

Оглавление

Глава 1. Пакетный менеджер	2
1.1 Основной пакетный менеджер в Альт Платформа	4
1.2 Система управления пакетами	4
1.3 Установка необходимых пакетов для процесса сборки	7
Глава 2. Основные команды пакетного менеджера	10
2.1 Просмотр недавно установленных пакетов	10

Глава 1

Пакетный менеджер

Операционная система состоит из разнообразных программ, библиотек, скриптов и приложений — число компонент может достигать тысячи единиц. В каждой из которых могут быть включены десятки файлов. Для удобства работы пользователя системные компоненты в Linux представлены в виде пакетов¹. Пакет объединяет в общий архив все файлы, используемые программой. Пользователь подбирает программы — устанавливает, обновляет, проверяет, удаляет их — только по имени пакета, не вдаваясь в отдельные детали подбора всех необходимых файлов. Работа с пакетом, как единым целым, позволяет управлять всеми данными программы.

Пакет — это специально подготовленный архив, содержащий программы, конфигурационные файлы, данные и управляющую информацию. Метаданные пакета содержат полное имя, номер версии, описание пакета, имя разработчика, контрольную сумму, зависимости от других пакетов. Управляющая информация пакета содержит контрольные суммы устанавливаемых файлов, зависимости устанавливаемого пакета от других пакетов, краткое описание, сценарии установки и удаления пакета, и прочую информацию, которую использует менеджер пакетов. Пакеты хранятся в специальном хранилище — **репозитории пакетов**.

Для удобства работы с пакетами придумали собственные форматы архивов:

- RPM (.rpm). Разработан компанией Red Hat. Применяется в системе Альт, Ред ОС, RHEL и CentOS.
- DEB (.deb). Формат пакетов дистрибутива Debian, а также Ubuntu.
- TAR.XZ. Применяется в дистрибутивах ArchLinux и Manjaro.

¹Курячий Г., Маслинский К. (2010). Операционная система Linux. Курс лекций. ДМК Пресс.

Каждый пакет определяется архитектурой системы (под которую он собран), именем программы, номером ее версии и номером релиза этой программы в дистрибутиве.

Например, `admc-0.15.0-alt1.x86_64.rpm`:

Имя:	<code>admc</code>
Номер версии:	<code>0.15.0</code>
Номер релиза:	<code>alt1</code>
Архитектура:	<code>x86_64</code>

Пакетный менеджер — это программа для управления установкой, удалением, настройкой и обновлением пакетов с различным программным обеспечением. Пакетные менеджеры упрощают для пользователя процесс управления программами в системе. Пакетный менеджер также ведет учет пакетов, установленных в системе. Существует также менеджер зависимостей — специальная программа, подбирающая пакеты, зависящие друг от друга, и загружающая эти пакеты из хранилища². Менеджер зависимостей подбирает правильные версии пакетов и определяет порядок их установки. При помощи менеджера зависимостей можно узнать, с каким пакетом поставляется тот или иной файл.

Задачи пакетного менеджера:

- **установка программ.** Позволяет устанавливать программы из центрального хранилища или из локальных источников;
- **обновление программ.** Позволяет обновлять установленные программы до последних версий, представленных в хранилище;
- **удаление программ.** Позволяет безопасно удалять программы и все связанные с ними файлы;
- **управление зависимостями.** Автоматически устанавливает и управляет зависимостями программ;
- **проверка целостности пакетов.** Предотвращает конфликты при установке новых программ, обеспечивая целостность системы.

Пакетный менеджер позволяет:

- узнать информацию о пакете;
- определить пакет, которому принадлежит установленная программа;
- определить список компонент, установленных из указанного пакета.

²Кетов Д. (2021). Внутреннее устройство Linux. 2-е изд., перераб. и доп. БХВ-Петербург.

Пакетные менеджеры делятся на две категории — низкоуровневые и высокоуровневые.

- **Низкоуровневые пакетные менеджеры.** Используются для установки локальных пакетов, загруженных вручную пользователем, или высокоуровневым пакетным менеджером.
- **Высокоуровневые менеджеры.** Применяются для поиска и скачивания пакетов из репозиториев. В процессе работы могут задействовать низкоуровневые менеджеры для установки загруженных программ.

