



## Оглавление

Вступительное слово.....	2
ROSA: вопросы и ответы.....	2
Desktop.....	2
Будни разработчиков - энергопотребление и обнаружение устройств.....	2
Интеграция DrakConf и KDE Control Center.....	3
Инструменты разработки.....	5
FBA: отчеты о замкнутости репозитория по бинарным символам и «Obsolete» пакетам....	5
Результаты тестирования бинарной совместимости RELS и RHEL.....	5
ROSA ABF.....	6
ABF: базовый API, встроенные комментарии.....	6
ABF console client.....	7
LinuxCon Europe 2012.....	9
Авторы выпуска.....	12

## Вступительное слово

Медленно но верно, мы добрались до третьего выпуска нашего бюллетеня. За время, прошедшее с момента предыдущего номера, мы представили релиз операционной системы ROSA Enterprise Linux Server, а в декабре планируем выпустить новый дистрибутив ROSA Desktop 2012. Очень надеемся, что это не помешает нам подготовить новогодний выпуск «Точки РОСЫ» и призываем вас принять в нем активное участие (читайте об этом первый материал текущего выпуска). Надеемся на вашу помощь и поддержку.

Не скроем, приятно получать от вас положительные отзывы и слова поддержки. Еще более полезно изучать критические замечания. Но что заставляет по-настоящему гордиться, так это волна аналогичных проектов в русскоязычном Linux-сообществе, которую мы вызвали своим примером. Если это еще не эпидемия, то вирус точно :-).

*С уважением,  
команда выпуска «Точки РОСЫ»!*

## РОСА: вопросы и ответы

В номере, который планируется выпустить к Новому году, мы хотим предложить вам кое-что новое. Если у вас есть вопросы к разработчикам или руководству компании, отправляйте их на [ask-rosa@rosalab.ru](mailto:ask-rosa@rosalab.ru). Вопросы будут переданы по адресу и ответы на самые интересные из них вы сможете прочесть в новогоднем выпуске «Точки РОСЫ». Отдельная просьба не писать вопросы в комментариях к новостям на публичных ресурсах, на форумах и т.д., а отправлять их на указанный адрес. Это не потому, что мы вредные, а потому что так сбор и анализ вопросов будет для нас гораздо проще.

## Desktop

### **Будни разработчиков - энергопотребление и обнаружение устройств**

При разработке ROSA Desktop 2012 мы провели массу преобразований и переход на новые версии ПО - kmod, grub2 и новые ядра. Мы также решили использовать последнюю версию X-сервера под номером 13, чтобы не отставать от остальных дистрибутивов (первым его использовала Убунту в 12.10, и он будет применен во всех новинках этой осени-зимы).

Сразу же встал вопрос, а как быть с обнаружением оборудования, почему у нас не так хорошо, как ранее, определяются устройства, как быть с поддержкой видеокарт и новинками, привносимыми ядрами > 3.4.x.

В результате была сделана ревизия программных средств дистрибутива и проведена достаточно большая работа по внесению всех новшеств в программный код. И в настоящий момент все компоненты полноценно развиваются в ROSA, git с ними вы можете найти на ABF (<https://abf.rosalinux.ru/soft>).

Теперь по порядку, что было сделано за последнее время: X-сервер был обновлен до версии 13, как и все модули и драйвера. Переход был сделан достаточно бескровно: единственная потеря - это free only драйвера для видеокарт AMD/ATI старше чем HD5xxx.

**ldetect-lst:** та часть, которая отвечает за правильное обнаружение оборудования в утилитах

ROSA. Здесь два важных изменения:

- обновление базы устройств, которого не было с весны 2011 года, а также новые модули получения информации - из списка модулей ядра в связи с переходом на KMOD;
- новый алгоритм работы с видеокартами AMD/ATI, который был переписан заново и позволяет теперь правильно выставлять драйвера (был удален устаревший драйвер radeonhd и заложен механизм для legacy драйверов, которые пока, увы, не работают с последним X-сервером).

**drakx-утилиты.** Здесь главная вещь, которая была сделана - восстановление работы детектирования режимов энергопотребления и переписывание их на работу с cripower. Теперь desktop ядра автоматически имеют включенный ondemand governor (что означает снижение частоты и потребления энергии процессора, когда это не нужно), и система верно определяет нужные модули ядра и загружает их при старте.

