

Для управления пакетами в разных версиях SUSE как самое высокоуровневое средство используется Yast, который на самом деле использует zypper (а если еще точнее, то его библиотеку libzypp). Причем, обратите внимание на то, что вторая буква в его названии это "Y", а то почему-то его название многие порываются написать, как zipper. Zypper - средство для управления пакетами в текстовом режиме. С SUSE версии 11 (включая энтерпрайзовые версии) zypper существенно прибавил в скорости.

По данному теперь он легко уделывает yum (в отличие от yum zypper написан на C) и не уступает (по субъективному ощущению) apt. Синтаксис его конфигурационных файлов достаточно прост, например, чтобы управлять разными репозиториями не нужно ломать голову в отношении их приоритетов (это камень в огород apt). Также zypper достаточно «всеяден» в плане подключения разных репозиториях - он понимает:

- «родной» формат репозиториях yast;
- репозитории yum (rpm gpo-md);
- iso-образы репозиториях (да-да, не нужно их распаковывать!);
- локальный каталог с rpm-пакетами;
- то, что писать уже практически не обязательно - разные сетевые источники репозиториях — http, ftp, nfs.

Об остальных интересностях я расскажу по ходу дела.

С версии openSUSE 11.2 в zypper наконец-то была добавлена опция, которая давно в него просилась, а именно, опции предварительного выкачивания пакетов при обновлении. Ранее zypper работал так. Например, нужно обновить с десяток пакетов. Zypper выкачивал их все и устанавливал по одному. В принципе, ничего страшного. Если канал в сеть хороший и надежный. А это в наших широтах не всегда встречается. В основном конфигурационном файле /etc/zypp/zypp.conf данное поведение описывается опцией commit.download.mode (опция закомментирована по умолчанию), которая имеет следующие варианты:

- *DownloadOnly* — опция, которая легко заменяется ключом --dry-run, т. е. выкачивание всех пакетов необходимых для обновления без их установки.
- *DownloadInAdvance* — сначала выкачать все пакеты, требующие обновления, затем начать процесс их установки.
- *DownloadInHeaps* — опция аналогичная представленной выше, но в данном случае загрузка и установка пакетов выполняется «порциями», не нарушающими целостность системы. Примерно также ведет себя пакетный менеджер в Mandriva.
- *DownloadAsNeeded* — традиционное поведение. Загрузка и установка осуществляется по одному пакету.

Следующей интересной возможностью zypper является сокращенный вариант его опций, т. е. для установки пакета можно написать zypper install foopackage, а можно zypper in foopackage. Далее я буду приводить именно сокращенный вариант опций, а полный вариант писать в скобках.

Поиск пакетов

Для того, чтобы найти нужный пакет в подключенных репозиториях используется следующая команда:

`zypper se foopackage` — выполнить поиск (se - search) пакета foopackage.

Если вы хотите найти какую-то программу, но не знаете в каком пакете ее искать, можно сделать так:

```
booka:/etc/zypp # zypper wp vi
Loading repository data...
Reading installed packages...
S | Name | Type | Version | Arch | Repository
---+-----+-----+-----+-----+-----
| gvim | package | 7.2-16.7 | i586 | openSUSE-11.2-Oss
i | vim | package | 7.2-16.7 | i586 | openSUSE-11.2-Oss
| vim-enhanced | package | 7.2-16.7 | i586 | openSUSE-11.2-Oss
```

Опция `wp` (what-provides) позволяет искать пакет по любому возможному параметру: имя исполняемого файла, путь до исполняемого файла, любой набор символов из описания пакета. В данном случае я попросил найти пакет, содержащий редактор `vi`.

Еще примеры:

Попробуем поискать в каком пакете у нас находится библиотека `libpng`:

```
host13:~ # zypper wp libpng
Loading repository data...
Reading installed packages...
S | Name | Type | Version | Arch | Repository
---+-----+-----+-----+-----+-----
i | libpng12-0 | package | 1.2.39-2.2 | x86_64 | openSUSE-11.2-Oss
v | libpng12-0 | package | 1.2.39-2.2 | i586 | openSUSE-11.2-Oss
```

Интересно, а в каком пакете находится файл `/bin/bash`?

```
Host13:~ # zypper wp /bin/bash
Loading repository data...
Reading installed packages...
S | Name | Type | Version | Arch | Repository
---+-----+-----+-----+-----+-----
+-----+-----+-----+-----+-----
i | bash | package | 4.0-18.4.1 | x86_64 | Updates for openSUSE
11.2-0
v | bash | package | 4.0-18.4.1 | i586 | Updates for openSUSE
11.2-0
v | bash | package | 4.0-18.3 | x86_64 | openSUSE-11.2-Oss
v | bash | package | 4.0-18.3 | i586 | openSUSE-11.2-Oss
```

Первый столбик показывает статус пакета, где `i` означает - установлен.

