Визитка

ДЕНИС СИЛАКОВ, к. ф.-м. н., старший архитектор ЗАО «РОСА», занимается автоматизацией разработки ОС «РОСА»

Что такое дистрибутив Linux?

Разработка дистрибутива Linux на примере РОСЫ

У многих людей, не очень знакомых с миром свободного ПО, слово «Linux» ассоциируется прежде всего с полноценной операционной системой — аналогом Windows, Mac OS X и других. Однако изначально слово «Linux» обозначало только ядро — необходимую часть ОС, но далеко не достаточную для полноценной работы с ней

Для обозначения всей системы более правильно использовать слово «дистрибутив». Дистрибутив Linux – это ОС, основанная на ядре Linux, библиотеках и инструментарии проекта GNU (таких как GCC и Glibc), графической подсистеме X Window System и большом наборе других программных компонентов (как правило, свободных и открытых, но это не обязательно).

Типичный современный дистрибутив Linux общего назначения состоит из нескольких тысяч приложений, утилит и библиотек, большая часть которых разрабатывается независимо от дистрибутива. Один дистрибутив от другого отличают уникальные компоненты, а также состав и версии сторонних продуктов.

Основные задачи создателей дистрибутива – это собрать воедино несколько тысяч компонентов, обеспечить их согласованную совместную работу и добавить ряд элементов, которые будут отличать данную систему от остальных.

Такой набор уникальных компонентов может включать в себя как приложения, так и наработки «непрограммного» характера – например, оригинальные дизайнерские решения. Поскольку создание чего-то нового обычно требует больших усилий, чем адаптация существующего ПО, то для большинства систем размер уникальных компонентов не превышает 1% от общего размера кода ОС.

Отбор и согласование сторонних компонентов, входящих в дистрибутив, также имеет свои сложности. Большинство из них обусловлено тем, что многие программы для Linux придерживаются политики частых релизов (release early, release often), в результате чего на свет появляется много версий одной и той же программы.

Среди этих версий необходимо выбрать наиболее подходящую для дистрибутива – самую стабильную либо наиболее функциональную, либо лучше других взаимодействующую с остальными компонентами ОС.

Далеко не всегда есть версия, удовлетворяющая сразу всем таким критериям. Более того, создателям дистрибутива нередко приходится дорабатывать сторонние продукты самостоятельно, чтобы они корректно функционировали в их системе.

В частности, среднестатистическое приложение требует для своей работы около десятка библиотек. Версии библиотек не всегда совместимы друг с другом, так что программа, работающая с одной версией, может отказаться работать с другой. Подобрать правильное сочетание версий библиотек и приложений – не очень простое занятие в ситуации, когда счет программных компонентов идет на тысячи.

Иными словами, разработка дистрибутива Linux – это непростое, но достаточно увлекательное занятие, подразумевающее решение множества задач различной природы и сложности.

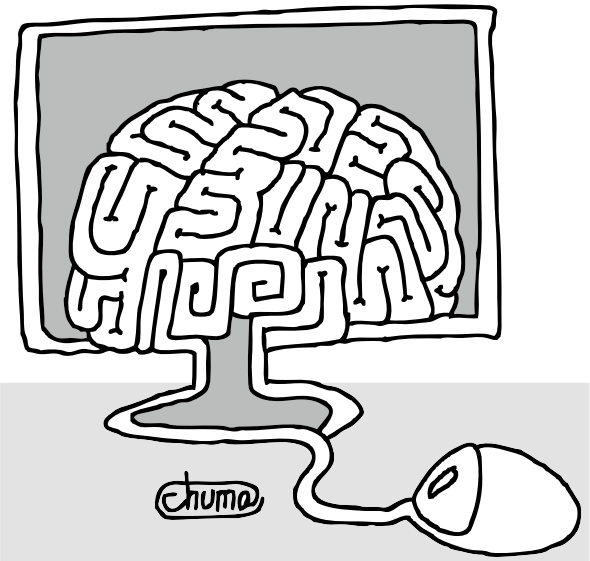
Далее в этой статье мы рассмотрим организацию процесса разработки и сопровождения семейства дистрибутивов РОСА для рабочих столов. Будут освещены вопросы организации разработчиков и сообщества, представлены основные направления работ, выполняемых непосредственно разработчиками РОСЫ, а также дан обзор ряда технических вопросов, связанных с инфраструктурой разработки и поддержки ОС.

Управление разработкой

В разработке большинства Linux-систем существенная роль принадлежит сообществу – добровольцам и энтузиастам, трудящимся на благо дистрибутива в свободное от своей основной работы время.

