над дистрибутивом ведется уже несколько месяцев, а идея его создания пришла вследствие опыта, полученного за годы участия в проекте Gentoo. Среди основных отличий от своего идеологического вдохновителя отмечаются новый формат пакетов (схожий с Gentoo ebuild) и некоторые проекты вроде измененной системы инициализации (init system).

На данный момент не существует образов для установки системы, однако дерево пакетов открыто для всех желающих. Впрочем, разработчики не рекомендуют даже помышлять о сколь-нибудь серьезном использовании Exherbo на данной стадии.

Openmoko переходит с GTK+ на Qtoria

Энтузиасты из проекта Openmoko объявили о существенных изменениях в программной базе своей платформы. В частности, сообщается о переводе основных приложений с GTK+ на Qtoria.

Проект Openmoko занимается разработкой программной Open Source-платформы на базе Linux для мобильных устройств и созданием первого полностью свободного смартфона Neo Freerunner (первая модель телефона была доступна для энтузиастов в виде Neo1973 в середине прошлого года). Qtoria — полноценная платформа для мобильных устройств, созданная в компании Trolltech на базе их другой разработки — фреймворка Qt.

Как стало известно от авторов Openmoko, теперь в качестве оконного менеджера вместо matchbox используется Enlightenment (E17), а основные приложения переводятся на Qtoria вместо собственных разработок на базе GTK+. Отдельно отмечается, что несмотря на это изменение, приложения на базе GTK+ будут полностью поддерживаться.

Trolltech еще в сентябре 2007 года портировала Qtoria на тестовый смартфон Neo1973 (на базе платформы Openmoko GTA01).

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

Обзор DragonFly BSD: полет стрекозы

Экскурс в историю

Проект под названием DragonFly BSD (<http://www.dragonflybsd.org>) появился на свет в июне 2003 года. Основателем стал Мэтью Диллон (Matthew Dillon), один из разработчиков операционной системы FreeBSD. DragonFly BSD по замыслу и словам его создателя должен был стать «логическим продолжением FreeBSD 4.x».

Такова официальная версия создания. Фактически же отправной точкой для создания проекта стали разногласия Диллона с другими разработчиками FreeBSD. Объектом спора было различие мнений

в вопросе мультипроцессорной поддержки в FreeBSD версии 5. Мэтью считал, что выбранный путь развития в конечном итоге приведет к неэффективному коду, который к тому же будет трудно поддерживать. В итоге Диллон лишил возможности вносить изменения в исходный код ОС. Результатом этих событий и стало появление данной замечательной ОС.

Некоторое время проект развивался практически незаметно. На официальном сайте можно было скачать исходные тексты, самостоятельно собрать их и ознаком-

иться с работой новой ОС. Тем временем в ядре системы оставалось все меньше и меньше от ее прародителя.

С начала 2004 года начали появляться первые образы ISO, еще не имевшие установщика, так что систему предполагалось перенести вручную. А в июне того же года появился предварительный релиз DragonFly BSD. Отличался он тем, что уже имел инсталлятор, упрощающий некоторые операции по установке системы. 12 июля 2004 года состоялся первый официальный релиз под номером 1.0 (<http://leaf.dragonflybsd.org/mailarchive/kernel/2004-07/msg00143.html>). С этого момента систему можно считать пригодной к использованию.

Проект достаточно быстро развивается. Первоначальный релиз содержал толь-

ко то, что входило в рамки базовой ОС: ни системы портов или пакетов, ни каких-либо дополнительных приложений не было. Сейчас в качестве репозитория используется система pkgsrc, заимствованная из проекта NetBSD. Существует и репозиторий бинарных пакетов, собранных из репозитория исходных текстов.

Особенности DragonFly BSD

Но что же может выделить «Стрекозу» (а именно так переводится название) из ряда ОС и делает ее не просто клоном FreeBSD, а уникальной системой?

Как театр начинается с вешалки, так и любая ОС начинается с установщика. Создатели DragonFly не стали изображать велосипед и взяли в качестве штатного инсталлятора BSD Installer (<http://www.bsdinstaller.com>). Надо заметить, что установочный диск одновременно является и LiveCD, что должно помочь, например, в случаях восстановления системы. Текстовый интерфейс инсталлятора достаточно удобен и позволяет быстро установить систему. Конечно, по возможностям он не дотягивает до установщика FreeBSD, зато превосходит по удобству OpenBSD. Интересно, что BSD Installer был разработан с целью стать универсальным инсталлятором для любой BSD-системы и впервые стал использоваться в DragonFly.

Для непосвященного пользователя установленная система практически ничем не отличается от любой BSD-системы. Все достаточно традиционно, утилиты на своих местах, идеология выдержана. Стандартная установка минимальна: в нее не входит ничего, кроме того, что нужно для первичного запуска системы, а все остальное доставляется пакетами по мере необходимости. Среди основных утилит в стандартной установке: GCC 3.4 и 4.1, Sendmail 8.4.12, OpenSSL 0.9.8g, OpenSSH 4.7p1.

Конечно, есть и различия. К примеру, в DragonFly существует возможность предварительной загрузки разделяемых библиотек в память, что ускоряет загрузку использующих их приложений. Другой особенностью является технология виртуального ядра, позволяющего выполнять ядро системы в пользовательском пространстве. Эти давно реализованные в Linux возможности впервые в мире BSD применены именно в DragonFly.

Еще одной уникальной технологией являются вариантные символические ссылки (varsyms). В отличие от традиционных символических ссылок varsyms позволяют включать в наименование исходного файла ссылки имя переменной,

в зависимости от значения которой ссылка будет указывать на различные файлы. Другой не менее уникальной возможностью является checkpointing – сохранение образа выполняемой программы с возможностью последующего продолжения с места сохранения на любой совместимой системе.

Для повышения надежности функционирования системы в целом был создан драйвер контроллера ECC-памяти, функционирующий аналогично драйверу для дисков, поддерживающих SMART. Драйвер, работая напрямую с контроллером памяти, опрашивает его каждую секунду, отслеживая появление невосстановимых ошибок.

Есть и другие утилиты и возможности, среди которых – новый почтовый агент dma (<http://leaf.dragonflybsd.org/mailarchive/submit/2008-02/msg00000.html>) и журналирование смонтированных файловых систем. Более полный список можно найти на Wiki-странице проекта (<http://wiki.dragonflybsd.org/index.cgi/DragonFlyTechnologies>).

Отличия ядра

Но основные изменения находятся под «капотом» – в ядре ОС. Здесь DragonFly BSD шагнула далеко вперед от своего предка.

Для начала остановлюсь на том, что ядро построено вокруг модели легковесных нитей ядра (LWKT, Light Weight Kernel Threads). Для каждого процесса в системе имеется ассоциированная с ним нить, а большинство процессов ядра – это фактически всего лишь чистые нити. Архитектура ядра приближается к микроядерным системам. Это значит, что традиционный механизм системных вызовов заменен на механизм сообщений и их очередей, похожий на тот, что можно найти в ядре MacOS.

У модели LWKT в ядре DragonFly BSD имеется ряд ключевых особенностей, независимых от архитектуры. Эти особенности были разработаны для устранения или уменьшения конкуренции между процессорами. Для каждого процессора в системе есть собственный LWKT-планировщик (scheduler). Нити умышленно привязываются к своим процессорам и могут быть перемещены на другой процессор только при наступлении некоторых особых условий. Нить никогда не будет вытеснена на другой процессор, пока она выполняется в ядре, и нить никогда не будет перемещаться между процессорами, когда она заблокирована.

Любая операция планирования LWKT на конкретном процессоре может выпол-

няться только непосредственно на этом процессоре. Это означает, что LWKT-планировщик может производить планирование и переключение нитей, не используя каких-либо блокировок. Таким образом, на уровне многопроцессорной системы кроме критических секций более никаких блокировок не используется.

В модели LWKT для коммуникации между процессорами используется система асинхронного обмена обработки сообщениями. Делая вызов функции, вы просто обеспечиваете целевой процессор ссылкой на функцию и данными, которые этот процессор обрабатывает асинхронно. Более подробно об особенностях реализации LWKT можно узнать на странице проекта (<http://www.dragonflybsd.org/threads.shtml>).

Последние тенденции и перспективы

Со времени первого анонса проекта к текущему официальному релизу 1.12.2 направление развития DragonFly также изменилось. Смещаясь по мере развития от системы общего назначения, на данный момент основной целью разработчиков является превращение DFly в систему, нацеленную на построение кластеров, обеспечивающую «прозрачную» кластеризацию через Интернет. В частности, активно ведется разработка файловой системы HAMMER, поддерживающей кластеризацию, журналирование и создание снимков файловой системы. С концепцией данной ФС можно ознакомиться в почтовых архивах (<http://leaf.dragonflybsd.org/mailarchive/kernel/2007-02/msg00097.html>).

Вообще основной целью последнего релиза была заявлена чистка и стабилизация кода. Но фактически в него вошло множество значительных изменений и новых возможностей. В частности, появилась поддержка Bluetooth, USB- и WiFi-стеки подверглись переработке, став более стабильными и получив поддержку новых устройств. Ядро получило начальную поддержку AMD64, одновременно избавившись от устаревших сетевых протоколов, таких как ARCNet, FDDI. Драйвер SATA теперь поддерживает и AHCI-контроллеры. Также в данный релиз вошла первая версия dma (DragonFly Mail Agent), призванная заменить в будущем Sendmail.

Сейчас идет работа по подготовке релиза версии 2.0, намеченного на июль этого года. Основным ожидаемым новшеством будет включение бета-версии файловой системы HAMMER. Другой же не менее ожидаемой вещью станет переработка сетевого стека для поддержки многопроцессорных систем.

Помимо этого в релиз должны войти такие изменения, выполняемые в рамках Google Summer of Code (GSoC), как новый планировщик, выделяющий ресурсы на основании динамического вычисления потребностей каждого процесса (proportional share userland scheduling algorithm); расширение поддержки многопроцессорных систем и очистка ядра от наследия в виде Big Giant Lock; разработки более эффективного планировщика дисковых операций и т. д. До этого момента планируется промежуточный

релиз для окончательной стабилизации системы.

К сожалению, несмотря на свои многочисленные достоинства, DragonFly BSD не может похвастаться большим числом разработчиков. Это, конечно, несколько сказывается на скорости развития, но в этом году проект получил нескольких студентов благодаря программе GSoC, что должно положительно сказаться на разработке.

