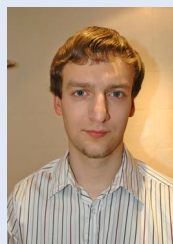


## Колонка главного редактора



В нашем издании назрели изменения совершенно разного характера. Виной тому – такое количество факторов, что рассказать о них абсолютно не представляется возможным. Поэтому – вкратце и по существу.

Во-первых, с рубрикой «FOSS Review» мы вынуждены попрощаться. Автор считает, что подобные материалы (обзоры Open Source-программ) его не развивают и приносят мало пользы читателям. Утверждение настолько же справедливое, насколько и спорное, а правда – в любом случае на стороне автора. Вместе с тем, он по-прежнему заинтересован в том, чтобы придумать новые способы реализации своих амбиций в «Open Source». Но конкретных идей о том, что и как писать, пока нет. Мы будем рады, если вы нам в этом поможете. Для связи используйте форум (<http://osa.samag.ru/forum>).

Во-вторых, рубрика вопросов и ответов по независимым от редакции причинам задерживается. Ваши первые вопросы уже получены и не останутся без внимания. Но это совсем не повод останавливаться! Присылайте и новые. E-mail прежний – [osa-question@samag.ru](mailto:osa-question@samag.ru).

И в-третьих, грядут совсем глобальные перемены, о которых говорить пока рано. Можно лишь предупредить, что, несмотря на кризис, мы будем стараться расти в меру наших сил и возможностей. Следите за обновлениями!

Главный редактор  
Дмитрий Шурупов  
([osa@samag.ru](mailto:osa@samag.ru))

### «Open Source»

электронное приложение к журналу  
«Системный администратор»  
№39, 20 февраля 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

### В Linux появилась базовая поддержка exFAT

Несмотря на то, что корпорация Microsoft не выпустила технические спецификации по своей новой файловой системе exFAT, Linux-энтузиаст уже представил первые тестовые патчи к ядру для поддержки чтения данных с exFAT.

Огава Хирофуми (OGAWA Hirofumi) опубликовал свои патчи в почтовой рассылке для разработчиков Linux-ядра (linux-kernel). После их наложения в ядре Linux появляется базовая поддержка файловой системы exFAT (только для чтения), которая получит развитие в ближайшее время. Это стало возможным благодаря процедуре обратного инжиниринга (reverse-engineer) дискового образа Windows Vista с exFAT.

exFAT – это новая файловая система от Microsoft, предназначенная для использования на мобильных устройствах вроде USB Flash-носителей большой емкости.

Корпорация активно занимается ее массовой популяризацией среди производителей различных устройств, однако не торопится с публикацией технических спецификаций, необходимых для возможности реализации поддержки exFAT в других операционных системах, таких как GNU/Linux.

### Куба подготовила свой Linux для противостояния США и Microsoft

Куба объявила о создании своей версии операционной системы на базе GNU/Linux. Этот шаг в первую очередь рассматривается как политический и является прямым продолжением государственной стратегии всестороннего сопротивления действиям США.

Кубинская версия Linux получила название Nova и была представлена на компьютерной конференции в Гаване. Новый продукт считается ключевым в стремлении кубинского правительства заменить проприетарное программное обеспечение американской корпорации Microsoft на своих компьютерах.

Использование американского ПО рассматривается кубинским правительством как потенциальная угроза безопасности страны: «Лучшее контролирование информационного процесса – дело особой важности», – заявил Рамиро Вальдес (Ramiro Valdes), министр коммуникаций Кубы, который возглавил комиссию, занимающуюся миграцией страны на ис-

пользование свободного программного обеспечения.

«Хочу думать, что через пять лет более половины компьютеров нашей страны будут переведены (на Linux)», – добавил Гектор Родригез (Hector Rodriguez), декан Школы свободного ПО в кубинском Университете информатики. По его же словам, на данный момент примерно на 20% компьютеров на Кубе установлено Linux. Стоит также отметить, что свободные продажи компьютеров для населения в этой стране начались лишь в прошлом году.

В основе кубинской системы Nova – популярный дистрибутив Gentoo, отличающийся использованием пакетов с исходным кодом, требующих компиляции (в отличие от бинарных пакетов, представленных во многих других популярных Linux-системах).

### Debian GNU/Linux 5.0 – популярный дистрибутив обновился

14 февраля 2009 года вышла новая стабильная версия популярного дистрибутива Debian GNU/Linux – 5.0 (под кодовым названием «Lenny»).

Из крупных изменений в новой версии Debian GNU/Linux стоит выделить прекращение поддержки 32-битной процессорной архитектуры SPARC, поддержку архитектуры armel, значительные продвижения в поддержке UTF-8, полную поддержку популярных портативных устройств ASUS Eee PC, соответствие LSB (Linux Standards Base) 3.2, появление Java-пакетов OpenJDK, утилит сборки Emdebian для кросс-компиляции пакетов Debian и сокращения их до размера, пригодного для запуска на системах ARM.

Среди пакетов программного обеспечения, вошедших в состав Debian GNU/Linux 5.0, – Apache 2.2.9, BIND 9.5.0, GCC 4.3.2, MySQL 5.0.51a, PostgreSQL 8.3.5, Python 2.5.2, PHP 5.2.6, OpenLDAP 2.4.11, OpenSSH 5.1p1. Среди десктоп-приложений – X.Org 7.3, GNOME 2.22, KDE 3.5.10, Xfce 4.4.2, OpenOffice.org 2.4.1, Icedove (Thunderbird) 2.0.0.19, Evolution 2.22.3, Pidgin 2.4.3.

Сообщается, что в Debian GNU/Linux «Lenny» намного больше пакетов ПО, чем в прошлом релизе («Etch»): всего их стало более 23200 (из них – более 7700 новых). Большая часть программ, входящих в дистрибутив, была обновлена: более 13400 пакетов, что составляет 72% дистрибутива «Etch». Также по различ-

ным причинами многие пакеты (более 3100) были удалены.

### Red Hat и Microsoft сотрудничают в области серверной виртуализации

Компании Microsoft и Red Hat подписали соглашения, в рамках которых они будут обеспечивать тестирование, сертификацию и совместную техническую поддержку общих заказчиков, использующих технологии серверной виртуализации.

Соглашения предусматривают участие компании Red Hat в программе Microsoft Server Virtualization Validation Program и участие Microsoft в программе Red Hat Hardware Certification List для обеспечения совместимости и поддержки виртуализации. Как заявляется, это позволит внедрять серверные операционные системы Microsoft Windows Server и Red Hat Enterprise Linux, виртуализированные соответственно на гипервизорах Red Hat и Microsoft, и получать поддержку от производителя соответствующей ОС.

### HTC и Vodafone представили HTC Magic – второй Android-смартфон

Компания HTC, подарившая миру первый смартфон с Android (T-Mobile G1), представила второе долгожданное устройство с мобильной Linux-платформой Google, которое в СМИ раньше часто называли как G2, – HTC Magic.

Новинка создана при участии крупнейшего в мире оператора сотовой связи Vodafone.

Презентация нового Android-устройства и демонстрация его возможностей состоялись на выставке Mobile World Congress, которая проходит в эти дни в Барселоне (Испания).

HTC Magic является полностью сенсорным смартфоном – в отличие от своего предшественника G1, физической клавиатуры у него нет. HTC позиционирует новинку как «сенсорный телефон для тех, кто борется за смелую сторону жизни».

Технические характеристики HTC Magic:

- ✓ GSM/GPRS/EDGE 850/900/1800/1900 МГц и HSDPA/WCDMA 900/2100 МГц;
- ✓ 3,2-дюймовый сенсорный TFT-дисплей с поддержкой разрешения 320x480 (HVGA);
- ✓ процессор Qualcomm MSM7201a (528 МГц);
- ✓ память: 512 Мб ROM и 192 Мб RAM, слот для microSD-карты;
- ✓ 3,2-мегапиксельная камера с автофокусировкой;
- ✓ аккумулятор на 1340 мАч: до 450 минут в режиме разговора, до 420 часов в режиме ожидания (для GSM).

Распространением HTC Magic будет заниматься мобильный оператор Vodafone. Начало продаж в Европе (Великобритания, Испания, Франция, Германия, Италия) намечено на весну этого года. Стоимость пока неизвестна.