Одна из задач создателей дистрибутива – организовать работу таких волонтеров и подсказать, в каких областях их усилия будут наиболее востребованы, ведь многие люди приходят просто с желанием помочь, не зная толком, чем именно. К тому же при работе большой команды необходим некоторый способ консолидированного принятия решений и разрешения конфликтов, чтобы не возникало анархии и ситуаций, когда разные части системы развиваются несогласованно.

В процессе разработки настольных дистрибутивов РОСЫ подобные ключевые роли в принятии решений принадлежат сотрудникам одноименной компании. Именно из их числа формируются два основных органа, управляющих развитием ОС, – Board и Технический комитет.



В разработке большинства Linux-систем существенная роль принадлежит сообществу — добровольцам и энтузиастам

Board

Во главе иерархии управления разработкой РОСЫ стоит так называемый Board – некоторый аналог совета директоров в коммерческих компаниях. Board – это команда из нескольких человек, которая осуществляет стратегическое руководство.

В сферу ответственности команды входят:

- стратегическое планирование разработки дистрибутива (постановка целей, определение бизнес-задач, не связанных с техническими вопросами) и донесение этой информации до разработчиков;
- обеспечение взаимодействия между руководством компании РОСА и коллективом разработчиков (ведь разработка РОСЫ финансируется одноименной компанией, интересы которой необходимо учитывать в первую очередь);
- назначение членов Технического комитета и регулирование его работы;
- разрешение конфликтов, которые не могут быть разрешены на более низких уровнях, и принятие решений, касающихся разработки дистрибутива, но не входящих в сферу ответственности кого-либо еще.

Технический комитет

Технический комитет (ТК) – также относительно небольшая команда (не более десяти человек, состав может варьироваться время от времени), осуществляющая контроль над всеми техническими аспектами разработки дистрибутива. ТК формируется из архитекторов и руководителей основных групп и направлений разработки.

Целью работы комитета является обеспечение процесса регулярного выпуска релизов дистрибутива, поддержки выпущенных релизов и процесса разработки в рамках задаваемых Board направлений развития и имеющихся ресурсов.

При необходимости к работе комитета привлекаются разработчики и аналитики, а наиболее активные участники разработки приглашаются на заседания ТК для обсуждения вопросов, в которых они являются экспертами.

Такая дополненная рядовыми разработчиками команда получила название Расширенного технического комитета. Следует отметить, что решения принимаются только основными членами ТК, остальные участники выступают в роли консультантов.

Официальные решения по тем или иным вопросам принимаются простым голосованием на заседаниях ТК, проходящих раз в неделю.

Ввиду физической удаленности членов ТК друг от друга, заседания эти являются чисто «виртуальными», то есть проводятся с использованием средств электронной коммуникации (как правило, используется формат видеоконференции).

Перед каждым заседанием среди Расширенного технического комитета публикуется повестка из тех ранее выставленных на общее обсуждение вопросов, по которым на этом заседании планируется принять решение.

После заседания все принятые решения публикуются среди расширенного комитета, и с этого момента они являются обязательными для выполнения всеми разработчиками дистрибутива.

В сферу ответственности ТК входит:

- принятие технических решений о всевозможных аспектах разработки дистрибутива в рамках, заданных Board. В частности, ТК занимается выработкой требований к сборочной системе, ратификацией расписаний релизов, принятием решений о включаемых в очередной релиз новшествах и версиях базовых компонентов (ядра, компилятора, системных библиотек), на которых будет основан релиз;
- делегирование части своих функций по техническому руководству путем создания команд и назначения ответственных по определенным вопросам;
- своевременное донесение до Board вопросов и проблем, требующих вмешательства последней;
- разрешение конфликтов, которые не могут быть разрешены на более низком уровне – как правило, это различные технические трения между разработчиками и мэйнтейнерами.

В организационном плане ТК является саморегулирующим органом – члены комитета сами занимаются выдвижением новых членов взамен неактивных или не справляющихся с обязанностями, а также составляют регламент своей работы (но, естественно, в эти процессы могут вмешиваться и представители Board).

Итак, Технический комитет дистрибутива сформирован из руководителей ключевых направлений его разработки. Они представляют на встречах точку зрения своих направлений и стараются вырабатывать компромиссные решения, устраивающие все команды.

Среди представителей ТК обязательно присутствует человек (или несколько людей), осуществляющий общую координацию работ по подготовке и поддержке конкретных версий (релизов) системы – релиз-менеджер.