В операционной системе **Альт** используется формат пакетов `.rpm`. Пакеты `rpm` хранятся в удаленном хранилище под названием **Sisyphus**. Для работы с такими пакетами применяется низкоуровневый пакетный менеджер `RPM` и консольная утилита `APT` (**A**dvanced **P**ackaging **T**ool).

1.1 Основной пакетный менеджер в Альт Платформа

В дистрибутивах **Альт** применяется пакетный менеджер `RPM`. **RPM Package Manager** — это семейство пакетных менеджеров, применяемых в различных дистрибутивах **GNU/Linux**. Практически каждый крупный проект, использующий `RPM`, имеет свою версию пакетного менеджера, отличающуюся от остальных.

Различия между представителями семейства `RPM` выражаются в:

- наборе макросов, используемых в `.spec-файлах`;
- различии сборки `rpm`-пакетов «по умолчанию» — при отсутствии каких-либо указаний в `.spec-файлах`, формате строк зависимостей;
- отличиях в семантике операций (например, в операциях сравнения версий пакетов);
- отличиях в формате файлов.

Для пользователя различия чаще всего заключаются в невозможности поставить пакет постороннего формата (например, `.deb`) из-за проблем с зависимостями другого формата пакета.

1.2 Система управления пакетами

Системой управления пакетами в дистрибутивах **Альт** является программа `APT` — **A**dvanced **P**ackaging **T**ool (усовершенствованный инструмент работы с пакетами). Программа `APT` способна автоматически устанавливать и настраивать программы в операционных системах **Альт** из предварительно откомпилированных пакетов и из исходных кодов. Утилита загружает пакеты из хранилища (репозитория), либо устанавливает с локального хранилища. Список

источников пакетов хранится в файле `/etc/apt/sources.list` и в каталоге `/etc/apt/sources.list.d/`. В системе Альт применяется графическая оболочка для `apt` — программа `Synaptic`. Утилита `apt` значительно упрощает процесс установки программ в командном режиме.

Команда `$ apt-get` выведет описание и возможности утилиты `apt-get`:

```
$ apt-get
apt 0.5.15lorg2 для linux x86_64 собран Jul 26 2023 18:10:41
Использование: apt-get [параметры] команда
apt-get [параметры] install|remove пакет1 [пакет2 ...]
apt-get [параметры] source пакет1 [пакет2 ...]
```

`apt-get` предоставляет простой командный интерфейс для получения и установки пакетов. Чаще других используются команды `update` (обновить) и `install` (установить).

Команды:

```
update - Получить обновлённые списки пакетов
upgrade - Произвести обновление
install - Установить новые пакеты
remove - Удалить пакеты
source - Скачать архивы исходников
build-dep - Установить всё необходимое для сборки исходных пакетов
dist-upgrade - Обновление системы в целом, см. apt-get(8)
clean - Удалить скачанные ранее архивные файлы
autoclean - Удалить давно скачанные архивные файлы
check - Удостовериться в отсутствии неудовлетворённых зависимостей
dedup - Remove unallowed duplicated pkgs
```

Параметры:

```
-h Краткая справка
-q Скрыть индикатор процесса
-qq Не показывать ничего кроме сообщений об ошибках
-d Получить пакеты и выйти БЕЗ их установки или распаковки
-s Симулировать упорядочение вместо реального исполнения
-y Автоматически отвечать 'ДА' на все вопросы
-f Попробовать исправить положение если найдены неудовлетворённые зависимости
-m Попробовать продолжить если часть архивов недоступна
-u Показать список обновляемых пакетов
-b Собрать пакет после получения его исходника
-D При удалении пакета стремиться удалить компоненты, от которых он зависит
-V Подробно показывать номера версий
-c=? Использовать указанный файл конфигурации
-o=? Изменить любой из параметров настройки (например: -o dir::cache=/tmp)
```

Более полное описание доступно на страницах руководства `man`:
`apt-get(8)`, `sources.list(5)` и `apt.conf(5)`.