Очень хочется надеяться, что «Стрекоза» будет развиваться и дальше, а дан-

ная статья хоть немного поспособствует популяризации данного проекта. Более подробно ознакомиться с данной ОС можно на официальном сайте проекта www.dragonflybsd.org. Там же можно найти ссылки на официальные списки рассылок. Существует и блог с RSS-каналом (<http://www.shiningsilence.com/dbsdlog>), где ежедневно освещается текущее состояние дел в проекте.

Дмитрий Комиссаров
(dxj@mail.ru)

FOSS Review 002

htop

- ✓ Версия: 0.7.
- ✓ Лицензия: GPL.
- ✓ Размер: 144 Кб (tar.gz).
- ✓ Сайт: <http://htop.sourceforge.net>.

Консольное приложение, являющееся хорошей заменой стандартному системному менеджеру процессов top. Главное отличие состоит в наглядности и интерактивности интерфейса. Больше не нужно запоминать PID процесса, чтобы его «убить» — можно пользоваться мышью.

Да и вообще — htop загружается быстрее своего «старшего брата», так как ему не требуется собирать системную информацию перед запуском.

PCMan File Manager

- ✓ Версия: 0.4.1.1.
- ✓ Лицензия: GPL.
- ✓ Размер: 971 Кб (tar.gz).
- ✓ Сайт: <http://pcmanfm.sourceforge.net>.

Легкий, быстрый однопанельный файловый менеджер. PCMan обладает широкими функциональными возможностями, такими как поддержка табов (как в Firefox), работа со сменными носителями, организация рабочего стола (с ярлыками и файлами папки Desktop), отрисовка миниатюр для изображений, просмотр файлов в виде списка. Интерфейс программы, написанный с помощью GTK+, очень похож на Nautilus, но, в отличие от последнего, PCMan не привязан к библиотекам среды GNOME.

На современном компьютере на запуск программы уйдет чуть более секунды.

Brasero

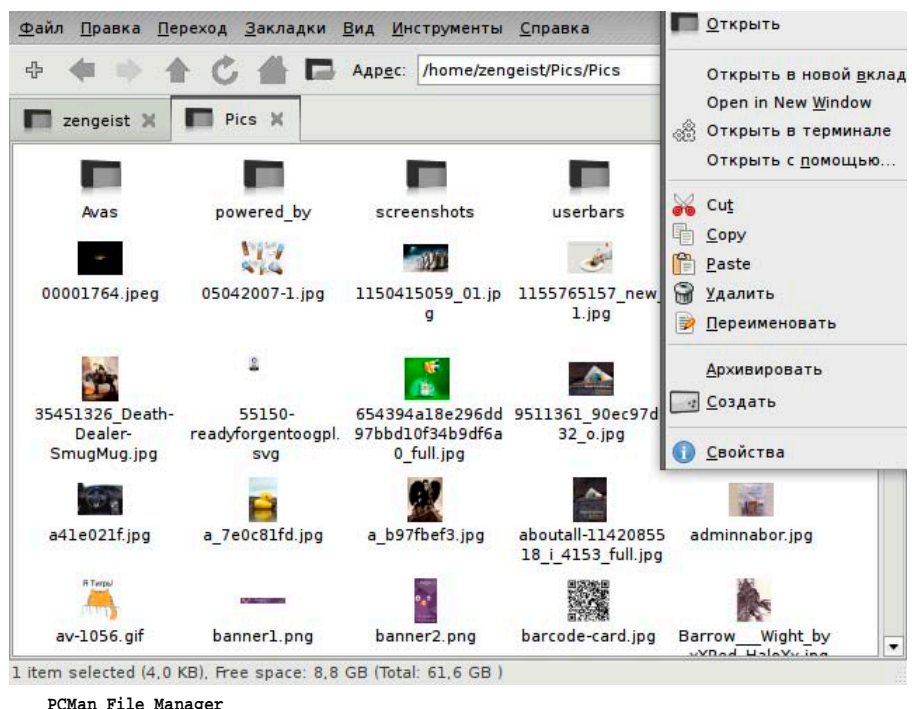
- ✓ Версия: 0.7.1.
- ✓ Лицензия: GPL.
- ✓ Размер: 1.5 Мб (tar.gz).

- ✓ Сайт: <http://www.gnome.org/projects/brasero>.

Tasks: 194 total, 3 running
Load average: 0.06 0.17 0.16
Uptime: 02:51:41

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
14392	root	20	0	458M	77112	11812	R	1.3	7.6	1:54.76	/usr/bin/X :0 -br
21772	zengest	20	0	19400	1324	992	R	1.3	0.1	0:00.14	htop
14623	zengest	20	0	433M	33560	17380	R	1.3	3.3	0:28.48	nautilus --no-defa
15015	zengest	20	0	269M	21840	10152	S	1.3	2.1	0:17.58	transmission
5356	zengest	20	0	24156	1172	1064	S	1.3	0.1	0:02.41	hald-addon-storage
15014	zengest	20	0	269M	21840	10152	S	0.7	2.1	0:13.82	transmission
14622	zengest	20	0	335M	35624	17216	S	0.0	3.5	0:13.82	gnome-panel --sm-c
14693	zengest	20	0	221M	23740	6000	S	0.0	2.3	0:14.78	/usr/bin/compiz.re
21753	zengest	20	0	207M	17396	6828	S	0.0	1.7	0:00.30	gnome-screenshot -
15676	zengest	20	0	690M	192M	26224	S	0.0	19.3	1:27.76	/usr/lib/firefox-3
16225	zengest	20	0	433M	33560	17380	S	0.0	3.3	0:01.17	nautilus --no-defa
14817	zengest	20	0	170M	16268	6560	S	0.0	1.6	0:01.04	/usr/lib/notificat
16235	zengest	20	0	274M	25704	12644	S	0.0	2.5	0:00.82	gnome-terminal
15567	zengest	20	0	407M	41940	17716	S	0.0	4.1	0:04.56	gedit file:///home
1	root	20	0	4020	620	536	S	0.0	0.1	0:00.98	/sbin/init
2746	root	16	-4	17184	488	384	S	0.0	0.0	0:00.30	/sbin/udevd --daem

htop



лочкой к стандартным консольным приложениям, таким как dvdrrtools, cdrtools, mkisofs и cdrdao.

Запись дисков по умолчанию осуществляется при помощи библиотеки libburn.

Помимо типичных для этого класса программ функций доступны такие, как запись сис-файла, копирование диска на другой носитель, просмотр изображения, аудио- и видеофайла перед записью, удаление содержимого перезаписываемого CD или DVD.

Возможности Brasero можно расширить при помощи плагинов.

Интерфейс программы, схожий с Nero и k3b, очень удобен.

Диалог записи может быть скрыт в системный трей, а после записи Brasero предложит проверить целостность данных на диске.

CoverFinder

- ✓ **Версия:** 0.1.1.
- ✓ **Лицензия:** GPL.
- ✓ **Размер:** 107 Кб (tar.gz).
- ✓ **Сайт:** <http://mookooh.org/coverfinder>.

Многие современные мультимедийные плееры показывают обложки музыкальных альбомов. Согласитесь: красиво, когда обложка есть. А что делать, если таковой нет?

Правильно: найти в сети. CoverFinder – небольшая программа, которая с легкостью справляется с поиском и сохранением на ваш жесткий диск обложек альбомов, используя Web API портала Amazon.com.

SoundConverter

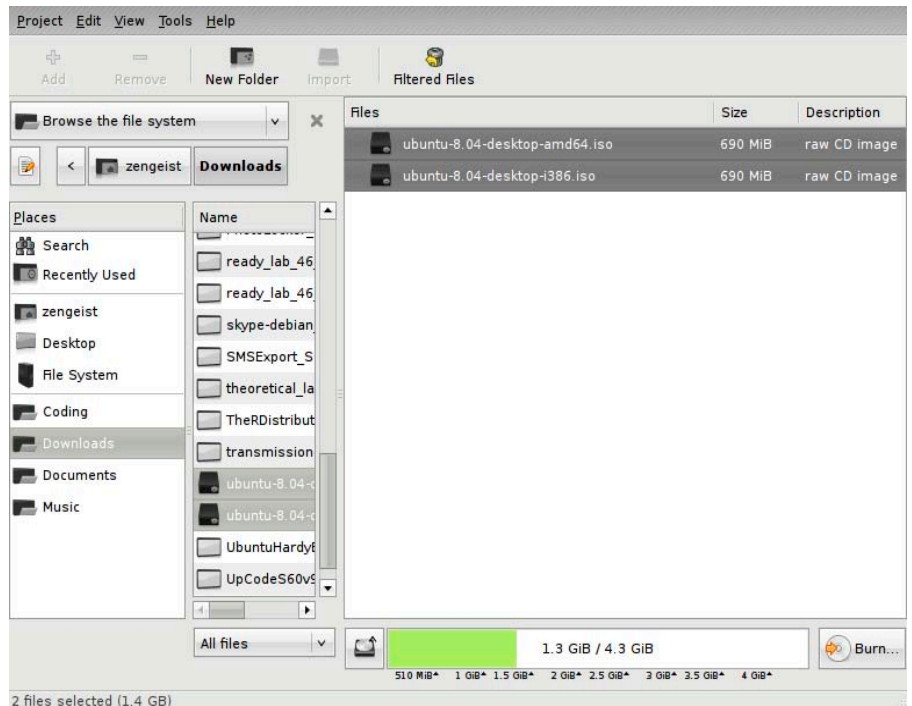
- ✓ **Версия:** 1.2.0.
- ✓ **Лицензия:** GPL.
- ✓ **Размер:** 253 Кб (tar.gz).
- ✓ **Сайт:** <http://soundconverter.berlios.de>.

Многие музыкальные плееры читают музыкальные файлы только в формате MP3. Если у вас есть музыка в отличном от MP3 формате, то прослушать ее в плеере не удастся. Решить проблему поможет SoundConverter – программа для перекодировки музыкальных файлов.

