

Бранчи и пакеты в Альте: от backport policy к disttag

Селезнев Владимир

Калуга, 2019



Sisyphus

- Sisyphus — проект, целью которого является развитие репозитория пакетов свободного ПО для удобной разработки на его основе дистрибутивов и других решений
- ALT Linux Team — международная, в основном русскоговорящая, команда разработчиков свободного программного обеспечения



Sisyphus

- Sisyphus — проект, целью которого является развитие репозитория пакетов свободного ПО для удобной разработки на его основе дистрибутивов и других решений
- ALT Linux Team — международная, в основном русскоговорящая, команда разработчиков свободного программного обеспечения

Бранчи

- Бранчи — стабильные ветки репозитория ALT
- Бранчи создаются путём форка Сизифа
- На основе бранчей создаются дистрибутивы и образы систем ALT

Пакеты

- Пакеты — типичная минимальная единица дистрибуции программного обеспечения в экосистемах Linux
- Помимо ПО в пакетах содержится метаинформация о пакете
- Часть метаинформации пакета предназначена для сохранения установленной системы в консistentном состоянии
- Над пакетами определены три операции: установка, удаление и обновление



Пакеты

- Пакеты — типичная минимальная единица дистрибуции программного обеспечения в экосистемах Linux
- Помимо ПО в пакетах содержится метаинформация о пакете
- Часть метаинформации пакета предназначена для сохранения установленной системы в консistentном состоянии
- Над пакетами определены три операции: установка, удаление и обновление

Пакеты делятся на два типа

- Исходные пакеты — содержат исходные коды, патчи, дополнительные файлы и инструкции сборки
- Бинарные пакеты — содержат откомпилированные программы, библиотеки и данные



Пакеты в ALT

- ALT использует пакеты формата rpm, управлением репозиториями на целевых системах и разрешением зависимостей занимается apt
- Бинарные пакеты можно разделить на два типа:
 - Архитектурно-зависимые
 - Архитектурно-независимые

Пакеты в ALT

- ALT использует пакеты формата rpm, управлением репозиториями на целевых системах и разрешением зависимостей занимается apt
- Бинарные пакеты можно разделить на два типа:
 - Архитектурно-зависимые
 - Архитектурно-независимые

Идентификация пакетов

RPM-пакет однозначно идентифицируется по
%name-[%epoch:]%version-%release, где:

- %name — имя пакета (N)
- %epoch — версия эпохи пакета в репозитории (E)
- %version — апстримная версия пакета (V)
- %release — версия сборки пакета (R)



Пример имён файлов пакетов

- `foo-1-alt1.src.rpm` — имя файла исходного пакета
- `foo-1-alt1.x86_64.rpm` — имя файла бинарного архитектурно-зависимого пакета
- `foo-doc-1-alt1.noarch.rpm` — имя файла бинарного архитектурно-независимого пакета



Пример имён файлов пакетов

- `foo-1-alt1.src.rpm` — имя файла исходного пакета
- `foo-1-alt1.x86_64.rpm` — имя файла бинарного архитектурно-зависимого пакета
- `foo-doc-1-alt1.noarch.rpm` — имя файла бинарного архитектурно-независимого пакета

- Из одного исходного пакета может быть собрано несколько бинарных пакетов (`foo`, `foo-doc`, `libfoo1`, `foo-devel` и т.д.)
- Раньше NEVRA бинарного пакета однозначно определял сборку этого пакета



Пример зависимостей

```
$ rpm -qR rpm
coreutils
popt >= 1.10.2.1
/bin/sh
/bin/bash
/bin/mv
/bin/rm
/bin/sh
/lib64/ld-linux-x86-64.so.2
/sbin/ldconfig
grep
libpopt.so.0()(64bit) >= set:jgtcU6BLBccTnteGxrE0
librpm7 = 4.13.0.1-alt12:sisyphus+234730.200.2.1
rpmlib(PayloadIsLzma)
```



Backport policy

Решаемые задачи

- Обновляемость: пакеты из одного бранча должны мочь обновиться до пакетов из более свежих бранчей
- Идентификация: однозначное определение, для какого бранча был собран пакет
- Наглядность: имя целевого бранча должен быть виден там, где виден номер релиза



Backport policy

Решаемые задачи

- Обновляемость: пакеты из одного бранча должны мочь обновиться до пакетов из более свежих бранчей
- Идентификация: однозначное определение, для какого бранча был собран пакет
- Наглядность: имя целевого бранча должен быть виден там, где виден номер релиза

Пример

foo-1-alt1 — пакет, собранный в Сизиф

foo-1-alt0.M80P.1 — пакет, собранный в p8



Современные проблемы бранчей

- Увеличилось число одновременно поддерживаемых бранчей
- Исчез линейный порядок между бранчами



Современные проблемы бранчей

- Увеличилось число одновременно поддерживаемых бранчей
- Исчез линейный порядок между бранчами

Что хотелось

- Собирать один и тот же релиз пакета в разные бранчи
- Пересобирать пакет без фактического изменения исходников ради обновления сборочных зависимостей



Бранчи в ALT

Современные проблемы бранчей

- Увеличилось число одновременно поддерживаемых бранчей
- Исчез линейный порядок между бранчами

Что хотелось

- Собирать один и тот же релиз пакета в разные бранчи
- Пересобирать пакет без фактического изменения исходников ради обновления сборочных зависимостей