Релиз-менеджмент

Релиз-менеджер заведует созданием и поддержкой конкретных версий дистрибутива. В область его ответственности входит:

- составление расписаний жизненного цикла релиза, содержащих время выпуска предварительных версий. Количество промежуточных релизов может варьироваться; ниже будут рассмотрены критерии, которым должен удовлетворять тот или иной промежуточный выпуск;
- обеспечение своевременной сборки образов дистрибутива;
- контроль над целостностью дистрибутива при обновлении пакетов в стабильном релизе начиная с момента его ответвления от нестабильной ветки разработки и заканчивая моментом прекращения поддержки;
- координация действий команд разработчиков при устранении ошибок;
- контроль над правами доступа конкретных разработчиков к обновлению конкретных пакетов, включенных в стабильную ветвь выпущенного или готовящегося релиза;
- поддержание баланса между необходимостью внесения исправлений и контролем целостности дистрибутива.

Отметим, что приблизительные сроки выпуска очередной версии задаются членами Board, но окончательную «отмашку» на выпуск дает именно релиз-менеджер. «Отмашка» дается на основе отчетов всех команд, участвующих в разработке.

Давайте посмотрим, какие основные направления (и соответствующие команды) присутствуют в процессе создания РОСЫ.

Ключевые направления работы

Как мы уже отметили в начале статьи, основными задачами разработчиков дистрибутива является создание собственных компонентов и интеграция в ОС продуктов стороннего производства. Главное направление собственных разработок в РОСЕ лежит в области графического интерфейса пользователя, и этим занимается отдельная команда – UXTeam.

Также в создании дистрибутива принимают участие системные программисты, занимающиеся как созданием ПО, непосредственно входящего в ОС, так и разработкой вспомогательного инструментария, используемого другими командами.

В РОСЕ существуют отдельная команда, отвечающая за своевременные обновления, связанные с безопасностью, группа контроля качества системы, а также команда локализации, занимающаяся переводом на разные языки различных частей приложений (в первую очередь элементов графического интерфейса).

Наконец, интеграция и адаптация стороннего ПО осуществляется силами мэйнтейнеров. Именно к этой группе относится наибольшее число добровольных помощников (хотя поддержка основных компонентов, критичных для функционирования системы, проводится сотрудниками компании). С рассмотрения этой самой многочисленной команды мы и начнем.

Мэйнтейнеры

Мэйнтейнер – это человек, отвечающий за сборку той или иной программы для дистрибутива и ее адаптацию к правилам системы.

Эти правила различаются в разных дистрибутивах и затрагивают такие аспекты, как расположение файлов в файловой системе, установку нескольких версий одной и той же библиотеки и многое другое.

Полный перечень таких правил для РОСЫ можно найти по ссылке – http://wiki.rosalab.ru/en/index.php/Category:Packaging_Policies.

Помимо адаптации программ, к основным задачам мэйнтейнера относятся:

- отслеживание выхода новых версий ПО и их сборка для дистрибутива;
- ответы на вопросы пользователей программы и реакция на сообщения в системе отслеживания ошибок;
- отслеживание обновлений, связанных с устранением безопасности (совместно с Security Team);
- взаимодействие с непосредственными разработчиками ПО – донесение до них пожеланий пользователей дистрибутива, отслеживание новостей и тенденций развития программы, помощь в разработке и исправлении ошибок.

На первый взгляд может показаться, что труд мэйнтейнера содержит достаточно много рутинных действий. Однако в РОСЕ используется ряд инструментов, позволяющих уменьшить рутину и сосредоточиться на более интересных задачах. Подробнее об этих инструментах мы расскажем в следующей части статьи.

Отметим, что интеграция и адаптация многих программ не требует больших усилий. Именно с поддержки таких программ обычно и начинают свой путь как разработчики, так и мэйнтейнеры дистрибутива.

Подобные несложные задания позволяют им познакомиться с инфраструктурой и инструментами, используемыми в процессе создания системы, и в то же время сразу внести некоторый вклад в разработку.

Команда User Experience (UXTeam)

Основной задачей Команды UX является улучшение процесса взаимодействия конечного пользователя с дистрибутивом.

В число задач команды входит:
- реализация фирменного стиля дистрибутива в темах оформления и пользовательских приложениях;
- отслеживание и систематизация пожеланий и идей пользователей по улучшению дистрибутива и сопутствующего ПО, доведение этой информации до ТК в сроки, достаточные для учета этих пожеланий при составлении расписания подготовки нового релиза;
- анализ нововведений и изменений в дистрибутиве с точки зрения UX;
- определение списка важных для пользователя приложений, на тестирование которых нужно обратить особое внимание.