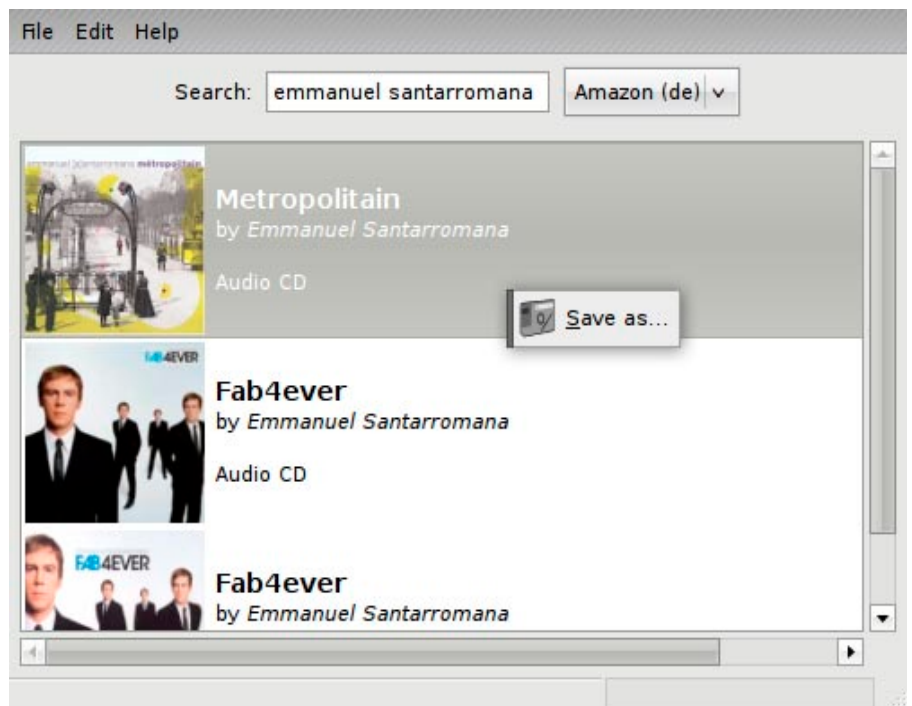
Приложение понимает все форматы файлов, которые может читать библиотека GStreamer (Ogg Vorbis, AAC, MP3, FLAC, WAV, AVI, MPEG, MOV, M4A, AC3, DTS, ALAC, MPC, Shorten, APE, SID и другие), перекодировав их в MP3, FLAC, WAV или Ogg Vorbis.

Transmission

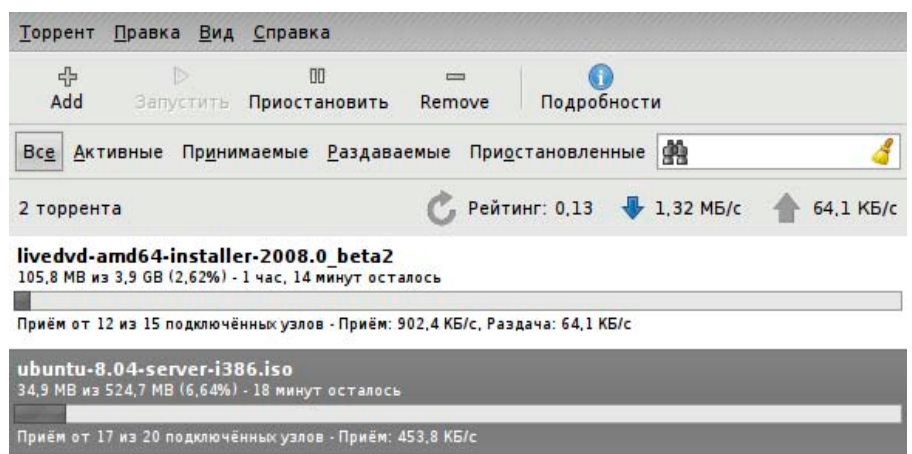
- ✓ **Версия:** 1.21.
- ✓ **Лицензия:** MIT.



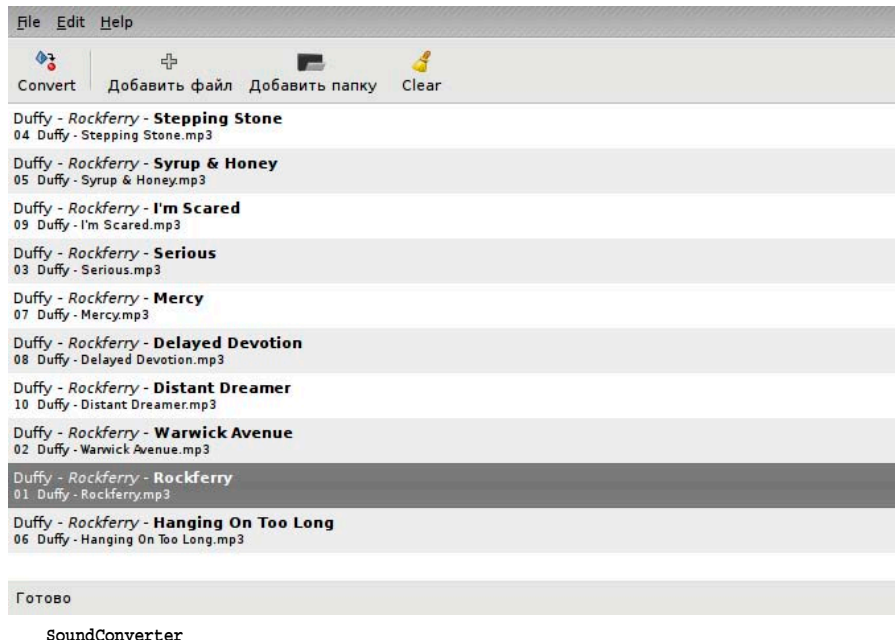
Brasero



CoverFinder



Transmission



- ✓ **Размер:** 3,8 Мб (tar.bz2).
- ✓ **Сайт:** <http://www.transmissionbt.com>.

Главные достоинства этого torrent-клиента – удобство, легкость и простота. Transmission не перегружен тяжелым, зачастую ненужным, функционалом и с блеском справляется с непосредственной задачей – загрузкой файлов.

Окно программы прячется в системный трей, а о завершении загрузки сообщит всплывающее напоминание.

Присутствует функция ограничения по скорости и подробная информация о torrent-файле, в последней версии добавилась поддержка многотрекерных торрентов.

Роман Комков
(r.komkov@gmail.com)

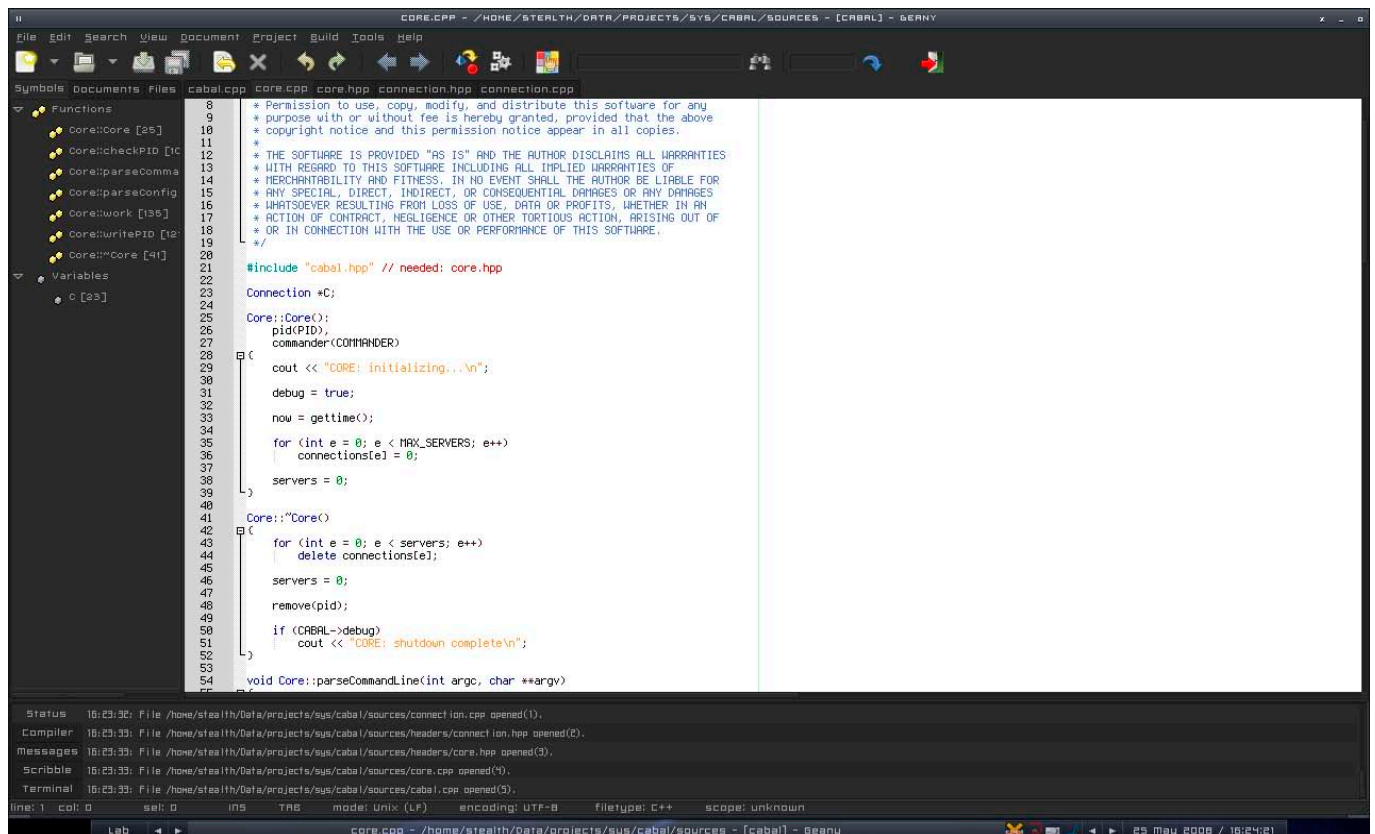
Geany: интегрированная среда разработки приложений

3 года назад мне потребовалась программа для написания курсового проекта по дисциплине «Информатика» в 11-м классе. Проект заключался в

построении сайта средствами XML + XSLT, который позднее был представлен на городском конкурсе «Искатель» и занял первое место. Однако найти такую програм-

му в далёком 2005 году было весьма не просто. Большинство приложений подобного рода (а мне была нужна именно IDE – Integrated Development Environment, «интегрированная среда разработки») были написаны на языке Java: знаменитый jEdit, молодой тогда ещё Eclipse, платный XML-редактор oXygen с некоторыми функциями IDE.

Неудобства заключались в следующем: постоянное «пожирание» памяти виртуальной машины Java на моём довольно немоощном ПК давало о себе знать



Общий вид Geany

примерно каждые полчаса – с большим количеством текста программы не справлялись; отсутствие дополнительных возможностей, таких как встроенный файловый менеджер, терминал, сказывалось на моей производительности, потому что каждый раз приходилось «бегать» по файловой системе в поисках нужной информации и использовать её для копировки.

