Утилита dnf / dnf5. Как пользоваться в Fedora?

Category: dnf / dnf5,GNU/Linux,утилиты
2026-02-02

Описание.

DNF (полное название *Dandified YUM*) — менеджер пакетов для дистрибутивов **GNU\Linux** на основе **RPM-пакетов**. Это следующее поколение приложения **YUM** (полное название *Yellowdog Updater, Modified*), которое было призвано решить недостатки предыдущего **YUM**: низкую производительность, высокое потребление памяти и низкую скорость итеративного разрешения зависимостей.

Существует и обновлённая версия — **DNF5**. Разработана как полная перепись **DNF**.

Официальная документация: dnf.readthedocs.io.

Особенности:

- В отличие от старого **DNF**, который в основном написан на **Python**, **DNF5** разработан на **C++** для лучшей производительности и меньшего использования ресурсов.
- Устраняет недостатки **DNF**, например, медленную производительность, сложную обработку зависимостей.

Основные функции **DNF** / **DNF5** включают:

- Поиск пакета в репозитории.
- Установка пакетов.
- Обновление пакетов.
- Удаление пакетов.
- Управление репозиториями.
- Кэширование.
- Транзакции.

Основные функции.

Установка утилиты.

Это может быть редкий случай, когда **DNF** / **DNF5** не установлен в системе **GNU/Linux**.

Если **DNF** / **DNF5** не установлен в вашем дистрибутиве на основе **RPM**, вы можете использовать **Yum** для установки **DNF** / **DNF5**.

```
$ sudo yum install dnf
$ sudo yum install dnf5
```

Версия утилиты.

Следующая команда показывает версию, установленную в вашей системе **GNU/Linux**.

```
$ sudo dnf --version
$ sudo dnf5 --version
```

Справка по утилите.

Опция справки по утилите.

```
$ sudo dnf --help
$ sudo dnf5 --help
```

Поиск пакета.

Позволяет найти пакет программного обеспечения в доступных репозиториях:

```
$ sudo dnf search имя_пакета
```

Поиск содержимого в пакетах.

Иногда требуется выяснить, в каких пакетах или исходных кодах содержится определенный исполняемый файл или имя пакета.

```
$ sudo dnf provides имя_пакета
```

Например, если требуется узнать, какие исходные файлы содержат mc, можно найти их, как показано в примере ниже.

Это одна из лучших функций **DNF** / **DNF5** при исследовании проблем зависимостей.

```
$ sudo dnf provides mc
```

Ответ:

```
[hamster@fedora ~]$ sudo dnf provides mc
Обновление и загрузка репозиториев:
Репозитории загружены.
mc-1:4.8.33-2.fc43.x86_64 : User-friendly text console file manager and visual shell
Repo         : @System
Matched From :
Provide      : mc = 4.8.33-2.fc43

mc-1:4.8.33-2.fc43.x86_64 : User-friendly text console file manager and visual shell
Repo         : fedora
Matched From :
Provide      : mc = 4.8.33-2.fc43

[hamster@fedora ~]$
```

Просмотр информации.

Позволяет получить информацию о доступных и установленных пакетах:

```
$ sudo dnf info имя_пакета
```

Ответ:

```
[hamster@ems ~]$ sudo dnf info mc
Обновление и загрузка репозиториев:
Репозитории загружены.
Установленные пакеты
Имя              : mc
Этап             : 1
Версия           : 4.8.33
Выпуск           : 2.fc43
Архитектура      : x86_64
Размер установки : 7.2 MiB
Источник         : mc-4.8.33-2.fc43.src.rpm
Из репозитория   : fedora
Сводка           : User-friendly text console file manager and visual shell
URL-адрес        : https://midnight-commander.org/
Лицензия         : GPL-3.0-or-later
Описание         : Midnight Commander is a visual shell much like a file manager, only with
                 : many more features. It is a text mode application, but it also includes
                 : mouse support. Midnight Commander's best features are its ability to FTP,
                 : view tar and zip files, and to poke into RPMs for specific files.
Поставщик        : Fedora Project
[hamster@ems ~]$
```

Проверить-обновить пакеты.

Параметр check-update проверяет наличие всех обновлений, доступных для вашей системы. В качестве параметра этого параметра также используется имя пакета. Однако, если имя пакета не указано, то проверяются обновления для всех установленных пакетов в вашей системе:

```
$ sudo dnf check-update
```

Список всех обновлений.

Чтобы просмотреть список всех обновлений в вашей системе, воспользуйтесь опцией списка:

```
$ sudo dnf list updates
```

Обновление пакетов.

Чтобы установить обновления для всей вашей системы, включите опцию обновления.

```
$ sudo dnf update
```

Обновим все установленные пакеты до их последних доступных версий с обработкой зависимости и пересборкой их, если это необходимо.

Дополнительно обновим метаданные репозиториев перед выполнением обновления:

```
$ sudo dnf upgrade --refresh
```

Обновление конкретного пакета.

Можно обновить конкретное приложение или пакет, указав название пакета в качестве параметра опции обновления:

```
$ sudo dnf update имя_пакета
```

Установка пакетов.

