



## Оглавление

Вступительное слово.....	2
Desktop.....	2
ROSA Popularity Contest - исследование популярности пакетов.....	2
Инструменты разработки.....	3
Сравнение репозиториев ROSA 2012 Desktop, Mandriva Cooker и Mageia Cauldron.....	3
Репозитории LSB доступны в формате rpmi.....	3
Urpm-heroclosure - анализ нескольких репозиториев и HTML-отчеты.....	4
ROSA ABF.....	5
Обновление ABF.....	5
Отдельное хранилище для архивов с исходным кодом.....	6
Pull Requests.....	7
Frontend Brother of ABF.....	8
Доступна первая версия консольного клиента для ABF.....	8
2Safe.....	9
Простой клиент 2Safe.com. Обзор API.....	9
Историческая ретроспектива.....	10
Подготовка релиза ROSA/Mandriva 2011: как это было.....	10
Авторы выпуска.....	11

## Вступительное слово

Всем отличного настроения, и да пребудет с вами удачная компиляция, не глючат программы, а ядро никогда не валится в панику!

Вот мы и добрались до второго выпуска нашего бюллетеня. В нем мы постарались учесть предыдущий опыт и высказанные пожелания в меру своих возможностей. В данном выпуске довольно-таки много новостей от команды разработчиков АВФ, за что им отдельное спасибо!

И, конечно же, огромное спасибо нашим разработчикам и инженерам, нашедшим время в своем плотном графике на то, чтобы написать и прислать статьи в «Точку РОСЫ».

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

## Desktop

### ***ROSA Popularity Contest - исследование популярности пакетов***

ROSA Desktop содержит тысячи пакетов с различными приложениями, библиотеками и утилитами. Для ряда пакетов мы гарантируем качественную поддержку, для некоторых - только обеспечиваем базовую функциональность.

Чтобы помочь разработчикам РОСЫ узнать, какие пакеты наиболее важны для пользователей, мы подготовили специальный инструмент - `rosa-popularity-contest` - который собирает статистику о пакетах, используемых в установленной системе. Раз в неделю инструмент посылает отчет на наш сервер, где производится подсчет статистики.

Чтобы принять участие в таком исследовании, достаточно просто установить пакет **rosa-popularity-contest**. Больше никаких действий со стороны пользователя не требуется - инструмент работает в полностью автоматическом режиме.

Анализ использования пакетов производится на основе сведений о времени последнего доступа (`atime`) к файлам пакета. Детали работы `rosa-popularity-contest` описаны на нашей вики: [http://wiki.rosalab.ru/ru/index.php/Исследование\\_популярности\\_пакетов\\_РОСЫ](http://wiki.rosalab.ru/ru/index.php/Исследование_популярности_пакетов_РОСЫ)

Отчеты по репозиториям публикуются здесь: <http://fba.rosalinux.ru/popcon/>

Для каждого пакета отчет содержит следующие численные показатели:

<b>Installed</b>	процент респондентов, у которых пакет установлен
<b>Vote</b>	процент респондентов, использовавших пакет в течение последнего месяца
<b>Old</b>	процент респондентов, не использовавших пакет
<b>Recent</b>	процент респондентов, недавно обновивших пакет; нельзя достоверно сказать, используют ли они его или нет
<b>No-files</b>	процент респондентов, для которых информация об использовании пакета недоступна (например, пакет установлен на ФС, примонтированную с опцией 'noatime')

Сейчас у нас есть только небольшое количество отчетов от разработчиков, но в будущем мы надеемся существенно расширить аудиторию.

## Инструменты разработки

### **Сравнение репозиториев ROSA 2012 Desktop, Mandriva Cooker и Mageia Cauldron**

При разработке дистрибутива одной из главных задач является выбор пакетов для него. При решении этой задачи полезно провести анализ репозиториев других дистрибутивов и узнать — что они посчитали важным для своих пользователей?

Для сравнения репозиториев urpmi у нас есть замечательный инструмент [urpm-repodiff](#). С его помощью можно, в частности, сравнивать репозитории РОСЫ, Mandriva и Mageia.

