

Колонка главного редактора



Все статьи, представленные в этом выпуске «Open Source», являются составляющими одного из циклов: обзор оконного менеджера Fluxbox, удобная работа в Linux-консоли, долгожданное сравнение GTK+ и Qt, интервью от SourceKibitzer. И несмотря на это, нам удалось сделать так, чтобы каждый материал был достаточно независимым, обладал своей «личной» ценностью – проще говоря, чтобы он с интересом читался и вне «сопутствующих» ему статей. Надеюсь, это не только мое субъективное восприятие. Возможно, это тоже субъективно, но в последнее время у меня все значительно растет скептицизм по отношению к некоторым компаниям, заигрывающим с Open Source, а иногда и на полном серьезе «продвигающим» его. Они создают приятную для невнимательного глаза видимость открытия исходного кода и дружелюбного отношения ко всему миру и обществу энтузиастов, глубокого понимания важности «свободы» по Столлману... Но в действительности лишь пользуются модным в индустрии направлением для достижения своих «корпоративных» целей. Я понимаю, что в целом в бизнесе такое поведение нормально. Но ведь должны быть какие-то рамки. Подобные тем, какими люди руководствуются в обществе для идентификации лицемерных личностей. Только вот где они?

Главный редактор
Дмитрий Шурупов
(osa@samag.ru)

«Open Source»

электронное приложение к журналу
«Системный администратор»
№24, 18 апреля 2008 г.

РЕДАКЦИЯ

Исполнительный директор

Владимир Положевец

Главный редактор

Дмитрий Шурупов

Верстка и оформление

Владимир Лукин

Сайт электронного приложения:

<http://osa.samag.ru>

За содержание статьи ответственность несет автор. Все права на опубликованные материалы защищены.

Новости мира Open Source

Веб-браузер GNOME Epirhany переходит на WebKit

Команда разработчиков веб-браузера Epirhany, используемого по умолчанию в графической среде GNOME, объявила о переходе на движок WebKit.

Веб-браузер Epirhany, ставший наследником Galeon, традиционно для рендеринга веб-страниц использовал популярный движок Gecko, являющийся трудом разработчиков из проекта Mozilla. Однако теперь разработчики Epirhany решили перейти на WebKit – Open Source-фреймворк, берущий свое начало из движкаKHTML веб-браузера Konqueror (стандартного для KDE) и в дальнейшем проработанный усилиями компании Apple.

По словам разработчиков Epirhany, зависимость браузера от Gecko породила множество неудобств. Среди плюсов WebKit они обозначили ряд существующих под него API, прямое использование им технологий GNOME (Cairo для графики и Pango для рендеринга, а также libsoup – для сети, GStreamer – для тегов аудио и видео в HTML5), синхронизированный с графиком GNOME 6-месячный цикл релизов, большой потенциал WebKit (что будет способствовать привлечению разработчиков к Epirhany), удобство использования Epirhany для других нужд GNOME (Yelp, Devhelp, редактор в Evolution вместо GtkHTML).

9000 ПК в школах Швейцарии полностью переводят на Linux

В швейцарских школах стартовала инициатива «Да здравствует свободное программное обеспечение!» (Long Live Free Software), в рамках которой 9 тысяч персональных компьютеров будут полностью переведены на Linux.

Ранее на всех этих машинах были установлены две операционные системы: Windows и Ubuntu Linux. Теперь, следуя рекомендации специализированного министерства (Department of Public Instruction), на школьных компьютерах остаются только Linux. Данный сюжет получил огласку в местной газете «Трибуна Женевы» (Tribune de Geneve).

Инициатива будет выполнена к сентябрю текущего года – с этого момента на всех 9000 компьютерах в школах будет установлен лишь дистрибутив Ubuntu Linux со свободным и открытым программным обеспечением. Представители власти рады достигаемой благода-

ря миграции на Linux экономии, а в министерстве считают, что таким образом будут обеспечены лучшие условия для обучения.

Выборы в Бразилии состоятся при участии Linux-устройств

Бразильский Верховный суд (Tribunal Superior Eleitoral, TSE), ответственный за выборы в стране, которые состоятся в 2008 году, объявил о переводе устройств для электронного голосования под управление операционной системой GNU/Linux.

Для процедуры электронного голосования в этом году в Бразилии будут использоваться 430 тысяч специализированных автоматических устройств. Ранее на них были установлены операционные системы VirtuOS и Windows CE, а теперь согласно официальному объявлению TSE от 4 апреля функционируют приложения с открытым кодом (Open Source), запущенные в среде ОС GNU/Linux.

Сообщается, что все установленное на устройства Open Source-ПО и все внутренние разработки будут оснащены цифровыми подписями.

Вышел Linux-дистрибутив Mandriva 2008 Spring

Французская компания Mandriva объявила о выпуске новой версии своего дистрибутива – Mandriva Linux 2008 Spring.

Среди ключевых новшеств отмечают: новая утилита родительского контроля, мультимедийный центр Elisa (единный интерфейс для работы с фотографиями, музыкой и видео), Codeina для автоматической установки кодеков, аудиофреймворк PulseAudio, медиасервер Coherence, поддержка нового субноутбука Asus Eee PC «из коробки».

Среди пакетов программного обеспечения в Mandriva 2008.1: Linux-ядро 2.6.24, GNOME 2.22, KDE 3.5.9 (в репозиториях доступен и релиз 4.0.3), Xfce 4.4.2, Compiz Fusion 0.7.2, OpenOffice.org 2.4.0, Mozilla Firefox 2.0 и Thunderbird 2.0.

Бесплатная версия дистрибутива, Mandriva Linux One 2008 Spring, доступна для свободного скачивания на <http://mandriva.ru/downloads>.

IDC: в 2011 году на Linux будут тратить 49 млрд USD

Исследователи из компании IDC предсказывают рост совокупных мировых затрат на Linux и связанное с открытой операци-

онной системой программное обеспечение с 21 млрд USD в 2007 году до 49 млрд USD в 2011 году.

Как отмечают аналитики IDC, в последнее время становится все более заметной тенденция расширения использования Linux-серверов в область ориентированных на коммерцию баз данных и систем управления предприятием. Эта сфера «исторически была территорией Microsoft Windows и UNIX».

Подсчеты IDC показывают, что на сегодняшний день ПО на платформе Linux приносит в общей сложности 10 млрд USD в год, что составляет около 4 процентов от всего рынка (242 млрд USD). Прогноз аналитиков таков: «Эта доля превысит 9 процентов к 2011 году, когда доходы от относящегося к Linux программного обеспечения составят 31 млрд USD из общих 330 млрд USD».

Мобильная платформа Nokia Maemo будет поддерживать Qt

Проект программной платформы для мобильных устройств Maemo объявил о том, что у него появится поддержка Qt.

Разработкой платформы Maemo (Internet Tablet OS) занимается Nokia, которая ранее в этом году сообщила о своем намерении приобрести норвежскую компанию Trolltech, подарившую миру Qt. Ранее в проекте использовался только набор утилит Hildon, основанный на

GTK+. Развитие Hildon не останавливается – этот фреймворк останется базовым для интерфейса приложений Internet Tablet OS.