Посмотреть информацию о пакете можно командой `zypper info <имя пакета>`

Установка и удаление пакетов

Установка пакетов из подключенных репозиториев выполняется командой:

```
zypper in foopackage
```

Данная команда (in - install) установит пакет foopackage со всеми его зависимостями.

Для удаления пакетов используется опция rm/remove:

```
zypper rm foopackage
```

Данная команда удалит пакет foopackage из вашей системы.

Для проверки целостности системы на предмет удовлетворения зависимостей существует команда verify:

```
zypper ve
```

Если чего-то не будет хватать - zypper предложит доустановить нехватящие пакеты (или даже удалить ненужные).

Также есть еще два ключа, которые не являются обязательными, но могут здорово помочь, если вы вызываете zypper из скриптов. Первый ключ «-у» заставляет пакетный менеджер отвечать на все вопрос «да»/«yes». Второй ключ «-l» (маленькая L) - имеет похожее значение, она заставляет zypper соглашаться с лицензионными соглашениями отдельных пакетов (например, таких как Adobe Flash).

Обновление

Выполнять обновление системы zypper позволяет двумя способами — на основе патчей и на основе пакетов. Первый способ рекомендуется для серверов. В данном случае производится только наложение патчей, исправляющих ошибки безопасности на установленное ПО, причем этот способ обновлений должен поддерживаться теми, кто ведет репозиторий, из которого вы обновляетесь. Стандартные репозитории openSUSE поддерживают данный способ. Второй способ выполняет установку новых пакетов в систему. Понятно, что в данном случае все определяется способом ведения репозитория майнтейнерами пакетов.

Стандартные репозитории openSUSE «замораживают» номера версий ПО, так что в принципе эти два способа равноценны (но только для них). Это все была теория. Теперь немного практики.

Обновить метаданные репозитория можно командой zypper ref и это необходимо делать всякий раз перед выполнением обновления (либо включить autorefresh для всех репозиториев - см.ниже).

Просмотреть список доступных патчей (list-patches) можно командой:

```
zypper lp
```

Просмотреть информацию о конкретном патче можно командой (про опцию -t ниже):

```
zypper info -t patch foopatch
```

Установить патчи можно командой `zypper patch` .

Обновление системы на основе пакетов выполняется командой:

```
zypper up [имя пакета]
```

Данная команда (`up/update`) выполняет обновление либо указанного пакета, либо всей системы.

Вторая команда предназначается для обновления системы между релизами (`dist-upgrade`):

```
zypper dup
```

Последняя команда имеет несколько интересных эффектов. Как известно все пакеты rpm имеют поле `Vendor`, в котором указан сборщик пакета. Пакеты из стандартных репозиториях имеют в данном поле `openSUSE` (или просто `SUSE` для энтерпрайзовых версий). Пакеты собранные на `openSUSE Build Service` имеют в данном поле слово `obs` с указанием вида репозитория. Так вот команда `zypper up` выполняет обновление таким образом, чтобы поле `Vendor` не менялось при обновлении. `zypper dup`, наоборот, может предпочесть изменить вендора пакета при обновлении.

Следующая интересная возможность касается тех, кто как я любит поэкспериментировать с системой. Допустим, вы хотите поставить последнюю версию KDE, подключаете репозиторий (об этом ниже) `KDE4:Factory` (данный репозиторий для разработчиков и тестеров), выполняете `zypper dup` (при этом `Vendor` меняется с `openSUSE` на что-то вроде `obs://build.opensuse.org/KDE/KDE4:Factory`) и получаете ее. Но потом, вы обнаруживаете, что в ней еще куча ошибок и вы хотели бы вернуться обратно. Что же делать? Неужели ничего нельзя поправить?!?! А ничего страшного! Убираете данный репозиторий (удаляете файл с его описанием или просто выключаете его) и опять выполняете `zypper dup`. При этом `zypper` вам предложит выполнить `downgrade` всех обновленных ранее пакетов. Т. е. `zypper dup` выполняет обновление системы таким образом, чтобы она всегда соответствовала подключенным репозиториям. Справедливости ради, стоит отметить, что `downgrade` не всегда проходит гладко. Иногда, при неблагоприятном положении звезд и планет, в системе остаются библиотеки от новых репозиториях, которые могут помешать работе программ. Так что возможно, придется затем позаниматься таким «увлекательным» занятием, как `troubleshooting`.

Управление репозиториями

Как отмечалось выше `zypper` всеяден в плане возможных репозиториях. Посмотреть что же у вас подключено в данный момент можно следующей командой:

```
host13:~ # zypper lr
```

В первой колонке приведен порядковый номер репозитория, во второй и третьей его название и описание. Четвертая и пятая колонки показывают включен ли данный репозиторий и включено ли его автообновление (`autorefresh`). Если последняя возможность включена, то при каждом своем запуске `zypper` будет проверять нужно ли обновление метаданных репозитория и, если нужно, выполнять его. В противном случае, вам нужно будет делать это собственноручно командой `zypper ref` (`refresh`).