Позволяет устанавливать новые программные пакеты из репозиториев.

```
$ sudo dnf install имя_пакета
```

Установка пакетов из определённого репозитория.

Если вы хотите выполнить установку из определенного репозитория, можно использовать параметр --enablerepo при выполнении команды.

```
$ sudo dnf --enablerepo=epel install имя_пакета
```

Переустановка пакетов.

Позволяет переустанавливать программные пакеты из репозиториев:

```
$ sudo dnf reinstall имя_пакета
```

Удаление пакетов.

Позволяет удалять программные пакеты с хоста:

```
$ sudo dnf remove имя_пакета
```

Установка пакета, который вы загрузили вручную.

Иногда требуется загрузить **rpm-пакет** локально и установить его.

Можно установить его используя команду **localinstall**, указав полный путь к файлу .rpm.

```
$ sudo dnf localinstall имя_пакета.rpm
```

Примечание! Приведенная выше команда должна устранить все зависимости при установке целевого пакета .rpm. Если нет, можно выполнить следующую команду.

```
$ sudo dnf --nogpgcheck localinstall имя_пакета.rpm
```

Другой способ установить локальный **rpm-пакет** – это использовать команду вида:

```
$ sudo dnf install имя_пакета.rpm
```

Понижении версии пакета.

Если нужно понизить версию пакета до его предыдущей версии, можно воспользоваться опцией понижения версии в **DNF / DNF5**.

Будьте очень осторожны при выполнении этой команды. Эта команда удаляет текущую версию пакета и устанавливает самую высокую из всех доступных предыдущих версий с более низкой версией.

```
$ sudo dnf downgrade имя_пакета
```

Групповые операции.

Одной из замечательных функций системы, основанной на **RPM**, является **группирование пакетов**.

Группа – это набор пакетов, логически сгруппированных вместе. Это позволяет установить их все за один раз, выполнив одну команду с

названием группы.

Некоторые команды **DNF**, которые используются для работы с группами.

1) Выводит список установленных и доступных групп.

Отфильтровать результаты можно с помощью параметров --installed и --available.

Параметр --hidden позволяет отобразить скрытые группы.

$ sudo dnf group list

Ответ:

```
[hamster@mygs ~]$ sudo dnf group list
Обновление и загрузка репозиториев:
Репозитории загружены.
ID                            Name                                          Installed
milkymist                     Milkymist                                        no
kde-desktop                   KDE                                             yes
headless-management           Headless Management                              no
kde-media                     KDE Multimedia support                          yes
kde-education                 KDE Educational applications                     no
kde-mobile                    KDE Mobile                                       no
kde-mobile-apps               KDE Mobile Apps                                  no
kde-pim                       KDE PIM                                         yes
kde-office                    KDE Office                                       no
kde-software-development      KDE Software Development                         no
libreoffice                   LibreOffice                                     yes
kf6-software-development      KDE Frameworks 6 Software Development            no
mate-applications             MATE Applications                                no
kde-apps                      KDE Applications                                yes
network-server                Network Servers                                  no
neuron-modelling-simulators   Neuron Modelling Simulators                      no
office                        Office/Productivity                              no
python-classroom              Python Classroom                                 no
python-science                Python Science                                   no
robotics-suite                Robotics                                         no
rpm-development-tools         RPM Development Tools                            no
security-lab                  Security Lab                                     no
swaywm-extended               Sway Window Manager (supplemental packages)      no
text-internet                 Text-based Internet                              no
vlc                           VideoLAN Client                                  no
window-managers               Window Managers                                  no
container-management          Container Management                             no
3d-printing                   3D Printing                                      no
audio                         Audio Production                                 no
games                         Games and Entertainment                          no
sound-and-video               Sound and Video                                  no
system-tools                  System Tools                                     no
authoring-and-publishing      Authoring and Publishing                         no
budgie-desktop                Budgie                                           no
budgie-desktop-apps           Budgie Desktop Applications                      no
c-development                 C Development Tools and Libraries                no
cloud-infrastructure          Cloud Infrastructure                             no
cloud-management              Cloud Management Tools                           no
compiz                        Compiz                                           no
admin-tools                   Administration Tools                            yes
cosmic-desktop                COSMIC Desktop                                   no
cosmic-desktop-apps           COSMIC Desktop Supplementary Applications        no
d-development                 D Development Tools and Libraries                no
desktop-accessibility         Desktop accessibility                           yes
design-suite                  Design Suite                                     no
development-tools             Development Tools                                no
domain-client                 Domain Membership                                no
editors                       Editors                                          no
education                     Educational Software                             no
electronic-lab                Electronic Lab                                   no
engineering-and-scientific    Engineering and Scientific                       no
freeipa-server                FreeIPA Server                                   no
[hamster@mygs ~]$
```

2) Выводит информацию о перечне пакетов, содержащихся в определённой

группе (как обязательных, так и необязательных).

```
$ sudo dnf group info имя_группы
```

3) Устанавливает группу пакетов по имени группы или с помощью идентификатора группы.

```
$ sudo dnf group install имя_группы
```

4) Обновляет пакеты из группы и саму группу.

```
$ sudo dnf group upgrade имя_группы
```

5) Удаляет группу пакетов по имени группы или по идентификатору группы.

```
$ sudo dnf group remove имя_группы
```

6) Отмечает указанную группу как установленную, при этом никакие пакеты не устанавливаются.

```
$ sudo dnf group mark install имя_группы
```

7) Отмечает указанную группу как удалённую, при этом никакие пакеты не удаляются.

```
$ sudo dnf group mark remove имя_группы
```

Очистка временных файлов.