Сообщается, что в ближайшее время Nokia начнет набирать штат программистов на Qt/C++, однако ожидать Qt-приложений от компании пока не приходится: первым реальным шагом по внедрению Qt в Maemo станет распространение библиотек Qt для разработки приложений для Maemo в течение четвертого квартала 2008 года.

В Аргентине подумывают о полном переходе на Open Source

Аргентина может стать первой в мире страной, все правительственные учреждения которой будут использовать только программное обеспечение с открытым кодом (Open Source). Об этом стало известно благодаря законопроекту, недавно принятому на рассмотрение в Национальном конгрессе Аргентины.

Инициатива полного перехода правительственных ведомств Аргентины на Open Source-ПО призвана стать частью широко развернувшейся кампании, направленной против пиратства в сфере программного обеспечения.

Как отметил Марио Алборноз (Mario Albornoz), директор Института социальных исследований науки и технологий (Institute of Social Studies of Science and Technology),

переход на Open Source обеспечит работой местных программистов и компаний-разработчиков ПО, однако также вызовет некоторые проблемы, связанные с необходимостью в установке и поддержке систем с открытым кодом.

Статистика использования Open Source будет подсчитана

Запущена новая инициатива Open Source Census, которая задается целью сбора статистики об использовании программного обеспечения с открытым кодом.

Создание Open Source Census – заслуга компании OpenLogic, занимающейся распространением утилит и сервисов по управлению Open Source-ПО. Анонс проекта состоялся еще в декабре.

OpenLogic представила (под открытой лицензией) автоматизированную утилиту оценки OSS Discovery. Приложение позволяет компаниям и частным лицам просканировать свои компьютеры на наличие на них программного обеспечения с открытым кодом и анонимно выслать данные на сервер сбора статистики. Затем желающие смогут узнать данные по использованию Open Source на своих компьютерах, а для всех будет доступна общая статистика.

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

Fluxbox: раптор среди динозавров X Window. Часть 2: Эволюция

Предисловие

В первой части статьи (см. «Open Source» 023) был представлен обзор оконного менеджера для X Window – Fluxbox. Пришло время установить его и настроить.

Для большинства дистрибутивов справедливо, что Fluxbox представлен в категории оконных менеджеров (wm) пакетов ПО, а установка не должна занимать больше двух-трех минут. На всякий случай напомню, что ссылки на архив с исходниками и пакеты для различных систем доступны на сайте проекта (<http://www.fluxbox.org/download.php>).

Ни для кого не секрет, что большинство современных оконных менеджеров обладает широким спектром конфигу-

рационных параметров, но, к сожалению, не всегда можно тонко подстроить под себя некоторые мелочи в оформлении. Fluxbox в этом плане совершеннее многих.

Интерфейс

Как и другие минималистские графические оболочки, Fluxbox состоит из двух основных элементов: меню и панели инструментов. Каждое запущенное окно также обрамляется элементами оформления Fluxbox и находится полностью под его контролем.

Меню может состоять из элементов, каждый из которых представляет собой ссылку на программу и её название, под-

меню, разделителей и служебных команд Fluxbox, таких как «exit» (выход), «workspaces» (рабочие места) и т.д.

Панель инструментов состоит из основной рабочей области для программ, места для даты/времени в формате strftime и переключателя рабочих мест.

Теперь рассмотрим поподробнее каждую из составляющих.

Меню

Конфигурация меню начинается с редактирования файла `~/fluxbox/menu`. Тег «begin» (начало) в квадратных скобках, за которым следует заголовок в круглых скобках определяет начало меню:

```
[begin] (Laptop Qtrl)
```

Все заголовки для элементов указываются в круглых скобках, определяющие элементы – в квадратных, названия программ или пути к ним – в фигурных, а графические значки – в угловых. Заканчивается же каждое из вложенных подменю тем «end» (конец), как и само меню:

```
[begin] (Laptop Qt1)
  [exec] (program 1) {/path/to/the/program1} ↵
    </path/to/the/icon/for/program1.png>
  [submenu] (Development)
    (program 2) {program2}
    [separator]
    [exit] (Quit)
  [end]
  [submenu] (Graphics)
    [exec] (The GIMP) {gimp}
  [end]
[end]
```

Как видно из примера, путь для графической пиктограммы является необязательным параметром, а полный путь к программе может быть заменён на название исполняемого файла, если он находится в одном из каталогов, описанных в глобальной переменной пользовательского окружения PATH.

Вложенность подменю практически безгранична, поэтому для удобства установленные программы можно разбить на множество категорий, которые станут названиями для подменю Fluxbox.

Для поддержки символов в Unicode необходимо заключить всё меню в следующие теги:

```
[encoding] {UTF-8}
..menu..
[/endencoding]
```

Некоторые специальные элементы меню:

- [workspaces]** – список рабочих мест;
- [restart]** – перезапуск Fluxbox;
- [reconfig]** – обновление конфигурационных файлов и их переименование к запущенному менеджеру;
- [stylesdir]** – список стилей;
- [include]** – подключение дополнительного файла или каталога;
- [exit]** – выход из Fluxbox.

Строки с комментариями во всех конфигурационных файлах Fluxbox начинаются с символа «#», за которым следует текст.

Панель инструментов

У Fluxbox классическая панель инструментов. Для её настройки в любой свободной части достаточно щёлкнуть правой кнопкой мыши, и появится контекстное меню, через которое можно изменять её конфигурацию: размеры панели и расположение на экране, уровень прозрачности, формат представления времени. Последний описан в руководстве к функции strftime() («man strftime») – например, «%d %b %Y / %T» приведёт к виду «17 Apr 2008 / 18:36:12».

Между тремя элементами панели инструментов находятся кнопки-переключатели. Левая пара отвечает за переключение между рабочими местами, а правая – за переключение между запущенными графическими приложениями, которые представляют собой окна.

Здесь же, в правой части расположен системный трей, куда сворачиваются поддерживающие его приложения.

Дополнительные возможности

Несмотря на простоту внешнего вида, Fluxbox обладает большими возможностями в тонкой подстройке «под себя», и его скрытый потенциал как довольно простого оконного менеджера весьма велик.

Свойства элементов оформления

Fluxbox управляет всеми графическими элементами и способен запоминать их параметры: размеры, положение и т.д. Все

сохранённые данные можно найти в файле ~/.fluxbox/apps. Соответственно, чтобы изменить местоположение (координаты) окна программы при его появлении на экране, достаточно отредактировать этот файл.

Глобальные настройки Fluxbox хранятся в файле ~/.fluxbox/init. Там можно найти всё: начиная от путей к темам и заканчивая размерами каждого элемента. Этот файл создаётся автоматически при первом запуске оконного менеджера – после этого можно приступить к его редактированию. Стоит отметить, что часть присутствующих здесь параметров регулируется с помощью элемента меню «[config]», а часть – доступна только в этом файле.

Полный список характеристик можно узнать в официальном руководстве («man fluxbox»).