Добавить репозиторий можно командой: `zypper ar URI alias`

```
host13:~ # zypper ar nfs://192.168.0.254/srv/ftp/sles11 sles11
Adding repository 'sles11' [done]
Repository 'sles11' successfully added
Enabled: Yes
Autorefresh: No
URI: nfs://192.168.0.254/srv/ftp/sles11
```

Удалить репозиторий можно командой `zypper rr`

```
host13: ~# zypper rr 13
Removing repository 'sles11' [done]
Repository 'sles11' has been removed.
```

Здесь я удалил репозиторий, указав его ID (то есть номер). Его можно увидеть в выводе команды `zypper lr`. Аналогичного эффекта я бы добился, указав `zypper rr sles11`. То есть в данном случае указывать нужно или ID репозитория, или его псевдоним.

Модификация параметров репозитория выполняется командой: `zypper mr [options]`. Список опций можно получить следующим образом:

```
host13:~ # zypper mr
Alias or an aggregate option is required.
modifyrepo (mr)
modifyrepo (mr)
```

Modify properties of repositories specified by alias, number, or URI, or by the
'--all, --remote, --local, --medium-type' aggregate options.

Command options:

- d, --disable Disable the repository (but don't remove it).
- e, --enable Enable a disabled repository.
- r, --refresh Enable auto-refresh of the repository.
- R, --no-refresh Disable auto-refresh of the repository.
- n, --name Set a descriptive name for the repository.
- p, --priority
- k, --keep-packages Enable RPM files caching.
- K, --no-keep-packages Disable RPM files caching.

- a, --all Apply changes to all repositories.
- l, --local Apply changes to all local repositories.
- t, --remote Apply changes to all remote repositories.
- m, --medium-type

Например, следующая команда включит параметр `autorefresh` для все репозиториях:

```
host13:~ # zypper mr -ra
```

Опция `-r` просит включить автообновление для репозитория (а `-R` выключит его), а опция `-a` говорит применить это ко всем репозиториям.

Репозитории могут иметь приоритеты, которые могут дополнительно указывать zypper ваши предпочтения (меньшее значение - больший приоритет). Работа с приоритетами аналогична тому, что происходит в yum. Только здесь не нужно ставить никаких дополнительных плагинов. Чтобы задать приоритет репозиторию можно воспользоваться командой zypper mg, но на мой взгляд, гораздо проще открыть файл .геро репозитория и дописать в нем, например, такую строчку - priority=100. Стандартные репозитории openSUSE имеют приоритет 90, а репозиторий Updates приоритет 20.

Имейте это в виду, когда будете задавать собственные приоритеты. После изменения приоритетов репозитория обязательно необходимо запустить, сначала zypper ref (если не включен autorefresh для репозитория), а затем zypper dup, для того, чтобы zypper установил пакеты в соответствии с высказанными вами предпочтениями.

Но, по-моему, гораздо удобнее для управления репозиториями использовать соответствующий модуль yast:

```
host13:~ # yast repositories
```

Любители графического интерфейса Yast могут воспользоваться либо его графическим меню, либо набрав в консоли:

```
host13:~ # yast2 repositories
```

Ну и самый простой способ подключения репозитория - это скачать файл с его описанием [отсюда](#). В каждом из репозитория есть текстовый .геро файл, например, для репозитория со свежими версиями Apache он лежит [здесь](#). Затем нужно поместить его в каталог /etc/zypp/repos.d/. Ну и отредактировать на предмет приоритетов, если это нужно.

Типы пакетов

Ну и самая интересная возможность zypper в том, что он позволяет использовать разные типы «пакетов» при установке. В данном случае под «пакетами» понимаются:

- собственно, пакеты (и если ничего не указывать специально, то имеются в виду именно они);
- патчи (patch) (репозитории могут содержать просто патчи, а не пакеты с обновлениями);
- шаблоны (pattern) - группы пакетов, устанавливающие ту или иную функциональность;
- продукты (product) - совсем редко встречающийся зверь - это группы пакетов для работы того или иного продукта;
- пакет с исходниками (srcpackage) - это обычный src.rpm.

Тип пакета указывается опцией -t . Например, получить список доступных шаблонов можно командой zypper patterns. И поставить тот, что нужен, командой:

```
host13:~ # zypper in -t pattern x11
```

Здесь pattern это тип устанавливаемого пакета, то есть мы указываем, что имеем в виду именно шаблон. x11 - имя устанавливаемого шаблона. Самое плохое при использовании шаблонов это то, что удалять пакеты шаблонами zypper пока не умеет.