Именно UXTeam разрабатывает основные уникальные компоненты РОСЫ, видимые конечным пользователям, – медиапроигрыватель ROSA Media Player, панель запуска приложений RocketBar, быстрый просмотрщик файлов всевозможных форматов Klook и другие.

Безопасность

Обеспечение безопасности дистрибутива в целом и поставляемого в его составе ПО является задачей Команды безопасности – Security Team.

В обязанности команды входит:
- выявление и отслеживание проблем с безопасностью, обнаруженных сторонними экспертами, в частности, отслеживание сообщений систем наподобие CVE (Common Vulnerabilities and Exposures, общедоступная система публикации известных уязвимостей в различных программных продуктах);
- оперативный поиск и создание патчей, исправляющих проблемы с безопасностью, привлечение к работе по исправлению проблем с безопасностью мэйнтейнеров соответствующих пакетов;
- определение важности и корректности обновлений для уже выпущенных релизов с точки зрения безопасности;
- консультация администраторов информационных систем разработки дистрибутива по вопросам безопасности.

Запросы на обновления, устраняющие проблемы с безопасностью, заводятся членами Security Team в системе учета ошибок РОСЫ, однако такие запросы являются приватными и видны только сотрудникам с соответствующими правами.

Такой подход является типичной практикой, применяемой во многих командах, ведь публикация сообщения о наличии уязвимости в ОС до появления исправления явным образом известит потенциальных злоумышленников о том, как можно атаковать уже установленные экземпляры этой ОС.

Контроль качества

Контроль качества дистрибутива и входящих в него продуктов осуществляет команда контроля качества (QA – Quality Assurance).

В число ее обязанностей входит:
- тестирование новых версий приложений и обновлений для уже выпущенных релизов;
- тестирование образов дистрибутива, предоставленных релиз-менеджером;
- установка приоритетов сообщений об ошибках;
- отслеживание того, чтобы все сообщения пользователей о дефектах были либо разрешены, либо занесены в систему отслеживания ошибок для разработчиков и прикреплены к конкретным командам; также при поступлении сообщения об ошибке команда QA должна определить, в каких из поддерживаемых версий дистрибутива имеется этот дефект.

В РОСЕ используются две системы сообщения об ошибках. Во-первых, это HelpDesk (http://support.rosalab.ru), через которую осуществляется официальная поддержка пользователей. Как правило, в HelpDesk с пользователями общаются только представители команды QA. Не секрет, что многие отчеты об ошибках в первичном виде выглядят просто как «у меня неожиданно перестала работать моя любимая программа».

Обычно такой информации недостаточно для локализации и исправления ошибки, и в обязанности QA входит получение дополнительных сведений, необходимых разработчикам (значений переменных среды, настроек приложений и системы, получение снимка образа памяти упавшей программы и тому подобного).

С такими сведениями представители QA идут в следующую систему отслеживания ошибок, за которой уже следят разработчики. В роли такой системы в РОСЕ в настоящее время используется Bugzilla (http://bugs.rosalinux.ru).

Предполагается, что в этой системе заводятся уже более осмысленные отчеты, содержащие достаточно информации для локализации проблем и поиска путей их разрешения. Bugzilla РОСЫ открыта, и заводить в ней отчеты об ошибках может любой желающий (после предварительной регистрации).

Однако делать это рекомендуется только опытным пользователям, способным общаться с разработчиками по техническим вопросам без посредничества команды QA.

Отметим, что, несмотря на свое название, системы отслеживания ошибок используются и для запросов на добавление новой функциональности.

Локализация

РОСА является системой с российскими корнями, и вполне естественно, что повышенное внимание при разработке системы уделяется переводу ее различных компонентов на русский язык. Не остаются забытыми и другие языки, но их поддержка отдается на откуп сообществу, в то время как поддержка русской локализации осуществляется отдельной командой из сотрудников компании.

Для подготовки переводов члены команды используют популярную систему Transifex, развернутую на сайте http://translations.rosalinux.com.

Здесь осуществляется перевод собственных разработок РОСЫ, а также ряда ключевых программ (в частности, компонентов графической среды KDE). В случае стороннего ПО готовые переводы передаются непосредственно разработчикам соответствующих программ.

Подготовка релиза

Новые версии РОСЫ выходят с определенной периодичностью. Основная работа над новой версией начинается сразу после выхода предыдущей.

В течение первого месяца после выхода релиза уточняется расписание подготовки следующего. При планировании учитываются пожелания Board, опыт подготовки предыдущих релизов, а также предварительно оценивается сложность предстоящих работ.

