

RPM5 — основные достижения и планы на будущее

Денис Силаков, Евгений Буданов

Москва

ROSA Lab.

ROSA, RPM5

<http://www.rosalab.ru>

Аннотация

Отличительной чертой многих дистрибутивов Linux является оригинальный подход к управлению программным обеспечением, основанный на формировании пакетов ПО и использовании специализированных систем управления такими пакетами. Распространенным является мнение, что подобные системы - это всего лишь архиваторы, однако в современном мире это далеко не так. Доклад посвящен RPM5 - одной из активно развивающихся систем управления пакетами, являющейся ответвлением широко распространенного инструментария RPM. Приводится обзор нововведений, реализованных в рамках проекта RPM5, а также улучшений, которые планируется добавить в будущем.

История проекта RPM5 насчитывает уже несколько лет. В настоящее время этот формат и инструментарий используется в качестве основного в ряде дистрибутивов (в частности, уже более года — в ROSA и Mandriva), и уже можно судить о том, какие из наработок RPM5 оказались полезными на практике, и какие из запланированных нововведений кажутся наиболее интересными для разработчиков и пользователей дистрибутивов.

Улучшения для мантейнеров

Одним из наиболее активно используемых мантейнерами новшеств являются файловые триггеры. Механизм триггеров позволяет избавиться от указания в `post-` и `pre-`скриптах повторения рутинных действий, необходимых при установке или удалении многих пакетов — например, запуска `ldconfig` для регенерации кэша `/etc/ld.so.cache` при изменении состава библиотек или обновления кэша иконок при изменении файлов в `/usr/share/icons`. В RPM5 вместо этого предлагается использовать триггеры — скрипты, запускаемые самим RPM при появлении или удалении файлов с определенными именами или местоположением.

Еще одной активно прорабатываемой областью является локализация описаний пакетов. Существующие в настоящее время подходы (помещение переводов в `срес-`файл, использование отдельного пакета `среспро`) на практике оказались не очень удачными. Разработчики RPM5 предложили использовать для хранения описаний пакетов специализированное хранилище (базу данных), работа с которой будет производится непосредственно инструментарием RPM.

Потенциально полезным также видится использование параллелизма при сборке пакетов — например, использование параллельного сжатия их содержимого. Это позволило бы ускорить процесс сборки на многоядерных машинах. Впрочем, технически это не всегда просто — например, параллельные алгоритмы сжатия существуют для `gzip` и для `bzip2`, а вот реализации для становящимся все более

популярным хз пока нет.

Из других изменений, облегчающих работу мантейнеров, можно выделить следующие:

- Возможность использования встроенных интерпретаторов — RPM5 может быть собран со встроенной поддержкой Ruby, Python, Perl, Tcl и Lua, а также с базовой поддержкой ODBC и языка запросов SQL.
- Автоматизация рутинных действий при сборке пакетов, позволяющая существенно уменьшить размер спес-файлов и сделать их более читаемыми. В частности, мантейнерам предоставляется богатый набор вспомогательных макросов, автоматические генераторы зависимостей и прочие удобства.
- Улучшенная работа с сетью — практически все операции, работающие с локальными файлами, могут работать и с файлами по сети (по протоколам HTTP и FTP).

Отметим, что достаточно много наработок в области сборки пакетов специфичны для конкретных дистрибутивов и реализуются их разработчиками как надстройки к RPM5 либо как части сборочных систем дистрибутивов.

Улучшения для пользователей

Ряд нововведений RPM5 будет заметен и конечным пользователям. К таким аспектам относится, прежде всего, транзакционное управление пакетами, позволяющее представить установку пакета как атомарную транзакцию (по аналогии с транзакциями СУБД), которую, в случае необходимости, можно откатить.

Также пользователи могут заметить прирост производительности при установке пакетов на многоядерных машинах, где RPM5 способен выполнять параллельно несколько действий на разных ядрах процессора. В настоящее время распараллеливание используется при обработке групп пакетов (для проверки их подписей, контрольных сумм и тому подобного). Возможность использования параллелизма на более низком уровне (например, для разархивирования содержимого пакета — одного из наиболее затратных по времени действий) исследуется, однако внедрение подобных механизмов сопряжено с существенными техническими трудностями и требует тщательного тестирования.

Наконец, встраивание в RPM минимальных возможностей по работе с системами контроля версий позволяет использовать такие системы для отслеживания модификаций конфигурационных файлов при обновлении пакетов. В настоящее время RPM умеет сохранять предыдущую версию файла с суффиксом `.rpm_save` (либо оставлять старую версию на месте, устанавливая новую с суффиксом `.rpm_new`). Использование систем контроля версий позволит отслеживать все изменения в файлах, не замусоривая файловую систему. При этом пользователь получает возможность сравнивать различные версии и даже производить их слияние (`merge`) помощью RPM.

Итоги

Сегодня уже можно сказать, что RPM5 доказал свою состоятельность и пригодность к использованию в промышленных масштабах. Проект активно развивается, при этом наряду с собственными улучшениями, производится перенос полезных наработок из RPM4, что позволяет сохранять совместимость с этим форматом. За развитием RPM5 можно следить на сайтах <http://rpm5.org/> и <http://launchpad.net/rpm>. Процесс разработки полностью открыт и принять в нем участие могут все желающие.