Конечно, были и родные аналоги: в Bluefish (<http://bluefish.openoffice.nl>) я сделал несколько простых веб-страниц, но для работы с серьёзным проектом её было явно недостаточно. Многие спросят о KDevelop или Anjuta, последняя из коих на то время был вообще в не-потребном состоянии. Но данные пакеты тянули за собой множество зависимостей из библиотек, а об их стабильности и говорить не приходилось. Кроме того, для того же создания графического интерфейса был разработан Glade (<http://glade.gnome.org>).

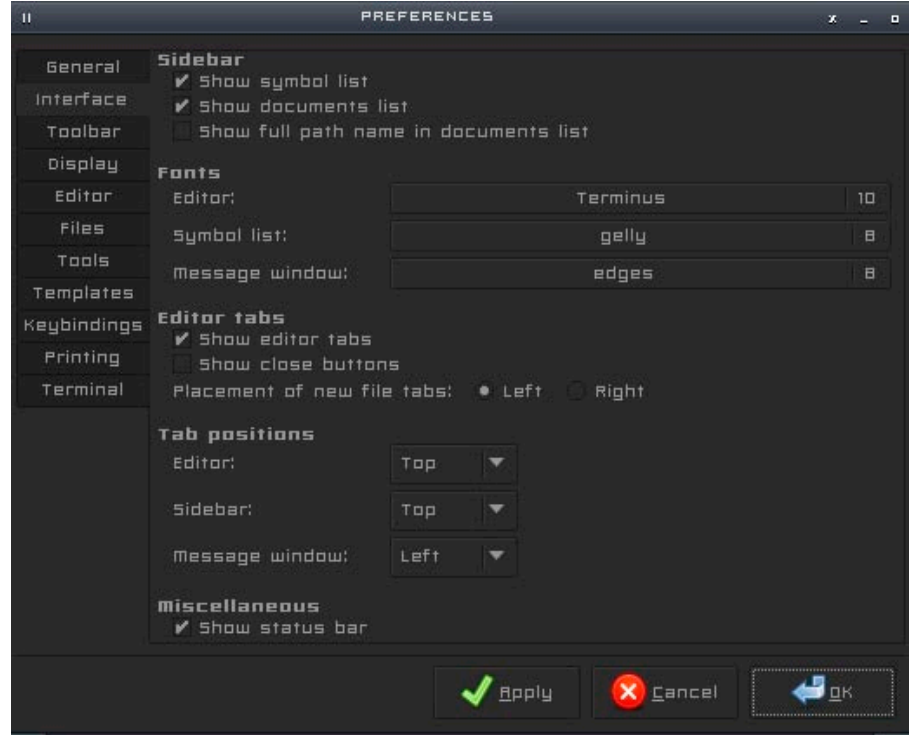
Первое знакомство

Желание найти что-то подходящее было сильнее лени, потому как это могло упростить работу не только над текущим проектом, но и в будущем по специальности вообще. Блуждая по просторам Сети, я наткнулся на релиз Geany 0.1. Тогда сайт (<http://geany.uvena.de>) смотрелся весьма примитивно и блекло на фоне домашних страниц многих других законченных проектов. В то же время он был удобен для навигации и обладал исчерпывающей информацией о программе и её возможностях. В первых версиях они были невелики, но очень чётко обозначены и имели некую логическую законченность: это редактор с подсветкой синтаксиса основных языков (C/C++, Java, Perl, PHP, XML); список открытых файлов (вкладок) для редактирования; поддержка разных типов окончания строки и кодировок; автоматическое дополнение в XML-тегах.

Но самое главное, что меня тогда привлекло в программе – это её размер при таком количестве возможностей. Всего несколько десятков килобайт. Этот довольно лёгкий и быстрый редактор с возможностями IDE сразу показал себя с хорошей стороны, когда я написал первую законченную часть проекта (он был посвящён Сталинградской битве, поэтому не сложно представить, сколько текстовой информации пришлось обработать и хранить одновременно в окнах Geany).

Основные возможности

С течением времени проект развивался (первые 5 релизов выходили с час-



Настройки Geany

тотой в 2-3 недели), и количество возможностей Geany постоянно увеличивалось. При этом сохранялся принцип KISS (Keep It Simple, Stupid), который так нелегко отыскать в современных проектах. Программа оставалась лёгкой, быстрой и функциональной.

Итак, на сегодняшний день Geany включает в себя такие возможности:

- ☑ подсветка синтаксиса множества языков программирования, скриптинга, разметки и даже запросов: ASM, C, C++, CAML, CSS, D, Diff, Fortran, Haskell, X(HTML), Java(Script), LaTeX, Lua, Make, Pascal, Perl, PHP, Python, Ruby, Sh, SQL, TCL, XML и др.;
- ☑ автоматическое определение конструкций кода, которые можно сворачивать (code folding);
- ☑ автодополнение известных конструкций, таких как if, for, while и т.д.;
- ☑ автодополнение тегов XML и HTML;
- ☑ автоматическое определение и создание отступов;
- ☑ списки символов функций, классов, объектов;
- ☑ поддержка построения проекта (compile, link, run);
- ☑ встроенный эмулятор терминала посредством VTE;
- ☑ дополнительные возможности: диалоговое окно выбора цвета, поиск, настройка «горячих» клавиш, поиск;
- ☑ поддержка встроенных и созданных пользователем шаблонов;
- ☑ менеджер проектов (обеспечивает лёгкую работу с несколькими работами);

- ☑ интерфейс для написания модулей к Geany.

Нужно сказать, что последнее – очень важная особенность, которой не хватает многим современным IDE. Благодаря ей программу легко «пичкать» новым функционалом от сторонних разработчиков.

Секреты мастерства

Конечно же, основа любого, даже самого простого, но удобного редактора для программиста – это подсветка синтаксиса. У Geany не свой движок, написанный с нуля. Энрико Трогер (Enrico Troger), замечательный программист, написавший несколько программных продуктов для рабочей среды XFCE, пошёл по пути наименьшего сопротивления: в Geany он использует наработки людей, специализирующихся на подсветке синтаксиса, – движок Scintilla (<http://scintilla.org>), которому очень много лет, но его разработка до сих пор идет в правильном направлении.

После того как проект встал на ноги, потребовалась очередная полезная возможность – наличие встроенного терминала. Используя библиотеку VTE, реализовать его было довольно просто. Зато после появления терминала Geany позволила пользователям отказаться от лишних окон для прямой работы с проектами.

Таким образом, Geany на первых стадиях, возможно, и не представила ничего сверхъестественного для IDE – она лишь вобрала в себя всё лучшее от существующих библиотек и предостави-

Электронное приложение «Open Source»

ла к этому понятный, простой и удобный интерфейс.

В одном из последних релизов появился модульный API, в результате чего некоторые функции (встроенный файловый менеджер, автосохранение, создание diff-файлов, экспорт и др.) были вынесены в модули, которые можно включать/отключать посредством менеджера плагинов в программе. Модульный интерфейс сделал программу очень мощным инструментом, благодаря которому теперь можно повышать не только свою производительность, но и скорость загрузки/работы самой Geany. Ненужные модули отключаются простым кликом мышки.

Настройка

Управление всем функционалом, который сейчас способна предоставить Geany, очень важно правильно организовать. Настроек действительно много.

Вот они:

- ✓ сессии и сохранение файлов внутри проекта;
- ✓ настройка сообщений;
- ✓ расположение основных элементов интерфейса (редакторское окно, панель инструментов, список открытых файлов, список символов);
- ✓ редакторское окно: отступы, шрифты, цвета;
- ✓ установка типов файлов и кодировки по умолчанию;

- ✓ основные параметры шаблонов;
- ✓ горячие клавиши;
- ✓ параметры печати;
- ✓ настройки встроенного терминала.

Шаблоны и модули

Развитие успешного проекта повлекло за собой интерес сторонних разработчиков. Моё первое письмо к Энрико заключалось в просьбе добавить подсветку синтаксиса для языка Lua и шаблон для лицензии BSD. С тех пор многие помогали улучшать программу и наращивать функционал.

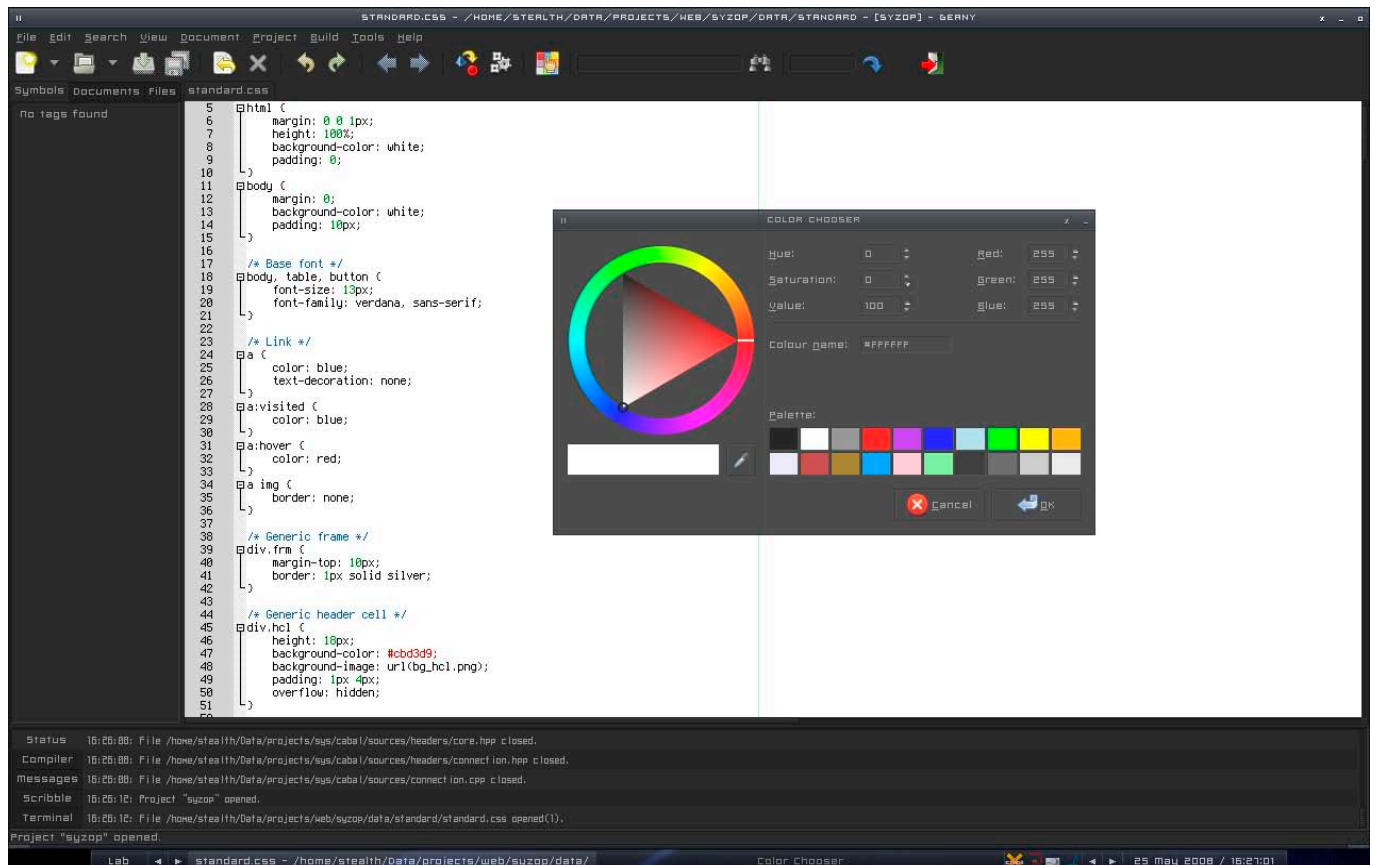
Помимо множества шаблонов, среди которых – различные лицензии, базовые конструкции программ, время/дата и другое, в настоящее время создано несколько проектов, занимающихся разработкой модулей для Geany:

- ✓ **GeanyLua** – плагин для использования скриптов на языке Lua (графические элементы, события) (<http://yetanothergeek.justfree.com/geanylua/>);
- ✓ **GeanyDebug** – плагин для использования GNU debugger (gdb) в Geany (<http://yetanothergeek.justfree.com/geanydebug/>);
- ✓ **GeanySendMail** – плагин для отправки текущего файла в качестве прикрепленного в выбранном почтовом клиенте (<http://frank.uvena.de/geany-plugins/geanysemail/geanysemail.php>);

- ✓ **geanyvc** – плагин для работы с системами управлением исходным кодом: CVS, Bazaar, GIT, Mercurial, Subversion, SVK (<http://users.cosmostv.by/yurand/geanyvc/>);
- ✓ **GeanyLaTeX** – плагин, улучшающий использование файлов LaTeX в Geany (<http://frank.uvena.de/geany/plugins/geanylatex/geanylatex.html>);
- ✓ **geanyprj** – альтернативный менеджер управления проектами (<http://users.cosmostv.by/yurand/geanyprj/>);
- ✓ **geanydoc** – позволяет выполнять команды при выборе слова рядом с текущей позицией курсора (<http://geany-plugins.sourceforge.net/geanydoc/>);
- ✓ **Spell Check** – проверка орфографии в Geany при помощи Aspell (<http://geany-plugins.sourceforge.net/spellcheck/>);
- ✓ **Backup Copy** – создание временных копий после сохранения файлов проекта (<http://geany-plugins.sourceforge.net/backupcopy/>);
- ✓ **Instant Save** – сохранение новых файлов без лишнего диалогового окна (<http://geany-plugins.sourceforge.net/instant-save/>).

Большинство из них написано энтузиастами, которым не хватало в Geany какой-либо важной для их пользовательского процесса мелочи.

Кроме того, существует список плагинов, представленный пользователями



Выбор цвета в Geany

в качестве пожеланий (<http://geany.uvena.de/Support/Plugins>) к следующим версиям. Многие сторонние разработчики берут из него идеи и создают целые серии плагинов для Geany.

Дополнительные возможности

Многие мелочи в Geany могут приятно удивить.

Например, кнопка с диалоговым окном выбора цвета будет особенно полезна веб-разработчикам, когда не нужно открывать графический редактор или вспомогательную программу для просмотра веб-интерпретации цветового оттенка.

Конечно же, поиск и автоматическая замена для любого редактора является обязательной. Geany здесь не исключение.

Переход к конкретной строке в файле в виде удобного окошка в панели инструментов также помогает при навигации. Автоматическая замена всех символов <tab> на пробелы может помочь разработчикам на языке Python правилам PEP-8 (<http://www.python.org/dev/peps/pep-0008>).

Поддержка

В настоящее время поддержка проекта осуществляется множеством людей: основными координаторами проекта, переводчиками (за время существования Geany программу перевели на 22 языка мира), высылающими патчи по улучшению возможностей и исправлению ошибок (их список можно увидеть здесь: http://geany.svn.sourceforge.net/viewvc/*checkout*/geany/trunk/THANKS).

У проекта существует множество средств связи: IRC-канал #geany

на Freenode (irc.freenode.net), списки рассылки для пользователей (<http://lists.uvena.de/geany>), переводчиков (<http://lists.uvena.de/geany-i18n>) и разработчиков (<http://lists.uvena.de/geany-devel>), а также, конечно, электронная почта.

Заключение

Вот так, спустя 3 года из крохотного проекта развилась полноценная IDE, занимающая высокие места в рейтингах и получающая отличные оценки среди специалистов (<http://linux.softpedia.com>). Но самое главное, что миру Open Source достался продукт, который вполне может конкурировать с тем же Komodo от ActiveState, превосходя последнего в производительности и функциональной мощи.

Влад Глаголев
(stealth@sourcemage.org)

Qt и GTK+: два кита. Часть 3

Наконец мы подошли к графическому интерфейсу, но вначале надо поглядеть, какой механизм обеспечивает его работу. Со времен Turbo Vision (кстати, он разрабатывается до сих пор) эволюция в этом плане не очень продвинулась вперед – разве что функции-обработчики стало удобнее привязывать к событиям.

Прим. ред. Это третья часть статьи, в которой приведено сравнение Qt и GTK+ глазами программиста. Ее первые две части были опубликованы в «Open Source» 024 и 025.

Причинно-следственный механизм

Обе библиотеки «живут» на основе механизма послышки сигналов и реакции на эти сигналы. В GTK практически у каждого визуального объекта есть свойства-сигналы, которые являются указателями на функции обратного вызова (callback functions). Например, у «кнопочного» объекта GtkWidget есть сигнал «clicked». Чтобы связать с нажатием на кнопку какое-то действие, мы назначаем этому сигналу функцию, реагирующую на возникновение сигнала:

```
g_signal_connect (button, "clicked", G_CALLBACK (cb_callback), data);
```

В этом примере:

- ☑ **button** – экземпляр класса GtkWidget;
- ☑ **cb_callback** – функция, вызываемая при нажатии на кнопку (т.е. при получении сигнала «clicked»);
- ☑ **data** – некие данные, которые можно передать в функцию обратного вызова.

Сама же функция может быть примерно такой:

```
void cb_callback (GtkWidget *button, gpointer user_data)
{
    // Тут что-то делаем, а в user_data можем получить значение,
    // переданное в параметре data при вызове функции
    // g_signal_connect
}
```

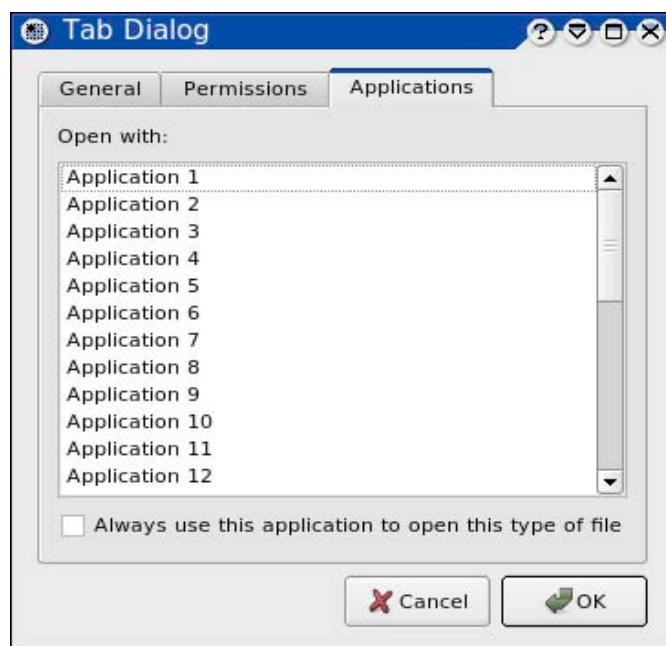
В Qt действует концепция сигналов и слотов. Сигнал в Qt примерно то же, что в GTK, – это «посыл», означающий некое событие. У кнопки класса QPushButton есть сигнал «clicked». Но вместо привязки к сигналу функции обратного вызова мы назначаем посылку этого сигнала в то, что именуется в Qt слотом. Слот в Qt – просто функция-член класса, которая объявлена в разделе slots.

Например, у нас есть кнопка bt_button. Допустим, мы создаем ее в конструкторе класса:

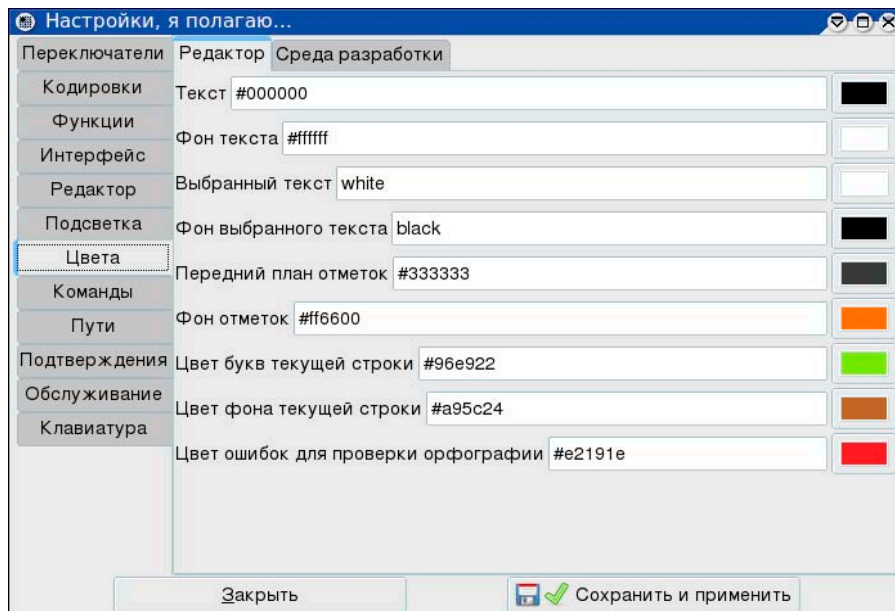
```
QPushButton *bt_button = new QPushButton ("Hello", this);
```

В описании того же класса нам нужно теперь объявить функцию, которая будет у нас вызываться при поступлении от кнопки сигнала clicked:

```
private slots:
void button_clicked();
```



