

## Колонка главного редактора



Несомненно, одним из главных событий последних дней стал первый предварительный выпуск операционной системы Google Chrome OS.

Ее исходный код уже доступен в рамках проекта Chromium OS ([www.chromium.org/chromium-os](http://www.chromium.org/chromium-os)). Если кто-то вдруг еще не знает, то данная ОС представляет собой связку из Linux-ядра, графического сервера X.Org и запущенного в нем веб-браузера Chrome, в рамках которого запускаются все приложения. С этой инициативой в Google намерены возглавить движение современных операционных систем, основанных на веб-технологиях. Их перспективы уже давно не дают покоя аналитикам, специалистам и предпринимателям.

Просматривая слайды с презентации Chrome OS, поймал себя на другой мысли, которая не покидает многие умы все последние годы... Google весьма аккуратно и уверенно движется в направлении создания нового мира ИТ, и ключевое место в нем, разумеется, занимают продукты самой компании. Вот и получается, что с одной стороны, разработку Chrome OS олицетворяют красиво нарисованные человечки, подписанные как «Open Source community», но с другой — среди иконок приложений, расположенных вокруг нетбука с Chrome OS, очевидным образом вырисовываются именно сервисы Google. Открытость, конечно, подразумевает свободу от любых привязок, но как окажется на самом деле?

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

### «Open Source»

электронное приложение к журналу  
«Системный администратор»  
№52, 20 ноября 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 статей ответственность несет автор. Все права на опубликованные материалы защищены.

## Новости мира Open Source

### Ubuntu Linux обновился до 9.10 «Karmic Koala»

29 октября состоялся новый релиз Linux-дистрибутива Ubuntu — 9.10 «Karmic Koala» в редакциях Desktop Edition и Server Edition.

Среди новшеств в десктоп-версии Ubuntu 9.10 «Karmic Koala» отмечаются:

- ☑ переход на систему загрузки Upstart, ускоряющую время старта операционной системы;
- ☑ новый центр программного обеспечения Ubuntu Software Center;
- ☑ графическая среда GNOME 2.28, где Empathy заменил Pidgin в качестве IM-клиента по умолчанию;
- ☑ инструмент Quickly, упрощающий процесс разработки приложений для Ubuntu;
- ☑ Linux-ядро 2.6.31;
- ☑ система hal объявлена устаревшей в пользу «DeviceKit-power», «DeviceKit-disks» и «udev»;
- ☑ в драйвере для видеокарт Intel метод ускорения заменен с EXA на UXA;
- ☑ ext4 стала файловой системой по умолчанию;
- ☑ GRUB2 стал загрузчиком ОС по умолчанию;
- ☑ улучшен процесс инсталляции Ubuntu на iSCSI.

### Canonical реформировала Ubuntu Shiplt

Компания Canonical объявила о том, что больше не будет рассылать бесплатные компакт-диски с Ubuntu Linux всем желающим.

Ввиду больших затрат на бесплатную рассылку дисков с Ubuntu во все страны в рамках акции Shiplt, в Canonical решили реформировать эту инициативу. Отныне диски с Ubuntu смогут бесплатно заказывать только официальные участники сообщества Ubuntu.

Всем остальным предлагается обновлять свои системы через Сеть или скачивать ISO-образы и записывать диски самостоятельно. Кроме того, можно купить комплекты CD с последним релизом Ubuntu (9.10 «Karmic Koala») на [shop.ubuntu.com](http://shop.ubuntu.com). Комплект из 5 дисков обойдется в 5,85 евро, а из 200 — в 158,35 евро.

За время функционирования Shiplt через этот сервис компанией Canonical были разосланы миллионы дисков с Ubuntu Linux пользователям всего мира, что сыграло значительную роль в популяризации Linux-дистрибутива.

### Французское правительство переходит на продукты Mozilla

Государственное казначейство Франции объявило о намерении перевести 130 тысяч десктопов на Open Source-приложения Thunderbird и Lightning. Продукты проекта Mozilla: Thunderbird для работы с электронной почтой и Lightning для календарей и совместной работы — заменят на правительственных настольных ПК Франции проприетарные аналоги от IBM (Lotus Notes) и Microsoft (Outlook). Об этом стало известно от французского сайта ИТ-новостей [01net](http://01net.fr). Переход на Open Source-приложения отчасти связан со слиянием французских ведомств DGP (Directorate General of Public Finance), DGI (Directorate General of Taxes) и DGCP (Directorate General of Public Accountancy). У каждого из них были свои лицензии на Lotus Notes и Microsoft Outlook, а теперь было решено, что массовый переход на Open Source-аналоги позволит сэкономить финансовые затраты на программное обеспечение и упростить работу с лицензиями на него.

### Релиз GNOME 3.0 перенесен на сентябрь 2010

Винсент Унц (Vincent Untz) из GNOME Release Team отправил в почтовую рассылку [gnome-announce-list](mailto:gnome-announce-list) письмо, в котором объявил о том, что следующий крупный релиз GNOME — 3.0 — состоится в сентябре 2010 года. Ранее планировалось, что GNOME 3.0 появится уже в марте, однако результаты недавнего опроса разработчиков этой графической рабочей среды показали, что далеко не у всех компонентов GNOME будут достаточно стабильные и отлаженные новые версии к этому сроку. Сообщается, что принятое решение позволит разработчикам подготовить два хорошо протестированных релиза GNOME: 2.30 — в марте и 3.0 — в сентябре. «Давайте сделаем 2010 год сказочным для GNOME!» — завершил свое письмо Винсент.

### Skype готовит Open Source-версию Linux-клиента

В компании Skype подтвердили слухи о том, что идет разработка Open Source-версии клиента для GNU/Linux. Впервые эта информация появилась в блоге одного из разработчиков Mandriva, которому стало известно о готовящейся Open Source-версии клиента Skype. После появления этих сведений на [slashdot.org](http://slashdot.org) в Skype решили опубликовать официальное объяснение.

Как сообщается на сайте Skype, сейчас действительно идут работы по созданию Open Source-версии Linux-клиента Skype. Ожидается, что этот клиент «станет частью более существенного предложения», но подробности о нем пока не разглашаются. В Skype полагают, что наличие GUI-приложения с открытым кодом поможет ему лучше адаптироваться в мире разношерстных Linux-дистрибутивов и на других платформах, что в конечном счете ускорит процесс разработки. Из этой информации осталось непонятно, останется ли закрытым сам протокол Skype, поскольку в заявлении говорится лишь об открытом коде приложения с графическим интерфейсом.

## Основатель KDE получил высший орден Германии

Маттиас Эттрих (Matthias Ettrich), основатель проекта свободной графической рабочей среды KDE, был награжден высшим орденом Германии – «За заслуги перед Федеративной Республикой Германия» (Der Verdienstorden der Bundesrepublik Deutschland).

Церемония вручения ордена состоялась 6 ноября в Сенате Берлина. Награда была присуждена основателю KDE за его вклад в мир свободного программного обеспечения как признание его работы над распространением инноваций и знаний для всеобщего блага. Сам Эттрих был настолько удивлен приглашением на вручение ордена ФРГ, что сначала принял электронное письмо за спам.

Датой основания проекта KDE считается 14 октября 1996 года, когда Маттиас Эттрих разослал электронное письмо с официальным анонсом KDE, предложив использовать Qt в качестве базовой библиотеки для виджетов и обозначив основные компоненты будущей графической среды Kool Desktop Environment.

## Simply Linux – новый дистрибутив на базе ALT Linux

Компания ALT Linux объявила о появлении «первого заметного релиза пользовательской системы, который сделан на основе ее технологий, но не ею самой» – Simply GNU/Linux 5.0.0.

Лидером и автором проекта Simply Linux является Денис Корявов – участник команды ALT Linux Team. Со временем к нему присоединились единомышленники, и теперь у дистрибутива есть свой логотип, дизайн, концепция, а сайт находится в стадии разработки. Заявляется, что Simply Linux ориентирован на начинающих пользователей.