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

## KDE 4.2: готов ли новый рабочий стол для конечного пользователя?

В январе 2008 года состоялся релиз принципиально новой среды рабочего стола – KDE 4.0 (см. его обзор в «Open Source» 021 от 08.02.2008. – Прим. ред.). Несмотря на версию, которую многие ошибочно приняли за стабильную, среда была предназначена для демонстрации возможностей и привлечения разработчиков и энтузиастов. После ее выхода в свет критика не заставила себя ждать: обсуждение новой среды поднималось на многих популярных западных и отечественных сайтах и форумах. Тем временем, в течение всего 2008 года шла активная работа над проектом, и в августе KDE «дорос» до версии 4.1, после чего я и стал его постоянным пользователем. До конца года было вышло еще несколько версий, направленных на повышение стабильности и исправления обнаруженных ошибок.

Труды команды разработчиков не были замечены. Так, например, читательское голосование журнала «Linux Format» определило победителем в номинации «Free Software Project of the Year» (Проект свободного ПО 2008 года) именно KDE 4.

И вот 27 января 2009 вышла новая версия KDE – 4.2. Согласно официальному анонсу, эта версия готова для конечного пользователя. Но так ли это на самом деле? Давайте посмотрим...

### Plasma

Наверное, многих в первую очередь интересует вопрос, «падает ли Plasma». (Plasma – один из ключевых компонентов KDE 4, представляющих рабочий стол этой среды. Именно ее нестабильная работа вызвала немало негативных отзывов пользователей. – Прим. ред.) Что ж – с нее и начнем.

Благо, разработчики потрудились на славу, и изменения затронули почти все элементы Plasma. Можно с уверенностью заявить, что панели в KDE постепенно возвращают утраченные в «четверке» возможности. Например, наконец-то было реализовано автоматическое скрывание панелей, чего так не хватало многим пользователям.

Однако здесь иногда возникают и трудности: однажды после перезагрузки у меня панель утратила данную опцию,

и пришлось снова ее включать, а в другой раз – две панели «обменялись» этой функцией, то есть перестала скрываться та, которая должна была это делать, и наоборот. В обоих случаях пришлось вручную исправлять огрехи.

Всплывающие подсказки с выходом новой версии перемещаются плавно и содержат иконку выполняемого приложения – впрочем, эта возможность многим покажется уже знакомой, поскольку она присутствовала в специальной сборке KDE 4.1.2 в составе Kubuntu 8.10.

Для тех, у кого обычно открыто много окон, предусмотрено два улучшения: группировка и отображение нескольких строк в панели. Количество отображаемых строк настраивается в параметрах виджета «Управление заданиями». Там же можно настроить правила для отображения, группировки и сортировки элементов. По умолчанию группируются все приложения, но эту возможность можно отключить в два клика через контекстное меню в панели.

Также предусмотрена временная разгруппировка по щелчку средней кнопки мыши в панели. Но не обошлось без проблем и здесь: разгруппированные приложения в панели могут наезжать на другие элементы (особенно это заметно при использовании нескольких строк), и другая проблема, возникающая в случае нескольких строк: не все разгруппированные элементы отображаются, а часть из них – просто пропадает (повторный щелчок мы-

шью возвращает все элементы обратно в сгруппированный вид). В общем, напрашивается логичный вывод – есть над чем работать.

Многим «ранним» пользователям KDE 4 должна быть знакома ошибка отображения иконок в трее, когда вместо требуемой прозрачности показывался разный шум или фоном для одной иконки становилась другая (соседняя) иконка. Как известно, проблема связана с попыткой разработчиков разобраться с особенностью функционирования GTK-приложений с альфа-каналом иконок в трее. Решение появилось в обновлении к библиотеке Qt, где до версии 4.5 и содержался проблемный код, но в KDE 4.2 используются «временные меры». И они оказались недостаточно эффективными: шум периодически все же появляется, и намного чаще его можно заметить, если использовать визуальные эффекты Kwin (о них – см. ниже).

Теперь, пожалуй, о самой интересной и необычной возможности... Системные уведомления всех приложений были перенесены в трей таким образом, что создают «очередь», вид которой унифицирован для любого приложения. Пояснить их работу лучше на конкретном примере. Допустим, закачивается torrent и одновременно с этим идет общение через Kopete:

- ☒ В какой-то момент приходит сообщение – вы тут же можете прочитать или ответить на него.
- ☒ После этого нужный контакт входит в сеть – по нажатию одной кнопки можно начать чат с ним.
- ☒ Одновременно с этим, заканчивается загрузка torrent-файла.

В результате, на рабочем столе (над или под треем – в зависимости от его расположения) будет создана очередь уведомлений, состоящая из трех сообщений одного вида. Для всех них в трее предусмотрена одна общая пиктограмма, по нажатию на которую можно скрыть сообщения.

Закачки, перемещения или удаления файлов в системе также имеют свою пиктограмму в трее, и могут в него сворачиваться, освобождая пространство для работы, а не для созерцания прогресса выполнения. Таким образом, рабочий стол «избавляется» от всех ненужных окон. К слову, что-то подобное хочет реализовать Марк Шаттлворт в грядущей версии Ubuntu (http://www.markshuttleworth.com/archives/253).

В настройках внешнего вида приложений тоже произошел сдвиг. Так, в па-

раметрах системы можно настроить тему, которая будет использоваться для того или иного элемента. Если есть несколько тем (кстати, они легко скачиваются по нажатию на Get New Theme), то любую из них можно выбрать в качестве основной, скажем, для диалога выключения. Эти настройки применимы к используемым на данный момент виджетам, что позволяет более гибко настроить внешний вид рабочего стола.

Сдвиги невооруженным глазом заметны и в управлении питанием, которое также настраивается через параметры системы, – теперь можно детально настраивать параметры энергосбережения и эффективно управлять ими с помощью нового виджета.

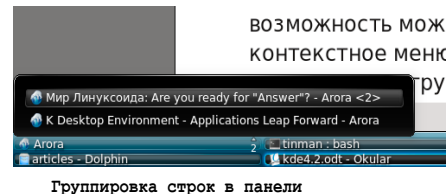
Немного о других виджетах, которые «подросли» в новой версии KDE. Важно, что были не только исправлены многие ошибки, но и расширены функциональные возможности (теперь виджеты могут отображаться поверх скринсейвера), а также добавлены новые виджеты.

Расстроило, что долгожданный погодный виджет работает только с двумя российскими городами.

Зато оказался удобным виджет быстрой публикации текста и изображений, использующий сервисы Paste/Img-Bin (вроде [Pastebin.com](http://Pastebin.com)).

Также были добавлены виджеты отображения системной информации: загрузка ЦП, мониторинг сети и жестких дисков...

Если этого мало, сторонние виджеты всегда можно найти на просторах Интер-

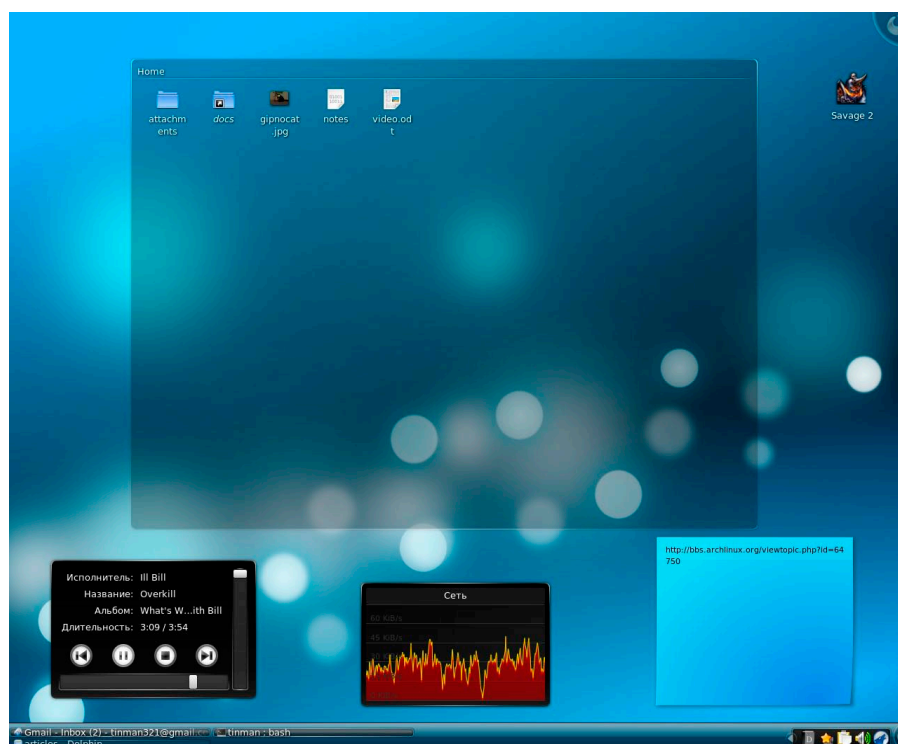


нета (например, на [kde-look.org](http://kde-look.org)), подключить GoogleGadgets или написать свои на одном из поддерживаемых языков (JavaScript, Python и Ruby).