Полный отчет о сравнении репозиториев ROSA 2012 Desktop (main+contrib), Mandriva Cooker и Mageia Cauldron можно найти [здесь](#). Отдельно можно посмотреть [список пакетов](#), отсутствующих в РОСЕ, но имеющихся в родственных дистрибутивах.

Отчеты обновляются ежедневно.

\*\*\*

Хром не любит котов?! Да за такое его не то что по умолчанию, его из репов надо выкидывать нафиг!  
– Обсуждение в #rosa-devel

\*\*\*

### **Репозитории LSB доступны в формате urpmi**

Стандарт Linux Standard Base (LSB), разрабатываемый одноименной рабочей группой консорциума The Linux Foundation, - это не только текстовый документ, но и набор различных программных продуктов, призванных упростить создание дистрибутивов и приложений, соответствующих стандарту.

Благодаря совместным усилиям разработчиков «РОСА» и инженеров The Linux Foundation, репозитории с программными продуктами LSB теперь доступны в формате urpmi и могут быть подключены в дистрибутивах ROSA.

Для подключения репозиториев с последними выпущенными версиями инструментария LSB в 32-битной системе необходимо в консоли с правами root выполнить следующую команду:

```
urpmi.addmedia lsb http://ftp.linuxfoundation.org/pub/lsb/repositories/rpm/lsb-4.1/repo-ia32/
```

Также можно подключить репозиторий со snapshot-версиями инструментов:

```
urpmi.addmedia lsb-snapshot  
http://ftp.linuxfoundation.org/pub/lsb/repositories/rpm/lsb-snapshot/repo-ia32/
```

Для 64-битных систем понадобятся следующие команды:

```
urpmi.addmedia lsb64  
http://ftp.linuxfoundation.org/pub/lsb/repositories/rpm/lsb-4.1/repo-x86\_64/  
urpmi.addmedia lsb64-snapshot  
http://ftp.linuxfoundation.org/pub/lsb/repositories/rpm/lsb-snapshot/repo-x86\_64/
```

После этого в системе будут доступны для установки пакеты LSB. Все эти пакеты можно разделить на три группы:

- инструменты для оценки переносимости приложений и их проверки на соответствие стандарту LSB; основным инструментом является [Linux Application Checker](#) (пакет `lsb-task-app-testkit`, запуск командой `/opt/lsb/app-checker/bin/app-checker-start.pl`);
- тесты для проверки соответствия дистрибутивов стандарту LSB и фреймворк для их запуска и анализа результатов – [Linux Distribution Checker](#); чтобы получить все тесты и фреймворк, необходимо установить пакет `lsb-task-dist-testkit`;
- инструментарий [LSB SDK](#) для разработки переносимых приложений (пакет `lsb-task-sdk`).

Инструменты LSB рассчитаны прежде всего на разработчиков приложений и дистрибутивов, однако могут быть интересны и обычным пользователям. Например, [Linux Application Checker](#) позволяет проверить возможность запуска того или иного приложения в основных дистрибутивах Linux.

Более детальную информацию по работе с репозиториями LSB можно найти на сайте The Linux Foundation: <https://wiki.linuxfoundation.org/en/RepositoryInfo>.

Отчет [Linux Application Checker](#) для пакета `vim-minimal` из ROSA 2012 Marathon – `vim-minimal` может также без перекомпиляции быть запущен в Mandriva 2010.1, Oracle 6, RHEL 6 и SLES 11 SP1, но не в Asianux 3.0, CentOS 5.2 или Fedora 7.

**Analysis Results for vim-minimal-7.3.285-2 on x86**

Some compatibility problems detected

- There are 4 of 76 distributions that provide all the required libraries and interfaces.
- The Application uses 1 external library incompatible with LSB 4.1.