В основе новой системы – репозиторий ALT Linux Sisyphus, а текущий релиз базируется на стабильной ветке «Пятая Платформа» (релиз «официальных» дистрибутивов ALT Linux на ее основе состоялся в конце октября). Среди программного обеспечения, включенного в состав Simply Linux, – легковесный оконный менеджер Xfce 4.6.1, браузер Firefox 3.5.3 с Яндекс.Баром, почтовый клиент Thunderbird 3.0pre4, IM-клиент Pidgin 2.6.3, офисный пакет OpenOffice.org 3.1.1, графический редактор GIMP 2.6.7, музыкальный плеер Sonata 1.6.2, мультимедийный плеер GNOME-MPlayer 0.98. Из системного программного обеспечения можно выделить Linux-ядро 2.6.30, Glibc 2.10 и GCC 4.4.1.

## Вышел Mandriva Linux 2010

7 ноября французская компания Mandriva разослала по электронному почте официальный анонс новой версии своего флагманского продукта – дистрибутива Mandriva Linux 2010 «Adelie».

Среди наиболее значимых изменений в релизе Mandriva Linux 2010 выделяются: отказ от устаревшей версии графической среды KDE 3 в пользу актуальной KDE 4 (предусмотрена миграция пользовательских настроек/данных из KDE3); переход на ext4 в качестве файловой системы по умолчанию; упро-

щение интерфейса инструмента разбиения жесткого диска, используемого в инсталляторе; создание гостевого пользовательского аккаунта по умолчанию (его домашний каталог монтируется как tmpfs); переход на Plymouth для управления заставкой загрузки системы вместо Splashy; новый фреймворк безопасности Tomyo, пришедший на смену AppArmor (оснащен графическим средством управления – tomyo-gui); новое средство сетевого мониторинга – net\_monitor; поддержка Moblin в качестве одного из доступных окружений.

## Опубликован исходный код Android 2.0 «Eclair»

Компания Google опубликовала исходный код новой версии своей мобильной Linux/Java-платформы – Android 2.0 «Eclair» – для свободного скачивания на [source.android.com](http://source.android.com).

До недавнего времени Android 2.0 распространялся только в качестве прошивки, предварительно установленной в смартфоны Motorola Droid, продаваемые сотовым оператором Verizon на территории США. Теперь исходники доступны для всех желающих модифицировать и использовать новую версию Android.

Тем временем автор сторонних сборок CyanogenMod, который недавно стал инициатором создания независимой организации Open Android Alliance, уже пересобрал исходники Android 2.0 для запуска этой прошивки на первом смартфоне с Android – T-Mobile G1. Раньше у энтузиастов были опасения, что новые версии Android не будут функционировать на G1, однако, по словам автора CyanogenMod, релиз 2.0 работает «быстро и гладко».

В HTC подтвердили информацию о том, что они готовят обновление с Android 2.0 для коммуникатора HTC Hero.

## Стартовал проект русскоязычной версии ОС MINIX 3

Российские энтузиасты объявили о запуске проекта по созданию русскоязычной версии операционной системы MINIX 3.

Основные цели проекта – обеспечить русскоязычных пользователей системы MINIX 3 удобной и полной документацией и локализацией, а также включить в дистрибутив дополнительное программное обеспечение. Создаваемая редакция MINIX 3 ориентирована не только на разработчиков и пользователей встроенных систем, но и на применение в качестве демонстрационной системы для обучения специалистов по информационным технологиям в высших учебных заведениях.

Первая русская редакция MINIX 3 будет базироваться на стабильном релизе 3.1.5 ревизии 5612. Дистрибутив будет распространяться в двух вариантах: LiveCD и LiveUSB – с возможностью установки на жесткий диск (также ожидаются образы для виртуальных машин VMware и QEMU). Дата выхода первого релиза приурочена ко Дню защитника Отечества – 23 февраля 2010 года.

## Linux-дистрибутив Fedora обновился до версии 12

17 ноября вышла новая версия популярного бесплатного Linux-дистрибутива Fedora – 12 «Constantine».

Среди ключевых новшеств в релизе отмечаются:

- ☑ Произведена смена основной 32-битной архитектуры с i586 на i686. Пересборка пакетов под более современные процессоры дает прирост в общей производительности системы.
- ☑ Улучшенная поддержка веб-камер (обработка видео в libv4l, многочисленные тестирования).
- ☑ Новый инструмент для автоматической отправки сообщений об ошибках – ABRT (Automatic bug reporting tool).
- ☑ Повсеместное использование файловой системы ext4 (даже для загрузочного раздела жесткого диска благодаря GRUB 2).

- ✓ Новый алгоритм сжатия в RPM-пакетах (XZ – новый формат LZMA).
- ✓ KDE 4.3, GNOME 2.28.1 и десктоп Moblin (для нетбуков).
- ✓ Ранняя (экспериментальная) версия GNOME Shell – ключевого компонента будущего релиза GNOME 3.0.

## eIRTE и eFinder – российские разработки на базе jQuery

Российские разработчики представили два своих Open Source-проекта, написанных на JavaScript с использованием jQuery UI: WYSIWYG-редактор eIRTE и файловый менеджер eFinder (<http://elrte.ru>).

Среди основных возможностей eIRTE – редактирование текста, изменение его стиля; вставка и управление свойствами различных элементов (изображения, таблицы, списки и т.д.); просмотр и редактирование HTML-кода; обычный и полноэкранный режим работы. Из особенностей выделяются расширенные возможности работы с CSS, небольшой размер, простота подклю-

чения и высокая скорость загрузки, поддержка локализации, интеграция с eFinder.

eFinder – файловый менеджер для Web, созданный «под вдохновением от простоты и удобства программы Finder в Mac OS X». Среди его возможностей и особенностей – все основные операции с файлами и каталогами на удаленном сервере; три вида отображения текущей директории (список, маленькие иконки и предпросмотр); мультизагрузка файлов; возможность настройки корневой директории, прав доступа и привилегий; встроенный редактор текстовых файлов.

Функционирование обоих продуктов тестировалось в веб-браузерах Firefox 3.5, Internet Explorer 7 и 8, Safari 4, Opera 10 и Chrome. eIRTE и eFinder распространяются под свободной лицензией BSD.

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

# gOS: операционная система в стиле Web 2.0

## Встречайте – gOS!

gOS (или «good OS», <http://www.thinkgos.com>) разработана Good OS LLC – одной из софтверных компаний Кремниевой долины в США. Последняя была основана в начале 2007 года, а уже 1 ноября 2007 года стартовали продажи компьютеров Everex Green gPC TC2502, работающих под управлением первой версии gOS (1.0.1). Тогда один из директоров компании, Дэйв Лью (Dave Liu), писал, что после встречи с представителями Open Source-сообщества он вдруг понял, что его мечту: максимально широко распространить использование технологий Web 2.0 – можно осуществить путем создания дистрибутива Linux, который бы облегчил пользователям доступ к Google и другим онлайн-приложениям. Именно этим были определены все особенности как первой, так и последующих версий операционной системы gOS. Интересующиеся могут познакомиться с историей выхода всех версий этой ОС и кратким описанием их особенностей ([http://en.wikipedia.org/wiki/GOS\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/GOS_(operating_system))).

Последняя версия, имеющаяся к моменту написания обзора, – gOS 3.1 Gadgets (SP1) «Афродита», представляет собой пересборку Ubuntu Hardy Heron (8.04.1) с GNOME в качестве рабочего окружения и предназначена для использования на нетбуках и неттопах. Эта версия была выпущена 3 января 2009 года и является обновлением (с фиксацией ошибок) предыдущего релиза gOS – 3.0. Последнюю версию в виде образа LiveCD можно скачать с сайта проекта. К слову, релиз Ubuntu Hardy Heron был использован в качестве базы еще для версии gOS Escape pod 2.9M, которая вышла в начале 2008 года. На тот момент времени это был самый последний релиз Ubuntu, имеющий к тому же статус LTS (Long Term Support, т.е. релиз с длительной поддержкой – в течение