**XFdrake.** При переходе на Grub2 у нас «отвалилась» возможность добавления нужной опции (pokmsboot) в строку загрузки. Мы опять же переписали эту часть, восстановив правильную установку и работу видеодрайверов (особенно проприетарных драйверов). Вместе с обновленным списком Card+ теперь обнаружение видеокарт должно быть гораздо лучше и правильней.

Проблемы, конечно же, остаются - не всегда гладко происходит переключение со свободных драйверов на проприетарные, новый X-сервер преподносит свои сюрпризы, пользователи P4 и старых Atom вынуждены для получения верного энергопотребления использовать обходные маневры (изначально «кривой» модуль теперь просто игнорируется), но главное, что их можно также поправить.

В целом проведена достаточно большая работа по обновлению подсистемы, которая не была затронута на протяжении как минимум последних двух лет.

\*\*\*

kda: Кто-то понял вообще что это?

Juliette: я читала англ. вику и не помню, поняла я или нет

Juliette: видимо не поняла

--Обсуждение новости о бозоне Хиггса в #rosa-devel

\*\*\*

## ***Интеграция DrakConf и KDE Control Center***

Одним из направлений развития дистрибутивов ROSA является избавление от устаревшего инструментария drakxtools. В частности, в ROSA 2012 Desktop мы планируем избавиться от ситуации, когда утилиты конфигурации системы находятся в двух местах - обертке DrakConf для различных программ из drakxtools (иконка «Configure Your Computer» в SimpleWelcome) и «Центре управления KDE» и соответствующих инструментах других DE («Configure Your Desktop»). При этом для части утилит из drakxtools центр управления KDE предлагает более удобные аналоги, однако для некоторых программ достойных альтернатив нет.

В приближающемся релизе мы планируем полностью убрать DrakConf, а утилиты, для которых нет альтернатив в KDE, интегрировать в центр управления KDE. Для этого каждая утилита будет обернута в соответствующий KCM-модуль. Процесс интеграции утилит сейчас в самом разгаре, и первые версии KCM-оберток для ряда утилит уже готовы:

kcm-drakauth

kcm-drakfirewall  
kcm-drakguard  
kcm-drakinvictus  
kcm-draksec  
kcm-harddrake  
kcm-rpmdrake-sources  
kcm-rpmdrake  
kcm-rpmdrake-update  
kcm-update-freq  
kcm-userdrake  
kcm-XFdrake

Вы можете также установить пакет drakconf-kde4, который установит те kcm-модули, планируемые ко включению в дистрибутив по умолчанию.

Hardrake будет доступен в секции Hardware, большинство других утилит - в секции System Administration.



## Инструменты разработки

### ***ФВА: отчеты о замкнутости репозиториев по бинарным символам и «Obsolete» пакетах***

На сайте <http://fba.rosalinux.ru> уже давно доступны регулярно обновляемые отчеты о замкнутости репозиториев ROSA по зависимостям. Однако наличие всех зависимостей не всегда гарантирует успешность работы приложения - ведь установленные по зависимостям пакеты могут не предоставлять каких-то необходимых для его работы элементов. Одним из часто встречающихся примеров является отсутствие требуемых приложению разделяемых библиотек и бинарных символов. Такая ситуация возникает, если пакет с нужной библиотекой не устанавливается по зависимостям либо содержит библиотеку слишком новой (или слишком старой) версии, в которой отсутствуют нужные функции.

Для автоматического отслеживания подобных случаев мы настроили на ФВА анализ всех ELF-файлов репозиториев РОСЫ с целью сопоставления предоставляемых и требуемых наборов бинарных символов и библиотек. В настоящее время такой анализ проводится только для репозиториев main; отчеты доступны здесь:

[http://fba.rosalinux.ru/test/elf\\_closure/](http://fba.rosalinux.ru/test/elf_closure/)

Еще одной проблемой, с которой приходится время от времени сталкиваться при работе над большими репозиториями, является наличие в одном репозитории пакетов, один из которых объявлен устаревшим («obsolete») в пользу другого. Зачастую это вынужденная мера, так как от устаревшего пакета могут зависеть другие проекты, и до их обновления удалить его из репозитория нельзя. Однако имея несколько тысяч пакетов, отслеживать текущую ситуацию с зависимостями и удалять ставшие ненужными пакеты нелегко. ФВА теперь помогает и в решении этой задачи, предоставляя отчеты об устаревших пакетах:

[http://fba.rosalinux.ru/test/repomanage\\_obsoletes/](http://fba.rosalinux.ru/test/repomanage_obsoletes/)

