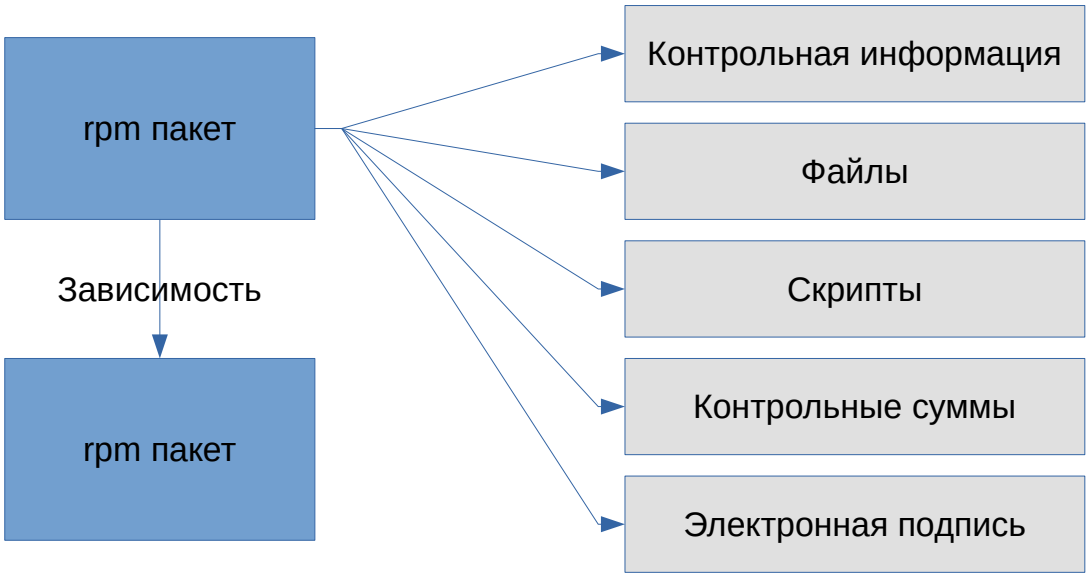
Установка и обновление программного обеспечения

Понятие пакетного менеджера



Изначально, в Linux-системах единственным способом установить новую программу была её компиляция из исходного кода (собственно говоря, этот способ никто не отменял и по сей день. Программы, которые распространяются в виде исходников, не такая уж редкость). Однако, в определенный момент времени разработчики решили предусмотреть альтернативный способ установки и обновления ПО. И появилось понятие программного пакета. Программный пакет – это разновидность архива, содержащая внутри себя устанавливаемое программное обеспечение, описательную информацию, а также специальные скрипты, которые выполняются при установке или удалении пакета.

При этом, в различных дистрибутивах реализована поддержка разных форматов программных пакетов. По типу поддерживаемых пакетов, дистрибутивы Linux можно поделить на несколько категорий: deb-based (Debian, Mint, Ubuntu), pacman-based (Arch Linux, Manjaro), RPM-based (RedHat, Fedora, CENTOS, OpenSUSE) и source-based (Slackware, Gentoo).