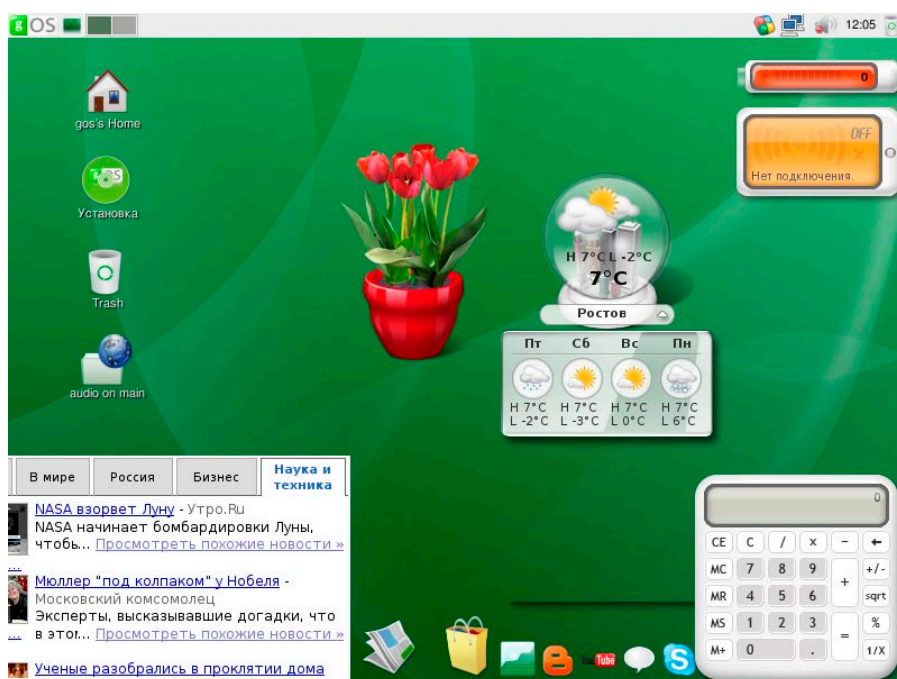
трех лет для десктопа). Скорее всего, именно из-за этого команда разработчиков gOS оставила эту версию Ubuntu в качестве базовой и для последующей версии gOS (3.0 Gadgets).

Сайт компании достаточно беден, и на нем практически отсутствует сколь-

либо полезная информация, кроме нескольких ссылок на зеркала для скачивания и не очень большого количества красивых скриншотов. Возможно, авторы решили, что простота установки и использования «хорошей ОС» не должна требовать чего-либо иного. Но на мой взгляд, полезной может оказаться хотя бы ссылка на неофициальные форумы gOS (<http://gosforums.org>), куда можно обратиться в случае возникновения проблем. Сразу следует отметить, что все, что есть на сайте и на форумах, – на английском языке.

## Перед установкой

Прежде всего познакомимся с требованиями к железу, на котором должна работать gOS. Заявленные минимальные (рекомендуемые) требования таковы:



Общий вид рабочего стола gOS



- ✓ процессор x86 с частотой не ниже 300 (700) МГц,
- ✓ 64 (384) Мб оперативной памяти,
- ✓ не менее 4 (8) Гб дискового пространства (для полной инсталляции и пространства для swap),
- ✓ графическая карта VGA, поддерживающая по меньшей мере разрешение 640 x 480 (1024 x 768),
- ✓ CD/DVD-привод или сетевая карта.

В общем, «аппетиты» у этой ОС – очень и очень скромные по сегодняшним меркам. Что логично – ведь авторы позиционируют gOS как легковесную систему, предназначенную для использования главным образом на нетбуках. О каких-либо дополнительных требованиях к аппаратному обеспечению на официальном сайте не сообщается, однако, побродив по форумам gOS, удалось выяснить, что без сложностей с «железом» не обходится. Но в них ничего необычного – обычный спектр проблем, которые возникают при установке любой не очень мощной ОС на широкий парк разнородных компьютеров.

Если вам не повезет и у вас во время установки gOS или при ее использовании возникнут сложности, а на форумах gOS вы не найдете решения, то ответ стоит поискать в Google Groups (<http://groups.google.com/group/goslinux>) или... на форумах Ubuntu (<http://ubuntuforums.org>). Не забывайте, что gOS 3.1 Gadgets (SP1) основывается на Ubuntu Hardy Heron (8.04.1), так что с большой вероятностью проблемы, возникающие в этих ОС, а также способы их решения будут аналогичными.

## Работа с LiveCD

При запуске LiveCD с gOS 3.1 есть возможность сменить язык интерфейса (59 вариантов, в том числе и русский) и раскладку клавиатуры (97 вариантов), а также выбрать режим установки: обычный, безопасный графический, использовать диск с дополнительными драйверами, OEM-установка (с созданием временного пользователя «oem» для предпродажной конфигурации системы производителями компьютеров). Кроме того, можно сразу включить специальные возможности: высокий контраст, лупа, программа чтения с экрана, интерфейс Брайля, клавиш-модификаторы, экранная клавиатура. Есть возможность указать параметры загрузки, необходимые для отдельных платформ и некоторых дисковых контроллеров. После того как операционная система загрузится, на экране монитора появляется стильный зеленый рабочий стол с гаджетами от Google: календарем, погодным информатором, калькулятором, новостной лентой, индикатором беспроводных подключений, индикатором заряда батареи

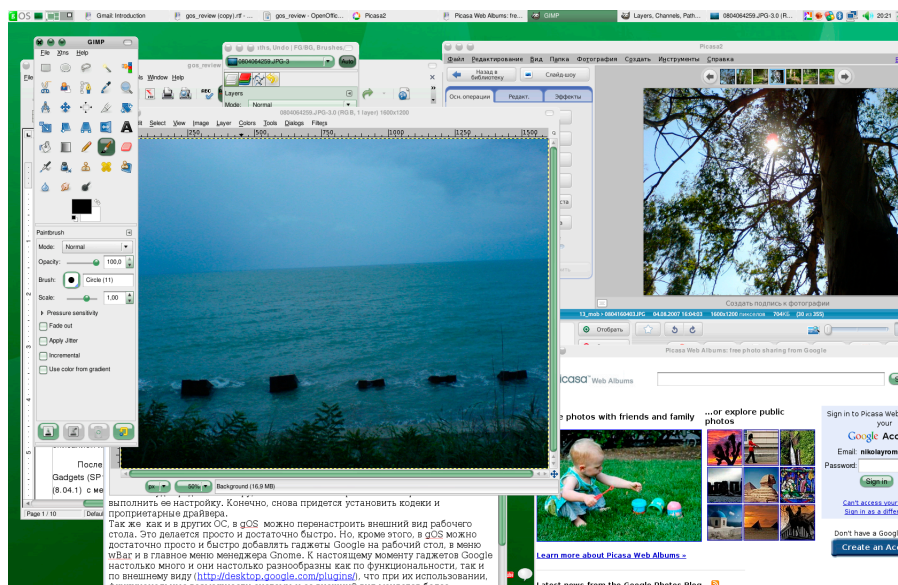
и виртуальными красными тюльпанами в таком же красном горшке. Действительно очень красиво! Теперь – подробнее о рабочем столе. В нижней части экрана – панель wBar с иконками в стиле MacOS X, увеличивающимися при наведении курсора мыши. Среди приложений в wBar по умолчанию – браузер Firefox, доступ к почте Gmail и ряду других сервисов от Google, IM-клиенты Skype или Pidgin, офис OpenOffice.org. В верхней части экрана расположена панель менеджера GNOME. Гаджеты Google, расположенные на рабочем столе, легко отключаются с помощью иконки, расположенной в правой части панели GNOME.

Запуск любой ОС как LiveCD – скорее для ознакомления с ОС, а не регулярного использования. Тем не менее, хотелось бы заострить внимание еще на некоторых аспектах работы gOS именно как LiveCD. Во-первых, удобство и скорость загрузки. Если у вас сравнительно «обычное» железо – проблем с запуском не будет: я это проверял на типичных офисных и домашних компьютерах, а также ноутбуках. Скорость загрузки gOS при сравнении с дистрибутивом Ubuntu 8.04.3 (Desktop, i386) была несколько быстрее, хотя на мощных современных компьютерах эта разница практически не видна. Тем не менее очевидно, что разработчики уменьшили количество установленных в систему пакетов и запускаемых по умолчанию сервисов.

Во время первоначальной загрузки ОС полезно не забыть выбрать русский язык. Это позволит вам работать с документами на русском языке. Что же касается русификации интерфейса, то она, конечно, присутствует, но в очень ограниченном виде. Простые надписи в интерфейсе – на русском языке, но если что-то посложнее – мы увидим английский текст. Многие окна с настройками гаджетов, апплетов и других компонентов системы – тоже на английском языке. Если же у вас вдруг возникнут проблемы и будет выдана какая-либо диагностика, то она тоже будет на английском. Следующий момент – простота подключения к сети. gOS в режиме LiveCD без проблем определяла используемые компьютерами сетевые платы/беспроводные адаптеры и выполняла всю необходимую настройку. (Среди «опробованного» сетевого железа преобладали изделия фирмы D-Link, а среди беспроводного – интерфейсы на основе Atheros.)