«Горячие» клавиши

Что сделает любой оконный менеджер максимально удобным? Конечно же, комбинации «горячих» клавиш (так называемых «hotkeys»). Их значения хранятся в файле ~/.fluxbox/keys.

Каждая строка состоит из нескольких частей:

```
<модификатор> <клавиша> :команда операция
```

Последняя часть – «операция» – не является обязательной, о чем см. ниже.

Список модификаторов можно просмотреть командой «xmodmap -pm», а названия клавиш (помимо стандартных) – в файле /usr/share/X11/XKeysymDB.

В качестве третьей части используются специальные команды Fluxbox, такие как «PrevWorkspace» (переключение на предыдущее рабочее место) и др.

Например, для переключения между окнами с помощью клавиш <Alt> + <Tab>, <Alt> + <Shift> + <Tab> (в обратную сторону) нужно внести следующие изменения:

```
Mod1 Tab :NextWindow
Mod1 Shift Tab :PrevWindow
```

«Mod1» соответствует левой клавише <Alt>.

Особый интерес вызывает команда «ExecCommand», которая используется для выполнения произвольных операций. Например, если вы используете ALSA, можно настроить специальные клавиши для управления звуком (я использовал комбинации, характерные для моего ноутбука и «подогнанные» с помощью xmodmap):

```
None XF86AudioMute :ExecCommand amixer -q sset "Master" toggle
None XF86AudioLowerVolume :ExecCommand amixer -q sset "PCM" 5-
None XF86AudioRaiseVolume :ExecCommand amixer -q sset "PCM" 5+
```

для включения/выключения, понижения и повышения уровня звука соответственно.

Кнопки мыши настраиваются практически аналогично:

```
OnDesktop Mouse1 :hideMenu
OnDesktop Mouse2 :workspaceMenu
OnDesktop Mouse3 :rootMenu
```

Скрипт инициализации

Иногда перед запуском оконного менеджера требуется выполнить предварительные команды. Установить задний фон, запустить системный монитор и т.п. В Fluxbox это делается посредством файла-скрипта ~/.fluxbox/startup.

У Fluxbox нет встроенных средств для установки заднего фона, поэтому можно воспользоваться просмотрщиком изоб-

Электронное приложение «Open Source»

ражений FEN и прописать путь к картинке в скрипте следующим образом:

```
fbsetbg -f /usr/share/wallpapers/true_wallpaper.png
```

fbsetbg – это команда из дистрибутива самого Fluxbox, представляющая собой скрипт по нахождению программ, способных установить фоновое изображение. При наличии программы FEN будет использована именно она.

Для запуска приложений нужно вбить их список в скрипт, заканчивая каждую строку символом амперсанда («&»), что означает запуск процесса в фоновом режиме:

```
conky &  
xcompngr &
```

После этого нужно обязательно добавить сам Fluxbox:

```
fluxbox -log ~/.xsession-errors
```

Автоматическая идентификация по SSH

Иногда требуется автоматическая идентификация SSH, чтобы не вводить каждый раз секретную фразу (например, при инициализации соединения с удалённым сервером). Чтобы автоматизировать этот процесс, достаточно добавить несколько строк перед запуском Fluxbox и после него – например, так:

```
id1=~/.ssh/identity  
id2=~/.ssh/id_dsa  
id3=~/.ssh/id_rsa  
  
if [ -x /usr/bin/ssh-agent ] && [ -f $id1 -o -f $id2 ]  
-o -f $id3 ]; then  
    eval `ssh-agent -s`  
    ssh-add < /dev/null  
fi  
  
fluxbox -log ~/.xsession-errors  
  
if [ "$SSH_AGENT_PID" ]; then  
    ssh-add -D < /dev/null  
    eval `ssh-agent -s -k`  
fi
```

Таким образом, если в системе присутствует графическая интерпретация программы «ssh-askpass», при входе в систему потребуется лишь раз ввести ключевую фразу, и подключение к удалённому серверу с использованием публичных ключей превратится в сказку.

Оформление

Теперь о самой интересной части настройки – графической. Для Fluxbox существует множество тем, но если у вас есть небольшой опыт работы с графикой, создать свою не составит большого труда. В состав дистрибутива уже входит пара десятков тем, а если этого недостаточно – рекомендую обратиться к следующим ресурсам:

- ☑ <http://www.fluxbox.org/themes.php> – список тем на официальном сайте;
- ☑ <http://themes.freshmeat.net/browse/962> – большая коллекция тем на портале Freshmeat.net;
- ☑ <http://www.opendesktop.org/?xcontentmode=7400> – постоянно обновляющиеся темы на openDesktop.

В этом разделе мы рассмотрим лишь базовую настройку, т.к. параметров для отдельной темы довольно много, и трудно найти лучшее подробное описание каждого из них, чем в официальной документации.

Темы, которые входят в состав дистрибутива Fluxbox, обычно устанавливаются в каталог /usr/share/fluxbox/styles. Каждая из них представляет собой один файл (либо каталог, если в теме много графических элементов). Такие графические элементы находятся в подкаталоге pixmap и имеют обычно формат XPM или PNG. В файле конфигурации темы (theme.cfg) описывается внешний вид.

Как можно было заметить, во Fluxbox все параметры очень похожи на объекты. В случае со стилями – аналогично. Например, для установки основного шрифта меню требуется следующее:

```
menu.frame.font: -artwiz-snap-normal-r-*-10-*-*-*-iso8859-1
```

Для изменения шрифта заголовков меню:

```
menu.title.font: -artwiz-kates-medium-r-*-14-*-*-*-iso8859-1
```

Параметров довольно много, поэтому приведу лишь некоторые:

- ☑ **menu.borderColor** – цвет границы элементов меню;
- ☑ **menu.frame** – эффекты для элемента меню (например, Raised Bevel1 Gradient Vertical);
- ☑ **toolbar.clock.font** – шрифт для отображения часов на панели инструментов;
- ☑ **toolbar.height** – высота панели инструментов в пикселях;
- ☑ **window.justify** – выравнивание заголовка окна;
- ☑ **window.title.focus pixmap** – картинка для отображения заголовка окна при наведении на него указателя мыши.

Полный список доступен по команде «man fluxstyle».

Стоит сразу заметить, что для отображения символов в Unicode в свойствах шрифта следует указать на кодировку, поэтому последние 2 значения должны выглядеть так:

```
-iso10646-1
```

А для window.font, скорее всего, потребуется следующее:

```
-*-*-*-*-*-*-*-*-*-*-u
```

Не стоит забывать, что системная локаль должна поддерживать UTF-8. Это устанавливается, например, так:

```
export LANG="ru_RU.UTF-8"
```

Зависит от предпочтений и наличия поддерживаемых кодировок в библиотеке glibc (не путать с glib!) вашего дистрибутива.

Разумеется, сам шрифт должен поддерживать символы Unicode. Наиболее известными в этом плане остаются Terminus и DejaVu.

Заключение

На этом обзор Fluxbox из двух статей считаю завершённым. Надеюсь, информация оказалась полезной как тем, кто давно знаком с Fluxbox, так и тем, кто о нём услышал впервые.