Виджет со вкладками в Qt



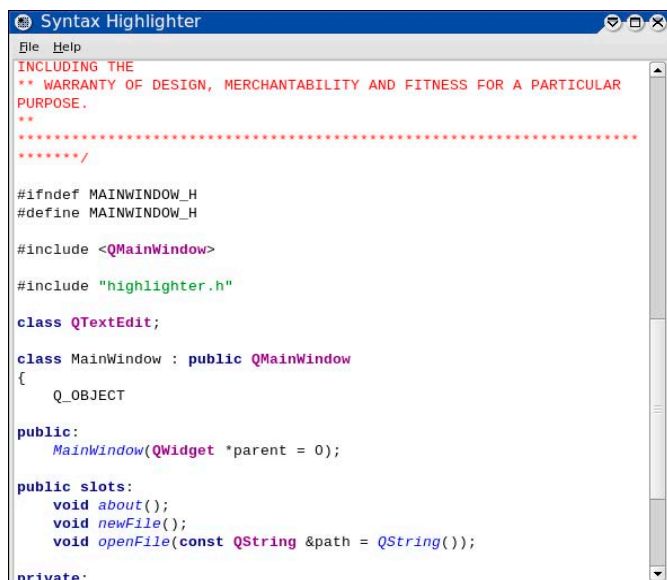
Метки, поля ввода и кнопки в GTK+

Как видно, пока это просто объявление функции – без привязки к конкретному сигналу. Просто мы объявили эту функцию в особой секции, в private slots (можно в public slots или private slots – эти «расширения» C++ на самом деле пропускаются потом через Qt-препроцессор moc). Само собой, надо будет еще написать код функции button_clicked, но это оставим вне нашего примера. Посмотрим теперь, как поставить вызов этой функции в соответствие посылке сигналу clicked. После создания кнопки в конструкторе класса пишем:

```
connect(bt_button, SIGNAL(clicked()), this, SLOT(button_clicked()));
```

Теперь при посылке кнопкой сигнала clicked будет вызвана функция-слот button_clicked. Само слово «слот» несколько путает, поскольку наводит на мысль о некоем разъеме для подключения чего-либо. На самом же деле слоты в Qt – такие же функции, как и остальные. Слоты могут быть виртуальными функциями. В классах есть предопределенные, готовые функции-слоты, их можно переопределять.

Вызов функции-слота после возникновения сигнала срабатывает примерно в 10 раз медленнее, чем вызов обычной callback-функции, о чем разработчики Qt честно предупрежда-



Виджет текстового редактора в Qt

ют в документации. Однако в современном состоянии технологии эта медлительность довольно относительно – на процессоре с частотой в 500 МГц механизм Qt работает на скорости около 2 миллионов сигналов в секунду.

Виджеты

GTK+ и Qt используют схожие подходы к построению интерфейса. Одни виджеты содержат в себе другие виджеты. В Gtk+ есть виджеты-контейнеры, куда можно упаковывать (термин из GTK+) виджеты. Из контейнеров наиболее популярны «коробочки»: GtkHBox и GtkVBox. Отличия состоят в том, что виджеты, помещенные в GtkHBox, выстраиваются по горизонтали, а в GtkVBox – по вертикали. Есть и другие контейнеры.

В Qt всё несколько иначе. Всякий виджет (потомок основного класса – QWidget)

может быть контейнером. Однако в отличие от GTK+ в Qt на расположение виджетов внутри контейнеров влияют не сами виджеты и методы упаковки, а layout-классы. Если мы хотим, например, выстроить виджеты по горизонтали, то кроме помещения виджетов в контейнер надо еще и добавить их в layout класса QHBoxLayout. Таким образом, контейнер содержит в себе виджеты, чье расположение управляется layout, которые бывают вложенными. В каждом из них есть функция addLayout, добавляющая layout к текущему. Это позволяет наполнять контейнеры виджетами в произвольном порядке.

MDI и «табы»

Классический многодокументный интерфейс (MDI), когда окна документов находятся внутри главного окна, в GTK отсутствует. Вместо этого можно либо плодить окна, которые плавают на рабочем столе (как в GIMP), либо использовать виджет GtkNotebook, инкапсулирующий «табовый» движок вкладок с поименованными «корешками». В корешок можно помещать свои виджеты – например, кнопку закрытия корешка.

У Qt есть и MDI (класс-контейнер QMdiArea и оконный класс QMdiSub Window), и вкладки с корешками – виджет QTabWidget. Вкладки имеют заголовок, а вот собственный виджет в этот заголовок поместить нельзя. Однако QTabWidget дает возможность всунуть пользовательский виджет (ту же кнопку закрытия) в угол «блокнота». И в Qt, и в GTK+ можно настраивать расположение корешков: сверху, сбоку, снизу. Кроме того, в Qt два вида корешков: округлые и острые.

Кнопки

Что интересного можно написать о кнопках? Вроде бы кнопки и кнопки, но не так всё просто. В GTK+ кнопки имеют доступ к общему для программ GTK+ ресурсу иконок – это стандартный набор иконок вроде глифов открытия файла, сохранения, разных восклицательных знаков и тому подобного. Всё это называется stock images. «Стоковые» изображения зависят от выбранной пользователем темы. И если вам в Qt перед созданием кнопки с глифом приходится озаботиться иконкой под нее, то в GTK+ достаточно написать нечто вроде:

```
GtkWidget *button = gtk_button_new_from_stock(GTK_STOCK_CANCEL);
```

Отмечу, что при создании такой «стоковой» кнопки мы по-

лучаем не только глиф, но и локализованную надпись. Стоковые идентификаторы могут быть использованы также при создании кнопок на панели инструментов (toolbar) в элементах меню.

Меню

В GTK+ механизм меню отделен от движка панели инструментов, а в Qt они как бы совмещены – экземпляры класса QAction используются и как пункты меню, и как кнопки на панели. Впрочем, технология actions есть и в GTK, и вы можете выбирать, с чем вам удобнее работать: с традиционными меню или с набором actions. Что такое action в GTK+ и Qt? Слово «action» переводится как «действие». В нашем случае мы имеем дело с понятием, обозначающим объект, который заданным вами образом откликается на вызов (например, нажатием кнопки или выбором пункта меню).

В GTK+ можно создать некое «действие», задать ему функцию-обработчик, метку, комбинацию клавиш и потом на основе этого «действия» создать пункт меню или кнопку на панели инструментов. В Qt «действия» сами служат пунктами меню и кнопками на панели.

И в GTK+, и в Qt (о чем мало кто знает) меню могут быть «оторваны» и плавать на экране в отдельных окнах.

Списки

Одна из важнейших задач программирования – вывод неких данных в развернутом списке (не меню), из которого можно выбирать элементы. В GTK+ для этого разработан мощнейший, но довольно низкоуровневый движок – класс GtkTreeView и набор вспомогательных классов и функций. GtkTreeView способен отображать как линейные списки, так и деревья, причем не только строк, но и картинок и вообще любых виджетов. Универсальность класса имеет один недостаток: для удобной работы с элементарным списком строк надо написать много кода. В свое время я написал для себя библиотеку, упрощающую вывод строчковых списков, поскольку использовать для этого GtkTreeView всякий раз напрямую неудобно. На сайте www.gtk.org в разделе документации лежит довольно объемная книжка по работе с GtkTreeView – советую прочитать её вдобавок к документации, идущей в составе самой GTK+.

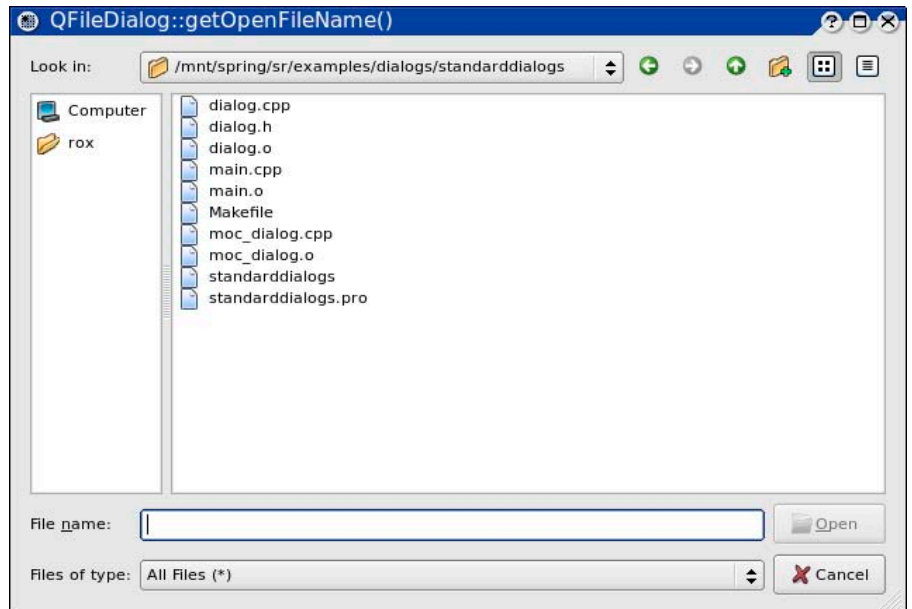
На коротком примере я покажу, как в GtkTreeView *list добавляется строка gchar *s:

```
// Объявляем модель представления данных
GtkTreeModel *model;
// Объявляем итератор, представляющий новый ряд в списке:
GtkTreeIter newrow;
// Получаем модель из списка (предполагается, что она задана ранее)
model = gtk_tree_view_get_model (GTK_TREE_VIEW (list));
// Добавляем в модель новый ряд
gtk_list_store_append (GTK_LIST_STORE (model), &newrow);
// Нулевой колонке нового ряда модели присваиваем значение,
// равное s:
gtk_list_store_set (GTK_LIST_STORE (model), &newrow, 0, s, -1);
```

Теперь то же действие в Qt:

```
list->addItem (s);
```

В Qt просто есть замечательный класс высокого уровня



Диалог выбора файла в Qt

QListWidget, выполняющий за программиста всю возню с моделью.

Работа с текстом

Возможности GTK+ по вводу и выводу текста примерно равны, хотя в GTK+ мы снова сталкиваемся с некоторой недоделанностью. Низкий уровень – пожалуйста, а вот выше – на усмотрение. Возьмем к примеру виджет текстового редактора GTK+ – GtkTextView. Вернее, там два связанных вместе класса: GtkTextView для визуального представления и GtkTextBuffer – для внутренних механизмов. Всё здорово! Текст в UTF-8, разные методы ввода, удобное хождение по тексту с помощью итераторов, возможность отмечать фрагменты текста иными атрибутами шрифта – можно даже виджеты в текст вставлять. Однако нет механизма Undo/Redo. А в Qt у аналогичного виджета QTextEdit такой механизм есть. И готовый класс для динамической подсветки синтаксиса тоже есть. Однако потенциал в редактировании текста что в Qt, что в GTK+ одинаков – просто разработчики Qt решили программистов порадовать.