## Рабочий стол

Выход KDE 4.0 сбил многих с толку, и одной из причин стала необычная концепция рабочего стола, суть которой сводилась к представлению рабочего пространства как площадки для запуска виджетов, а традиционный десктоп просто отображал файлы из каталога Desktop и содержал фоновый рисунок. Многим эта концепция сильно не понравилась и они решили, что «KDE 4 не нужен». И вот в новой версии разработчики учли пожелания пользователей и вернули эту возможность, предоставив выбор: привычный или новый вид. Первое подразумевает использование абсолютно любой директории для отображения, что опять же позволяет отказаться от каталога Desktop, в котором нет объективной необходимости. Так же, как и виджет «Просмотр папки», этот режим позволяет выбрать, какие типы файлов вы хотите отображать, а какие – нет.

Еще одной интересной особенностью являются «activities» («занятия» в русской локализации), которые позволяют созда-



Рабочие области (activities) в KDE 4.2





Рабочие столы на сторонах куба

вать рабочие области (см. скриншот). На них могут быть размещены свои виджеты и применены свои настройки (например, фоновая картинка). Так, например, один может использовать традиционный вид, а другой – новый, однако панели и окна для них будут общими.

## Эффекты KWin

Улучшения проводились и в области привлекательности для обычного пользователя. Появились новые эффекты, а старые были доработаны. Так, появились возможности, которые довольно давно существуют в Compoz: при наведении курсора мыши на один из углов в области края экрана и его задержке в таком положении на некоторое время можно выполнить одну из доступных возможностей: показать все окна (все или с данного стола), активировать визуальные 3D-эффекты смены рабочего стола.

Появились новые эффекты отображения виртуальных столов: на стороне куба, сферы или цилиндра. Причем, если количество виртуальных рабочих столов отлично от четырех, то куб автоматически становится призмой. Были добавлены и другие эффекты, среди которых, например, сворачивание в стиле «Магической лампы». Что касается производительности, то у меня она оставалась на адекватном уровне только при использовании куба – остальные эффекты работают с задержками в прорисовке.

## Приложения KDE 4.2

KDE – это не только рабочая среда, но еще и приложения, которые совершенствуют-

ся от версии к версии. Если говорить об интернет-среде, то это хорошо всем известный браузер Konqueror, который получил улучшения в рендеринге веб-страниц и векторной графики, оптимизацию, поиск в стиле Firefox и отображение закладок на первоначальной странице. Кстати, существует виджет для запуска различных профилей Konqueror – для интернета, обзора файлов, для разработки и так далее. Почтовый клиент KMail приобрел возможность гибкой настройки отображения цепочек в каталогах.

Если говорить о файлах – это в первую очередь Dolphin и Ark. В первом – добавлены возможности масштабирования отображаемых иконок, предварительного просмотра файла во всплывающей под-

сказке, а также просмотр архивов. Кроме того, теперь в Dolphin, как и в виджете примонтированных устройств, можно посмотреть объем и используемое пространство съемного накопителя. Наконец стало возможно полноценное использование Ark как основного архиватора: добавлены действия в контекстное меню Dolphin по извлечению/созданию архивов.

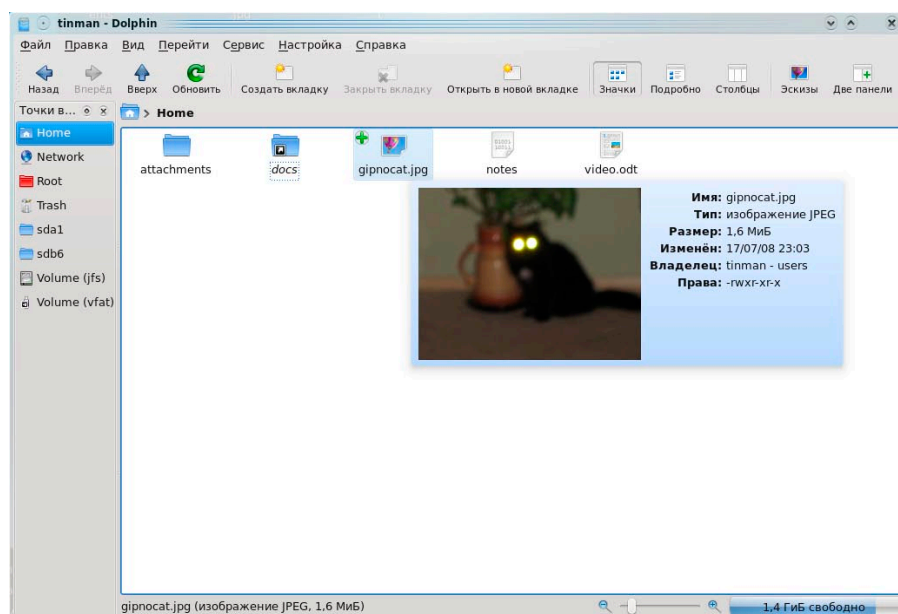
Помимо основных приложений, входящих в состав KDE, обновились и так называемые Extragear-программы, среди коих – KTorrent и Amarok. Очень радуют известия о возобновлении активной разработки k3b для KDE 4 (подробности об этом – см. на форуме: <http://forum.kde.org/temporary-replacement-for-k3b-t-30795.html>).

## Заключение

После выхода KDE 4.2 по всему миру прошли так называемые «Release Parties». Был анонсирован Qt 4.5 RC1, в котором провели оптимизацию, исправили трей, обновили QWebKit и многое другое. Также начались и активизировались работы по переводу KDE на другие языки – в том числе и русский.

Возвращаясь к вопросу, который был поставлен в начале, отмечу, что всего в статье не описать. Тем более, то количество деталей, что появились в новой версии KDE. Очень сложно передать впечатления от данного выпуска, и хочется сказать каждому: «Просто попробуй!». Однако ряд деталей и недоделок в KDE 4.2 не позволяют мне заявить, что эта среда наконец готова для повсеместного использования.

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



# Обзор мультимедийного плеера Miro

**M**iro – это... очередной видеоплеер, скажет любой, кто впервые открыл страницу проекта (<http://www.getmiro.com>). И будет весьма не прав. Приложение для загрузки видео с YouTube? Не только и не столько. Bittorrent-клиент? И это тоже. Так что же тогда? Если сформулировать кратко, Miro – приложение, которое позволит вам находить, скачивать, смотреть и организовывать видеоконтент, будь то подкаст, поисковый запрос к видеохостингу или сериал с одного из торрент-трекеров. В этом плане он несколько напоминает другой Open Source-проект – подкаст-агрегатор gPodder (<http://gpodder.org>). Так уж получилось, что я являюсь участником данного проекта и по-сему буду невольно проводить параллели. Принципиальная разница в том, что gPodder сосредоточен на аудиоконтенте и его передаче на различные устройства, а Miro призван работать с видео на десктопе (увы, полностью игнорируя факт существования мобильных устройств, способных его воспроизводить). Для этого у Miro есть всего понемногу от каждой технологии и от каждого типа приложений, объединённых в нём. Так, на видео-проигрыватель отведены три настройки, на битторрент-клиент – десяток, и все это – против километрового мануала MPlayer и вызывавшего трепет окна настроек Azureus. Во главе угла – простота и интеграция, дающие в результате синергетический эффект.

## Под капотом

Чтобы предоставить всю эту красоту пользователю, Miro использует множество сторонних открытых компонентов. Построенный на XULRunner, с интерфейсом на GTK+ (в Linux-версии), логикой на Python, работой с изображениями через ImageMagick и проигрыванием видео через VLC, Xine и GStreamer, Miro служит наглядным примером UNIX-way.