Остаётся лишь пожелать, чтобы количество его пользователей увеличивалось с каждым днём, а IRC-канал #fluxbox на Freenode.net пополнялся новыми интересными идеями.

Влад Глаголев
(stealth@sourcemage.org)

Жизнь в консольном Linux-десктопе.

Часть 2: Мультимедиа

Наличие разнообразных приложений для работы в командной строке UNIX-подобных систем ни для кого не новость. Очевидно, что основная их часть не ориентирована на решение задач десктоп-пользователя. Но ведь это еще не означает, что таких нет вовсе...

В продолжении к начатой в прошлом выпуске «Open Source» (#023) статье собрана информация о мультимедийных программах, обладающих (насколько это возможно) простым в освоении и удобным в использовании консольным интерфейсом. Поскольку основное внимание в материале уделено именно таким программным проектам, сразу напомним о том, что подборка носит субъективный характер, а я, как автор, не претендую на то, что обозначил лучшие (или самые популярные, или самые функциональные и т.д.) продукты. Включенные в обзор программы просто соответствуют моим представлениям о том, с какими консольными приложениями сможет без особых неудобств работать более-менее «рядовой» пользователь десктоп-системы».

Создание скриншотов: fbgrab

Одной из наиболее частых проблем, возникающих у пользователей UNIX-консоли и относящихся к мультимедийной области, является создание снимка экрана. В качестве однозначного решения рекомендуется утилита fbgrab [1]. Несмотря на то что она не обновлялась с 2004 года, программа отлично справляется с теми задачами, для которых она создавалась.

Среди возможностей стоит выделить такие необязательные параметры, как размеры и разрядность изображения (ключи «-w» и «-h» для ширины и высоты, «-b» — для глубины цвета), задержка перед снятием скриншота (задается в секундах, флаг «-s»). Пример создания скриншота:

```
$ fbgrab -w 800 -h 600 -b 16 screen.png
```

Результатом выполнения такой команды станет 16-битное изображение размера 800x600 в формате PNG (screen.png) с экраном консоли во время запуска fbgrab. Если запустить fbgrab в фоновом режиме, добавив «&» в конце строки, и добавить аргумент «-s» с произвольным це-

лочисленным значением, то в течение заданного (в секундах) временного интервала можно совершать те действия, результат выполнения которых вы хотите запечатлеть на скриншоте:

```
$ fbgrab -s 42 screen.png &
```

В этом примере fbgrab будет запущен в фоне, освободив командную строку для дальнейшей работы пользователя, и сделает скриншот консоли лишь через последующие 42 секунды.

И последнее замечание о fbgrab: с помощью этой утилиты сделаны скриншоты, приводимые во всех частях данной статьи.

Просмотр изображений: fbida

Следующий логичный шаг — просмотр изображений. Для этого рекомендую обратиться к наработкам проекта fbida [2], создавшего ряд небольших утилит, интерес среди которых в данном контексте представляют fbi и fbgs. fbi (framebuffer imageviewer) — это непосредственно просмотрщик изображений, а fbgs — скрипт к нему для отображения документов в форматах PDF/PostScript.

В fbi поддерживаются следующие форматы файлов: GIF, JPEG, PNG, BMP, PPM, TIFF, XWD, PhotoCD. При открытии

других производится попытка конвертирования с помощью ImageMagick.

В fbi предусмотрены режимы слайд-шоу (флаг «-t» со значением в секундах для автоматической смены картинки), чтения текста («-P») и «редактирования» изображения («-e»). Второй режим предназначен, например, для отсканированных документов и просмотра PDF (через fbgs). В нем просмотрщик не производит вертикального сдвига при открытии больших изображений, а осуществляет его по нажатию на пробел — причем на количество пикселей, чуть меньшее, чем высота текущего экрана, чтобы пользователь смог продолжить чтение с нужной строки. Режим редактирования ограничен функциями поворота картинки по часовой стрелке (и против нее; но все это доступно только для JPEG) и удаления файла, а также оснащен возможностями создания резервных копий файла и сохранения времени модификации файла.

Для открытия в fbi изображений в режиме автоматического масштабирования (т.е. с «подгонкой» под текущий размер экрана) достаточно выполнить:

```
$ fbi -a ~/images/*.jpg file2.jpg -l /path/to/file3.jpg
```

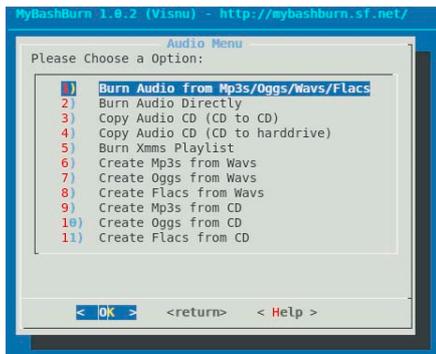
Во время просмотра можно увеличивать/уменьшать масштаб («+», «-»), переходить между картинками (PgUp, PgDn), просматривать информацию EXIF. Справка о доступных действиях вызывается нажатием на клавишу «h».

Из других подобных приложений выделю:

- ☑ **fbv** [3] — простой просмотрщик, использующий libungif для поддержки GIF, libjpeg — для JPEG, libpng — для PNG;



Просмотрщик изображений fbi



Меню «Аудио» в MyBashBurn

ставляющая: программа позволяет записывать CD/DVD с данными и многосессионные CD, AudioCD, поддерживает работу с ISO (создание и запись на CD/DVD), BIN/CUE (для записи AudioCD). Для работы с аудио предусмотрены также запись диска из плейлиста XMMS, кодирование из AudioCD или WAV в MP3/OGG/FLAC.

Существует и еще одна надстройка к традиционным утилитам для записи дисков в консоли – burncenter [17]. Ее «интерфейс» – это череда вопросов, на кото-

рые предстоит выбрать ответы для записи диска. Функционал близок к MyBashBurn, хотя и уступает ему по некоторым возможностям. Для копирования аудиодисков на жесткий диск есть также утилита abcde. Она позволяет кодировать звуковые дорожки CD в форматы Ogg Vorbis, Ogg Speex, FLAC, MP3, Musepack, M4A. Поддерживаются теги ID3/ID3v2, CDDb (как интернет, так и локальная база) для определения метаданных о диске, нормализация звука, возобновление начатого ранее процесса копирования.

В следующей серии

Как и сообщалось ранее, в следующей, заключительной, части статьи будут рассмотрены консольные интернет-приложения, среди которых – веб-браузеры, FTP-клиенты, IM/IRC-клиенты, email-клиенты, ed2k-клиент и RSS-агрегатор.

Я по-прежнему принимаю пожелания по включению в статью информации о каких-либо программах – присылайте их мне на e-mail.