\* \* \*

- Да этот ваш Mac неудобен - я не смог в нем найти консоль! --  
Обсуждения в офисе

\* \* \*

### ***Результаты тестирования бинарной совместимости RELS и RHEL***

Не секрет, что за основу нашей операционной системы ROSA Enterprise Linux Server (RELS) был взят самый известный и стабильный дистрибутив Red Hat Enterprise Linux (RHEL). В этот дистрибутив были добавлены фирменные приложения от компании «РОСА», чтобы улучшить функциональность дистрибутива для наших заказчиков. При этом изменения были внесены и в дистрибутив, и в системную часть для обеспечения работы данных приложений. Как всем известно, такие изменения могут нарушать бинарную совместимость между дистрибутивами, т.е. невозможность переноса приложений между дистрибутивами без необходимости их перекомпиляции.

Чтобы обеспечить такую совместимость, мы применяли специальные инструменты в процессе разработки. Одним из таких инструментов является ABI Compliance Checker [1], разрабатываемый на данный момент в компании «РОСА». Наиболее часто причиной нарушения бинарной совместимости между дистрибутивами являются несовместимые изменения в одной или нескольких системных библиотеках. Данный инструмент

предназначен для анализа таких изменений на совместимость.

К сожалению, совместимость может быть нарушена не только из-за изменений в библиотеках. Причиной нарушения совместимости могут послужить также изменения в любых, так называемых, интерфейсных файлах системы, которые могут быть использованы приложениями, т.е. в модулях Perl и Python, конфигурационных файлах и т.д. Для детального полностью автоматизированного анализа изменений в таких файлах у нас пока нет инструментов. Однако специально для такого случая у нас есть инструмент для визуализации изменений - PkgDiff [2], с помощью которого можно быстро просмотреть изменения в соответствующих установочных пакетах.

Для визуализации изменений во всех пакетах дистрибутива мы сделали специальный инструмент DistDiff, который запускает PkgDiff на изменившихся пакетах, содержащих интерфейсные файлы, и выводит статистическую информацию об изменениях в специальный отчет. Финальный отчет об изменениях [3] был проанализирован экспертами по бинарной совместимости из компании «РОСА», в результате чего большинство изменений были признаны совместимыми, а несовместимые изменения были исправлены до релиза. Таким образом совместимость RELS и RHEL составляет ровно 100%.

[1] <https://github.com/lvc/abi-compliance-checker>

[2] <http://pkgdiff.github.com/pkgdiff>

[3] [http://upstream-tracker.org/compatibility/RHEL\\_6.3\\_to\\_RELS\\_2012/x86\\_64/changes\\_report.html](http://upstream-tracker.org/compatibility/RHEL_6.3_to_RELS_2012/x86_64/changes_report.html)

## ROSA ABF

### **ABF: базовый API, встроенные комментарии**

В октябре команда ABF представила реализацию API, а также возможность использования встроенных комментариев.

Теперь все основные операции с ABF, за исключением взаимодействия с базой майнтейнеров и сборкой продуктов (созданием ISO-образов), могут производиться посредством ABF API. Документация доступна здесь: <http://abf-doc.rosalinux.ru/>. В настоящее время ABF API предоставляет около 60 функций. API находится в стадии «бета» и в будущем возможны изменения, несовместимые с текущей реализацией.

Основным продуктом, где ABF API уже активно используется, является консольный клиент ABF.

Следующее новшество ABF это встроенные комментарии: теперь ABF позволяет вам комментировать как коммиты целиком, так и обсуждать конкретные строки кода. Теперь можно более конкретно обсуждать спес-файлы, патчи или новый код.

\*\*\*

- Нет денег? Это называется cash flow.
- Это называется немного по-другому.
- Нет, нет, я про русский язык.
- Обсуждение на совещании ТК

## ABF console client

В предыдущем выпуске приводилась информация, что у нас появился новый способ взаимодействия с ABF - консольный клиент. За последний месяц много усилий разработчиков ABF было потрачено на разработку API, что дало больший простор и для разработки клиента. Правда, пока был задействован далеко не весь потенциал API, в основном упор был сделан на улучшение работы с базовой функциональностью. Итак, зачем же нужен этот консольный клиент? Рассмотрим жизненный цикл пакета. Пользователь ABF захотел взять проект, что-то изменить, собрать проект и опубликовать. Начнем по порядку.