Что еще обычно нужно от компьютера? Конечно, возможность поработать над документами. Для этого в gOS имеется офисный пакет OpenOffice.org версии 2.4.1, а также предлагается воспользоваться онлайн-сервисом Google Docs. Для графических изображений есть классический редактор в виде Gimp и онлайн-сервис – Picasa. Монтирование носителей осуществляется автоматически при выборе соответствующего устройства в основном меню в пункте Places. В системе есть драйверы для чтения/записи данных на разделах NTFS, так что не возникнет проблем с доступом к файлам в Windows. Кроме того, установлен и WINE, позволяющий запускать Windows-приложения.

Теперь о мультимедийных компонентах. В gOS есть все для прослушивания музыки и просмотра видео, но нужно сделать одно уточнение. Если для прослушивания аудиозаписи достаточно просто двойного щелчка мыши по иконке файла с аудиозаписью, то для просмотра видеоматериала потребуются сделать немного больше. Дело в том, что в gOS изначально не установлены видеокодеки. Когда вы попытаетесь запустить просмотр видео, от-



GIMP, Picasa и другие приложения в gOS

кроется окно, в котором предложат загрузить недостающие кодеки из репозитория. После вашего согласия они будут скачаны и установлены. Стоит также напомнить, что, если вы выйдете из gOS, а потом снова загрузитесь с LiveCD, кодеки придется устанавливать заново. Аналогично при выходе из gOS не сохраняются настройки гаджетов Google и некоторые другие системные параметры. Все вышесказанное в полной мере относится и к случаям, когда вам для какого-нибудь устройства потребуются проприетарные драйверы. К сожалению, в стандартной комплектации gOS LiveCD не предусмотрено какого-либо простого средства для сохранения текущих настроек системы и гаджетов, например, на флешке или на CD/DVD.

Если вы хотите более глубоко ознакомиться с возможностями gOS, следует запустить менеджер пакетов Synaptic. Обратите внимание на то, что, кроме уже перечисленных средств работы с офисными документами, доступа к сетям, мультимедийных возможностей, в gOS есть классический набор сетевого инструментария, позволяющий проверять работоспособность сетей и выполнять несложную их настройку. И пару слов о пользователях: при загрузке LiveCD вы получаете доступ как root, однако ряд привилегий этого пользователя (администрирование системы, автоматический доступ к внешним носителям и некоторые другие) по умолчанию отключен. При необходимости их достаточно просто включить.

Сравнительно медленная загрузка LiveCD, необходимость сохранять текущую настройку, а затем не забывать ее восстанавливать – все это, конечно, добавляет хлопот, но, увы, такова особенность работы с «живыми дисками». Для постоянного использования gOS, разумеется, лучше установить ее на жесткий диск.

### Устанавливаем gOS 3.1

Для того чтобы установить gOS 3.1 Gadgets (SP1) на жесткий диск, нужно после полной загрузки LiveCD выбрать иконку «Установка» на рабочем столе или одноименный пункт главного меню (System Preferences → Администрирование → Установка). Так вы запустите инсталлятор системы. Сама установка практически не отличается от инсталляции аналогичных дистрибутивов: вы выбираете язык и часовой пояс, затем указываете, где на жестком диске будет размещена система, создаете регистрационную запись, еще раз выбираете русский язык и русскую раскладку клавиатуры и, нажав на кнопку Install, запускаете процесс установки. Инсталляция происходит в автоматическом режиме, а подробности о ее ходе, к сожалению, скрыты от пользователя. На финальной стадии система обращается к сети: пытается найти зеркала репозитариев и скачать обновленные версии приложений и системных пакетов. При этом закачиваются и языковые пакеты. После полной установки системы еще некоторые английские надписи в интерфейсе системы будут заменены русскими (но, к сожалению, не все).

Итак, система установлена – зайдём в нее. В принципе на рабочем столе все то же самое, что и в LiveCD, но система по понятным причинам работает быстрее. Поскольку gOS теперь установлена для постоянной эксплуатации, имеет смысл более полно настроить внешнюю конфигурацию. Так же, как и в других ОС, в gOS можно изменить внешний вид рабочего стола – это делается просто и достаточно быстро. Кроме этого, в gOS можно легко добавлять гаджеты Google на рабочий стол, в меню wBar и в главное меню менеджера GNOME. К настоящему моменту гаджетов Google настолько много и они настолько разнообразны как по функциональности, так и по внешнему виду (<http://desktop.google.com/plugins>), что при их использовании функциональные возможности системы и ее внешний вид меняется более радикально, чем при традиционной смене тем, обоев, шрифтов и дру-

гих обычных настроек. Для добавления веб-приложений на рабочий стол в системе gOS используется Mozilla Prism, что существенно повышает стабильность работы приложений, – это достигается за счет их изоляции в Prism (падение одного из них не приводит к падению других, как в «обычных» браузерах Mozilla). Весь интерфейс веб-приложения размещается в таких же окнах, что и «родные» программы, установленные в самой gOS, а не в окне браузера так, как мы это привыкли видеть.

Поскольку теперь мы имеем дело с ОС, установленной на жестком диске, логично заняться администрированием системы – завести пользователей и группы, выдать им права и разграничить между ними ресурсы. Также можно более тонко настроить доступ к проводным, беспроводным сетям, Bluetooth-устройствам и, если потребуется, использовать для этого шифрование. Для всего этого в gOS есть соответствующий инструментарий, и все это делается приблизительно так же, как и в других Linux-системах. Еще раз подчеркну, что gOS – сетевая ОС, и для нее крайне важно подключение к Интернету по хорошему каналу связи. Из Сети она берет текущие данные, используемые в гаджетах Google: сводки погоды, новости, сообщения с бирж и другое, через Сеть закачиваются обновления и патчи (система постоянно пользуется репозиториями для собственного обновления, и при достаточно широком канале все это делается в фоновом режиме, не мешая пользователям).

### Резюме о «хорошей ОС»

Система gOS 3.1 Gadgets (SP1) достаточно хороша в том смысле, что не выдвигает для своей работы каких-либо специфических требований. Она, скорее всего, будет без проблем работать на компьютерах, предназначенных для Windows, и с носителями, подготовленными для работы с Windows. Она хорошо скомпонована – в ней есть полный «джентльменский» набор ПО, позволяющий использовать компьютер в качестве типового офисного или в качестве несильно «навороченного» домашнего компьютера. Также она хороша потому, что не потребует от вас специальных знаний – вариант установки системы по умолчанию удобен и не нуждается в дополнительной настройке. Однако, если вам все же что-то не понравилось, есть возможность все переделать под свой вкус и свои нужды. Удобно, что есть LiveCD, но если эта ОС пришлась вам по душе, то, скорее всего, просто установите ее на жесткий диск, а «живой диск» будете запускать у своих друзей для демонстрации, как эта система работает на их технике.

Поскольку система рассчитана в основном на потребительский рынок, а не на специалистов ИТ, многие ее настройки изначально выполнены в стиле Windows. Возможно, это не очень понравится энтузиастам Linux, но надо признать, что пользователю, который до этого момента использовал одну из версий Windows, будет проще осваиваться в gOS. Linux-гуру достаточно быстро все перенастроят по-своему, а для тех, кто новичок в Linux, не придется методом долгих проб и ошибок искать, как, например, изменить раскладку клавиатуры. Это тоже хорошо.

Однако самое ценное в системе – ее ориентация на использование технологий Web 2.0. Благодаря этому она занимает промежуточное положение между «обычными» ОС, рассчитанными на использование только тех приложений, которые установлены в самой ОС, и новым поколением ОС, в которых, как предполагается, будут использоваться исключительно веб-приложения, установленные и работающие на просторах Интернета где-то там – «в облаке». «Облачным» технологиям пророчат радужные перспективы, так что в этом аспекте gOS даже опережает другие дистрибутивы своего класса.