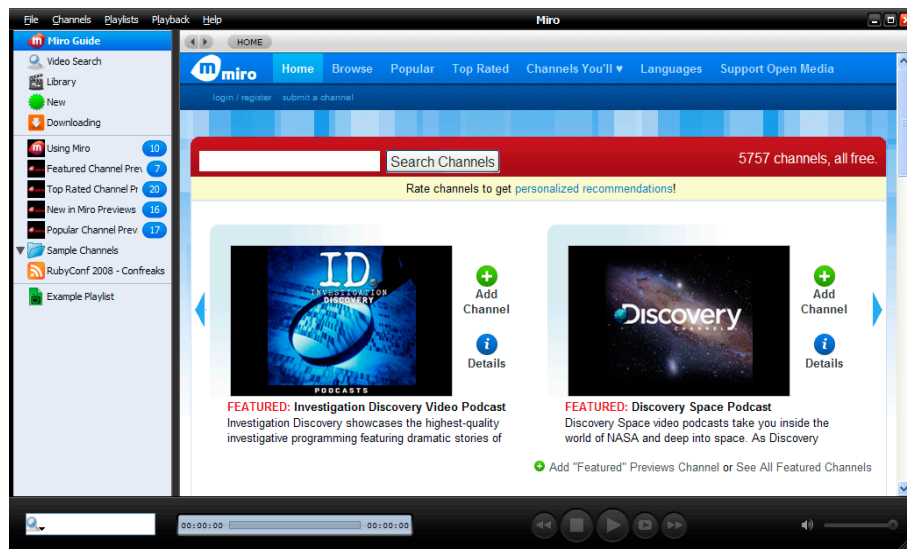
## Установка

Пожалуй, стоит посвятить пару слов установке. Miro доступен для всех основных платформ: Windows, Mac OS X, GNU/Linux. Причём для последней есть пакеты под ряд дистрибутивов (для не попавшего в их список Gentoo нашёлся ebuild в оверлее sunrise). Использование открытых кроссплатформенных библиотек

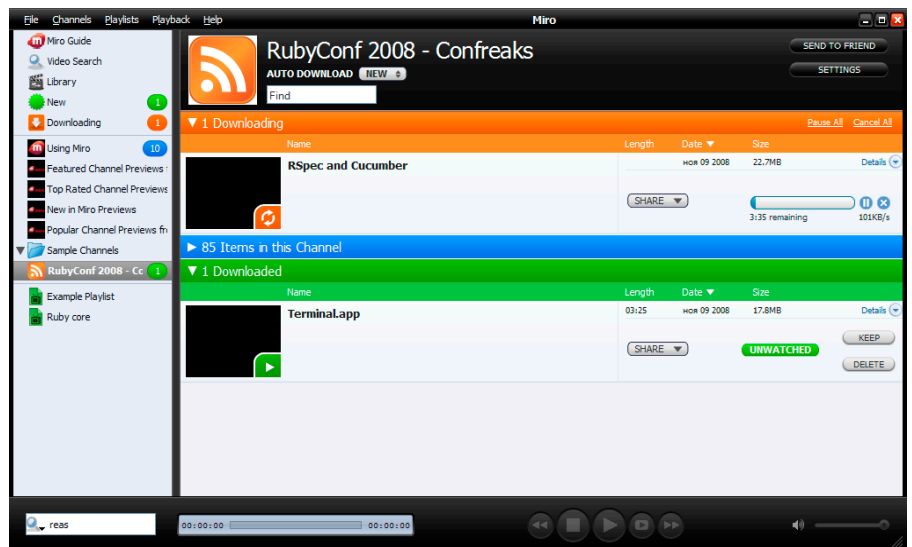
и программ позволяет надеяться на возможность запуска Miro и в других окружениях (например, во всём множестве POSIX-совместимых ОС). Межплатформенность достигается за счет использования XULRunner в качестве каркаса для

разработки приложений. Этим обусловлена и разница во внешнем виде приложения под разными платформами. Родной интерфейс в данном случае мне кажется скорее недостатком, чем преимуществом – достаточно сравнить версии GTK+ и Windows (приведённые в статье скриншоты сделаны в Windows).

Остальные зависимости Miro широко распространены и не так тяжелы: в репозиториях найдётся всё необходимое. А пользователям Windows вообще не при-



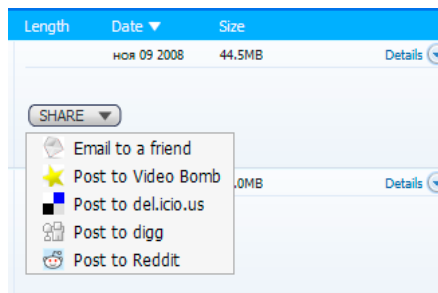
Главное окно Miro



Просмотр RubyConf2008 в Miro



Настройки RSS-канала в Miro



Возможности поделиться видео

дётся беспокоиться об этом – установщик Miro под Windows содержит всё в себе.

## Первые впечатления

В Miro есть на что посмотреть – особенно для пользователя, не пугающегося английского языка. Можно сразу после запуска программы открыть вкладку Miro Guide и найти там тысячи каналов на самую разную тематику, отсортированных по категориям. В открытой «Linux & Free Software» их нашлось 27.

Я же, как человек, интересующийся языком программирования Ruby, захотел посмотреть что-нибудь тематическое и нашёл много уже знакомых каналов: Learning Rails, Railscasts и несколько других. Репрезентативность впечатляет. С каналами из каталога всё понятно – находится быстро, добавляются в один клик.

Но вот на habrahabr.ru подсказали (<http://habrahabr.ru/blogs/ruby/48312>), что можно посмотреть видео с RubyConf2008 (<http://rubyconf2008.confreaks.com>). Зайдя

на страничку с записями выступлений, увидел ссылку на RSS видеоподкаста (путно заметив, что скачивать рекомендуют именно Miro) – в самый раз. В Miro: «Каналы → Добавить канал» – и последнее видео из него начинает закачиваться. После загрузки посмотреть его можно тут же... правда, повторяя про себя мантру «это не видеоплеер» и не обращая особого внимания на ограниченность функций (например, неработающий полноэкранный режим). Впрочем, особенно приятная для просмотра скринкастов функция видеоплеера – запоминание текущей позиции в ролике – здесь присутствует.

У каждого канала можно переопределить настройки (удобно, например, «устаревание» видео в применении к новым каналам: старые выпуски будут сами удалены), искать по каналам, а особо интересным видео можно поделиться с друзьями самыми разными путями – отправить ссылку по почте или опубликовать её на одном из социальных сервисов (delicious, reddit, video bomb, digg). Возможность определить свой способ «поделиться» для пользователя отсутствует, а для разработчика кроется в простых XML-файлах (как и многое другое).

К месту и поиск, охватывающий с десяток видеохостингов. К примеру, в своё время перед покупкой Nokia N810 мне хотелось взглянуть на планшет, увидеть его в действии, виртуально «пощупать», дабы не покупать зря. Miro поможет и с этим:

два клика – и вот они, ролики про интересующее устройство: найдены, отсортированы, готовы к просмотру. С сортировкой вообще всё хорошо: она возможна в любом списке по размеру, длине, дате – ещё один способ не пропустить что-то новое для тех, кто держит руку на пульсе. Жаль, не все хостинги отдадут размер и длину роликов в RSS.

Есть тут и плейлисты, и библиотека, но они не представляют собой ничего особенного – простые линейные списки роликов, ещё один способ их организации.

## А хорошо бы...

Есть и простор для дальнейшего усовершенствования. Например, поиск – вернее, поисковые движки. Возможность их добавления есть – лишь бы видеохостинг умел выдавать RSS с ссылками на ролики... Но хоть они и описываются простым XML, добавить свой оказалось затруднительно. Предварительные изображения для роликов показываются, но только если они присутствуют в виде картинок в той же RSS, а ведь любой современный файловый менеджер давно научился делать это самостоятельно (правда, для локальных файлов). Ограничение скорости загрузки есть только для торрентов, а очередь загрузки отсутствует совсем: файлы качаются все скопом. К слову, в gPodder есть и то, и другое, хотя торренты там отданы на откуп сторонним программам.



Поиск по видеохостингам в RSS



Интерфейс Miro частично переведён (за что спасибо Вячеславу Куренышеву), но за помощью в этом можно и должно обратиться к самим пользователям. Кроме того, неплохо бы иметь возможность просматривать видео сразу, с начала его загрузки – так, как это сделано на сайтах самих видеохостингов. Благо, и до провинций дошли каналы, достаточно широкие для того, чтобы смотреть интернет-видео без буферизации, да и с буферизацией можно было бы найти разумный компромисс, вычислив примерное время скачивания. Скорее всего, возникли бы проблемы с торрентами: ведь кусочки там качаются не по порядку, но ведь на то и встроенный торрент-клиент, чтобы можно было заставить его качать последовательно.