**Distribution Compatibility** | App Components | External Libraries | External Interfaces | LSB Certification

The table below shows the compatibility status of your application with the distributions analyzed by the Linux Foundation. Your Application will run on the "green" distributions without loader problems. Compatibility with the "yellow" distributions can be easily achieved by excluding unneeded libraries from the dependencies of your application. Making the Application compatible with the "red" distributions may require more effort to avoid using missing libraries/interfaces or by supplying them as a part of your application package. Please note that functional correctness is not guaranteed by this analysis.

Summary		Missing Libraries		Missing Interfaces		Comments
Distribution	Status	Common	Unknown	Common	Unknown	
<a href="#">Mandriva 2010.1</a>	OK	none	none	none	none	
<a href="#">Oracle Linux 6</a>	OK	none	none	none	none	
<a href="#">RHEL 6</a>	OK	none	none	none	none	
<a href="#">SLES 11 SP1</a>	OK	none	none	none	none	
<a href="#">Asianux 3.0</a>	INCOMPATIBLE	none	none	1 (list...)	none	
<a href="#">CentOS 5.2</a>	INCOMPATIBLE	none	none	1 (list...)	none	
<a href="#">Fedora 7</a>	INCOMPATIBLE	none	none	1 (list...)	none	

## Urpm-repoclosure - анализ нескольких репозиториев и HTML-отчеты

Одним из инструментов, используемых для контроля состояния репозиториев ROSA, является [urpm-repoclosure](#), проверяющий замкнутость репозиториев по зависимостям.

Изначально **urpm-repoclosure** умел проверять замкнутость одного конкретного репозитория или заданного набора пакетов. Однако структура репозиториев ROSA такова, что «самодостаточным» репозиторием является только *main*; при анализе замкнутости *contrib* и *non-free* следует учитывать, что пакеты из этих репозиториев могут использовать и пакеты из *main*. Кроме того, необходимо учитывать наличие обновлений в ветках *updates*, которые могут замещать пакеты из веток *release*.

До сих пор для анализа чего-то кроме *main/release*, мы копировали все нужные пакеты в одно место с их последующей обработкой [urpm-repomanafe](#) (чтобы оставить только самые свежие версии пакетов). Но такой подход подразумевает работу непосредственно с пакетами, что является ресурсоемким занятием. В то же время *urpm-repoclosure* поддерживает существенно более быстрый способ работы - на основе файлов *synthesis.hdlist* (опция **-hdlist**).

Недавно мы добавили в *urpm-repoclosure* поддержку работы с несколькими файлами *synthesis.hdlist* - теперь он может самостоятельно их «объединять», убирая устаревшие версии пакетов. Более того - можно указать, какие из дополнительных репозиториях содержат обновления анализируемых пакетов (файл со списком соответствующих *synthesis.hdlist* передается опции **-update-hdlists**), а какие служат лишь для удовлетворения их зависимостей **-dep-hdlists**). Теперь анализ репозитория *contrib* можно провести, используя всего четыре файла *synthesis.hdlist* (из *main/release*, *main/updates*, *contrib/release* и *contrib/updates*):

```
echo "synthesis.hdlist-contrib.updates" > updates.list
echo "synthesis.hdlist-main.release" > dep.list
echo "synthesis.hdlist-main.updates" >> dep.list
```

```
urpm-repoclosure -hdlist synthesis.hdlist-contrib.updates -update-hdlists
updates.list -dep-hdlists dep.list
```

Схожим образом можно проверять замкнутость персональных репозиториях на ABF.

Наконец, для удобства отображения *urpm-repoclosure* теперь умеет генерировать отчеты в формате HTML. Никаких дополнительных опций для этого указывать не надо - HTML-отчет создается автоматически вместе с текстовым. Пример HTML-отчетов можно посмотреть на ежедневно обновляемой страничке с отчетами по состоянию репозиториях ROSA 2012 Marathon и ROSA 2012 Desktop: [http://upstream-tracker.org/repoclosure\\_logs/](http://upstream-tracker.org/repoclosure_logs/)

\*\*\*

В Советском Союзе пытались воспитывать людей, про которых можно сказать: честный, умный, коммунист.

Но как ни старались, в одном человеке уживалось только два из этих признаков. Либо честный коммунист, либо умный коммунист, либо честный и умный, но не коммунист...