В ОС Альт утилита `apt-get` использует основной пакетный менеджер RPM `Package Manager` — RPM для установки, обновления, удаления пакетов (программ), управления зависимостями. Обе утилиты RPM и APT одинаково позволяют установить, обновить или удалить пакет.

Отличия RPM и APT:

- APT учитывает зависимости устанавливаемого пакета;
- APT умеет работать с репозиторием в целом:
 - искать пакеты;
 - вычислять список обновлений — находить разницу версий пакетов, установленных локально и хранящихся в репозитории;
- APT получает информацию из пакетов, используя RPM.

RPM подразумевает работу с конкретными пакетами. Удовлетворение их зависимостей остаётся на усмотрении пользователя. APT позволяет вычислять, какие пакеты из репозитория нужно установить, чтобы удовлетворить зависимости, которые указаны в каждом конкретном RPM пакете. APT сам не устанавливает пакеты, он использует для этого RPM.

Примечание:

Установка пакетов в Альт Линукс происходит с помощью утилиты APT

Репозиторий — актуализируемое хранилище электронных данных. В данном документе речь идет о репозитории как об инфраструктуре разработки операционных систем, включающих, помимо системного ПО, любые программы пользовательского и серверного назначения («репозиторий пакетов»). Основная задача репозитория этого рода — интеграция разных пакетов программ в единую систему. Объектом хранения в таких репозиториях выступают пакеты программ, где каждое наименование ПО (будь то ядро операционной системы, служебная библиотека, текстовый редактор, сервер для обслуживания электронных сообщений или медиа проигрыватель) представлено в виде отдельного пакета. В операционных системах ALT репозиторий пакетов носит название **Sisyphus**.

Репозиторий пакетов — это множество исходных пакетов, собранных из них бинарных пакетов, и мета информация об этих пакетах.

Наличие репозитория пакетов и APT автоматизирует процессы управления установкой, обновления и удаления программного обеспечения; исключают риск случайного повреждения целостности операционной системы и прикладных программ.

Программа APT по запросу пользователя получает метаданные из репозитория, рассчитывает зависимости, получает от пользователя информацию о том, какие именно пакеты он хочет обновить или установить. Утилита предложит

пути решения — например, загрузит новые пакеты из репозитория, установит дополнительные или обновит имеющиеся пакеты. АРТ, в зависимости от настроек, может использовать удаленный репозиторий с помощью сетевого протокола (например, `ftp`), или использовать локальный репозиторий (например, на жестком диске).

Примечание:

Для обновления практически всего программного обеспечения (за исключением ядра операционной системы) на локальном компьютере до новой версии необходимо выполнить одну команду:

```
apt-get update
apt-get dist-upgrade
```

При использовании АРТ и обновляемого стабильного репозитория операционная система может функционировать на компьютере годами, плавно обновляясь до новых версий без переустановки системы.

1.3 Установка необходимых пакетов для процесса сборки

Сборка — формирование пакета на основе сборочных инструкций.

Сборка программного обеспечения в RPM упрощает распространение, управление и обновление программного обеспечения. Упаковка программного обеспечения в пакеты RPM, а не обычный архив, имеет ряд преимуществ³:

- Пользователи могут использовать средства управления пакетами (например, `Yum` или `PackageKit`) для установки, переустановки, удаления, обновления и проверки пакетов RPM.
- Пакетный менеджер RPM предполагает наличие базы данных, которая с помощью специализированных утилит позволяет получать информацию о пакетах в системе.
- Каждый пакет RPM содержит метаданные, описывающие компоненты пакета, версию, выпуск, размер, URL проекта, установочные инструкции и т. д.
- RPM позволяет брать оригинальные источники программного обеспечения и упаковывать их в пакеты с исходным кодом (`.src.rpm`) и бинарные пакеты (`.rpm`). В пакетах с исходным кодом хранятся оригинальные исходные данные вместе со всеми изменениями (`*.patch`), а так же сборочные инструкции (`.spec`) и дополнительная информация. В бинарных пакетах вместо исходного кода упакованы подготовленные файлы и скрипты установки, но нет сборочных инструкций. Еще существуют случаи распространения

³<https://rpm-packaging-guide-ru.github.io/#Why-Package-Software-with-RPM>

пакетов без исходного кода и бинарных данных, в таких пакетах присутствуют скрипты для скачивания и модификации файлов, необходимых для работы приложения.