В завершение стоило бы сделать хотя бы на начальном уровне работу с устройствами. Например, экспортом всего видео определённого канала в заданный пользователем каталог, как это сделано в gPodder для совсем безнадёжных случаев и неподдерживаемых устройств. В идеале, конечно, нормальная синхронизация – ведь все необходимые для этого библиотеки есть и могут быть использованы из Python, а также можно воспользоваться наработками gPodder, раз уж они написаны на одном языке.

## Недостатки

Не обошлось и без недостатков. Основной недостаток один – некоторая сыроватость. Если в Windows-версии программа работает прекрасно, то в Linux-варианте – «падает» без видимых причин.

Одно большое недоразумение под Linux и воспроизведение видео – оно банально не работает. Под Windows безальтернативно используется VLC, под Linux же был предложен выбор из Xine и GStreamer – одинаково безуспешный.

## Пока готовился номер

10 февраля вышла новая версия программы – Miro 2.0 (а вскоре после этого – и исправления 2.0.1, 2.0.2), принеся с собой множество изменений и улучшений. Среди наиболее интересных: Linux-версия программы стала выглядеть значительно лучше, появилась новая секция для аудио в боковой панели (что делает Miro более универсальным), добавлена возможность смотреть видео в отдельном ок-

не и продолжать пользоваться приложением, представлены подписка на канал пользователя YouTube и загрузка HD-варианта ролика с него, полноэкранный режим для второго монитора (для домашних кинотеатров), исправлено более 600 багов.

Для меня же самым важным стал другой момент – в новой версии наотрез отказался работать поиск, что ставит вопрос о целесообразности обновления.

Установленный по горячим следам VLC не был обнаружен программой. Виной всему оказалась нестабильная ветка дистрибутива, которую я использовал. После отката Xine на стабильную версию видео заработало, падения пропали, однако полноэкранный режим, под Windows просто не перекрывающий панель задач, здесь показывал скопище GTK-виджетов. (С пакетом Miro в Ubuntu Linux 8.10 не замечено проблем ни с воспроизведением видео, ни с полноэкранном режимом. – Прим. ред.)

Странно реализовано позиционирование в ролике: перемотка возможна только перетаскиванием ползунка. Некоторые места интерфейса не дружат с русским. В целом – ничего критичного: мелкие огрехи выглядят неопрятно, но удовольствия не портят.

## Резюме

Если вы хотите смотреть видео из Интернета, не заботясь о том, как качать, куда сохранять и что потом с ним делать, вам определённо стоит попробовать Miro. Благодаря «Open Source» (ведь лучший способ что-либо изучить – рассказать другому) я это и сделал. Теперь точно знаю, чем смотреть очередную конференцию или скринкаст. Как всегда, интересные идеи в рассмотренной области я буду рад видеть в своём почтовом ящике.

Леонид Пономарев  
([lp\\_x@mail.ru](mailto:lp_x@mail.ru))



Странная поддержка русского в Miro

# Ее PC и Ubuntu Linux. Часть 5.2: Как можно быстрее – любыми способами!

## Предисловие

Еще два года назад разгорались жаркие споры о том, что лучше: телефон с Bluetooth и КПК или только коммуникатор. У каждой стороны полемики были свои веские аргументы, и каждый считал долгом найти еще одну повседневную задачу, с которой не смог бы справиться

с соперник. Но время прибило на корню то консервативное решение, приверженцем которого был и я. Сейчас относительно многие используют коммуникаторы и не задаются вопросами двухгодичной давности: их устраивает практически все – лишь бы хватило заряда аккумулятора...

Выпуск Ее PC (и других субноутбуков той же ценовой категории) подтолкнул пользователей мобильных устройств к новым рассуждениям о «связках» и «молонитах». Действительно, на этом маленьком и доступном ноутбуке можно делать многое из того, чем оснастили современные коммуникаторы, но с большим комфортом. Ведь относительно большой экран не заставляет напрягать зрение для чтения книг, просмотра веб-страниц, написания и редактирования офисных документов, а встроенная клавиатура позволяет отказаться от покупки внешней и медленного, с большим количеством ошибок, набора на стандартной. И это –

не говоря о запуске на устройстве требовательных к ресурсам приложений или того, у чего вовсе нет аналогов под ARM.

Все это позволяет сделать вывод, что связка из современного телефона, на котором можно запустить Opera Mini и текстовый редактор для «быстрых набросков», и Еее PC вполне может быть хоть пока и гипотетической, но альтернативой «монопольному» коммуникатору для тех, кто не любит подход «все в одном и понемногу».

## Введение

В прошлой части цикла (см. «Open Source» 036 от 09.12.2008) была рассмотрена проблема уменьшения времени загрузки Еее PC при помощи «косметических» изменений. Нам удалось добиться неплохого результата в 53 секунды, что составило около 60% от первоначального показателя (86 секунд). Как уже сообщалось ранее, цели добиться того же результата, что и инженеры из компании Intel, не было, однако приблизиться к нему полезно. Итак, попробуем осуществить более глубокую перестройку системы...

## Оптимизация внутри системы. На смену Upstart: Finit

Как отмечалось в комментарии к графику bootchart в предыдущей статье, для уменьшения времени загрузки полезно поменять стандартную систему инициализации Ubuntu (Upstart) на что-то более легкое, быстрое и ориентированное на задачи конечного пользователя. Я полагаю, что это должен быть Finit (<http://hellabs.org/finit>). Вы, наверное, уже замечали, насколько быстро загружается Еее PC с родным дистрибутивом Xandros. Так вот: Finit был разработан как Open Source-альтернатива закрытой системе инициализации Fastinit, созданной Asus для Xandros. Изначально существовала редакция Finit только для дистрибутива Mandriva на Еее PC. Сейчас же можно найти версию и под Ubuntu – в ней не все идеально, но постараемся решить возможные проблемы.

Основной принцип, который был положен в основу Finit, заключается в максимально быстрой загрузке интерфейса пользователя с продолжающимся запуском различных компонентов системы в фоне. Вдобавок, Finit лишили более половины стандартных сервисов, оставив только самые необходимые. Впрочем, это не мешает добавить свои службы в случае необходимости.

Для установки Finit потребуется загрузить пакет с исходными текстами программы, доступный в сети (<http://www.smurfy.de/files/finit25062008.tar.gz> – редакция от пользователя [smurfy@forum.eeeuser.com](http://forum.eeeuser.com)). В директории распакованного архива перейдите в каталог smurfy, откройте в любом текстовом редакторе (например, gedit) файл finit-alt.c и внесите правки в код между 233 и 234 строками так, чтобы это выглядело следующим образом (правки – от [javierrivera@forum.eeeuser.com](mailto:javierrivera@forum.eeeuser.com)):

```
if (startx) {
    mkdir("/dev/input", 0755);
    chardev("/dev/urandom", 0666, 1, 9);
    chardev("/dev/ptmx", 0666, 5, 2);
    chardev("/dev/null", 0666, 1, 3);
    chardev("/dev/mem", 0640, 1, 1);
    chmod("/dev/null", 0667);
    chmod("/dev/mem", 0640);
    chardev("/dev/tty", 0666, 5, 0);
    chardev("/dev/tty3", 0666, 4, 3);
    chardev("/dev/input/mice", 0660, 13, 63);
    chardev("/dev/input/event0", 0660, 13, 64);
    chardev("/dev/agpgart", 0666, 10, 175);
    blkdev("/dev/loop0", 0600, 7, 0);
    xpid = spawnxserver();
}
```

Это необходимо для запуска X-сервера до загрузки та-

ких служб, как udev, dbus и hal. Они будут исполняться в фоне во время инициализации графической оболочки. Теперь можно начинать процесс компиляции выполнением команды make из этой же директории. Полезно предварительно убедиться, что в системе установлены пакеты libconfig0 и libconfig0-dev (если нет, то установка тривиальна: «sudo apt-get install libconfig0 libconfig0-dev»).