--- Из разговоров в офисе

\*\*\*

## ROSA ABF

### Обновление ABF

Мы рады представить три новых возможности ABF:

1. Email-уведомления о сборках
2. Поддержка разметки комментариев и сообщений об ошибках
3. Новый вид страницы со списком проектов (<https://abf.rosalinux.ru/projects>)

Немного деталей:

- Email-уведомления о сборках  
Оповещение по Email являлось одной из самых запрашиваемых возможностей. С настоящего момента, помимо автоматически обновляемой страницы мониторинга и

ленты активности с поддержкой RSS, вы можете получать электронные письма с уведомлениями о завершении сборочных заданий. Вы можете подписаться на разные типы оповещений, выбрав события, при возникновении которых будет посылаться письмо, а также включать/отключать получение уведомлений в Центре Подписки (<https://abf.rosalinux.ru/settings/notifiers>).

- Поддержка разметки комментариев и сообщений об ошибках  
Комментарии и описания задач являются важной составляющей работы, и полезно предусмотреть возможность их удобного оформления. Теперь в ABF у нас есть простой способ размечать ваши сообщения, используя Markdown - простой и легковесный язык разметки. Мы также подготовили небольшую заметку о Markdown с полезными советами по его использованию (спасибо github!). Вы можете увидеть эти советы рядом с любым комментарием или сообщением об ошибке.
- Новый вид страницы со списком проектов  
До недавнего момента пользоваться страницей со списком проектов было неудобно, если вы участвовали в тысяче проектов. Но теперь стало возможным ограничить число отображаемых проектов по их владельцам. Вы можете просматривать все проекты (приватные или открытые) заданной группы или фильтровать их по названиям с помощью панели поиска. Перейдите на <https://abf.rosalinux.ru/projects> и посмотрите на новый механизм в действии.

Подсказка: одновременно можно задать несколько параметров!

## ***Отдельное хранилище для архивов с исходным кодом***

Традиционно в сборочных средах различных Linux-систем все файлы, необходимые для сборки пакета, хранятся в репозитории некоторой системы контроля версий. В случае сборки RPM-пакетов к таким файлам относятся прежде всего архив с исходным кодом программы, срез-файл с инструкциями по сборке для rpmbuild и различные патчи. В ABF изначально все файлы, относящиеся к пакету, помещались в git-репозиторий соответствующего проекта. Однако практика показала, что хранение архивов с исходным кодом в git-репозитории нецелесообразно - ведь это бинарные файлы, для которых многие достоинства СКВ (например, отслеживание изменений в файлах и слияние модификаций от разных пользователей) неактуальны. А поскольку архивы нередко имеют достаточно большой размер, то клонирование git-репозитория со всеми ветками и историей может занимать существенное время.

Альтернативой является получение архивов с внешних источников (например, непосредственно с сайта разработчиков) - rpmbuild в ROSA предоставляет такую возможность. Однако использование внешних источников не представляется надежным решением - в момент сборки необходимый сайт может оказаться недоступен либо его скачивание займет существенное время из-за медленного соединения, файл с исходным кодом может быть модифицирован или перемещен, и так далее.

Для снижения нагрузки на git и ускорения работы с репозиториями без необходимости задействовать внешние источники, мы предоставили возможность вынести архивы с исходным кодом в специальное файловое хранилище непосредственно на ABF - <http://file-store.rosalinux.ru/>. Чтобы воспользоваться этим решением, вам необходимо:

- загрузить файл с исходным кодом на <http://file-store.rosalinux.ru/>. Сервис вернет вам контрольную сумму (SHA1) загруженного файла;
- в проекте на ABF необходимо создать файл .abf.yml следующего содержания:

sources:

"<имя файла>": <контрольная сумма>

(пример - <https://abf.rosalinux.ru/import/libreoffice/blob/rosa2012ts/.abf.yml>)

После этого ABF будет брать архив с исходным кодом из файлового хранилища.