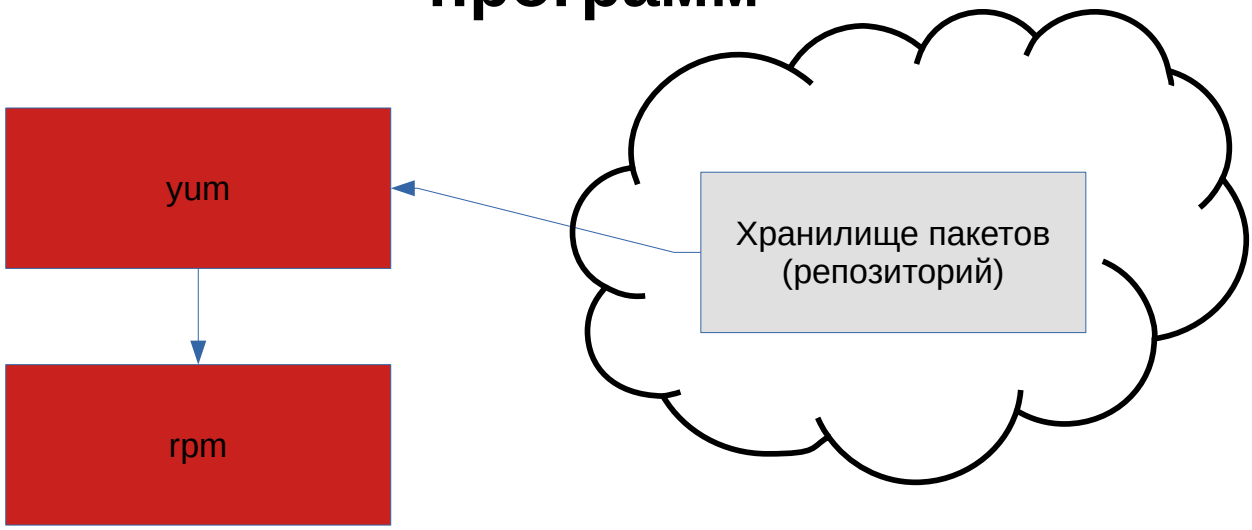
Наша система относится к категории систем, поддерживающих RPM-тип пакетов. Аббревиатура RPM расшифровывается как Red hat Packet Manager1 (менеджер пакетов Red Hat).

Имя RPM-пакета обычно состоит из нескольких частей: название программы, версия программы, номер релиза (количество пересборки программы одной и той же версии), тег, обозначающий название дистрибутива и архитектура, под которую собран пакет, например: bash-4.2.45-5.el7.x86_64.rpm

Иногда в пакет входят исходные коды. Такие пакеты не содержат информации об архитектуре, она заменяется на src: libgnomeuimm2.0-2.0.0-3.src.rpm

Также вместо архитектуры может присутствовать «noarch». Это означает, что пакет не привязан к какой-либо конкретной архитектуре и может быть установлен везде. Скажем, пакет содержит документацию и не имеет внутри исполняемых откомпилированных программ.

Установка, обновление и удаление программ



Для работы с rpm-пакетами был создан одноимённый менеджер – программа с названием rpm, но в настоящее время чаще прибегают к её специальному front-end'у, утилите с названием yum, обладающей более удобным функционалом. Ключевым отличием yum является возможность использования репозиториев.

В свою очередь, репозиторий – это место, где хранятся какие-либо данные (в нашем случае пакеты). Эти данные доступны по сети, по ним может осуществляться поиск. С определёнными оговорками, установку пакетов из репозитория можно сравнить с установкой ПО из Play Market в ОС Android. Давайте попробуем воспользоваться механизмом работы с пакетами. Для начала попробуем выполнить команду nmap:

[demo@localhost ~]$ nmap localhost bash: nmap: command not found...

Команда с таким именем отсутствует в системе. Т.к. количество пакетов в репозитории может варьироваться от нескольких десятков до нескольких тысяч, то вполне очевидно предположить, что в системе присутствует механизм, позволяющий осуществлять по нему поиск. И такой механизм действительно существует - это внутренняя команда yum - "search" (справку о всех внутренних командах yum можно получить, набрав "yum help"):

[demo@localhost ~]$ yum search nmap
Loaded plugins: langpacks
========================== N/S matched: nmap ==========================
nmap-frontend.noarch : The GTK+ front end for nmap
nmap-ncat.x86_64 : Nmap's Netcat replacement
nmap.x86_64 : Network exploration tool and security scanner

Приведённая команда осуществляет поиск в именах пакетов и их кратком описании. Если нужно, чтобы выполнялся по другим полям, например, "детальное описание" или "сайт разработчика", то необходимо использовать запрос "search all", т.е. команда с рис.2 имела бы вид: "yum search all nmap".

Менеджер обнаружил необходимый нам пакет, можно осуществить его установку:

```
[demo@localhost ~]$ sudo yum install nmap
Loaded plugins: langpacks base
Resolving Dependencies
--> Running transaction check
---> Package nmap.x86_64 2:6.40-7.el7 will be installed

Dependencies Resolved

================================================================
 Package          Arch          Version              Repository          Size
================================================================
Installing:
 nmap             x86_64        2:6.40-13.el7        base                3.9 M

Transaction Summary
================================================================
Upgrade  1 Package

Total size: 3.9 M Is this ok [y/d/N]:
```

Отключить запрос подтверждений можно, добавив опцию "-y". Это особенно полезно при установке пакетов из shell-сценариев.