Николай Ромоданов  
([nromodanov@gmail.com](mailto:nromodanov@gmail.com))

## Свободный паскаль. Часть 2: Lazarus

Это вторая часть материала, посвященного «свободному паскалю». Первую – про Free Pascal и GNU Pascal – см. в выпуске «Open Source» 050 (30.09.2009). – **Прим. ред.**

### Введение

Pascal, в его объектно ориентированном варианте Object Pascal, вышел на пик своей популярности с появлением визуальной среды разработки Borland Delphi. Delphi кардинально упростила процесс создания GUI для Windows и позволила абстрагироваться от чрезмерно сложного WinAPI, чем и снискала себе большую популярность. Вышедшая в 2002 году Delphi 7 стала по сути эталоном и стандартом для визуальных сред и зачастую используется по сей день.

Однако дальше последовала серия неудач. Delphi, будучи продуктом, ориентированным сугубо на систему Windows (о попытках ее адаптации к Linux ниже), оказалась заложником активного продвижения фирмой Microsoft платформы .NET и языка C#. Delphi 8 для .NET и Delphi 2005 получились неудачными и нестабильными. Delphi 2006 стала мультязычной IDE для C#, Delphi.NET, Delphi Win32 и C++, однако такое усложнение не оправдало себя. В 2006 году компания Borland по сути отказалась от Delphi, выделив ее в отдельную компанию, а затем и продав на фоне спада интереса к продукту. Сейчас Delphi поддерживается компанией Embarcadero Technologies и ее популярность невелика.

Еще в 2001 году компания Borland создала «Delphi для Linux» под названием Kylix. Проект не оправдал надежд (как в техническом, так и в коммерческом смысле) и был заброшен после третьей версии. Попытка создать кроссплатформенную библиотеку визуальных компонентов CLX, работающую и в Windows, и в Linux, также провалилась после того, как Kylix был заброшен.

Несмотря на то, что Delphi практически сошла со сцены как популярный коммерческий продукт, прекрасные идеи, заложенные в эту среду, перекечевали в ее свободный клон Lazarus.

Проект Lazarus (<http://www.lazarus.freepascal.org>) был основан в 1999 году. С технической точки зрения Lazarus – это IDE и обширная библиотека классов для компилятора fpc, которые выглядят и работают как клон Delphi. Lazarus работает в Windows, Linux, \*BSD и Mac OS.

Программы, созданные в Lazarus, без изменений компилируются во всех поддерживаемых системах (если явно не используются функции, зависящие от системы). Lazarus использует библиотеку визуальных компонентов LCL (Lazarus component library – аналог VCL от Borland), которая является универсальной «оберткой» для API различных операционных систем и различных GUI-библиотек (наборов «виджетов»). Программисту достаточно изучить универсальные компоненты LCL, не вникая в особенности системных библиотек. В каждой системе программа автоматически собирается с нужными библиотеками, а все элементы GUI преобразуются в те виджеты, которые доступны для данной системы. В Windows могут использоваться виджеты Win32/Win64 и Win CE, в Linux – GTK 1/2 или Qt, в MacOS – Carbon или Cocoa. Справедливости ради нужно сказать, что поддержка большинства наборов виджетов находится на преальфа-, альфа- и бета-стадиях (стабильными считаются только Win 32 и GTK 1). Имеется также экспериментальный набор виджетов fpGUI, созданный специально для Free Pascal, и независимый набор poGUI (оба – на стадии преальфа).

В этой статье будут использоваться виджеты GTK 2, поддержка которых даже на стадии беты в большинстве случаев не вызывает нареканий.

### Инсталляция

Lazarus присутствует в репозиториях всех популярных Linux-дистрибутивов, однако его версия может оказаться устаревшей или несовместимой с версией fpc из той же системы. Поэтому самый надежный способ установки – сборка из исходных кодов. Для Ubuntu намного проще использовать специальный репозиторий (версия из Ubuntu 9.04 у меня оказалась неработоспособной) – исчерпывающие инструкции о нем приведены в wiki ([http://wiki.lazarus.freepascal.org/Getting\\_Lazarus#Getting\\_Lazarus\\_from\\_our\\_Ubuntu\\_repository](http://wiki.lazarus.freepascal.org/Getting_Lazarus#Getting_Lazarus_from_our_Ubuntu_repository)).

Установленный таким образом Lazarus скомпилирован с набором виджетов GTK 1 и, соответственно, выглядит архаично. Хуже того, данная сборка не поддерживает юникод, и возникают проблемы с кириллицей. Однако это поправимо: Lazarus способен сам откомпилировать собственную IDE, т.к. среда полностью написана на Free Pascal. Это позволяет менять интерфейс в зависимости от предпочтений пользователя.

Чтобы перекомпилировать IDE с GTK 2, нужно сначала установить соответствующую версию библиотек (в Ubuntu это libgtk2.0-dev). Как оказалось, дальнейшие шаги в официальной документации описаны недостаточно подробно, поэтому пришлось обратиться к всезнающему Google.

В Ubuntu 9.04 успешной оказалась такая последовательность действий:

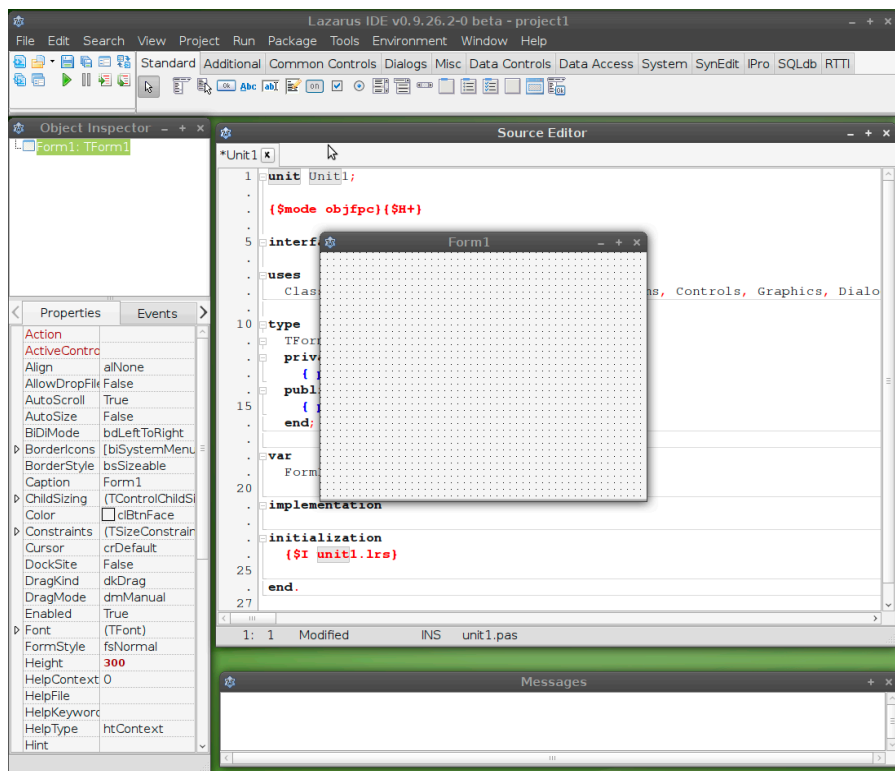


Рисунок 1. Lazarus с интерфейсом GTK 2



- ☑ Компилируем соответствующую библиотеку интерфейса:

```
cd /usr/lib/lazarus/lcl/interfaces/gtk2
sudo make
```

- ☑ Запускаем Lazarus с правами root:

```
sudo startlazarus
```

- ☑ В Tools → Configure → Build Lazarus устанавливаем флажок Build IDE without packages. В списке справа выбираем gtk2. Затем переходим на вкладку Advanced Build Options и также выбираем в списке gtk2.
- ☑ Нажимаем Build.

После компиляции Lazarus автоматически перезапускается уже с интерфейсом GTK 2 и полной поддержкой юникода (см. **рис. 1**).

Осталось исправить ошибку с размещением файла документации, допущенную при создании deb-пакета:

```
sudo ln /usr/share/doc/lazarus/index.html ↵
      /usr/lib/lazarus/docs/index.html
```

Я не устанавливал Lazarus в других дистрибутивах и на других системах, где могут возникнуть свои специфические проблемы. В этом случае сборка из исходных кодов может оказаться самым простым вариантом.