В частности, разработчики предлагают новшества, которые можно включить в следующий релиз. Предложения сохраняются в публичной wiki РОСЫ (http://wiki.rosalab.ru). Новшества анализируются и одобряются или отклоняются ТК.

Новые предложения поступают на разных этапах разработки, однако в расписании подготовки релиза устанавливаются крайние сроки, по истечении которых изменение тех или иных компонентов возможно только по указанию Board или в связи с невозможностью исправить критический дефект иным способом.

Расписания подготовки релизов могут отличаться друг от друга, но в целом они придерживаются одного и того же шаблона и подразумевают выпуск нескольких промежуточных версий.

Примерно за неделю до предполагаемой даты выпуска очередной версии вносится запрет на добавление изменений, не исправляющих какой-либо критический дефект. После стабилизации компонентов и достижения требований к выпускаемой версии релиз-менеджер предоставляет команде QA собранный iso-образ промежуточной версии и уточненные требования к ней (в частности, указывается, на какие аспекты необходимо обратить особое внимание при тестировании, а какие ошибки уже известны и планируются к исправлению в следующих версиях).

После этого команда QA тестирует образ, заносит выявленные недостатки в Bugzilla, выставляя им приоритеты сообразно требованиям к версии. По результатам тестирования, члены ТК проводят совещание о возможности выпуска данной версии. В случае если версию решено выпускать, в запланированную дату образ публикуется на основных зеркалах, после чего выпускается соответствующий анонс на сайте компании.

Как правило, все промежуточные версии являются публичными, так что в их тестировании могут принять участие и члены сообщества.

Техническое превью

Первой контрольной точкой при подготовке релиза является выпуск Технического превью (Technical Preview) по истечении нескольких месяцев после старта разработки.

Для выпуска превью пакетная база должна быть полностью пересобрана с использованием тех версий инструментария и системных библиотек, которые планируется использовать в релизе. Эта версия составляется исключительно для внутреннего пользования, однако ее уже можно установить с iso-образа.

Зачастую разработчики устанавливают ее в виртуальную машину и уже в ней ведут дальнейшие работы по подготовке релиза.

Альфа-версия

К моменту выхода альфа-версии должен быть сформирован список версий ключевых системных компонентов, которые войдут в финальный релиз.

В этот список могут включаться как уже вышедшие компоненты, так и те, релиз которых намечен их разработчиками не позже, чем на срок второй бета-версии.

Среди основных требований к альфа-версии:

- возможность установить систему хотя бы одним способом;
- запустить ее в виртуальной машине4
- подключиться к сети;
- запустить графическую оболочку (возможно, вручную – неполадки с автоматическим запуском X-сервера на этой стадии не исключены);
- открыть веб-браузер и другие основные программы.

Первая бета-версия

К моменту выхода первой бета-версии должен быть сформирован список версий пользовательского ПО, которое войдет в финальный релиз. В этот список может включаться как уже вышедшее ПО, так и версии, релиз которых намечен его разработчиками не позже, чем на срок первого релиз-кандидата.

Бета-версия должна уметь устанавливаться и запускаться как в виртуальной машине, так и на реальном оборудовании и автоматически запускать графическую оболочку пользователя.

Вторая бета-версия

Во второй бета-версии должны присутствовать окончательные версии ключевых компонентов, но допускаются бета-версии пользовательского ПО.

Основные требования к версии – установка на различные устройства и аппаратные конфигурации и корректная работа системы в целом.

После выхода этой версии начинается массовое тестирование и работа над локализацией тех приложений, обновление которых не планируется (таких должно быть большинство).

Релиз-кандидат

В эту промежуточную версию должны войти окончательные версии всего ПО, которое планировалось к выходу в финальном релизе, должен быть закончен пользовательский интерфейс.

В этой версии допускается наличие одного или двух критических дефектов. За исключением исправлений серьезных и критических ошибок, никакие изменения в ПО после выпуска релиз-кандидата не допускаются.

При необходимости внести какое-то изменение обновленные программы должны проходить отдельную процедуру проверки QA и помещаться в специальный репозиторий для обновлений.

Финальный релиз

В большинстве случаев релиз-кандидат и становится релизом – возможно, с рядом незначительных изменений. Если в нем обнаруживаются какие-то проблемы, то либо эти проблемы решаются и выпускается еще один релиз-кандидат, либо они заносятся в список известных проблем релиза с возможными путями обхода.

В следующей части статьи будут рассмотрены технические аспекты разработки дистрибутивов РОСЫ. EOF