Команда "yum install" сработает только в том случае, если пакет nmap не был установлен. Если пакет уже установлен и требуется его переустановить (например, из-за повреждений входивших в него файлов), используется команда yum reinstall.

Если требуется просто скачать пакет, не устанавливая его в систему, можно воспользоваться специальной командой yumdownloader. Соответствующий rpm-пакет будет скачан и помещён в текущую директорию.

Иногда, перед установкой пакета требуется уточнить какую-либо информацию, касающуюся версии или его описания. Это можно сделать с помощью специального запроса yum info:

```
[demo@localhost ~]$ yum info nmap
Loaded plugins: langpacks
Loading mirror speeds from cached hostfile
Available Packages
Name        : nmap
Arch        : x86_64
Epoch       : 2
Version     : 6.40
Release     : 7.el7
Size        : 16 M
Repo        : installed
From repo   : base
Summary     : Network exploration tool and security scanner
URL         : http://nmap.org/
License     : GPLv2 and LGPLv2+ and GPLv2+ and BSD
Description : Nmap is a utility for network exploration or security auditing.
                : It supports ping scanning (determine which hosts are up), many
                : port scanning techniques (determine what services the hosts are
                : offering), and TCP/IP fingerprinting (remote host operating system
                : identification).
```

Обратная задача – удалить пакет. Это делается следующим образом:

```
[demo@localhost ~]$ sudo yum remove -y nmap
```

Другие запросы к yum

yum list installed	Получить список всех установленных пакетов
yum list installed bash	Определить, установлен ли в системе пакет с именем bash
yum list installed bash*	Получить список всех установленных пакетов, имя которых начинается со слова bash
yum list available	Получить список всех пакетов, доступных для установки через подключенные репозитории
yum list	Получить список всех пакетов, известных менеджеру

Yum имеет свой собственный log-файл, в котором сохраняется информация о манипуляциях с пакетами:

```
[demo@localhost ~]$ sudo  tail  -n 5  /var/log/yum.log  [sudo] password for demo:
Apr 18 04:56:02 Updated: 1:net-snmp-libs-5.7.2-28.el7_4.1.x86_64
Apr 18 04:56:03 Updated: iwl6000g2b-firmware-17.168.5.2-58.el7_4.noarch
Apr 18 05:30:14 Installed: iotop-0.6-2.el7.noarch
Apr 20 04:02:46 Installed: kernel-devel-3.10.0-693.21.1.el7.x86_64 May 25 10:44:07
    Installed: libXaw-1.0.13-4.el7.x86_64
```

Помимо лога, yum ведёт историю изменений. С её помощью можно посмотреть, кто и когда вносил изменения в систему путём установки, удаления или обновления пакетов, а также сделать откат. Посмотреть историю yum:

```
[demo@localhost ~]$ sudo yum history [sudo] password for demo:
Loaded plugins: langpacks
ID     | Login user             | Date and time    | Action(s)       | Altered
-------------------------------------------------------------------------------
    7  | root <root>            | 2018-03-23 08:21 | I, U            |   56
    6  | root <root>            | 2018-03-23 07:58 | Install         |    4
    5  | root <root>            | 2018-03-07 06:11 | Install         |    1
    4  | root <root>            | 2018-03-07 04:51 | Install         |   41 EE
    3  | root <root>            | 2018-03-07 02:48 | Erase           |    2
…
```

У каждого события есть свой набор свойств: идентификатор, имя пользователя, дата и время, действие и количество пакетов, которое оно затронуло. Посмотреть информацию о событии:

[demo@localhost ~]$ sudo yum history info 7

Откатить событие

[demo@localhost ~]$ sudo yum history undo 7

Yum способен не только устанавливать и удалять пакеты, но и обновлять их. Получить список всех обновлённых пакетов:

[demo@localhost ~]$ yum list updated

Если в репозиториях присутствуют пакеты, более новой версии по сравнению с установленной, они будут отображены в виде списка. Обновить пакеты:

[demo@localhost ~]$ sudo yum update

Yum обновит все пакеты. Для того, чтобы обновить какой-то конкретный пакет, нужно указать его имя, например:

[demo@localhost ~]$ sudo yum update -y firefox

При выполнении обновления старая версия пакета заменяется новой. Если, к примеру, в системе был установлен firefox версии 50, а в репозитории был более новый пакет с номером версии 52, то после выполнения команды обновления версия 50 будет заменена на версию 52. Исключением из этого правила является пакет, содержащий обновление ядра системы.