## Использование

При написании программ с нуля Lazarus мало отличим от Delphi 7 – интерфейс клонирован практически идеально. В этой статье нет смысла подробно описывать приемы работы с Lazarus – для этого прекрасно подойдет любая книга или статья по Delphi версий 3-7. Совсем вкратце технологию программирования в Lazarus можно описать так:

- ☑ перетаскиваем на форму нужные компоненты GUI;
- ☑ выстраиваем их нужным образом;
- ☑ задаем нужные свойства в инспекторе свойств;
- ☑ вписываем код в автоматически сгенерированные методы обработки различных событий GUI (например, события нажатия кнопок или выбора элементов списка).

Отличия между Delphi и Lazarus, естественно, есть. Все расширения файлов, которые в Delphi начинались с буквы «d» (от Delphi), в Lazarus начинаются с буквы «l» (от Lazarus). Например .dpr → .lpr, .dfm → .lfm. Все названия, в которые входит «VCL», в Lazarus преобразуются в «LCL». Отличаются директивы компилятора для включения форм и файлов ресурсов. Если в Delphi использовалась директива {\$R \*.DFM} перед разделом implementation модуля, то в Lazarus ее нет, но используется {\$I \*.lrs} в разделе initialization. Все эти тонкости несущественны, если программа разрабатывается в Lazarus с нуля, но проявляются при переносе проектов из Delphi. Состав библиотек компонентов различается более существенно.

В Lazarus отсутствуют многие привычные компоненты Delphi и добавлены многочисленные новые (например, для работы с базами данных MySQL разных версий). Описать эти отличия в подробностях в короткой статье нельзя – их слишком много. К счастью, они не затрагивают базовые визуальные компоненты GUI (кнопки, списки и т.п.), если не считать отсутствия в Lazarus старых компонентов Delphi, унаследованных еще от Windows 3.11. Полный список отличий между Lazarus и Delphi имеется

на официальном сайте ([http://wiki.lazarus.freepascal.org/Lazarus\\_For\\_Delphi\\_Users](http://wiki.lazarus.freepascal.org/Lazarus_For_Delphi_Users)).

## Документация

Самым слабым местом Lazarus (как и многих других больших открытых проектов) на сегодняшний день является документация. Она очень неполная. Зачастую найти описание какого-либо метода или компонента удастся только через Google и то не сразу. Еще хуже обстоит дело с описаниями ошибок – они зачастую вообще не документированы и их приходится искать на форумах и списках рассылки. Нужно отметить также отсутствие контекстной помощи по идентификаторам, вызываемой в Delphi из редактора по <Ctrl> + <F1>, и контекстной помощи по многим компонентам и их свойствам. Впрочем, на официальном сайте проблемы с документацией честно признаются.

Несмотря на сумбур с официальной документацией, wiki-раздел сайта Lazarus (<http://wiki.lazarus.freepascal.org>) очень информативен. Там можно найти ответы на многие вопросы по отдельным областям программирования (базам данных, двумерной и трехмерной графике, сети, Интернету, интерфейсам популярных библиотек и т.п.). Главные страницы wiki даже переведены на русский, но сами статьи, как правило, существуют только на английском языке.

## Миграция с Delphi

Пожалуй, самый актуальный вопрос для поклонников Delphi – это перенос проектов из Delphi в Lazarus. Для этого предусмотрены специальные инструменты, делающие основную «черную» работу по конвертации форм и файлов проекта, однако, как справедливо указано на официальном сайте, они не всегда справляются со своим заданием на 100%. Для теста я взял свой старый проект, написанный на Delphi 5, и попробовал перенести его в Lazarus. Эта программа не использует никаких специфических системных вызовов (кроме запуска браузера при обращении к помощи), поэтому перенос обещал быть достаточно безболезненным.

Итак, попробуем конвертировать весь проект: Tools → Convert Delphi project to Lazarus project... и выбираем .dpr-файл, созданный в Delphi. Первым появляется предупреждение о том, что модуль Windows недоступен (что вполне понятно) и предложение закомментировать все упоминания о нем. Соглашаемся. А вот следующее сообщение уже менее приятно – не найден модуль DirOutln. Этот модуль содержит компоненты TDriveComboBox, TFileListBox и TdirectoryListBox, отображающие дерево каталогов и диски в «старом стиле» Windows 3.11. В Delphi они бывали удобны при создании специализированных сложных диалогов открытия и сохранения файлов. Их придется реализовывать по-другому. Других проблемных модулей конвертер не находит и переходит к анализу форм. Там обнаруживаются второстепенные свойства, которые не поддерживаются в Lazarus (вроде oldCreateOrder). Соглашаемся с тем, чтобы убрать их из файла формы. Дальше конвертер опять натывается на злосчастные TDriveComboBox, TFileListBox и TdirectoryListBox. Убираем и их.

После окончания конвертации открываем формы, чтобы проверить, как прошло преобразование. Убеждаемся, что расположение и форматирование компонентов несколько «съехало», но в целом все прошло успешно (см. **рис. 2**).



Рисунок 2. Искажения форматирования форм после преобразования

Кнопки панели инструментов, использующие списки изображений, отображаются с неправильным фоновым цветом (см. **рис. 3**). Открыв редактор компонента TImageList (который, к слову, работает безобразно медленно), убеждаемся, что проблема в неправильном прозрачном цвете в палитре, поменять который можно, только удалив из списка все изображения и перезагрузив их с диска заново. Не очень приятно, особенно если в списке десятки изображений.

Попробуем скомпилировать наш проект. Сразу же получаем предупреждение, что в коде нет обработчиков некоторых методов. Видимо, они пропали из-за привязки к TDriveComboBox и его «товарищам». Соглашаемся удалить эти вызовы. Заодно комментируем все экземпляры этих классов в коде. Следующим сюрпризом оказываются неопределенные виртуальные коды клавиш (символы VK\_RETURN, VK\_F1 и т.п.). После поиска в документации оказывается, что они определены в модуле LCLType. Добавляем его в uses. Далее оказывается, что конвертер нашел далеко не все места, где упоминается несуществующий модуль Windows, и не убрал из некоторых файлов не нужные в Lazarus директивы компилятора {\$R \*.DFM}.

Наконец компиляция завершается успешно – переходим к борьбе с ошибками времени выполнения. Первой проявляется No widgetset object. Она устраняется включением модуля Interfaces в .lpr-файл. Затем возникает ошибка в файле winapi.inc, который, по идее, вообще не должен компилироваться в Linux. Она исчезает после удаления пути к файлу помощи в стиле Windows, который остался от проекта Delphi в .lpr-файле. Дольше всего длился поиск причины ошибки Unknown property: bidimode. Это не поддерживаемое свойство пришлось вручную искать и удалять из одного из .lfm-файлов. После этого программа наконец-то запустилась, хотя и не была функционирующей из-за «потерянных» при конвертации компонентов. Кроме того, сразу бросилась в глаза медлительность интерфейса по сравнению с оригинальной версией, запускаемой в Delphi. Чем именно это вызвано, сказать сложно, но в принципе понятно, что «обертка» над GTK 2 в виде классов LCL не добавляет производительности. Скорость компиляции в Lazarus также оказалась ниже, чем в Delphi, но это «заслуга» не компилятора fpc, а медленного компоновщика ld.

Таким образом, импорт сколь-нибудь сложных проектов в Lazarus не обходится без пресловутых «танцев с бубном». Формы, содержащие нестандартные компоненты, и модули, использующие WinAPI, придется переделать, но в конце концов затра-

ты времени при этом несоизмеримо меньше, чем при переписывании всего проекта с нуля.