- **Клонирование git-репозитория.** Пусть имя проекта мы знаем. Что нужно делать дальше: лезть в web-часть, вводить имя проекта в поиск, выбирать нужный, переходить на страницу проекта и брать адрес проекта в ABF, выполнять «git clone URL». Хотя погодите, а ведь не нужно! Достаточно выполнить «abf get PROJECT». PROJECT - имя проекта, возможно, с указанием владельца (например, akirilenko/abf-console-client. Если владелец не указан, берется группа по умолчанию).
- **Внесение изменений.** После того как нужные файлы были изменены, можно выполнить «abf put MSG» и выполнится «git add --all», «git commit -m MSG», «git push». Также нужно все архивы предварительно загрузить на File-Store и изменить содержимое .abf.yml файла, это тоже долгий и рутинный процесс, особенно если речь идет о такой обработке сотни пакетов в день. Клиент же это делает автоматически при выполнении команды put.
- **Отправка на сборку.** Для этого нужно зайти на web-страницу, найти проект, тыкнуть несколько галочек и отправить на сборку. Казалось бы, не так уж долго. Но если эти однообразные действия нужно проделывать сотни раз в день - проще это делать через консоль, тем более что можно написать простой скрипт, делающий это в автоматическом режиме. Да здравствует автоматизация!
- **Процесс сборки.** Для проверки текущего состояния сборки нужно заходить на страницу в ABF, искать нужное сборочное задание (если их много, еще и держать в голове ID заданий). А можно сделать проще - выполнить «abf buildstatus ID», где ID можно опустить и получить информацию о последнем отправленном с консольного клиента задании (подробнее об этом написано далее).
- **Публикация.** Опциональный шаг, так как, во-первых, можно собирать с автоматической публикацией, во-вторых, публикация не всегда нужна. В любом случае, «abf publish ID» опубликует собранное задание без проблем.

Как видим, наличие консольного клиента несколько упрощает жизнь, особенно майнтейнерам дистрибутива. Поэтому мы стараемся, чтобы он шел в ногу со временем и развивался в соответствии с требованиями пользователей. Что было сделано в последней версии:

- В процессе работы с git-репозиториями ABF клиент запоминает их пути в системе. Также, если у вас уже много репозитория и вы хотите кэшировать их пути разом - достаточно выполнить «abf locate update-recursive -d PATH», где PATH - директория с репозиториями. Для чего нужно помнить положение репозитория? Например, теперь в консоли можно выполнить «abf cd PROJECT» (PROJECT - имя проекта, с группой вначале - import/abf-console-client - или без. Если группа отсутствует - берется группа пользователя по умолчанию). Можно просто узнать директорию проекта - «abf locate -p PROJECT». Также в будущем команда «abf backport» сможет переносить файлы не только между ветвями одного проекта, но и между разными проектами.

- В первых версиях консольный клиент имел лишь начальную поддержку консольного автодополнения, теперь же дополняется практически все: имена опций, ветви git, в «abf build» дополняются имена репозитория для сборки и для сохранения (причем набор последних зависит от проекта, и потому дополняются только если проект был указан ранее в строке опцией --project/-p). Как результат, работать с консольным клиентом становится гораздо приятнее, ведь не нужно каждый раз писать длинные названия или вспоминать, что же можно указать в --save-to-repository. В первые разы дополнение работает с задержками (порядка 0.5-1 сек), но со временем данные кэшируются и задержки сильно уменьшаются.
- Появилась проверка спес-файла. Можно выполнить "abf clean" в директории git-репозитория, и будет выведен список проблем, найденных в спес-файле. В случае если один из source или patch файлов не может быть найден (если файл был указан не по URL, отсутствует в .abf.yml и отсутствует в директории со спес-файлом), будет выдано сообщение об ошибке. Так же могут быть выданы некоторые предупреждения, например, если файл одновременно указан в .abf.yml и присутствует в директории. Так же выдаются предупреждения о 'лишних файлах', которые присутствуют в директории или в .abf.yml, но не требуются для сборки. Команда "abf clean --auto-remove" удалит эти файлы.

Данный тест запускается автоматически при отправке на сборку из директории с проектом. Будут напечатаны все предупреждения, а в случае наличия ошибок задание не будет отправлено (Отменить выполнение проверки при отправке на сборку можно опцией --skip-spes-check).