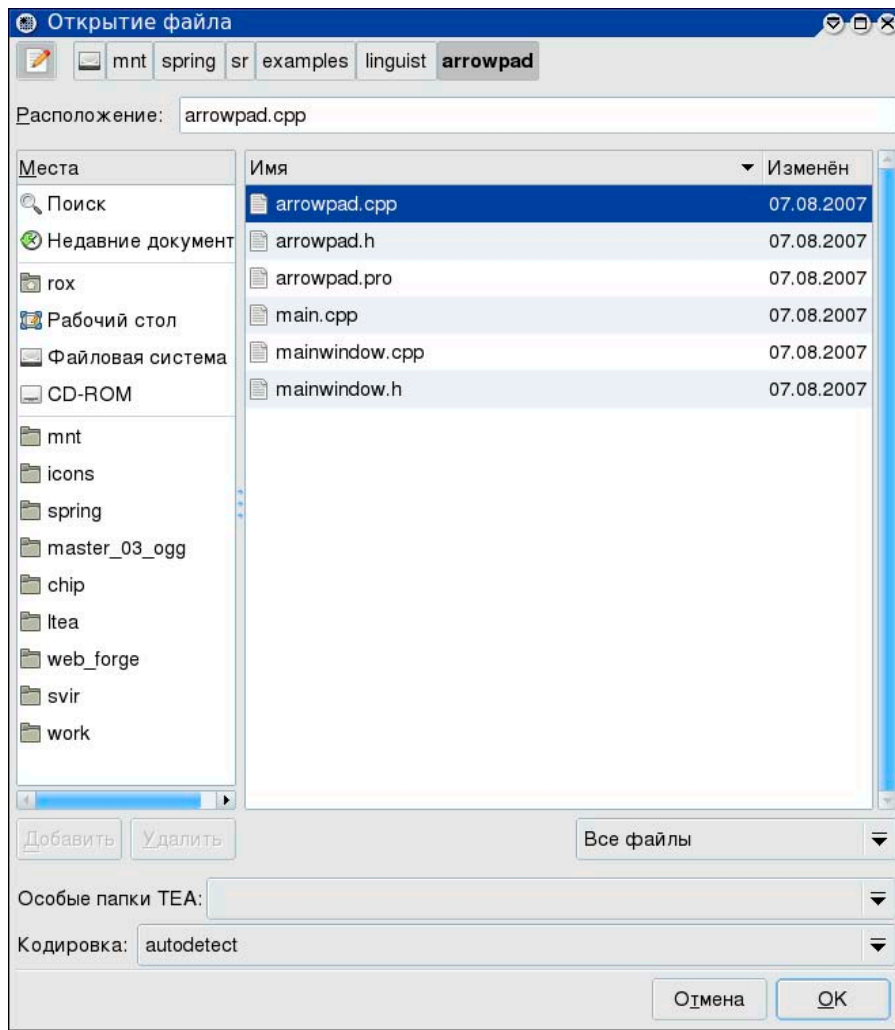
QTextEdit работает в двух режимах: обычный текстовый редактор и Rich Text. Внутри последнего режима лежит HTML. Фактически QTextEdit можно использовать как простенький HTML-браузер, хотя для этого предусмотрен другой класс – QTextBrowser, который, кроме прочего, поддерживает переходы по локальным ссылкам вида «#якорь». Таким образом, QTextBrowser может быть использован как движок для просмотра документации. Вот как запросто загружается в виджет документация из ресурса:

```
man = new (QTextBrowser);
man->setSource (QUrl("qrc:/manuals/ru.html"));
```

Примечание: протокол qrc указывает на то, что URL ссылается на файл внутри ресурса программы.

В новой версии Qt, которая вышла в начале мая, появился полноценный встроенный движок Web-браузера – WebKit. Это ответвление от KHTML с поддержкой HTML 4.01, XHTML 1.1, CSS 2.1, JavaScript 1.0 и свойства Canvas, SVG, а также частично CSS. В примере ниже я покажу, как с помощью WebKit открыть страницу во встроенном в вашу программу браузере:

```
QWebView view;
view.load ("http://webkit.org");
view.show();
```

Подстроенный под нужды программы диалог выбора файла в GTK+

Отмечу, что, кроме такого вот использования браузера «в лоб», доступна вся его внутренняя часть, все тонкости, так что теперь можно ожидать появления многих новых браузеров (и не только их), основанных на этой технологии.

Но вернемся ко вводу текста: однострочные текстовые поля. Обе библиотеки предоставляют для них движки автоматического дополнения. В Qt удобно сделана проверка вводимого текста – виджета однострочного ввода QLineEdit можно установить движок проверки – экземпляр класса QValidator. У последнего есть несколько готовых классов-потомков. Например, QRegExpValidator позволяет указать регулярное выражение, которое будет определять формат вводимого текста. Ниже показано, как ограничить ввод числами от 1 до 9999:

```
QRegExp r ("[1-9]\\d{0,3}");
QValidator *validator = new QRegExpValidator(r, this);
QLineEdit *edit = new QLineEdit (this);
edit->setValidator (validator);
```

Графика

GTK+ умеет отображать графику во всех популярных форматах, начиная PNG и JPEG, SVG и заканчивая TIFF. Для некоторых форматов доступны не только чтение, но и запись. Есть также набор графических примитивов: рисование точки, линии, дуги и тому подобное. Используя функции одной только GTK+, можно написать вполне приличный просмотрщик картинок.

В Qt, как обычно, кроме основных возможностей (отображение картинок и графические примитивы) есть и многое другое. Например, градиенты и интерфейс для вывода графики че-

рез OpenGL. Кроме того, для удобной работы с двумерной графикой Qt дает нам два класса: QGraphicsScene, который «заведует» двумерной сценой, наполненной картинками, текстом или примитивами, а также QGraphicsView, отображающий содержимое сцены. Сцена может быть больше размеров выводимой области, и текущую область отрисовки можно свободно перемещать по всему пространству сцены.

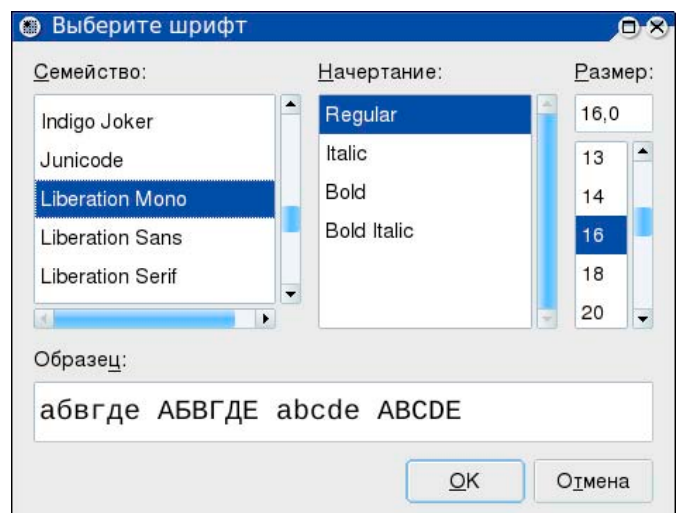
Не следует считать эти классы готовым движком платформы для игр вроде «Братьев Марио», хотя графические средства у Qt очень мощные.

А для простейшего вывода картинки в нужном месте окна достаточно и виджета-метки QLabel, который одинаково хорошо справляется с выводом графики и текста. У QLabel есть поле pixmap класса QPixmap. А QPixmap – это и поддержка популярных форматов файлов, и создание масок, и масштабирование, и работа с альфа-каналами – что хотите.

В GTK+ картинки отображаются в отдельном виджете: GtkImage с более скромными возможностями, чем в QPixmap. Хотя, учитывая «децентрализацию» в архитектуре GTK+ и разброс ее функций, в целом основных функций работы с графикой примерно столько же: есть масштабирование, вращение, работа с альфа-каналом и масками.

Диалоговые окна

Набор диалоговых окон в GTK+ и Qt примерно одинаков: выбор файла, шрифта, цвета, настройки печати. В GTK+, однако, даются еще и отдельные элементы многих таких диалогов: например, не «обрамленный» диалоговым окном виджет выбора цвета или полноэкранный виджет выбора шрифта (в Qt отдельно есть только выпадающий список со шрифтами). Как я уже писал ранее, диалоги в GTK+ удобнее подстраивать под свои нужды, да и функциональность некоторых этих диалогов выше. Сравните, например, диалоги выбора файлов в GTK+ и Qt: в первом и удобная панель навигации, и закладки, и строка поиска, а в Qt



Диалог выбора шрифта в GTK+

более пусто, хотя тоже есть автоматическое дополнение, а еще история последних посещенных каталогов.

Выводы

На сайте Qt можно прочесть слоган «code less, create more» — пиши кода меньше, а создавай больше. В самом деле, эта фраза хорошо описывает библиотеку Qt. То, что при использовании GTK+ требуется написания многих строк кода, в Qt порой решается всего парой строк. Дело не только в том, что полноценное ООП вместе с тщательно взвешенной архитектурой влияет на лаконичность программы. Просто GTK+ даёт набор основных средств и

оставляет вам доделывать остальное. Кирпичи есть — строй!

Qt однозначно охватывает больший спектр технологий, чем GTK+. Речь идет не о декорациях, а о насущном. Например, в Qt есть классы работы с протоколами FTP и HTTP. Конечно, никто не отменял (не будем пока говорить о будущем GIO) популярную библиотеку libcurl, которую легко использовать совместно с GTK+, но ведь это уже другая библиотека. Скажем так: в GTK+ нет таких важных вещей, которых не было бы в Qt. Но это правило лишено обратного действия: в Qt много важных вещей, которых просто нет в GTK+.

С другой стороны, GTK+ — это такая рабочая лошадь, неказистая с виду, но чертовски практичная в работе. Ежели в ней чего нет — подспорьем служат сторонние библиотеки. Я не могу сказать, что GTK+ так уж «хуже» Qt — понятия «хуже» и «лучше» в программировании вообще могут рассматриваться только в контексте решения конкретной задачи. Важно, что обе библиотеки в ближайшее время — основа большинства программ в мире UNIX-подобных систем, равно как и языки, на которых эти библиотеки написаны.