### Проекты, реализованные с помощью Lazarus

Существует официальный список активных проектов, написанных с помощью Lazarus ([http://wiki.lazarus.freepascal.org/Projects\\_using\\_Lazarus#Original\\_contributors\\_and\\_changes](http://wiki.lazarus.freepascal.org/Projects_using_Lazarus#Original_contributors_and_changes)). Среди них много компонентов и библиотек для самой Lazarus. Встречаются и серьезные коммерческие проекты (например, ERP-системы). Вот несколько заслуживающих внимания свободных разработок:

- ☑ **Cactus Jukebox** (<http://cactus.hey-you-freaks.de>) – кроссплатформенный аудиопроигрыватель с возможностью создания базы данных аудиозаписей. Имеет скромные системные требования и легко обрабатывает очень большие коллекции аудиофайлов.
- ☑ **Double Commander** (<http://sourceforge.net/projects/doublecmd>) – интересный двухпанельный файловый менеджер. Похож на Total Commander для Windows и способен загружать его плагины, но при этом полностью кроссплатформенный.
- ☑ **Fit Easily** (<http://fiteasily.com>) – online-сервис для построения моделей экспериментальных данных (фитовки). Полностью написан на Lazarus (включая CGI-модуль).
- ☑ **WinFF** ([http://winff.org/html\\_new](http://winff.org/html_new)) – кроссплатформенный графический интерфейс для конвертера видеофайлов FFMPEG.
- ☑ **OctaveGUI** (<http://code.google.com/p/octave-gui>) и **Xoctave** (<http://xoctave.webs.com>) – интерфейсы для популярного свободного математического пакета Octave.

### Выводы

На мой взгляд, на сегодняшний день Lazarus является самой простой в использовании свободной кроссплатформенной средой визуального программирования. Lazarus унаследовала у Delphi прекрасную концепцию визуального проектирования, которая позволяет даже новичкам писать программы со сложным оконным интерфейсом. Lazarus позволяет перевести проекты, написанные на Delphi, в ранг свободных и независимых от коммерческой среды разработки, а также сделать их кроссплатформенными с относительно небольшими усилиями. Можно по-разному относиться к самой идее подобных сред и к языку Object Pascal, положенному в основу Lazarus, но сам факт наличия инструмента такого класса можно смело назвать большим достижением движения Open Source. Lazarus еще не достиг версии 1.0, и в нем еще много недоработок, но он уже вполне пригоден для практического использования. Остается пожелать проекту успешного развития!

Семен Есипевский  
([yesint4@yahoo.com](mailto:yesint4@yahoo.com))

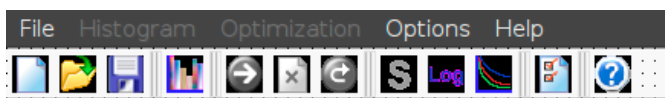


Рисунок 3. Неправильный фон кнопок на панели инструментов

## Путевка в жизнь для Open Source-проекта. Часть 1

### Предварение

Создание и «ведение» проекта с открытым кодом: вот вы задумали такое дело, но с чего начать? Первым делом объявите о своем решении друзьям и знакомым, живьем, в блоге или на каком-нибудь форуме. Сразу найдутся люди, которые посоветуют взять наперевес рашпиль и броситься «допиливать» некую уже существующую программу, подобную задуманной вами.

У советчиков в запасе целый список таких программ и заветное слово «лучше».

Значение разработчиков, которые шлют патчи или вливаются в уже существующие коллективы, конечно же, велико. Однако ни одна программа не возникает сама по себе. Есть, правда, коллек-

тивы, которые ничего не делают, но годами собираются писать какой-нибудь сверхпроект. Распределяют должности, рисуют всякие блок-схемы, создают утопии в своем воображении. У таких людей всё время уходит на болтовню вместо написания кода. Да, сложную программу не создать без тщательного продумывания архитектуры, но проект начинает жить только во вре-



мя его воплощения в коде, хотя бы в базовых классах. И чтобы всё это компилировалось. Есть два подхода разработки: сольный и групповой. В каждом свои плюсы и минусы. В сольном приходится самому поддерживать сайт, вывешивать сообщения о новой версии на разных ресурсах и тому подобное. Именно такая возня, лишенная творчества, утомляет и порой заставляет откладывать выпуск новой версии. С другой стороны – вы сами себе голова, развиваете проект в том направлении, которое считаете нужным. В групповом проекте надо считаться с мнением других. Иногда спорить. Уметь работать со средствами вроде CVS или Git. Однако плюсы в том, что обычно кто-то берет на себя связи с общественностью, кто-то рисует кнопки, кто-то полирует интерфейс и пишет документацию. Совместная работа над кодом – еще один плюс. Не можете что-то написать или исправить вы – сможет кто-то другой.

## У истоков

Семь раз отмерь – один отрежь. Прежде чем ваша алгоритмическая мысль станет воплощаться в строках кода, надо проявить себя большим стратегом. Сейчас мало кто пишет программы с нуля – обычно их основывают на какой-то популярной библиотеке виджетов. Далее буду называть это понятие ёмким и уродливым словом «тулkit», ибо в русском языке ему соответствует разве что длинное «набор инструментов». Итак, тулkit обычно включает в себя не только графические элементы управления (виджеты), но и огромную «невидимую» прослойку различных полезных функций и классов – контейнеры, списки и тому подобное. Если вы пишете программу для себя, не имеет значения ни язык программирования, ни тулkit. Берите seed7 – отличный язык – и пишите на здоровье. Но если вы хотите, чтобы вашей программой пользовались другие, то надо выбрать два краеугольных камня: тулkit и лицензию. Да, даже не язык программирования, а именно тулkit. Чем менее популярен тулkit – тем меньше будет пользователей у вашей программы. В некоторых легковесных Linux-дистрибутивах нет даже Qt. Я не агитирую за GTK – сам сейчас перешел на Qt, но GTK все же более распространена, чем Qt. Выбирая тулkit, задумайтесь кроме прочего над следующими вопросами:

- ☒ Работает ли тулkit на различных системах (\*nix, Windows, Symbian), если вам нужна многоплатформенность, и сколько легко достигается последнее?
- ☒ Какими средствами можно локализовать программы с тулkitом?
- ☒ Какая версия тулkита должна быть в «минимальных требованиях к системе»?

Выбор тулkита и, подчеркиваю, лицензии автоматически открывает вам доступ к исходному коду программ под этой же лицензией для этого тулkита. Допустим, в программах для Qt редко заимствуют код из GTK-проектов, и наоборот. Однако какие-то алгоритмы без привязки к тулkitу (например, мультимедийные) запросто переходят из программы в программу под одной и той же лицензией. Начальный этап рождения программы напоминает мне создание персонажа в ролевой игре – надо выбрать класс и тому подобные вещи, которые в будущем будут оказывать громадное влияние на развитие героя.

Наиболее «портируемыми» (подразумеваю удобство, а не потенциальную возможность) тулkitами мне представляются Qt и wxWidgets. GTK больше склонен к UNIX-подобным системам. (Примечание: кстати, Nokia сейчас вновь начинает продвигать Maemo, отходя от платформы S60, а GTK-приложения хорошо живут в среде Maemo, ибо у Maemo сейчас интерфейс завязан именно на GTK+. Впрочем, будущее Maemo – всё-таки в Qt

(<http://qt4.garage.maemo.org>) – не зря же Nokia купила Trolltech. Однако поддержку GTK+ никто не отменит.) FLTK удобно использовать, если ваша программа завязана на OpenGL. Fox Toolkit живет своей жизнью и, кажется, через какое-то время выйдет на новый уровень распространения, встав рядом с Qt.

Чаще всего получается, что, выбирая тулkit, лицензию в любом случае надо выбрать – иногда это диктуется разработчиками тулkита, а иногда приходится, как я уже писал, для доступа к исходникам других программ. К слову, я вообще против лицензий, поскольку всякая лицензия подразумевает в пользователе злоумышленника, ожидая, что он хочет работать с вашей программой «неправедным» путем. Вот поглядим даже на GPL. С одной стороны, конечно, справедливый подход: взял чужой код – так поделись же своим кодом с другими. Но почему это должно быть предписано лицензией, а не здравым смыслом? Есть исходники со статусом «общественное достояние» (public domain) – обычно такой код имеет корни в научной среде. Создавая программу с нуля, можно и свой код открыть таким образом. А вот если вы используете части других программ под такими-то лицензиями, то придется этим лицензиям угождать.