1. <http://hem.bredband.net/gmogmo/fbgrab>.
2. <http://linux.bytesex.org/fbida>.
3. <http://s-tech.elsat.net.pl/fbv>.
4. <http://www.svgalib.org/rus/zgv>.
5. <http://aa-project.sourceforge.net/aview>.
6. <http://www.mpg123.de>.
7. <http://mpg321.sourceforge.net>.
8. <http://www.xiph.org>.
9. <http://mp3blaster.sourceforge.net>.
10. <http://moc.daper.net>.
11. <http://www.red-bean.com/~decklin/software/lastfmsubmitd>.
12. <http://www.mplayerhq.hu>.
13. <http://www.mplayerhq.hu/design7/info.html>.
14. <http://linux.bytesex.org/xawtv>.
15. <http://mybashburn.sourceforge.net>.
16. <http://bashburn.sourceforge.net>.
17. <http://alx14.free.fr/burncenter>.
18. <http://www.hispalinux.es/~data/abcde.php>.

Дмитрий Шурупов
(osa@samag.ru)

Qt и GTK+: два кита. Часть 1

Перед созданием программы в UNIX-подобных системах программист сталкивается с выбором языка программирования и библиотеки, которая ляжет в основу программы. Библиотека подразумевает под собой не просто набор красивых виджетов, но и «невидимую» часть, которая предоставляет работу со списками данных, со строками, файлами и тому подобным. В этой статье я постараюсь сравнить основные возможности двух популярнейших библиотек: GTK+ и Qt – не для того, чтобы вынести приговор, какая из них лучше. У каждой есть свои плюсы и минусы. Я хочу лишь сравнить их возможности по некоторым критериям, а выбор остается за вами.

Почему именно Qt и GTK+? Почему бы не прибавить к ним достойные библиотеки вроде FOX Toolkit или wxWidgets? Причины две: статья и без того большая (стоит из 3 частей, которые появятся в ближайших выпусках «Open Source»), а кроме того, Qt и GTK+ всё же гораздо более известны и популярны, чем другие библиотеки.

Некоторые предварительные замечания. Когда в статье будет говорить о сборке проекта, подразумеваются классические системы сборки, а именно: autotools для GTK+ и qmake для Qt.

Я не преуменьшаю значения stake и Scons – просто надо давать примеры, а как писал мифический Козьма Прутков, «нельзя объять необъятное». Примеры Qt будут на C++, а для GTK+ – на обычном Си – все согласно языкам, на которых эти библиотеки написаны.

Перейдем к сравнению библиотек. Многое останется за бортом, но основные возможности библиотек я постараюсь осветить. Крупные библиотеки можно уподобить айсбергам – ведь у айсберга 90 процентов объема скрыто под водой. Две части статьи посвящены такой подводной части: классам и функциям, которые составляют основу любой толковой библиотеки и не имеют непосредственного отношения к графическому интерфейсу. И лишь в третьей части появятся окошки да кнопки.

Распространение и лицензии

Qt и GTK+ включены в популярные дистрибутивы GNU/Linux и *BSD.

Qt, кроме самой библиотеки, снабжает программиста примерами в исходниках (также впечатляет бинарная демка, доступная в Win32-версии), обучающими статьями, документацией (ее можно просматривать в HTML-браузере либо в «фирменном» браузере от Qt – Qt

Assistant), а также системой сборки qmake и утилитами Qt Designer (визуальный конструктор интерфейса) и Qt Linguist (редактор файлов локализации). Таким образом, Qt – очень самодостаточная библиотека, несущая с собой всё нужное для сборки программ, написанных под нее.

GTK+ тоже идет с примерами и иногда критикуемой документацией. В самом деле, некоторые важные вещи (например, работа с отображением списков) описаны в руководстве GTK+ не самым лучшим образом, из-за чего появляются замечательные обучающие статьи от сторонних авторов. Эти статьи доступны с сайта GTK+ (www.gtk.org). В состав GTK+ не входят особые дополнительные утилиты, если не считать программ вроде перевода растровых картинок в массивы на языке Си. Эти массивы вставляются прямо в код и отрисовываются так же, как обычные картинки.

Программы, написанные под Qt и GTK, могут быть собраны в UNIX-подобных системах и в Windows. Qt-программы собирать под Windows проще: qmake и файлы конфигурации исходника работают одинаково на всех платформах. Вы можете писать и собирать свою программу полностью в Linux, а если решите выпустить ее под Windows, то скачайте себе Windows-версию Qt и той же qmake пересоберите свой проект в этой системе фактически без правки каких-либо файлов. При установке Qt под Windows автоматически устанавлируется (из сети) Win32-порт компи-

лятора GCC. И в Windows Qt остается самодостаточной системой.

GTK+ лицензирована под LGPL, а Qt имеет двойную лицензию: GPL и особую Qt Commercial License. Вы можете писать коммерческие программы в Qt, и не используя последнюю, однако тогда вы должны распространять свой продукт под GPL. А Qt Commercial License – платная, но позволяет не открывать код.

Архитектура

GTK+ по большому счёту состоит из нескольких частей.

За виджеты отвечает собственно библиотека GTK+, внося своим именем некоторую путаницу: выходит, что в библиотеку GTK+ входит библиотека GTK+.

К ней напрямую примыкает Pango – движок рендеринга текста. Виджеты GTK+ используют для своей отрисовки библиотеку Cairo, которая в состав GTK+ не входит. Зато входит GDK – прослойка ниже уровня для взаимодействия с X-сервером, Cairo, устройствами ввода, а также отвечающая за работу с изображениями, шрифтами и буфером обмена. Есть также вспомогательная библиотека gdk-pixbuf, непосредственно отвечающая за загрузку, сохранение и обработку картинок. В самой основе GTK+ лежит библиотека Glib (не путать с GNU C library), обеспечивающая очередь событий, ввод-вывод, нити, работу со строками, списками, хэш-таблицами, выделение памяти и многое другое.

Всё это обилие библиотек дает повод думать, что Qt, в противовес GTK+, представляет собой эдакий монолит и что достаточно подключить мифический модуль Qt, как всё сразу станет доступным и работоспособным. На деле это не совсем так. И даже при компиляции в файле настройки системы сборки надо подключить отдельные части Qt.

Например, чтобы работала поддержка сетевых протоколов и обработка XML, надо в файл проекта добавить примерно такую строку:

```
QT += network xml
```

Qt – строгая ООП-среда. В ней всё – кроме, пожалуй, чисел – классы. Строка – класс QString. Цвет – QColor. Единичный символ и тот – класс QChar. GTK+ – тоже ООП, но в меньшей степени. Средствами обычного не-ООП Си в GTK+ создана иерархия объектов. Живдется это дело на структурах вместе с макросами. У объектов в GTK есть свойства, которые можно читать функцией `g_object_get` и записывать с помощью `g_object_set`. У объек-

тов нет своих функций-членов – вместо этого сами объекты передаются как параметры функциям, которые с ними работают. Фактически каждый класс-потомок в GTK+ – это структура, унаследовавшая все поля своих предков. Понятное дело, что о множественном наследовании речь не идёт.