- При отправке заданий на сборку консольный клиент не только выводит на экран ID отправленных заданий, но и запоминает их и связывает с проектом. Теперь "abf buildstatus" напишет информацию о последних заданиях, а при указании опции --project/-p (или если Вы в директории git-репозитория) - о последних для данного проекта.
- Что касается работы с API - существенно уменьшено количество ненужных запросов. Дело в том, что запрос, например, репозитория, дает так же частичную информацию о платформе, в которой этот репозиторий находится. Часто этой информации достаточно. Раньше из этого использовался только id, по которому выполнялся еще один API запрос для загрузки всех данных. В результате раньше первый запуск клиента порождал десятки ненужных запросов. Сейчас же создается новый объект "platform" и в него помещается вся доступная информация, а сам объект помечается как "stub". Если же попытаться получить значение поля, которое еще не загружено но должно присутствовать в классе - будет выполнен новый вызов API и вся информация будет загружена.
- Еще одна интересная особенность нового клиента - использование ETag ([http://en.wikipedia.org/wiki/HTTP\\_ETag](http://en.wikipedia.org/wiki/HTTP_ETag)) для кэширования данных сервера с их автоматической валидацией. Как результат, без опасности работать с устаревшими данными, получаем ускорение работы. В данный момент ABF не настроен для полной поддержки этой технологии, но в скором времени обработка таких кэшированных запросов будет действительно быстрой, а также такие запросы не будут влиять на счетчик запросов (напомним, что сейчас установлен лимит в 500 запросов к API в час).

Как видите, развитие продукта не останавливается. Все ваши отзывы (как положительные, так и отрицательные), предложения и пожелания приветствуются.



\*\*\*

konstantin\_vlasov: kda: Помню, помню, как в рассказе Беляева был упомянут диоксидаминоарсенобезнзолдихлоргидрат...

kda: konstantin\_vlasov: :) и ты даже запомнил как пишется?

konstantin\_vlasov: Да ты что, я был так им очарован, что зазубрил наизусть! :-)

-- Обсуждения в #rosa-devel

\*\*\*

## LinuxCon Europe 2012

*В блоге «РОСЫ» на Facebook был опубликован рассказ о наших продуктах, представленных на LinuxCon Europe 2012. А вот - некоторые впечатления о других участниках и о конференции в целом.*

Конференция была очень насыщенной — доклады шли в пять параллельных потоков. При этом нередко параллельно проходили доклады, предназначенные для схожих целевых аудиторий — так что приходилось делать нелегкий выбор, на что сходить. Частично это компенсировалось доступностью слайдов (сейчас многие презентации можно найти на сайте Linux Foundation) и возможностью «отловить» докладчиков в перерывах или вечером на каком-нибудь мероприятии от организаторов — где проходило немало дискуссий на технические темы в неформальной обстановке. Впрочем, и на самой конференции обстановка была далека от формальной — люди приходили, чтобы пообщаться на близкие им темы и узнать что-то новое (а не за футболками, ручками и прочими сувенирами, хотя всяких подарков раздавали множество).

Доклады были как сугубо технические, так и организационно-философского характера. Из популярных технических тем:

- облачные вычисления (спектр представленных компаний и проектов был очень велик — Citrix, Apache Cloudstack, OpenNebula, IBM, HP, и даже Microsoft);
- близкие к ним вопросы виртуализации (в основной части конференции рассказывали и про KVM, и про Xen, а заодно и про решения Parallels, а после основной части прошла серия семинаров про KVM и oVirt);
- отдельная секция была посвящена Tizen (который также активно рекламировался сотрудниками Intel на их стенде);
- в течение целого дня шли семинары про распределенную файловую систему GlusterFS.

Безусловно, лидерами по количеству посетителей являлись доклады приглашенных участников — не только потому, что параллельно с ними не шло никаких презентаций, но и потому, что темы были актуальны, а докладчики — интересны.



Линус Торвалдс обошелся без презентации и какой-то конкретной темы («Я же не знаю, что именно вам интересно» — «Nvidia!» — «На этот вопрос я уже дал исчерпывающий ответ :)»), вместо этого получился развлекательно-познавательный диалог с ведущим и залом. Конечно, не все вопросы носили шуточный характер — например, была затронута проблема взаимодействия разработчиков ядра с создателями Android, которые временами сильно затягивают с открытием своих наработок, не говоря уже об их передаче в основную ветку разработки. Создатель ядра Linux не выразил большой озабоченности этим вопросом, предложив вспомнить ситуацию десятилетней давности, когда разработчики ведущих дистрибутивов (RedHat, SUSE) предпочитали поддерживать множество собственных патчей к ядру, а не работать над общей кодовой базой. Постепенно положение дел нормализовалось — хотя разработка многих патчей по-прежнему происходит внутри дистрибутивов, они активно сотрудничают с другими командами — в частности, непосредственно с разработчиками ядра. Торвалдс выразил уверенность, что и в случае с Android будет выработан процесс, выгодный для обеих сторон.