- Для обеспечения верификации подлинности RPM-пакетов используется механизм электронных цифровых подписей GPG. Он позволяет подписать RPM пакет или обновить цифровую подпись: `rpm -addsign package.rpm` и `rpm -resign package.rpm`.
- Вы можете добавить свой пакет в RPM репозиторий, что позволит клиентам легко находить и устанавливать ваше программное обеспечение.

Задача сборки пакета начинается со сбора всех необходимых компонентов и завершается этапами сборки и тестирования.

Классическая сборка пакетов rpm состоит из следующих этапов⁴:

- поиск исходных данных;
- написание инструкций сборки;
- сборка пакета.

Примечание:

Для сокращения команд, встречающихся в тексте, будет использоваться нотация:

- команды **без административных привилегий** начинаются с символа "\$";
- команды **с административными привилегиями** начинаются с символа "#".

Примечание:

Необходимые инструменты для сборки rpm-пакетов устанавливаются в системе через пакетный менеджер apt командами:

```
# apt-get install gcc rpm-build rpmlint make
python gear hasher patch rpmdevtools
```

В наборе параметров команды `install` перечислены имена пакетов, необходимых для сборочной инфраструктуры⁵:

- `gcc` — набор компиляторов для различных языков программирования, разработанный в рамках проекта GNU;
- `rpm-build` — содержит сценарии и исполняемые программы, которые используются для сборки пакетов с помощью RPM;

⁴<https://alt-packaging-guide.github.io/>

⁵https://www.altlinux.org/Технология_сборки_пакетов_RPM

- **rpmlint** — инструмент для проверки распространенных ошибок в пакетах **rpm**. Можно проверить бинарные и исходные пакеты;
- **make** — инструмент GNU, упрощающий процесс сборки для пользователей;
- **python** — интерпретируемый интерактивный объектно-ориентированный язык программирования;
- **gear** — этот пакет содержит утилиты для сборки пакетов **RPM** из **GEAR**.репозитория и управления **GEAR**.репозиториями;
- **hasher** — современная технология создания независимых от сборочной системы пакетов;
- **patch** — программа исправлений применяет патчи к оригиналам;
- **rpmdevtools** — пакет содержит скрипты и файлы поддерживающие (X)**Emacs**, помогающие в разработке пакетов **RPM**.

Глава 2

Основные команды пакетного менеджера

2.1 Просмотр недавно установленных пакетов

Чтобы получить список последних установленных пакетов, введите следующую команду:

```
rpm -qa -last | head
```

Вывод:

smplayer-23.6.0-alt2.10169.x86_64	Пт 01 дек 2023 12:45:39
qt5-wayland-5.15.10-alt1.x86_64	Пт 01 дек 2023 12:35:44
qt5-tools-5.15.10-alt2.x86_64	Пт 01 дек 2023 12:35:44
qt5-sql-mysql-5.15.10-alt1.x86_64	Пт 01 дек 2023 12:35:44
qt5-dbus-5.15.10-alt2.x86_64	Пт 01 дек 2023 12:35:44
libqt5-xmlpatterns-5.15.10-alt1.x86_64	Пт 01 дек 2023 12:35:44
libqt5-x11extras-5.15.10-alt1.x86_64	Пт 01 дек 2023 12:35:44
libqt5-webenginewidgets-5.15.15-alt1.x86_64	Пт 01 дек 2023 12:35:44
libqt5-test-5.15.10-alt1.x86_64	Пт 01 дек 2023 12:35:44
libqt5-quickparticles-5.15.10-alt1.x86_64	Пт 01 дек 2023 12:35:44

Команда `rpm -qa -last` используется для вывода списка всех установленных пакетов, отсортированных по времени их установки. Пакеты будут отсортированы в порядке убывания времени установки — самые последние установленные пакеты отобразятся в верхней части списка.

Фильтрация вывода: утилита `grep` отфильтрует вывод и поможет найти искомый пакет. Например, следующая команда выведет информацию только о тех пакетах, название которых содержит «`kernel`»:

```
rpm -qa -last | grep kernel
```