При работе с GTK+ создание собственных классов – редкость. Чаще всего используются уже готовые классы. Вот тут проявляется важное отличие Qt от GTK+: Qt полна готовых решений. Прекрасные отполированные классы, самостоятельные единицы – глыбы. А GTK+ дает нам много мелких объектов – кирпичей. В Qt мы строим из глыб и кирпичиков, в GTK+ – только из кирпичиков. Со всеми вытекающими последствиями. С одной стороны, то, что «в Qt» делается парой строк кода, в GTK+ может обратиться не одним десятком строк, а то и потребует написания целой вспомогательной библиотеки (об этом см. дальше). С другой стороны, Qt в некоторых случаях не дает нужной гибкости.

Простой пример: в GTK+ в диалоговом окне выбора файла предусмотрен виджет предварительного просмотра. Это вдобавок к тому, что можно добавлять виджеты в само окно диалога. Но особо предусмотрено помещение виджета предварительного просмотра. В Qt такого нет. В Qt дан `QFileDialog` – пользуйтесь! Конечно, в самый низ его окна можно добавлять свои виджеты, например, вот так туда помещается метка класса `QLabel`:

```
QFileDialog dialog (this);
QLabel *l = new QLabel ("test");
dialog.layout()->addWidget(l);
```

Но если я хочу сделать традиционный, как в GIMP, предварительный просмотр картинки справа от списка файлов, то в Qt, используя стандартный диалог, это невозможно.

Ресурсы

Большинство программ, отличных от хрестоматийной «Hello, world!», нуждаются в разных дополнительных файлах с данными. Это могут быть глифы кнопок и картинки, документация, шаблоны, скрипты. Само собой, они включаются в дистрибутив программы либо в собранный с нею пакет и при установке копируются в некий каталог. Например, при установке из исходника данные традиционно помещаются в `/usr/local/share/имя_программы`. А переменная внутри исходника указывает на этот путь.

Утопия разрушается, когда мы начинаем приспосабливать программу

под ее сборку в различных дистрибутивах и системах. Правила сборки пакетов для Debian требуют помещать картинки в `/usr/share/pixmaps/`, а документацию в `/usr/share/doc/имя_пакета_данных_программы`. Да, именно «имя пакета данных», ибо в Debian любят разбивать продукт на два пакета: с бинарными файлами и с данными (документацией, картинками и так далее). В Darwin и OpenBSD – тоже свои тонкости. И надо всё предусмотреть. Для этого в исходник загоняется множество инструкций «`#if defined`» и «`#ifdef`», которые только и делают, что проверяют, в каких условиях собирается программа и какие пути к файлам с данными надо использовать.

В Qt существует движок ресурсов, позволяющий помещать нужные вам файлы в виде «ресурсов» прямо в исполняемый файл. «Всё своё ношу с собой». В исполняемый файл Qt-программы можно внедрить что угодно: файлы локализации, картинки, документацию. Механизм этого дела очень прост: достаточно задать `qrc`-файл с перечислением файлов, которые вы хотите включить в программу в качестве ресурсов. Вот пример файла описания ресурсов:

```
<!DOCTYPE RCC><RCC version="1.0">
<qresource>
  <file>images/copy.png</file>
  <file>translations/ru.qm</file>
  <file>manuals/ru.html</file>
</qresource>
</RCC>
```

Здесь объявляются ресурсами файлы `copy.png` (пусть это будет глиф для кнопки «Копировать»), `ru.qm` (файл с русской локализацией) и `ru.html` – файл с русской версией руководства. Файл описания ресурсов надо подключить к проекту – при использовании системы сборки `qmake` в файле `.pro` добавляется строка:

```
RESOURCES = application.qrc
```

где `application.qrc` – файл описаний ресурсов. Файлов может быть несколько, тогда пишем примерно так:

```
RESOURCES = buttons.qrc \
manuals.qrc
```

Утилита `qmake` «подхватывает» такие файлы автоматически. Можно и вручную, с помощью компилятора ресурсов `rcc` (он обычно – в `/usr/lib/qt4/bin/rcc`). Теперь самое удивительное о ресурсах – доступ к ним. Qt позволяет открывать файлы из ресурсов так же просто, как и внешние файлы, только путь к файлу-ресурсу надо предварять двоеточием. Например, чтобы загрузить иконку `copy.png`, в ко-

де загрузки картинки указывается путь <code></code>:/images/cut.png):

```
QIcon *icon = new QIcon („:/images/cut.png“);
```

Более того, в дереве каталогов ресурса можно использовать все встроенные в Qt способы работы с файловой системой. Допустим, необходимо получить список включенных в ресурс файлов с локализациями (которые в Qt имеют расширение .qm). Класс QDir дает средства для получения такого списка. Объявляется экземпляр класса и устанавливается начальный путь <code></code>:/ – корневой каталог дерева каталогов ресурсов:

```
QDir dir;
dir.setPath („/“);
```

После этого в нем изменяется текущая директория на каталог <code></code>translations):

```
dir.cd („translations“);
```

Теперь создается фильтр по маске <code></code>*.qm) и назначается экземпляру QDir:

```
QStringList filters;
filters << „*.qm“;
dir.setNameFilters(filters);
```

Наконец, можно получить список всех файлов из текущего каталога, соответствующих этому фильтру, и поместить его в экземпляр QStringList – files.

```
QStringList files = dir.entryList ();
```

Надо помнить, что файлы-ресурсы раздувают исполняемый файл. Увеличивается потребление памяти за счет ненужных пользователю «лишних» локализаций и тому подобных вещей. Логика подсказывает, что в ресурс удобнее всего включать что-нибудь небольшое, вроде глифов для кнопок. Но если пренебречь расходом памяти, то включение в ресурс всего, что может понадобиться программе, существенно упрощает и даже сводит

к минимуму заточку проекта под различные дистрибутивы и системы. Всё «скидывается» в ресурс, а там уже известно расположение каталогов – одно на все случаи одинаковое.

Что до GTK+, то в ней под «ресурсами» подразумеваются rc-файлы, задающие стили отображения виджетов – для работы с такими rc-файлами есть довольно развитое API.

Локализация

Вообще считается, что обеспечение программы файлами перевода интерфейса правильно называется «интернационализацией». Но это слово слишком длинное, и повторять его много раз я не хочу. Поэтому здесь буду писать везде «локализация». В GTK+ интерфейс переводится средствами стандартного gettext. Поэтому в Glib входят привычные макросы: подчеркивание (_) и прочие. Нужно поместить строку для перевода:

```
_ („Some text“)
```

На основе этих строк по команде «make update-po» в каталоге с файлами перевода файлы обновляются и содержат в себе записи вида:

```
#: src/interface.c:166
msgid „File“
msgstr „файл“
```

Такой файл открывается в любом редакторе, и каждому msgid ставится в соответствие перевод msgstr. Я не буду углубляться в подробности – читайте документацию по gettext. После сборки проекта из исходных файлов перевода получаются gmo-файлы особого формата – они-то и подгружаются gettext для перевода строк в программе.