Мейнтейнеры стали пытаться обходить эти проблемы



Современные проблемы бранчей

- Увеличилось число одновременно поддерживаемых бранчей
- Исчез линейный порядок между бранчами

Что хотелось

- Собирать один и тот же релиз пакета в разные бранчи
- Пересобирать пакет без фактического изменения исходников ради обновления сборочных зависимостей

Мейнтейнеры стали пытаться обходить эти проблемы

Необходим был способ однозначной различия различных сборок бинарных пакетов



Идентификация на основе хэшсуммы

- Некая контрольная сумма от неэффективной части заголовка rpm-пакета
- Строгие зависимости нового типа:
.hash-%name-%version-%release



Изменение идентификации бинарных пакетов

Идентификация на основе хэшсуммы

- Некая контрольная сумма от неэффективной части заголовка rpm-пакета
- Строгие зависимости нового типа:
.hash-%name-%version-%release

Строгие зависимости

```
$ rpm -qR foo-1-alt1
```

Было:

libfoo1 = 1-alt1

Стало:

.adc83b1-libfoo1-1-alt1



Изменение идентификации бинарных пакетов

Идентификация на основе хэшсуммы

- Некая контрольная сумма от неэффективной части заголовка rpm-пакета
- Строгие зависимости нового типа:
.hash-%name-%version-%release

Строгие зависимости

```
$ rpm -qR foo-1-alt1
```

Было:

libfoo1 = 1-alt1

Стало:

.adc83b1-libfoo1-1-alt1

Проблемы с такой схемой

noarch-подпакеты могут иметь строгие зависимости на arch-зависимые



Идентификация на основе информации сборочного задания

- Формировать строку на основе сборочного задания:
`branch.task.subtask.iter`
- Записывать эту строку вместо хэша в строгую зависимость
- Для наглядности записывать эту строку в RPM тег
`DistTag`

Изменение идентификации бинарных пакетов

Идентификация на основе информации сборочного задания

- Формировать строку на основе сборочного задания:
`branch.task.subtask.iter`
- Записывать эту строку вместо хэша в строгую зависимость
- Для наглядности записывать эту строку в RPM тег
`DistTag`

Проблемы

`apt` стал воспринимать такой вид строгих зависимостей как виртуальный, логика обработки их поменялась



Изменение идентификации бинарных пакетов

```
# apt-get install ntp
Reading Package Lists... Done
Building Dependency Tree... Done
Some packages could not be installed. This may mean
that you have
requested an impossible situation or if you are using
the unstable
distribution that some required packages have not yet
been created
or been moved out of Incoming.

...
The following packages have unmet dependencies:
ntp: Depends:
.sisyphus.213754.100.1.1-ntpd-4.2.8p12-alt1
E: Broken packages
```



Изменение идентификации бинарных пакетов

Новый формат более строгих зависимостей

```
$ rpm -qR foo-1-alt1
```

Было:

```
libfoo1 = 1-alt1
```

Стало:

```
libfoo1 = 1-alt1:sisyphus+123456.100.1.1
```

Изменение идентификации бинарных пакетов

Новый формат более строгих зависимостей

```
$ rpm -qR foo-1-alt1
```

Было:

```
libfoo1 = 1-alt1
```

Стало:

```
libfoo1 = 1-alt1:sisyphus+123456.100.1.1
```

Поломанная функциональность

Строчки из spec-файла:

```
Conflicts: foo > %{EVR}
```

```
Conflicts: foo < %{EVR}
```



Обновление пакетов внутри бранча

rpm в ALT учитывает buildtime:

```
# rpm -q foo
```

```
foo-1-alt1
```

```
# rpm -q -qf '%{buildtime} foo
```

```
123456789
```

```
# rpm -q -qf '%{buildtime} -p foo-1-alt1
```

```
234567890
```

```
# rpm -U foo-1-alt1
```

```
# rpm -q -qf '%{buildtime}
```

```
234567890
```

Проблема обновления бранчей

Проблемы

- Линейный порядок бранчей утрачен (пример: t7 и p7)
- У бранчей, имеющий линейный порядок ($p7 < p8$) при сборке в менее новый бранч buildtime получается больше, чем у более свежего



Через механизм apt_preferences(5)

```
$ cat /etc/apt/preferences.d/sisyphus
Package: *
Pin: release l=sisyphus
Pin-Priority: 1024
```

Через механизм apt_preferences(5)

```
$ cat /etc/apt/preferences.d/sisyphus
Package: *
Pin: release l=sisyphus
Pin-Priority: 1024
```

Недостатки

- Сильная изменение логики обновления
- Завязка на работу весом apt'a

Через механизм apt_preferences(5)

```
$ cat /etc/apt/preferences.d/sisyphus
Package: *
Pin: release l=sisyphus
Pin-Priority: 1024
```

Недостатки

- Сильная изменение логики обновления
- Завязка на работу весом apt'a

Отклонено

Через rpm

- Научить rpm учитывать DistTag



Проблема обновления бранчей

Итоговое решение

Был введён новый rpm-макрос `%_priority_distbranch`, который указывает на самый приоритетный бранч:

```
$ grep %_priority_distbranch /usr/lib/rpm/macros  
%_priority_distbranch sisypus
```



Спасибо за внимание!