После компиляции перенесите полученный файл finit-alt в директорию /sbin, а поддиректорию finit этого архива – в /etc. Кроме того, потребуется модифицировать /boot/grub/menu.lst для того, чтобы Еее PC смог использовать полученный finit-alt вместо стандартного Upstart. Создайте копию рабочей опции загрузки системы и добавьте к строке с kernel параметр init=/sbin/finit-alt. Результирующая запись в конфигурационном файле GRUB будет выглядеть примерно следующим образом:

```
title Ubuntu 8.04.1, kernel 2.6.24-20-eeepc FINIT
root (hd0,0)
kernel /boot/vmlinuz-2.6.24-20-eeepc
root=UUID=8e94a793-9f1a-4167-8fb8-6abaaad409d5 ro quiet splash
clocksource=hpet init=/sbin/finit-alt
initrd /boot/initrd.img-2.6.24-20-eeepc
```

Перейдем к настройке Finit на Еее PC. Для нормального функционирования внутренних устройств ноутбука необходимо добавить некоторые модули в автозагрузку. Откройте для редактирования файл /etc/finit/prexserver, который будет исполняться перед стартом X-сервера в режиме автологина, и добавьте туда строки перед «/etc/init.d/hwclock start»:

```
/sbin/modprobe pciehp pciehp _debug=1 pciehp _force=1
/sbin/modprobe eeepc _acpi
```

Мы уже делали подобное в первой части цикла (см. «Open Source» 030 от 29.08.2008), но в файле /etc/modules.

Если пропала возможность переключать раскладки клавиатуры, то необходимо настроить эту функцию через /etc/X11/xorg.conf, а не средствами графической среды (они могут не стартовать). Рекомендую этот вариант, так как он наиболее универсален. Один из способов такой настройки можно посмотреть, например, на <http://rus-linux.net/MyLDP/BOOKS/MDKman/ch02s02.html>.

Для того чтобы менеджер сети wicd заработал, необходимо в конец файла /etc/finit/postxserver добавить новую строку:

```
/etc/init.d/wicd start
```

Если же планируется часто использовать Bluetooth, то можно включить и этот сервис:

```
/etc/init.d/bluetooth start
```

Проблемы с системным временем (если в BIOS выставлено местное время) можно решить, присвоив значение «по» параметру «UTC» в файле /etc/default/rcS. При последующей загрузке исправьте часы в BIOS.

Теперь необходимо понять, каким образом запускать X-сервер. Можно включить автоматический вход и startx на имя своего пользователя, а можно использовать GDM. В любом случае для уменьшения времени загрузки необходимо будет выбрать одного пользователя для автоматического входа в систему. Важно помнить, что при использовании GDM будет невозможно запускать сервисы в фоне – это, конечно, не в лучшую сторону скажется на времени загрузки, но работа системы в целом будет стабильнее.

Разберем оба варианта.



## Автологин и startx

Откройте для редактирования `/etc/finit/finit.conf` и в поле `loginuser` после знака «=» между кавычками введите имя своего пользователя. Например, так:

```
loginuser = "serge"
```

Этого уже достаточно для загрузки с X-сервером. Для запуска XFCE вместо GNOME достаточно создать в домашней директории файл `.xsession` со следующим содержанием:

```
#!/bin/sh
exec /usr/bin/startxfce4
```

Наделите этот файл правами выполнения (выполнив команду «`chmod +x ~/.xsession`»). Он исполняется во время `startx` после запуска `/usr/bin/X`.

К сожалению, X-сервер во время работы не может обновить информацию об уже подключенных устройствах, из-за чего тачпад не будет распознан корректно после старта `udev`. По этой причине невозможно прокрутить документ вверх-вниз при помощи правой части тачпада, хотя «в целом» работать он будет.

## GDM

Для использования этого менеджера входа в систему необходимо в `/etc/finit/finit.conf` параметру `startx` присвоить значение `false`, а в конце файла `/etc/finit/postxserver` дописать:

```
/etc/init.d/gdm start
```

Проверьте после перезагрузки – теперь тачпад работает полностью.

## Время автономной работы и запуска

Стоит отметить, что такая существенная переделка не могла не повлиять на другие характеристики системы. Так, например, должно пострадать время автономной работы. Это заметно по показателям `powerop`: просыпаний в секунду стало больше на 20-25, а время пребывания процессора в «спящем» состоянии сократилось до 20-50 миллисекунд. Кроме того, теперь не получится загрузить модуль `eee` для управления частотой FSB:

процесс «`modprobe eee`» не может завершиться корректно. Впрочем, уменьшения времени автономной работы на увеличенной батарее я не ощутил: около 6-7 часов мне вполне достаточно.

После установки `Finit` в режиме автологина и `startx` время загрузки составило 41 секунду. В режиме `GDM` – на 4 секунды больше.

## Эксперимент: KDrive вместо X

Вы можете визуально заметить, что во всей цепочке загрузки системы только один компонент сильно теряет время на свою инициализацию – X-сервер. Для него существует облегченная версия, называемая `KDrive`, которую принято собирать из исходников с самостоятельным выбором необходимых компонентов. Подробную информацию по сборке можно прочитать на <http://www.pps.jussieu.fr/~jch/software/kdrive.html>. Для упрощения я воспользовался уже готовым `deb`-пакетом с того же сайта ([http://www.pps.jussieu.fr/~jch/software/kdrive/xvesa\\_0.20031107-2\\_all.deb](http://www.pps.jussieu.fr/~jch/software/kdrive/xvesa_0.20031107-2_all.deb)). Его можно установить обычным способом (например, через `Gdebi`). Данная сборка вполне стабильна, использует (как указано в названии пакета) драйвер `vesa` для видеокарты, но не лишена некоторых недостатков:

- ☑ Тачпад ноутбука не будет функционировать полноценно и в режиме `GDM` тоже.
- ☑ Переключение в консоль по `<Ctrl + Alt + F>` будет приводить к зависанию.
- ☑ Для работы от `startx` файлу `Xvesa` требуется присвоение `SUID`.
- ☑ В графической консоли отсутствует русский язык: буквы не отображаются, но смена шрифта должна решить эту проблему.
- ☑ Пожалуй, самый главный недостаток: отсутствие модуля `xkb`, в следствие чего для переключения раскладки клавиатуры придется использовать `xmodmap` и ему подобные утилиты.

Для постоянной работы `KDrive` я переименовал `/usr/bin/X` в `/usr/bin/X.old` (чтобы можно было вернуться к обычному `Xorg`), создал файл `/usr/bin/X`, наделил его правами запуска и добавил старт `KDrive` в ука-

занием разрешения, DPI и глубины цвета. Пример моего файла:

```
Xvesa -screen 800x480x32 -dpi 96
```

Далее установил пакет `915resolution`:

```
$ sudo apt-get install 915resolution
```

И добавил в самый конец файла `/etc/finit/prexserver` строку:

```
/etc/init.d/915resolution start
```

Эта операция позволит выставить максимальное разрешение экрана на стандартном драйвере видеокарты при использовании `GDM`. В режиме `startx` нельзя получить разрешение выше `640 x 480` без пересборки `KDrive` с драйвером «`i810`», который также включит и аппаратное ускорение. К сожалению, сам я такой операции не производил.

Переключение русской раскладки осуществляется через `xmodmap` и `xbindkeys`. За основу были взяты эти конфигурационные файлы `xmodmap` (<http://www.zaurus.ru/board/download.php?id=9242>). Я не буду приводить свои файлы в силу того, что для повседневной работы текущий `KDrive` не приспособлен.

Время загрузки составило 35 секунд.

## Закключение

Как видно из всех тестов пятой части цикла статей, время загрузки составило до 40% (в случае `KDrive`) от первоначального результата в 86 секунд. Стоит отметить, что внутрисистемная оптимизация позволила добиться не такого значительного уменьшения времени загрузки, как косметическая. Это говорит о неплохой изначальной оптимизации системы. Однако субъективно сама `Ubuntu` на `Eee PC` стала загружаться существенно быстрее, чем после косметических улучшений. Я остановился на варианте с обычным `Xorg` и автологином вместе с `startx`.

Этой статьей я заканчиваю цикл об использовании `Ubuntu` на `Eee PC`. До новых встреч!

Сергей Гулин  
([sugia@yandex.ru](mailto:sugia@yandex.ru))

# Тестирование программного обеспечения Open Source-средствами. Часть 1: JUnit

Существует множество методов тестирования программного обеспечения, позволяющих тестировать приложение на различных этапах разработки. В этой статье, разбитой на 3 части, вы познакомитесь со средствами модульного тестирования с открытым исходным кодом: JUnit, NUnit и CppUnit.

## Модульное тестирование

Модульное тестирование – это процесс, позволяющий проверить на наличие дефектов и ошибок отдельные модули программного обеспечения, обычно являющиеся отдельными классами. Такое тестирование активно используется при разработке по методике экстремального программирования, когда широко применяется подход так называемой разработки через тестирование. Последний заключается в том, что разработчик пишет тесты для программы до написания самой программы. Также модульное тестирование активно применяется при регрессионном тестировании, цель которого выявление дефектов, которые программист мог внести в уже проверенный ранее код.