В Qt используется свой движок перевода. Строки, которые нужно перевести, помечаются функцией tr():

```
tr(„some text to translate“)
```

Утилита lupdate создает файл-шаблон для перевода, а также обновляет существующие файлы с переводами. Файлы – тоже текстовые, однако по формату отличны от нужных gettext. Файлы локализации Qt имеют расширение .ts и содержат данные в формате XML. ts-файлы можно, конечно, править в текстовом редакторе, однако Qt предоставляет довольно удобный специальный редактор QtLinguist. На основе текстовых ts-файлов утилита lrelease создает бинарники с расширением .qm, которые и подгружаются движком перевода Qt. Примерно вот так:

```
// Создаем и устанавливаем «переводчик»
// для локализации самой Qt:
QTranslator qtTranslator;
qtTranslator.load(„qt_ “ + „
    QLocale::system().name());
app.installTranslator(&qtTranslator);

// А теперь – переводчик для строк
// из программы:
QTranslator myappTranslator;
myappTranslator.load „
    („:/translations/hello_ “ + „
    QLocale::system().name());
app.installTranslator(&myappTranslator);
```

Обратите внимание, что в примере загружается файл перевода из ресурса. Имя файла составляется из «hello_» и строки, возвращенной функцией QLocale::system().name(). При русской локализации у нас получается hello_ru. Но где же расширение .qm? Хотя оно опущено, экземпляр QTranslator всё равно загрузил файл, добавив нужное расширение.

Для полноты картины – вот как выглядит загрузка локализации в GTK+:

```
#ifdef ENABLE_NLS
bindtextdomain (GETTEXT_PACKAGE, „
    PACKAGE_LOCALE_DIR);
bind_textdomain_codeset „
    (GETTEXT_PACKAGE, „UTF-8“);
textdomain (GETTEXT_PACKAGE);
#endif

gtk_set_locale ();
```

Петр Семилетов
(tea@list.ru)

Цикл интервью от SourceKibitzer. Эпизод #6: Денис Харламов (GridGain)

Денис Харламов:
«Хотел стать ресторатором, но пошёл в Open Source»

В своём блоге Денис Харламов культивирует приготовление свежемолотого ко-

фе, пишет стихи, мечтает об одиночестве, тишине и ничегонеделании на диване. И практически ничем не выдаёт себя – разработчика продукта GridGain, ломающего стереотипы элитности и недоступ-

ности грид-систем.

Ликбез: «Грид» спешит на помощь

Идея грид-систем очень проста: не хватает ресурсов для выполнения сложного задания – раздели его на более мелкие задачи, реши их по отдельности, собери результаты воедино и получи ответ. Если памяти слишком мало – используй кеш данных. Если же задание перегружает процессор, то нужен вычислительный грид. Классическая грид-система пред-

Серия интервью с разработчиками ПО с открытым кодом: «А как же деньги?»

Что такое «русское» Open Source-ПО и можно ли вообще говорить об этом явлении в отдельно взятом регионе или стране? Чем живут его разработчики? Какие препятствия им не под силу? И за счет каких средств они, в конце концов, существуют?

лагают одновременное решение для обеих проблем.

Как давно вы попали под влияние идеологии OS и сколько OS-проектов уже реализовали?

Под влияние «идеологии» я попал в конце 90-х, когда состоялось мое первое знакомство с Linux. Не скажу, что оно было удачным, но впечатление эта программа произвела: можно было заглянуть в код и проследить ход мыслей разработчика системы. Что же касается проектов с открытым кодом, то это мой первый серьезный продукт. До этого были лишь небольшие программы и наработки.

Какие проблемы решает GridGain? Кто его использует?

Если бы я был менеджером по продажам, то сказал бы, что любые, и в первую очередь финансовые. И это правда, потому что в конечном итоге наш продукт ведёт к минимизации затрат.

А точнее, GridGain нужен при решении проблем, связанных с нехваткой ресурсов при выполнении сложных вычислительных задач. Области использования – финансовая аналитика, трехмерная графика и обсчет 3D-сцен, научные исследования. Долгое время понятие «грид-системы» ассоциировалось исключительно с научным и очень сложным в использовании продуктом, который могли позволить себе лишь немногие организации. Мы решили развеять этот миф и создали простой и доступный грид, позволяющий развернуть вычислительную систему на тех мощностях, что уже имеются. И в этом наше неоспоримое достоинство.

Приведу имена некоторых наших пользователей и области применения GridGain: научные вычисления в Гданьском технологическом университете, область биотехнологий – FivePrime Therapeutics, Inc, финансовый анализ в Moody's KMV, симуляторы различных систем в InterPSS.

Изначально продукт был ориентирован на сектор малого и среднего бизнеса, который не может позволить себе существенных инвестиций в сверхмощные компьютеры и системы. Однако мы обра-

Чтобы лучше узнать мир героев нашего времени, портал SourceKibitzer совместно с электронным приложением «Open Source» к журналу «Системный администратор» запустил серию интервью с русскоязычными разработчиками ПО с открытым кодом. Мы хотим рассказать о тех, кто до сих пор был скрыт за экраном монитора.

тели внимание, что и крупные корпорации начали использовать GridGain. Мы до сих пор гадаем, что их к этому подтолкнуло: качество продукта или открытость его кода.

Какой процент от клиентов GridGain составляют российские фирмы?

К сожалению, мы изначально не ориентируемся на российский рынок. Нам кажется, что в силу экономического развития, а также молодости рынка программных продуктов, местные компании не готовы использовать Open Source-продукты и платить деньги за их поддержку. Ориентируясь на статистику по скачиваниям, могу отметить, что Новосибирск, Дубна, Москва и Петербург проявили интерес к GridGain, что скорее свидетельствует об использовании его в научных исследованиях, нежели в коммерческих целях.

Как так получилось, что компания была основана в Калифорнии? И как вы в неё попали?

Компания основана двумя живущими в США людьми, долгое время занимавшимися разработкой коммерческих грид-систем: Никитой Ивановым и Дмитрием Сетракяном.

А моя история такова. Работая в фирме StarSoft, я в какой-то момент решил, что необходимо что-то кардинально изменить в жизни. Незадолго до ухода из компании занялся вплотную Linux, чтобы закрыть пробелы в образовании. И стал приглядываться к Open Source-компаниям. У меня было несколько вариантов, но выбрал GridGain, о котором я узнал от друга и коллеги.

Меня никогда не привлекали корпорации по производству программного обеспечения, а GridGain отличается неформальностью, и люди общаются между собой не как конкуренты, а как члены одной семьи.

Какую роль в развитии продукта вы сыграли?

Я пришел в компанию около года назад, когда были готовы существенные наработки и про-

глядывались очертания будущего продукта. Помимо активной роли разработчика я занимаюсь тестированием, выявлением наиболее сложных ошибок, общением на форуме и внесением предложений по поводу функциональности программы.

Есть ли у GridGain конкуренты и почему вы лучше?

Из конкурентов отмечу двух: GigaSpaces и Terracotta. Это наиболее значимые игроки на рынке коммерческих грид-систем, но это не означает, что они единственные.