Примеры проектов, уже использующих новую возможность:

- Libreoffice: <https://abf.rosalinux.ru/import/libreoffice/tree/rosa2012.1>

- Flightgear-data: <https://abf.rosalinux.ru/warpc/flightgear-data>

В будущем мы планируем перевести сборку всех пакетов в ABF на использование файлового хранилища и автоматически загружать архивы с исходным кодом при импорте проектов в ABF. Также для работы с файловым хранилищем будет представлен внешний API, а в разрабатываемый консольный клиент ABF будет добавлена функциональность по работе с архивами с исходным кодом.

## Pull Requests

После долгого периода разработки мы рады представить вам новую возможность ABF - поддержку Pull Requests!

- Что такое Pull Requests?  
Pull Requests («запросы на включение изменений») позволяют вам сообщить другим разработчикам об изменениях, которые вы поместили в git-репозиторий проекта на ABF. Как только такой запрос отправлен, заинтересованные участники проекта могут просмотреть список изменений, обсудить их и при необходимости внести дополнительные модификации.
- Зачем они нужны?  
До реализации таких запросов единственной доступной в ABF моделью совместной разработки была модель разделяемого репозитория. Эта модель используется преимущественно в небольших командах и организациях для совместной работы над внутренними проектами. При этом каждый разработчик имеет права на внесение изменений в общий репозиторий, а для изолированной работы используются отдельные ветки.

Теперь в ABF поддерживается еще одна модель: Fork & Pull. Эта модель позволяет всем клонировать проект и вносить изменения в свое ответвление без запроса прав на родительский проект. После внесения изменений они переносятся в исходный проект поддерживающими его разработчиками. Эта модель сокращает различные трения, возникающие с приходом новых участников, и популярна в открытых проектах, поскольку позволяет работать независимо, без выделенного координатора.

- Как их использовать?  
Склонировать проект, внесите желаемые изменения, при необходимости проведите сборку и отправьте запрос на включение изменений в родительский проект, обсудите изменения с другими участниками, поправьте код, если к нему есть замечания.  
Нет необходимости запрашивать доступ к исходному проекту или обсуждать каждый патч в почтовых рассылках. Все члены проекта и заинтересованные пользователи могут видеть изменения, помочь с написанием кода и тестированием или просто проголосовать за включение изменений или высказать критику.

- Мы разрабатываем проект на ABF. Для чего нам нужны Pull Requests? Вы можете использовать Pull Requests не только между разными проектами, но и для переноса изменений между разными ветками одного проекта!

Отличное решение для инспекций кода, попробуйте!

## **Frontend Brother of ABF**

ABF является мощной системой сборки дистрибутивов и разработки проектов. В частности, с помощью ABF мы собираем ROSA (как серверный, так и десктопный варианты). Однако при сборке дистрибутива возникают некоторые задачи, решить которые средствами самого ABF достаточно сложно в силу его универсальности и нацеленности на поддержку многих платформ. Например, нам интересны замкнутость репозитория ROSA по зависимостям, сравнение набора пакетов ROSA с родственными дистрибутивами и многое другое.

Для сбора и публикации подобной статистики мы разработали отдельный сайт - <http://fba.rosalinux.ru/>. FBA производит мониторинг репозитория ROSA, размещенных на ABF и их анализ с помощью инструментов `urpm-tools` и [Upstream Tracker](#).

В настоящее время доступны следующие виды отчетов:

- замкнутость репозитория по зависимостям;
- сравнение версий пакетов с родственными дистрибутивами и с апстримом;
- наличие в репозиториях устаревших пакетов;
- отчеты ROSA Package Popularity о наиболее используемых пакетах;
- отчеты Upstream Tracker об обратной совместимости различных версий библиотек.

## **Доступна первая версия консольного клиента для ABF**

Основным способом взаимодействия с ABF до последнего времени являлся веб-интерфейс. Однако несмотря на красоту и богатые функциональные возможности, использование веб-интерфейса ABF не всегда удобно - например, если нужно отправить на сборку сразу несколько пакетов поочередно, необходимо посетить страницы каждого из них, нажать там кнопку «New Build», выбрать целевую платформу и совершить ряд сопутствующих операций.