Разработкой тестов в зависимости от применяемой в компании методики разработки ПО могут заниматься как сами программисты, так и инженеры по тестированию ПО. В методике экстремального программирования разработка тестов возложена на самого программиста, в то время как в других методиках (таких, как Microsoft Solution Framework и Rational Unified Process) этим занимаются отдельные специалисты, а программисты сосредоточены на реализации по точному техническому заданию.

Существует множество средств, облегчающих создание модульных тестов для программ на различных языках программирования. В данной статье рассмотрены три библиотеки модульного тестирования на основе открытого исходного кода:

- ✓ **JUnit** – средство модульного тестирования модулей, написанных на языке программирования Java.
- ✓ **NUnit** – средство модульного тестирования модулей, написанных на языке программирования C# (платформы .NET).
- ✓ **CppUnit** – средство модульного тестирования модулей, написанных на языке программирования C++.

При рассмотрении всех библиотек тестирования сначала будем создавать класс для последующей проверки (на соответствующем языке программирования). Затем будем разрабатывать тест и выполнять его с помощью библиотеки. При разработке класса для тестирования будем следовать всем правилам языка программирования, так как библиотеки не могут нарушать правил использования классов, таких как вызов закрытых и защищенных методов<sup>1</sup>.

## JUnit

Первая часть статьи посвящена JUnit – библиотеке для модульного тестирования программного обеспечения, написан-

ного на языке Java. Существуют две активные версии данной библиотеки:

- ✓ **3.x** – применяется для тестирования приложений, написанных для ранних версий платформы Java (1.3.x, 1.4.x);
- ✓ **4.x** – применяется для тестирования приложений, написанных для активных версий платформы Java (1.5.x, 1.6.x), и использует современные возможности языка, такие как аннотации<sup>2</sup>.

JUnit поставляется в виде отдельной библиотеки, упакованной в JAR, которую вы можете добавить к проекту для обеспечения возможности модульного тестирования. Кроме того, с сайта разработчиков (<http://www.junit.org>) можно скачать ZIP-архив с библиотекой и tar.gz-архив с ее исходными кодами. На том же сайте доступна документация по классам, входящим в состав библиотеки и обеспечивающим возможность тестирования.

## Использование версии 3.x

Для создания тестов данная версия библиотеки активно использует интерфейсы<sup>3</sup> Java. В данном случае они применяются для определения обязательного для реализации набора методов, которые будут вызваны библиотекой при тестировании. В этом случае на разработчика возлагается задача по написанию полностью законченного класса теста. Для того чтобы написать тест с использованием данной библиотеки, необходимо создать класс, реализующий интерфейс Test, или написать класс, расширяющий абстрактный класс TestCase.

Для начала напомним класс, который будем тестировать:

```
class Money {
    private int fAmount;
    private String fCurrency;

    public Money(int amount, String currency) {
        fAmount = amount;
        fCurrency = currency;
    }

    public int amount() {
        return fAmount;
    }

    public String currency() {
        return fCurrency;
    }

    public Money add(Money m) {
        return new Money( amount() + m.amount(), currency() );
    }

    public boolean equals(Object anObject) {
        if (anObject instanceof Money) {
            Money aMoney = (Money) anObject;
            return aMoney.currency().equals(currency()) &&
                amount() == aMoney.amount();
        }
        return false;
    }
}
```

Данный класс делает не так уж много: хранит в себе обозна-

1. В языке Java защищенные методы и поля доступны не только потомкам класса, но и классам из того же пакета, что и данный класс.
2. Аннотация – нововведение платформы Java 1.5, являющееся специальной конструкцией, позволяющей помещать дополнительные метаданные в классы. Эти метаданные могут быть прочитаны инструментальными средствами и во время выполнения программы средствами отражения (более развитый аналог RTTI в языке C++).
3. Интерфейс – это абсолютно абстрактный класс, все методы и поля которого являются открытыми.

чение валюты и ее количество. Вместе с тем мы написали метод, который позволяет прибавлять к имеющейся сумме дополнительную. Теперь напишем тест для данного класса. При этом не будем реализовывать интерфейс, а расширим существующий класс `TestCase`. Так мы избегаем необходимости реализации специальных методов, отвечающих за инициализацию и выполнение теста.

```
public class MoneyTest extends TestCase {
    public void testSimpleAdd() {
        // Создаем два объекта, которые будем тестировать,
        // и третий, который будет эталоном для проверки
        // результата операции
        Money m12CHF = new Money(12, "CHF");
        Money m14CHF = new Money(14, "CHF");
        Money expected = new Money(26, "CHF");
        // Производим действие сложения
        Money result = m12CHF.add(m14CHF);
        // Проверяем тестовое условие (являются ли равными
        // два объекта)
        Assert.assertTrue(expected.equals(result));
    }
}
```

В случае если при проверке условия произойдет ошибка, библиотека сообщит об этом и укажет на тест, который был провален. Теперь можно тестировать объект созданного класса.

Если необходимо изменить порядок выполнения тестов, то следует «перегрузить» специальные методы, которые отвечают за исполнение тестовых методов и сбор статистики по выполнению тестов. Чтобы сделать это, необходимо иметь общее представление о паттернах проектирования и ознакомиться с разделом документации JUnit, посвященным созданию тестов с помощью реализации интерфейсов и перегрузки соответствующих методов.

В данном примере показан тест, который проверяет два объекта на равенство. Существует множество наборов инструкций, с которыми можно познакомиться, прочитав документацию на класс `Assert`, содержащий статические методы проверки различных условий, показанных в **таблице 1<sup>4</sup>**.

После того как тест написан, необходимо его выполнить. Для этого потребуется откомпилировать исходные коды класса и самого теста.

## Запуск теста

После успешной компиляции теста и класса можно произвести тестирование. Для этого необходимо выполнить их, загрузив в `TestRunner`. Для отображения процесса и результатов выполнения теста можно использовать два вида:

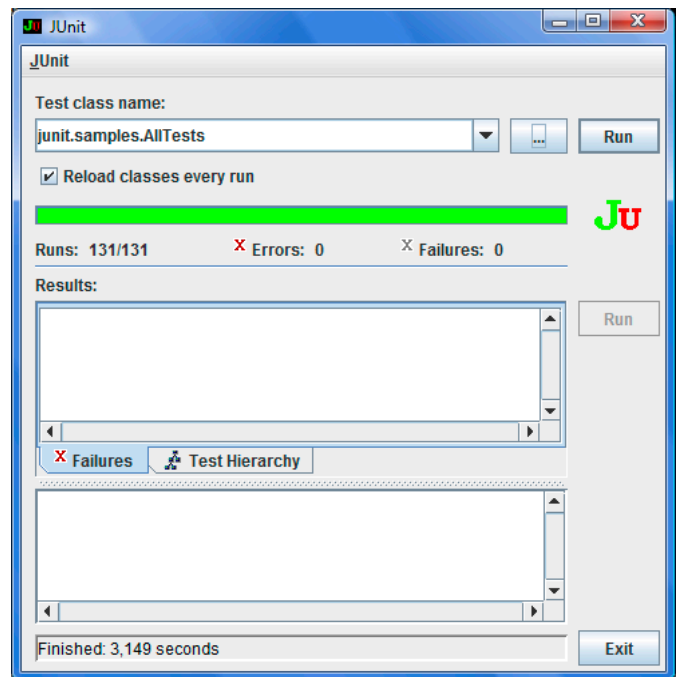
- ☑ **текстовая консоль** – результаты теста будут поступать на стандартный вывод;
- ☑ **графическая консоль** – графический интерфейс, с помощью которого можно выбирать тесты для выполнения.

Графический интерфейс удобен, если вы используете утилиту как независимую программу. В случае если вы хотите встроить тестирование в программу (оболочку, среду разработки и т.п.), текстовый вариант будет более предпочтительным.

Для запуска утилиты необходимо:

- ☑ Добавить к переменной окружения `CLASSPATH` путь к JAR-архиву с библиотекой.
- ☑ Выполнить:
  - для запуска в текстовом режиме стандартный вывод:

```
java junit.textui.TestRunner <класс.теста>
```



Графический интерфейс утилиты для выполнения тестов

- для запуска в графическом режиме AWT:

```
java junit.awtui.TestRunner [класс.теста]
```

- для запуска в графическом режиме Swing:

```
java junit.swingui.TestRunner [класс.теста]
```

## Использование версии 4.x

Данная версия библиотеки была разработана с учетом новых возможностей платформы Java, начиная с версии 1.5 (Tiger), и может использовать аннотации для указания специальных атрибутов для тестов, которые не использовались в третьей версии JUnit. Кроме того, необходимо отметить отсутствие графической консоли тестирования в данной версии – доступна лишь текстовая консоль, которая может быть запущена по аналогии с третьей версией.