Действительность такова, что при нынешнем положении вещей вам скорее всего придется выбрать для своей программы некую лицензию. Но какую? Соображения о том, что тулkit диктует лицензию, оставим сейчас в стороне – ведь если тулkit диктует, надобно подчиниться и следующие ниже рассуждения вам ни к чему. Будет ли ваша программа относиться к Open Source или Free Software? В чем разница? Free Software можно считать подкатегорией Open Source, поскольку это программы с открытым кодом, которые, помимо прочего, попадают под определение «свободного», с точки зрения Ричарда Столлмана и проекта GNU. Наиболее популярны сейчас лицензии GNU GPL, MPL (Mozilla Public License), MIT License и BSD-подобные лицензии, а также zlib/libpng License (она, кстати, не требует открытости кода). Смысл GPL можно выразить так: взял в программу код из GPL-программы – лицензирую свою программу тоже под GPL. GPL обязывает открывать код: око за око, зуб за зуб. MPL чаще сравнивают с LGPL, поскольку обе эти лицензии не так строги, как «чистая» GPL. MPL-части программы можно спокойно комбинировать с не-MPL – при условии, что MPL-код остается под лицензией MPL. GPL же обязывает, чтобы весь код программы был под GPL, если хотя бы какая-то часть этого кода лицензирована под GPL. LGPL – грубо говоря, вы можете использовать (методом динамического связывания) LGPL-библиотеки в своей программе, не выкладывая саму программу под LGPL. Конечно, если вы берете код из такой библиотеки и вставляете в свой проект, то начинает действовать «инфекционная» природа лицензии и ваш код уже придется лицензировать под LGPL или GPL (LGPL-код можно перевести под лицензию GPL).

В общем, для некоммерческих программ я советую GPL v3. Во-первых, чтобы не морочить себе голову, а во-вторых, у нее красивый баннер. Черновик перевода GPL v3 на русский язык доступен по адресу <http://code.google.com/p/gpl3rus>. Об отличиях GPL второй и третьей версий толково написано на <http://www.permlug.org/wiki/quickguide/gplv3>. К теме выбора тулkита и лицензии примыкает выбор использования сторонних библиотек. Вам придется позаботиться о лицензионной совместимости – это раз. Во-вторых, необходим очень взвешенный подход к зависимости от библиотек, особенно если они не входят в популярные дистрибутивы и сложны в сборке из исходников. Если какие-то задачи решаются на уровне тулkита или вы можете решить их сами – не прибегайте к помощи сторонних библиотек, какими бы заманчивыми они вам ни казались. Если библиотека невелика – включите её в дистрибутив своей программы.

## Средства разработки

Можно кричать, что ничего не нужно и достаточно текстового редактора да командной строки для вызова компилятора и отладчика. С одной стороны, это справедливое утверждение – тем более, что некоторые редакторы нынче мало отличаются от простых IDE (интегрированных сред разработки). Однако IDE предоставляют приятные мелочи, от которых трудно отказаться – это и удобные переходы к месту, где находится определение функции, и подсказки к параметрам тех же функций, и многое-многое другое. И везде люди спрашивают: «Какую IDE посоветуете?». Какой смысл спрашивать? Надо брать и пробовать. Вообще можно сразу понять, под что «заточена» IDE, по её «тулkitной» природе. Например, Qt Creator заточен для Qt и включает в себя не только редактор и интерфейс к отладчику, но и удобнейшее средство просмотра документации. У Qt Creator великолепный, смелый интерфейс – смелый потому, что разработчики отказались от диалоговых окон, столь тормозящих работу и являющихся пережитком допотопных времен. KDevelop третьей версии был универсальной средой разработки без особой привязки к чему-либо. Я его использовал для autotools-проектов на GTK+. Однако KDevelop 4 в текущем состоянии – это среда со строгой заточкой под C++ и KDE. Очень много внимания уделено автоматическому дополнению и тому подобным вещам – кажется, больше уже и придумать нечего в этом плане. Между тем такую простую и полезную вещь, как закладки на уровне проекта, почему-то еще не сделали, а в третьей версии она есть. Да много чего еще там было и что покамест не переключалось в четвертую... Если бы я выбирал между 3-й и 4-й, то сейчас выбрал бы третью, но в итоге остановился на QtCreator.

Anjuta – классическая IDE для программ, написанных под GTK+ и GNOME. Не столь тяжеловесна, как KDevelop, и предоставляет умеренный, взвешенный набор функций. Кроме прочего, оснащена встроенным Devhelp. Многие близкие к GTK+ библиотеки поставляют документацию в формате Devhelp, а сам Devhelp – это просмотрщик гипертекстовой документации. Поддерживает плагины. Обратите внимание, что в модуль anjuta-extras входят редактор на основе движка Scintilla, а также под-

держка Valgrind и gprof. Ну и, конечно, истинные любители консоли будут использовать в качестве IDE редактор vim. Вообще не так важно, что именно, а главное – чтобы было удобно и вдохновляюще. Кстати, связка редактор + консоль зачастую очень удобна...

## Где поселить проект?

Можно пойти простым путем и наспех сделать официальный сайт на каком-нибудь бесплатном хостинге вроде [narod.ru](http://narod.ru), а можно купить хостинг, доменное имя и тоже радоваться. Но существуют проекты бесплатного хостинга, ориентированные на открытое и свободное ПО. Речь идет о [savannah.gnu.org](http://savannah.gnu.org) (только для СПО), [sourceforge.net](http://sourceforge.net), [fireforge.ru](http://fireforge.ru), [www.berlios.de](http://www.berlios.de). Вообще сравнение подобных хостеров смотрите на [http://en.wikipedia.org/wiki/Comparison\\_of\\_open\\_source\\_software\\_hosting\\_facilities](http://en.wikipedia.org/wiki/Comparison_of_open_source_software_hosting_facilities).

Лично мой опыт (как человека ленивого и пользующегося тем, что на слуху и под рукой) сводится к многолетнему общению с SourceForge (далее SF) – причем задействую я только малую часть предоставляемых им функций. SF поддерживает колоссальный набор сервисов: и хостинг сайта, и размещение файлов вашей программы без сайта (собственно, это два разных сервиса – хранилище файлов и сайт), CVS/Subversion/Git/Mercurial, MediaWiki, трекер ошибок, форум, список рассылки и многое другое. Загрузка файлов программы на SF идет через веб-интерфейс (хотя можно иначе), а файлов сайта – по ssh (для работы с SF у вас должны быть ключи ssh, которые можно создать командой `ssh-keygen`). В большинстве случаев последнее выглядит как запуск `sftp`, заход в удаленный каталог и загрузка обычными FTP-командами вроде `put`.

Во второй части статьи (см. следующий выпуск «Open Source») будут рассмотрены такие аспекты, как общение с пользователями, работа над ошибками, перевод интерфейса и документации, распространение программы.

Петр Семилетов  
([tea@list.ru](mailto:tea@list.ru))

## «Open Source» приглашает к сотрудничеству!

Электронное приложение «Open Source» всегда открыто для сотрудничества с новыми авторами, с читателями и их конструктивными предложениями по улучшению издания, обоснованной критикой и любыми отзывами, с компаниями, занимающимися разработкой и продвижением программного обеспечения с открытым кодом. Приветствуются все энтузиасты, желающие опубликовать у нас свои статьи. Тематика нужных материалов очевидна из предназначения приложения, то есть FOSS (Free and

Open Source Software): теория и практическое применение; исторические сведения, анализ сегодняшнего положения, прогнозы на будущее и другие аспекты, связанные с открытым ПО.

Среди наиболее интересных на данный момент общих тем можно выделить:

- ☑ общие обзоры новых и/или интересных проектов Open Source и конкретных приложений, свежих версий дистрибутивов Linux, \*BSD и других систем;
- ☑ советы и рекомендации новичкам в GNU;

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

Желательный объем статей: 6000 или 12000 символов (с пробелами). Примеры актуальных сейчас тем для статей публикуются на <http://osa.samag.ru/todo>. Но не стоит строго ограничиваться приведенными выше рамками!

Публичное обсуждение «Open Source» проводится на форуме сайта журнала «Системный администратор» по адресу: <http://osa.samag.ru/forum>. Связаться с редакцией можно по электронной почте [osa@samag.ru](mailto:osa@samag.ru).

P.S. За статьи мы платим.

## Подписные индексы:

**20780\***

+ диск с архивом статей  
2008 года

**81655\*\***

без диска

по каталогу агентства  
«Роспечать»

**88099\***

+ диск с архивом статей  
2008 года

**87836\*\***

без диска

по каталогу агентства  
«Пресса России»

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

# Подписка на журнал «Системный администратор»

## Российская Федерация

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

## СНГ

В странах СНГ подписка принимается  
в почтовых отделениях по националь-  
ным каталогам или по списку номенкла-  
туры «АРЗИ»:

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

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