Для решения таких задач больше подходит консольный клиент, который можно использовать в различных скриптах. Благодаря появлению API ABF, создание такого клиента стало возможно, и в настоящее время активно ведется его разработка.

Первой версией уже можно пользоваться - она предоставляет базовую функциональность по работе с пакетами ABF.

Установить пакет `abf-console-client` можно отсюда:

(32 бит)

[http://abf.rosalinux.ru/downloads/akirilenko\\_personal/repository/rosa2012lts/i586/main/release/](http://abf.rosalinux.ru/downloads/akirilenko_personal/repository/rosa2012lts/i586/main/release/)

(64 бит)

[http://abf.rosalinux.ru/downloads/akirilenko\\_personal/repository/rosa2012lts/x86\\_64/main/release/](http://abf.rosalinux.ru/downloads/akirilenko_personal/repository/rosa2012lts/x86_64/main/release/)

О том, какие действия можно совершать с помощью клиента, можно узнать, запустив `"abf -h"`. Для каждой команды можно получить более подробную справку, выполнив `"abf help <имя_команды>"` (например, `"abf help build"`).

В ближайшем будущем функциональность клиента будет постоянно расширяться и дорабатываться, а документация по его использованию станет подробнее.



## 2Safe

### Простой клиент 2Safe.com. Обзор API

Вот уже почти полтора года как сервис облачного хранения [2Safe](#) работает в режиме тестирования, и очень не многие знают о том, что этот сервис умеет. Попробуем разобраться по порядку. Текст рассчитан на узкий круг специалистов, и в нем преобладают специфические термины и примеры кода. В общем, Developers only.

Сам по себе API представляет из себя набор функций сервиса, при помощи которых можно полноценно с сервисом работать. В том числе выполнять всяческие проверки, регистрировать и удалять учетные записи, управлять файлами и многое другое. Полный список всех функций представлен в нашей статье на wiki - [2Safe\\_API](#).

А мы попробуем неспешно написать простенький консольный клиент. Пример может оказаться вполне полезным, скажем, для ~~скриптования~~ автоматизации наиболее часто выполняемых операций.

Сразу договоримся, что писать мы будем на [Perl](#) и будем использовать библиотеку [perl-lib2safe](#). Библиотека доступна в репозитории contrib ROSA 2012 Marathon и ROSA 2012 Desktop.

Итак, устанавливаем:

```
$ sudo urpmi perl-lib2safe
```

В библиотеке самую малость изменены названия функций из оригинального API, но они не перестали быть понятными. Например `list_dir` -> `ls`, `copy_file` -> `cp` и т.д. Читаем справку:

```
$ man lib2safe
```

Открываем наш любимый текстовый редактор:

```
$ vi test2safe-client.pl
```

И пишем мега программу:

---

```
#!/usr/bin/perl

use lib2safe;

my $psync = new lib2safe (URL => 'https://api.2safe.com/'); #Кстати можно и без
параметров вовсе. https://api.2safe.com/ есть дефолтовое значение урла.

my $mylogin = $psync->login(login=>'mylogin', password=>'*****'); #Вот так вот
просто авторизовываемся уже зарегистрированным ранее пользователем.

#Проверяем статус авторизации
die('Error Authorize') unless
    ($$mylogin{response}{success} && $$mylogin{response}{success} eq 'true');

my $ls = $psync->ls(); #без параметров просматривается содержимое корневой
директории авторизованного пользователя.

my $new_dir = $psync->mkdir(dir_id => $$ls{response}{root}{id}, dir_name =>
'new_dir'); #Создали в корне новую папочку с именем "new_dir"

my $testfile = $psync->put(dir_id => $$new_dir{response}{dir_id}, file =>
"test2safe-client.pl", name => '2safe-client.pl'); # А тут заливаем во вновь
```

созданную папочку наш скрипт но под другим именем.

---

Ради справедливости следует заменить, что в примере приведены лишь несколько функций. Всего в библиотеке их реализовано более 40, перечислю лишь разделы:

- Пользовательские функции.
- Функции по работе с файловой системой.
- Функции по работе с блокировками.
- Функции расшаривания и публикации объектов.
- Функции по работе с версионностью.