Петр Семилетов
(tea@list.ru)

Интервью: Андрей Шитов, основатель YAPC::Russia

Андрей Шитов (<http://shitov.ru>) — заметный деятель русскоязычного Интернета, ныне проявляющий особую активность в сообществе Perl-программистов. Он успел поработать в Студии Артемия Лебедева, причем не один год выполняя там функции технического директора, создать множество сайтов различной тематики и организовать группу Perl-программистов Moscow.pm 2.0 (<http://moscow.pm.org>), а также ряд мероприятий, которые по достоинству оценили любители языка Perl в России и Украине. Последние проводятся в рамках проекта YAPC::Russia (<http://yapcrussia.org>) при поддержке Moscow.pm и ведущих отечественных интернет-компаний.

Мне удалось познакомиться с ним в прошлом году, во время подготовки первого российского Perl-мероприятия «Perl Today», обзор которого можно найти в журнале «Системный администратор» за ноябрь 2007 года. На этом активная деятельность Андрея не закончилась, и теперь, после прошедшей в Москве конференции «May Perl» (<http://event.perlrussia.ru/yr2008>), вниманию читателей «Open Source» предлагается интервью с инициатором продвижения Perl в русскоязычных странах.

Вопросы по YAPC::Russia

Что побудило тебя проводить в СНГ мероприятия, посвященные Perl? Каковы их цели?

Импульс дала европейская конференция YAPC::Europe 2007 в Вене. Кстати, эта конференция была одна из самых крупных за последнее время и воодушевила многих других участников — после нее,

например, начался новый виток в развитии Perl 6.

Мероприятия, посвященные Perl, долгое время проводятся по всему миру, но не у нас. Про цели можно сказать сухо: популяризация Perl или пафосно: дать возможность посетить такие мероприятия большему числу программистов. Можно

сказать цинично: заполнить нишу. Все это соответствует действительности, но мне больше по душе другая формулировка: я делаю мероприятия, которые мне хотелось бы посетить самому.

Участие в них бесплатно. А откуда средства на организацию? Есть какие-то заинтересованные в этом спонсоры?

У меня нет задачи заработать на проведении мероприятий, и поэтому вопрос с финансированием решается просто. Во-первых, в Москве есть несколько крупных компаний, которые нуждаются в Perl-про-



Андрей Шитов на YAPC::Russia 2008 (фотография Алексея Капранова)

граммистах (достаточно заглянуть на сайты вакансий), и одновременно с этим существует дефицит рабочей силы. Поэтому мероприятия, на которые одновременно собираются программисты всех уровней, не могут не интересовать компании, которым нужны кадры. Во-вторых, есть компании, которые готовы предоставить более ценное, чем просто выделить определенную сумму.

Например, мы с самого начала сотрудничаем с Рамблером и Линуксцентром. Рамблер выступил генеральным спонсором на всех московских мероприятиях, а, например, первый воркшоп, прошедший в Москве в октябре 2007 года, был проведен в помещении, которое предоставил Линуксцентр вместе с презентационным оборудованием и даже компьютерами для участников.

Это уже вторая конференция в Москве, а еще одно мероприятие прошло в Киеве. На сайте сообщается, что скоро также состоятся конференции во Владивостоке и в Минске. Чем обусловлена такая география? Почему в списке нет известного своим программистским потенциалом Питера и что насчет других городов России и прочих стран СНГ?

География определяется очень просто: там, где есть активность и достаточное число потенциальных участников, там мероприятия и проходят. После первого воркшопа в течение нескольких недель появились Perl monger-группы во Владивостоке, Казани, Ростове-на-Дону и в Минске. Вот и готовый список для новых планов. Удаленные воркшопы удобнее организовывать локальными силами, и такие силы пока нашлись только во Владивостоке и Минске. Оба, кстати, имеют особое значение: Владивосток ставит флаг на дальневосточной карте, а Минск – на карте стран бывшего СССР, где до сих пор активно пользуются русским языком.

Питерской РМ-группы до сих пор не существует, хотя этот город числится в опубликованных планах на проведение воркшопов в ближайшее время. Кроме того, там есть филиалы заинтересованных московских компаний. Из Санкт-Петербурга в ЖЖ-сообществе `ru_perl` числится около 40 человек, что явно больше читателей, например, из Владивостока. Так что Питер не выпал из поля зрения.

Организация мероприятий в других городах заведомо осложнена не только малым числом Perl-программистов, но и тем, что там будет труднее собрать достаточное число участников, которые приедут из крупных городов.

Про страны СНГ могу сказать то же самое: мысли есть, но обнародовать их рано. Причем не обязательно ограничиваться пространством СНГ. Например, мне очень хочется охватить Литву и Латвию, интересно сотрудничать с людьми из Восточной Европы: в большинстве этих стран Perl-воркшопов не было никогда, но заинтересованные в них программисты там есть. В этом году, например, был организован первый хорватский воркшоп в Загребе.

Личное отношение к Perl

За что ты любишь Perl?

За то, что язык создан человеком, которому интересно не только программирование, но и лингвистика. Это как раз тот случай, когда язык интересен не только как инструмент для достижения цели, а как среда для повседневной жизни. А еще у Perl есть замечательное сообщество.

В последнее время стало модно говорить, что Perl мертв. Каковы, на твой взгляд, его основные области применения сегодня? Как ты оцениваешь его положение в этих сферах?

Взгляните на календарь мероприятий про Perl. За два месяца три конференции: 15-16 мая – YAPC::Asia, 17-18 мая – YAPC::Russia, 16-20 июня – YAPC::NA (Северная Америка), два воркшопа: 24-25 мая – Nordic Perl Workshop в Стокгольме, 6-7 мая – Португальский воркшоп и один хакатон QA Tools Hackathon 22-25 июня в Швеции. Иными словами, за эти два месяца состоятся мероприятия из каждой ступени в классификации Perl-мероприятий. Уже этого достаточно для того, чтобы понять, что высказываемое мнение не верно. И я еще не упомянул более 30 встреч местных групп, которые пройдут в эти же два месяца.

Сегодня (собственно, как и все предыдущие 10-20 лет) у Perl две большие области для применения: веб-программирование и системное администрирование. Если говорить о веб-программировании, то сейчас Perl распространен больше на крупных сайтах, нежели на небольших подделках. И это тоже неплохо.

А что с трудоустройством Perl-программистов в мире? В России? И какие у них перспективы?

За весь мир не скажу, а в России Perl-программисты чувствуют себя достаточно хорошо: относительно малое число самих программистов и относительно высокий средний возраст дают возможность получать относительно высокие зарплаты. Хо-

тя при этом нужно быть особенно осторожным, чтобы не спровоцировать «синдром Кобола», то есть когда остается настолько мало программистов, что найти их становится крайне сложно и дорого.

Плохо лишь то, что многие работодатели идут на поводу у безликого общественного мнения и не признают Perl или просто не знают о его достоинствах.

Про Perl 6 ты уже писал в журнале «Системный администратор» (№11 за 2006 год и №03 за 2008). Каково его нынешнее состояние? Кем он сейчас используется?

С начала этого года Perl 6 вступил в очередной виток развития, за которым уже виден релиз. Применять пока особо нечего, хотя еще в 2004 году я сделал крошечный сайт (<http://real.perl6.ru>) с несколькими примерами типичных задач, возникающих при создании сайтов. И сам сайт, и все примеры работают на одной из первых реализаций языка. Кроме того, с Perl 6 уже экспериментируют в Рамблере.

Есть ли у него будущее в enterprise-применении? Когда это станет возможным?

Промышленного применения тоже можно ожидать, но массовым оно станет не ранее чем через пару лет после первого релиза Perl 6. Кстати, выход Perl 6 не отменит и все, что работает на Perl 5, поэтому мы получим две собственные «торговые марки», за которыми стоит одно и то же Perl-сообщество.

Open Source

Как ты оцениваешь роль открытого исходного кода для Perl?

Для компиляторов Open Source вообще незаменимая вещь. Для Perl 6 это еще более ценно, потому что изначально предполагается, что определение языка будет дано не через его реализацию (единственную), а через подробное описание грамматики, на основе которой могут создаваться сразу несколько реализаций.

А какова роль Open Source в мире программного обеспечения в целом? В чем, по-твоему, главные преимущества Open Source над проприетарным ПО?

Главное, что есть у Open Source, – возможность в любой момент изменить ход проекта или создать на его основе что-либо новое. Если отвлечься от программирования и посмотреть на это иначе, то проекты можно сравнить с физическими явлениями. Многие объекты в природе описываются с помощью фрактальных структур,

а их развитие – хаотическими процессами, так что Open Source довольно близок к тому, что естественно для мира, в котором мы живем.

При этом я совсем не против проприетарного ПО и не выставляю априори дополнительных бонусов программам с открытым кодом. Все зависит от области задачи, которую решает программа. Например, я с удовольствием пользуюсь коммерческими компьютерными словарями на ноутбуке, хотя на моих серверах работает только Open Source.

Какие ты видишь перспективы развития Open Source в России?

Open Source (и не только в России) мешает повышенная жажда зарабатывать на том, что не несет никакой ценности.

Разработчики иногда считают, что если опубликовать свой код в открытом доступе, то от этого обязательно должен быть доход. При этом забывают о косвенной выгоде: например, если взять Perl, у которого существует большой архив CPAN, то когда разработчик публикует очень хороший модуль и его начинают использовать другие, это устанавливает мини-стандарт на соответствующую область – отличный способ влиять на ход истории (а это куда интереснее, чем просто зарабатывать деньги).

Пользуешься ли ты другим ПО с открытым кодом? Каким?

Все, что у меня работает на веб-серверах, по умолчанию существует с открытым исходным кодом: это компиляторы, веб-

серверы, Wiki-движки и даже framework для сайта Perl-конференций.

И последнее. Твое напутствие русскоязычным Perl-программистам.

У меня два пожелания: не нужно принимать близко к сердцу негативные высказывания о языке, его состоянии и перспективах, но в то же время не следует упускать из своих рук возможность доминирования. Решения просты: публикуйте код на CPAN, создавайте локальные РМ-группы, приходите на мероприятия и (главное) не бойтесь прийти туда не как участник, а как докладчик, начав, например, с пятиминутного блиц-доклада.

Дмитрий Шурупов
(osa@samag.ru)

Подписные индексы:

20780*
81655**

по каталогу агентства
«Роспечать»

88099*
87836**

по каталогу агентства
«Пресса России»

* **годовой**
** **полугодовой**

**Стоимость подписки
через редакцию:**

900* руб.
за 6 номеров

1800* руб.
за 12 номеров

Подписка на журнал «Системный администратор»

Российская Федерация

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

СНГ

В странах СНГ подписка принимается в почтовых отделениях по национальным каталогам или по списку номенклатуры «АРЗИ»:

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

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