Некоторые сессионные доклады также собрали приличное количество слушателей (так что многим пришлось даже стоять) — например, рассказ Джеймса Боттомли (James Bottomley) о ситуации с UEFI (Джеймс пообещал, что в скором будущем Linux Foundation договорится с Microsoft и подписанный ключом MS загрузчик, который можно будет использовать для загрузки основной системы). Тьяго Масиэйра (Thiago Macieira) с докладом про Qt Project также собрал немало слушателей — многие с интересом следят за постепенным открытием процесса разработки Qt.

По опыту можно сказать, что, например, открытый баг-трекер для Qt — действительно полезная вещь, располагающая к привлечению сторонних участников — имея дело с тестами LSB, мы иногда находим потенциальные ошибки в самом Qt, но еще года два-три назад процесс обработки сообщений о таких ошибках был абсолютно непрозрачен. Временами складывалось впечатление, что некоторые сообщения просто исчезают где-то за интерфейсом заведения отчета об ошибке — без дополнительного общения с разработчиками Qt нельзя было ни посмотреть его статус, ни узнать, работает ли кто-то над его исправлением. Теперь Qt стал гораздо ближе к сообществу. Правда, на вопрос Тьяго: «кто из присутствующих подписан на список рассылки разработчиков Qt Project?», руку поднял только Ларс Нол (Lars Knoll) — ведущий архитектор проекта.

Помимо докладов, немало информации можно было получить на стендах компаний и проектов. Было много стендов, связанных с облачными вычислениями и виртуализацией, некоторые стенды демонстрировали дистрибутивы и связанные с ними решения — помимо РОСЫ, в эту категорию попадали Fedora/RedHat, SUSE, WindRiver (с прицелом на встраиваемые системы) и ряд других. Представители Oracle рассказывали не только о своем дистрибутиве на основе RedHat, но и о других открытых проектах (в частности, MySQL), заверяя всех, что корпорация нацелена на взаимовыгодное сотрудничество с сообществом.

Усилиями Intel и WindRiver много внимания было сосредоточено на встраиваемых системах под управлением Linux. Intel вовсю рекламировал Tizen и демонстрировал прототип IVI на его основе. Многие посетители смотрели на Tizen настороженно, помятуя о судьбе его предшественников — Moblin и MeeGo, — фактически канувших в небытие.

Также представители Intel вовсю расписывали достоинства использования архитектуры x86 для создания устройств на основе Android — причем одним из ключевых моментов были не достоинства архитектуры как таковой, а удобство разработки. Ведь создание приложений под Android обычно производится на обычной настольной машине (которая в подавляющем большинстве случаев имеет архитектуру x86), а тестирование осуществляется в эмуляторе QEMU. За счет использования аппаратной поддержки виртуализации, производительность QEMU при эмуляции платформы x86 на порядок лучше, чем в случае ARM, что существенно ускоряет тестирование и обкатку приложений (да и просто делает эти процессы более удобными и не столь мучительными).

Интересно, что именно на стенде Intel наиболее активно рекламировался Android — стенд Google также был посвящен этой ОС, но был существенно скромнее, и каких-то агрессивных маркетинговых акций представители интернет-гиганта не проводили.

В общем и целом LinuxCon Europe 2012 получился очень познавательным и насыщенным (даже рассказ о наиболее запомнившихся аспектах растянулся на пару страниц :)).

- У нас есть технические писатели?
- Ну, да, есть взвод побывавший в боях.
- В смысле очень качественные писатели?
- Нет, просто осталось из восьми трое...
- Разговоры в офисе

## **Авторы выпуска**

Александр Казанцев

Андрей Пономаренко

Антон Кириленко

Антон Чернышов

Денис Силаков

Константин Кочерешкин

Свои отзывы и пожелания по интересным темам, недочетам и идеям по развитию отправляйте на [rosa-point@rosalab.ru](mailto:rosa-point@rosalab.ru)