Также замечу, что официальный сайт проекта использует точно этот же список функций. То есть, разработчики сайта не используют более никаких вызовов к сервису, кроме перечисленных в документе. Вы можете с теми же усилиями создать, например, собственный сайт с аналогичным функционалом с минимальными затратами.

Да простят меня перлокодеры 80+ лвл за мой французский.

\*\*\*

- У разработчика нет денег на интернет? Давайте сделаем ему кнопочку "Donate"!

--- С совещания Технического Комитета «РОСЫ»

\*\*\*

## Историческая ретроспектива

### **Подготовка релиза ROSA/Mandriva 2011: как это было**

Возможно, вас улыбнут рабочие будни скромных тружеников grmbuild'a и Kenobi, а также некоторые проблемы, с которыми мы сталкивались на заре существования нашей компании (вероятны ошибки в сроках, но мы с Сашей Бурмашевым старались все вспомнить тщательно). Просьба отнестись к тексту как к исторической исповеди смеющихся над «собой в прошлом» людей. Сейчас уже, к счастью, многие вещи и полученный опыт позволили выстроить процессы более правильным образом и диверсифицировать риски. Но тогда было «весело» :).

Итак:

1. Перед Technology Preview Женя Додонов и Саша Бурмашев долго не могли собрать образы, потому что наши коллеги первое время работали «ща все новое загрузим, а что сломается починим». Сломалось почти все, что можно...
2. Перед alpha1 умерли 2 узла в кластере сборочной системы, и некоторое время потребовалось на восстановление всего процесса. Сборка новых пакетов шла с перебоями.
3. Перед beta1 «упал» сервер репозитория, 2 недели не было доступно никаких обновлений. Разработчики делали все «вслепую». В дальнейшем, практически перед каждым релизом, в самый последний момент кто-то из разработчиков загружал в репозитории что-то, что ломало всю сборку.
4. Перед beta2 «упал» сервер репозитория и пропали все наработки за последние 2 дня (самое большое количество важных изменений было загружено именно в эти 2 дня...).

Вдобавок ко всему, чтобы квест казался нам интересней, вообще «упал» сам провайдер, к

которому был подключен наш дата-центр во Франции. Это еще не все. В связи с тем, что скорость скачки iso образов в тот момент у Жени Додонова была на уровне 12-15 Кб/сек (образы приходилось качать из Франции), мы их выложили без проверки - образ 64-bit оказался нерабочий...

Между beta2 и beta3 загрузили новый gpm в репозитории, который не мог устанавливать пакеты. Для того чтобы это починить, нужно было загрузить старый gpm, а как это было сделать если текущий сломан?

5. Перед beta3 один хороший человек загрузил новый binutils и поломалась сборка образов, никто не мог понять, почему.

6. Перед RC1 просто нереальными усилиями всей команды был загружен весь планируемый функционал, но тестирования не было, поэтому образ вышел даже более «забагованным», чем beta3. В образ также попала Sphere, которая мешала перезагружать компьютер.

7. Перед RC2 ушел Женя Додонов, и создали ветку 2011, но «недосоздали» - в нее по-прежнему попадали новые пакеты, а старые не удалялись. Для её очистки Bogdano написал мега-скрипт, который удалял старые версии пакетов. Проблема была всего в одном лишнем « - », которое добавил Пер Ойвинд в пакетный менеджер RPM5...

8. Перед самым релизом (за день до) умер еще один сервер (n4) в сборочном кластере. Вся сборочная система полностью «встала». Релизы вышли с одним подложенным вручную пакетом.

Но тем не менее релиз был успешно выпущен! Большинство указанных проблем починил Bogdano Arendartchuk - бразилец, с украинскими корнями.

## **Авторы выпуска**

Александр Бурмашев

Владимир Шаршов

Денис Корявов

Денис Силаков

Сергей Соколов

Антон Чернышов

Константин Кочерешкин