Зато для новой версии существует иной способ запуска тестов:

- ☑ для запуска из командной строки:

```
java org.junit.runner.JUnitCore [имя_класса.класс]
```

- ☑ для запуска из приложения Java: встроив такой код в свою программу, можно с легкостью проводить тестирование модулей и компонентов; причем может быть выполнено сразу несколько тестовых классов:

```
org.junit.runner.JUnitCore.runClasses(имя_класса.класс, ...)
```

Используя пример, который был написан для третьей версии, перепишем только участок кода, отвечающий за тест:

```
@Test
public void simpleAdd() {
    Money m12CHF= new Money(12, "CHF");
    Money m14CHF= new Money(14, "CHF");
```

4. Если вы загляните в документацию, то обнаружите, что на самом деле методов больше, но они являются перегруженными версиями приведенных в таблице методов. Именно поэтому все методы в таблице приведены без точной сигнатуры.



```
Money expected= new Money(26, "CHF");
Money result= m12CHF.add(m14CHF);
assertTrue(expected.equals(result));
}
```

Как видно из примера, перед методом, производящим тестирование, появилась аннотация `@Test`, говорящая о том, что данный метод является тестом. Важное отличие в том, что теперь можно просто отмечать метод как тестовый, а не расширять класс `TestCase` и не реализовывать интерфейс `Test`.

Кроме того, появилась возможность с помощью аннотаций проверять возникновение ожидаемых исключений:

```
@Test(expected= IndexOutOfBoundsException.class)
public void empty() {
    new ArrayList<Object>().get(0);
}
```

Таким образом, мы проверяем возникновение ожидаемого исключения в случае, если индекс выйдет за границы массива.

Ниже в **таблице 2** приведены наиболее часто используемые аннотации.

В классе `Assert` есть и изменения: некоторые методы объявлены устаревшими, некоторые – удалены полностью (`failSame`, `failNotSame`, `failNotEqual`). Вдобавок, введен метод для тестирования массивов: `assertArrayEquals`, который вызывает ошибку, если все элементы одного массива не равны всем элементам второго массива. Самое значительное изменение в классе `Assert` – появление условного метода проверки `assertThat`,

который тестирует переданный объект на соответствие условию, реализующему интерфейс `Matcher`. Основным плюсом такого решения является то, что вы можете разрабатывать свои (специфичные) условия и использовать их повторно при написании других тестов.

В дополнение к методам проверки, вызывающим ошибку в случае невыполнения условия, появился новый класс `Assume` (предположение). Методы проверки данного класса отличаются тем, что в случае невыполнения условия ошибка не возникает, а появляется информационное сообщение о том, что предположение не исполнилось. Методы проверки предположений приведены в **таблице 3**.

## Заключение по JUnit

В результате рассмотрения JUnit были показаны основные принципы работы с ней, две ее версии и различия между ними. Очевидно, что новая версия библиотеки (версия 4) обеспечивает более простой и понятный способ написания тестовых классов благодаря механизму аннотаций. В связи с тем что старая версия платформы Java (1.3.x-1.4.x) постепенно уступает место более новым (1.5.x-1.6.x), то и старая версия библиотеки постепенно уступает свое место новой. Также были рассмотрены основные классы и их методы, которые используются при написании тестов: класс `Assert` и новый класс `Assume`, появившийся в библиотеке версии 4.

Библиотека JUnit имеет широкое распространение в среде программистов на языке Java и поэтому интегрирована во многие среды разработки (в частности, `NetBeans` и `Eclipse`), где создание тестов предельно упрощено, а их

результаты привязываются к исходному коду тестируемого класса, что позволяет быстро переходить к методу, содержащему ошибку. Благодаря интеграции можно производить и пошаговую отладку тестов в отладчике среды.

Использование библиотеки значительно облегчает разработку качественных классов, которые затем применяются при построении более сложных модулей:

- ☑ проверяется поведение класса и его соответствие техническому заданию;
- ☑ проверяется наличие ошибок при работе отдельных методов класса (кроме `private`-методов);
- ☑ проверяется открытый интерфейс класса и его недоработки;
- ☑ библиотека активно применяется при подходе разработки через тестирование.

Теперь и вы можете попробовать применить данную библиотеку в своем проекте на языке Java и оценить те плюсы, которые приносит применение модульного тестирования.

В следующих выпусках «Open Source» будут рассмотрены `Open Source`-средства тестирования `NUnit` и `CppUnit`.

Александр Шайхразеев  
([alexander.shaykhrzeev@gmail.com](mailto:alexander.shaykhrzeev@gmail.com))

Таблица 1. Статические методы проверки различных условий

Тип теста	Метод	Описание
Равенство	<code>assertEquals</code>	Проверяет равенство объектов, сравнивая их поля. Ошибка возникает, если объекты не равны
Сравнение ссылок	<code>assertSame</code>	Сравнивает ссылки двух объектов. Ошибка возникает, если ссылки не равны
	<code>assertNotSame</code>	Сравнивает ссылки двух объектов. Ошибка возникает, если ссылки равны
Условие	<code>assertNull</code>	Проверяет ссылку на объект на равенство <code>null</code> . Ошибка возникает, если ссылка не равна <code>null</code>
	<code>assertNotNull</code>	Проверяет ссылку на объект на неравенство <code>null</code> . Ошибка возникает, если ссылка равна <code>null</code>
	<code>assertTrue</code>	Проверяет результат логической операции на истинность. Ошибка возникает, если значение не является истиной
	<code>assertFalse</code>	Проверяет результат логической операции на ложность. Ошибка возникает, если значение является истинным
Вспомогательные	<code>fail</code>	Позволяет прекратить выполнение теста с сообщением об ошибке
	<code>failSame</code>	Позволяет прекратить выполнение теста по условию равенства ссылок
	<code>failNotSame</code>	Позволяет прекратить выполнение теста по условию неравенства ссылок
	<code>failNotEquals</code>	Позволяет прекратить выполнение теста по условию неравенства объектов

Таблица 2. Наиболее часто используемые аннотации

Название	Описание
<code>Before</code>	Данной аннотацией помечают метод класса, захватывающий системные ресурсы, такие как файлы, дескрипторы доступа к БД и т.п.
<code>After</code>	Данной аннотацией помечают метод, отвечающий за освобождение ресурсов, которые были захвачены у системы для проведения теста
<code>Test</code>	Данной аннотацией помечают методы класса, производящие тесты над испытуемым классом

Таблица 3. Методы проверки предположений

Название	Описание
<code>assumeNoException</code>	Предполагает, что проверяемый метод завершится, не вызвав исключений. В случае исключения сообщает об этом
<code>assumeNotNull</code>	Предполагает, что передаваемый аргумент не является ссылкой на <code>null</code>
<code>assumeThat</code>	Предполагает выполнение условия. Аналогично тому, как работает метод проверки <code>assertThat</code> , но в случае невыполнения условия ведет себя иначе
<code>assumeTrue</code>	Предполагает, что значение передаваемого аргумента – истина

# «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](mailto:osa@samag.ru).

## Подписные индексы:

**20780\***

+ диск с архивом статей 2008 года

**81655\*\***

без диска

по каталогу агентства «Роспечать»

**88099\***

+ диск с архивом статей 2008 года

**87836\*\***

без диска

по каталогу агентства «Пресса России»

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

## Подписка на журнал «Системный администратор»

### Российская Федерация

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

### СНГ

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

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

- ☑ **Казахстан** – по каталогу «Российская пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»
- ☑ **Беларусь** – по каталогу изданий стран СНГ через РГО «Белпочта» (220050, г. Минск, пр-т Ф. Скорины, 10)
- ☑ **Узбекистан** – по каталогу «Davriy nashrlar», российские издания через агентство по распространению печати «Davriy nashrlar» (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)
- ☑ **Армения** – по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г. Ереван, пл. Сасунци Давида, д. 2) и ЗАО «Контакт-Мамул» (375002, г. Ереван, ул. Сарьяна, 22)
- ☑ **Грузия** – по списку номенклатуры «АРЗИ» через АО «Сакпресса» (380019, г. Тбилиси, ул. Хошараульская, 29) и АО «Мацне» (380060, г. Тбилиси, пр-т Гамсахурдия, 42)
- ☑ **Молдавия** – по каталогу через ГП «Пошта 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