Здесь надо различать так называемую виртуализацию (создание единой, прозрачной для разработчика виртуальной машины на основе нескольких реальных VM) и собственно грид. Terracotta относится к первой категории – виртуализации – и именно поэтому обладает рядом недостатков. GigaSpaces вырос из data-grid и возможности его использования как вычислительного грида ограничены его же прошлым. К тому же это платные продукты.

Говоря о GridGain, я бы хотел отметить открытость его архитектуры и подчеркнуть две уникальные черты, делающие его очень простым в использовании. Во-первых, это Java-аннотации, позволяющие распределить вычисления буквально одной строчкой кода. Во-вторых, это т.н. peer class loading, благодаря которому вам нет необходимости заботиться о загрузке вашего кода на все компьютеры в гриде – это происходит автоматически, как только ваш код начинает выполняться.

Команда GridGain из 6 человек разбросана по всему миру: США, Россия, Украина. Как часто вы встречаетесь в реальном времени? Не препятствуют ли расстояния развитию фирмы?

Встречаемся мы редко, однако общение происходит как минимум один раз в день по Skype, что ни в коей мере не мешает нашей работе и развитию фирмы.

Почему была выбрана модель Open Source? Какие её недостатки?

На наш взгляд, Open Source-модель – это следующий этап развития рынка программного обеспечения. Да и по сути, с точки зрения потребителя, разница в open source и proprietary или close source невелика. Из недостатков можно отметить бо-

Название проекта	GridGain
Сайт	http://www.gridgain.com/product_10reasons.html
Статистика	За 10 месяцев существования продукта – более 10000 загрузок

лее позднее получение прибыли (с поддержки), но и это в случае отсутствия изначальных инвестиций в продукт.

Какое самое большое достижение за историю проекта?

Самое большое – это рост популярности продукта за столь короткий период, что говорит о его востребованности. Все наши вложения вернулись сторицей в виде клиентов и растущего числа наших пользователей. Первый релиз грида состоялся около 10 месяцев назад, сам продукт разрабатывался более года. На текущий момент мы перешли черту 10000 загрузок, и согласно нашей статистике GridGain запускают по всему миру каждые 4 минуты.

Дальнейшие планы по развитию GridGain?

Как раз недавно мы выпустили новую версию 2.0.2. Ряд существенных улучшений для более эффективного использования грида и большей его управляемости должен порадовать наших пользователей и, надеемся, привлечь новых. Этот релиз был действительно трудным для нас, поскольку мы стали пионерами в реализации ряда аспектов, только еще запланированных в Java 7, и в не настолько полном объеме, как удалось это сделать нам.

Из наиболее интересного можно отметить планы по интеграции с Google Android

и новый виток грид-вычислений на основе мобильных устройств. А также реализацию простой и гибкой консоли для управления продуктом и полную интеграцию с Amazon EC2.

Из глобальных планов есть только один – доказать, что грид-вычисления доступны всем и не являются чем-то экстраординарным.

Как вы думаете, что будет с OS-общиной в мире через 3 года? Через 8 лет?

Развитие, развитие и еще раз развитие. Естественный эволюционный процесс должен вытеснить более костную закрытую модель, а доверие к Open Source-модели через её понимание должно возрасти. Тогда знак равенства между Open Source и бесплатной и между Open Source и некачественной обязательно уйдёт.

Уже сейчас этот подъем заметен. Так, в ряде компаний требования по безопасности вынуждают предоставлять код продукта. Или часто перед началом разработки создаются т. н. пилотные проекты для принятия правильного и адекватного решения по используемым технологиям и существующему ПО от других компаний. И опять-таки open source здесь имеет преимущество, поскольку его лицензия обычно позволяет использование в коммерческих продуктах и не ограниченной по функциональности версии, а полноценной.

Какой, на ваш взгляд, самый успешный OS-продукт на данный момент?

В последнее время появилось достаточно много серьезных Open Source-продуктов, но, отдавая дань традиции, я бы выделил Linux. Хотя не могу не упомянуть Eclipse, JBoss AS и линейку продуктов Apache.

Хотелось ли когда-то попробовать себя в другой специальности?

Очень! Давняя мечта – стать ресторатором – так со временем и не воплотилась, несмотря на все идеи по ее реализации. Видимо, недостаточно работаю в этом направлении.

В своём блоге вы говорите: «Мне почему-то всегда хотелось, чтобы моя жизнь была похожа на танго». Что общего у танго с программированием? И какие шаги надо сделать, чтобы жизнь действительно стала похожей на танго?

У танго с программированием? Ничего. Танго – это стиль жизни, скольжение по ней, открытое и честное, страстное и увлеченное. Но это недостижимый идеал, далекий от воплощения. Окружающий социум никогда не позволит отдельному индивиду делать то, что ему заблагорассудится. Да и быть идеалистом опасно...

Олеся Быкова
(olesja_b@yahoo.com)

«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 мире 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! Если у вас есть свои темы и предложения, присылайте – рассмотрим все вопросы.

Редакция постоянно стремится к раз-

витию и совершенствованию своего издания, поэтому мы рады активным читателям и их комментариям как относительно публикуемых материалов, так и самой структуры приложения, его оформления и приоритетов. Публичное обсуждение «Open Source» проводится на форуме сайта журнала «Системный администратор» по адресу: <http://osa.samag.ru/forum>.

Мы заинтересованы в сотрудничестве с участниками свободных проектов, непосредственно разработчиками программного обеспечения с открытым кодом и с компаниями, специализирующимися на создании продуктов на базе технологий с открытым кодом, предоставляющими те или иные услуги по Open Source, занимающимися его распространением. Кроме того, мы готовы освещать различные тематические события, проводимые на территории России и посвященные Free Software, выступать в качестве информационных спонсоров выставок, конференций, презентаций и других мероприятий.

Связаться с редакцией можно по электронной почте osa@samag.ru.

Подписные индексы:

20780*

81655**

по каталогу агентства
«Роспечать»

88099*

87836**

по каталогу агентства
«Пресса России»

* годовой

** полугодовой

Стоимость подписки
через редакцию:

900* руб.
за 6 номеров

1800* руб.
за 12 номеров

Подписка на журнал «Системный администратор»

Российская Федерация

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

СНГ

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

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

- ☑ **Казахстан** – по каталогу «Российская пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»
- ☑ **Беларусь** – по каталогу изданий стран СНГ через РГО «Белпочта» (220050, г. Минск, пр-т Ф. Скорины, 10)
- ☑ **Узбекистан** – по каталогу «Davriy nashrlar», российские издания через агентство по распространению печати «Davriy nashrlar» (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)
- ☑ **Армения** – по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г. Ереван, пл. Сасунци Давида, д. 2) и ЗАО «Контакт-Мамул» (375002, г. Ереван, ул. Сарьяна, 22)
- ☑ **Грузия** – по списку номенклатуры «АРЗИ» через АО «Сакпресса» (380019, г. Тбилиси, ул. Хошараульская, 29) и АО «Мацне» (380060, г. Тбилиси, пр-т Гамсахурдия, 42)
- ☑ **Молдавия** – по каталогу через ГП «Пошта Молдовой» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
по списку через ГУП «Почта Приднестровья» (MD-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