Дело в том, что ядро является одним из ключевых компонентов операционной системы. Его работа довольно сильно завязана на взаимодействие с оборудованием, что может привести к проблемам с какими-то аппаратными составляющими системы. На случай, если с обновленным ядром возникнут проблемы, пакетный менеджер не заменяет старую версию ядра новой, а обеспечивает возможность их параллельного существования.

В меню загрузчика при этом добавляется новый пункт, позволяющий осуществить выбор между текущей и предыдущей версией ядер.

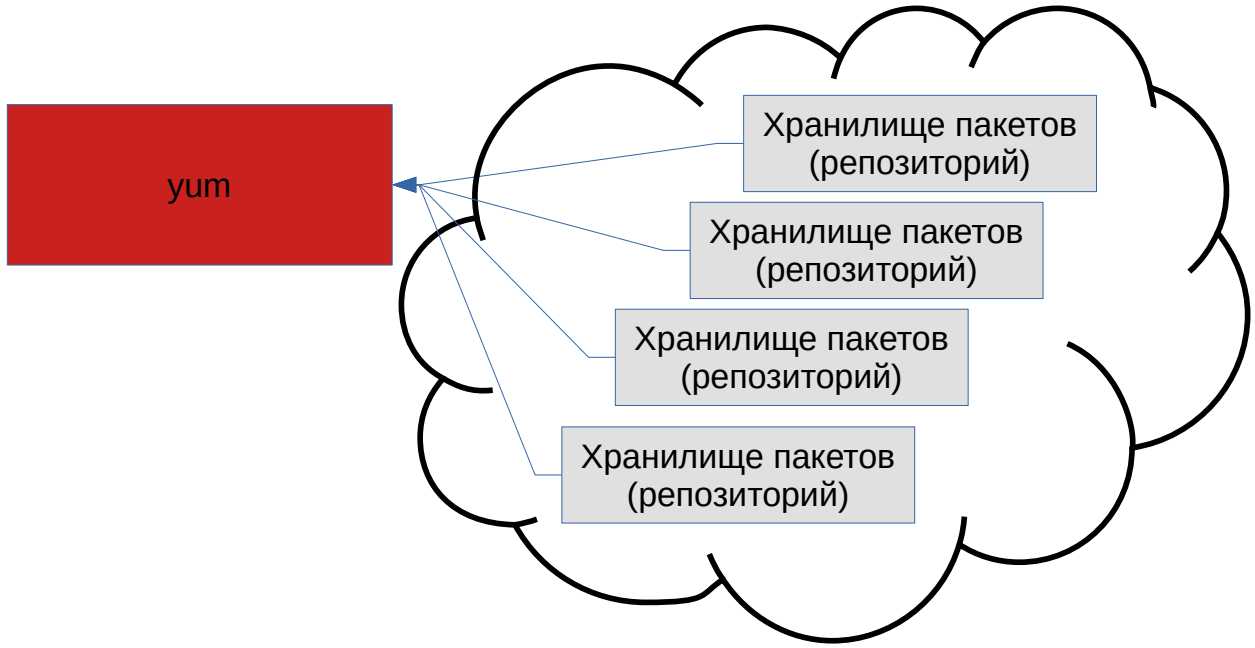
Посмотреть, какие версии ядра в настоящий момент установлены в системе можно так:

```
[demo@localhost ~]$ yum list kernel
Loaded plugins: langpacks Installed Packages
kernel.x86_64              3.10.0-693.el7                @anaconda
kernel.x86_64              3.10.0-693.21.1.el7           @updates
Available Packages
kernel.x86_64              3.10.0-862.3.3.el7            updates
```

Обновить ядро системы:

[demo@localhost ~]$ sudo yum update -y kernel

Управление репозиториями



Для работы с менеджером пакетов yum нам необходимо, чтобы к системе был подключен хотя бы один репозиторий. Как говорилось ранее, репозиторий – это место, где хранятся пакеты и некая дополнительная информация, позволяющая менеджеру с этими пакетами работать. Обычно репозиторий находится где-то в пределах сетевой видимости, но может располагаться и локально. Получить список репозиториев можно следующим образом:

```
[demo@localhost ~]$ yum repolist
Loaded plugins: langpacks
repo id                    repo name                          status
base                       CentOS-7 - Base                     9,911
extras                      CentOS-7 - Extras                    313
updates                    CentOS-7 - Updates                  695
repolist: 10,919
[demo@localhost ~]$
[demo@localhost ~]$ yum repolist all
Loaded plugins:  langpacks
repo id                     repo name                          status
base                        CentOS-7 - Base                     enabled: 9,911
base-debuginfo             CentOS-7 - Debuginfo               disabled
base-source                 CentOS-7 - Base Sources             disabled
extras                      CentOS-7 - Extras                   enabled:    313
updates                     CentOS-7 - Updates                  enabled:   695
updates-source             CentOS-7 - Updates Sources         disabled
repolist: 10,919
```

Репозитории могут находиться в одном из двух состояний: ключен и выключен. По умолчанию команда yum работает только с репозиториями, имеющими статус "enabled". Для того, чтобы задействовать отключенный репозиторий, в команде yum используется опция --enablerepo (есть противоположная ей по смыслу опция --disablerepo). Пример:

```
yum install -y xz-debuginfo --enablerepo=base-debuginfo
```

Информацию об известных ему репозиториях менеджер пакетов хранит в специальной директории - /etc/yum.repos.d Там располагаются файлы, имена которых заканчиваются на ".repo" В одном файле может содержаться информация сразу о нескольких репозиториях. Формат файла обычно следующий:

```
[adobe-linux-x86_64]
name=Adobe Systems Incorporated
baseurl=http://linuxdownload.adobe.com/linux/x86_64/
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-adobe-linux
```

Директивы, представленные на рисунке 22 можно поделить на две категории: обязательные и необязательные. Без обязательных репозиторий работать не будет. В квадратных скобках задаётся идентификатор репозитория. Это обязательная директива. Идентификатор должен быть уникальным, именно через него мы можем ссылаться на репозиторий в командах управления пакетами. Директива "name" содержит краткое описание репозитория. Это необязательная директива. Далее идет директива baseurl (обязательная) определяющая месторасположение репозитория. Директива "enabled" позволяет включать (1) или отключать (0) описанный репозиторий. Если директива "enabled" отсутствует, то поумолчанию репозиторий будет считаться включенным. При формировании программного пакета издатель обычно подписывает его своей цифровой подписью. Благодаря этому, существует возможность проверки целостности пакета на этапе его установки в систему. Эта проверка включается обязательным параметром "gpgcheck=1". Если этот параметр включен, то для обеспечения его работоспособности необходим второй параметр "gpgkey", указывающий путь к ключу.

```
[adobe-linux-x86_64]
baseurl=http://linuxdownload.adobe.com/linux/x86_64/
gpgcheck=0
```

Задание 1

1) Проверьте, работает ли команда mc
2) С помощью поиска yum определите, присутствует ли в репозиториях консольный файловый менеджер Midnight Commander. Если да, установите его.
3) Проверьте, работает ли команда mc.
4) Завершите работу mc и удалите его пакет из системы
5) Снова проверьте, работает ли команда mc.
6) Посмотрите, как ваши действия отразились в логах yum

1. Проверьте, работает ли команда mc
[demo@localhost]$ mc
2. С помощью поиска yum определите, присутствует ли в репозиториях консольный файловый менеджер Midnight Commander. Если да, установите его.
[demo@localhost]$ yum list mc
[demo@localhost]$ sudo yum install -y mc
3. Проверьте, работает ли команда mc.
[demo@localhost]$ mc
4. Завершите работу mc и удалите его пакет из системы
[demo@localhost]$ sudo yum remove -y mc
5. Снова проверьте, работает ли команда mc.
[demo@localhost]$ mc
6. Посмотрите, как ваши действия отразились в логах yum
[demo@localhost]$ sudo tail /var/log/yum.log