Чтобы удалить все временные файлы для включенных репозиториев в вашей системе, используйте опцию clean:

```
$ sudo dnf clean all
```

Если вы хотите удалить определенный временный файл, используйте различные опции, описанные ниже.

Удаляем файлы кэша для метаданных репозитория:

```
$ sudo dnf clean dbcache
```

Удаляем локальные файлы **cookie**, содержащие сигнатуру времени загрузки пакетов для каждого репозитория:

```
$ sudo dnf clean expire-cache
```

Удаляет все метаданные репозитория:

```
$ sudo dnf clean metadata
```

Удаляет все кэшированные пакеты:

```
$ sudo dnf clean packages
```

Со временем система загружается большим количеством приложений и пакетов, установленных пользователем. Следующий параметр autoremove удаляет все конечные пакеты, установленные в качестве зависимостей для любых приложений, установленных пользователем, которые больше не нужны. Таким образом, их можно безопасно удалить для освобождения места на диске.

Примечание! Бывает что оно удаляет одиночные утилиты, которые ставятся из одного единственно пакета и не содержат никаких зависимостей с другими пакетами. Надо быть внимательным.

```
$ sudo dnf autoremove
```

Управление репозиториями.

Утилита **DNF** / **DNF5** также поддерживает управление репозиториями в системах на базе **RPM**.

Добавление нового репозитория.

Можно добавить репозиторий вручную, создав файл repo_name.repo в каталоге /etc/yum.repos.d/.

Для добавления нового репозитория используется команда:

```
$ sudo dnf config-manager addrepo --
from-repofile=https://example.com/repo_name.repo
```

Пример создания локального файла репозитория:

```
$ sudo mcedit /etc/yum.repos.d/repo_name.repo
```

Примерное содержимое репозитория:

```
[my-repo]
name=My Custom Repository
baseurl=https://example.com/repo/
enabled=1
gpgcheck=1
gpgkey=https://example.com/repo/RPM-GPG-KEY
```

Просмотр списка репозиториев.

Выполним команду:

```
$ sudo dnf repolist
```

Ответ:



Для отображения всех репозиториев, включая отключенные:

```
$ sudo dnf repolist all
```

Включение и отключение репозитория.

Можно временно включить или отключить репозиторий без его удаления.

Отключить репозиторий:

```
$ sudo dnf config-manager setopt <repository-id>.enabled=0
```

Включить репозиторий:

```
$ sudo dnf config-manager setopt <repository-id>.enabled=1
```

Удаление репозитория.

Чтобы удалить репозиторий, удалите соответствующий файл repo_name.repo из каталога /etc/yum.repos.d/.

```
$ sudo rm /etc/yum.repos.d/repo_name.repo
```

Настройка приоритетов репозиториев.

Если в системе несколько репозиториев, можно настроить их приоритеты с помощью плагина **dnf-plugins-core**.

Установим плагин, если он отсутствует:

```
$ sudo dnf install dnf-plugins-core
```

Затем создадим файл my-priority-repo.repo в каталоге

/etc/yum.repos.d/ с указанием приоритета:

```
$ sudo mcedit /etc/yum.repos.d/my-priority-repo.repo
```

Примерное наполнение файла:

```
[my-high-priority-repo]
name=High Priority Repository
baseurl=https://example.com/high-priority/
enabled=1
priority=1

[my-low-priority-repo]
name=Low Priority Repository
baseurl=https://example.com/low-priority/
enabled=1
priority=10
```

Чем меньше число в priority, тем выше приоритет репозитория.

Обновление списка репозиториев.

После изменения репозиториев обновите метаданные:

```
$ sudo dnf clean all
$ sudo dnf makecache
```

Пример установки, отключения и включения стороннего репозитория.

Пример добавления репозитория **EPEL** (*Extra Packages for Enterprise Linux*):

```
$ sudo dnf install epel-release
```

После этого пакеты из **EPEL** станут доступны для установки.

Проверим идентификаторы репозиториев:

```
$ sudo dnf repolist
```

Отключить **EPEL** можно вот так:

```
$ sudo dnf config-manager setopt epel.enabled=0
```

Проверим идентификаторы репозиториев вместе со отключенными репозиториями:

```
$ sudo dnf repolist all
```

Включить обратно **EPEL** можно вот так:

```
$ sudo dnf config-manager setopt epel.enabled=1
```

Удалить насовсем можно вот так:

```
$ sudo dnf remove epel-release
```

Важно! После отключения/включения репозитория рекомендуется запустить команды:

```
$ sudo dnf clean all
$ sudo dnf makecache
```

Информация о конкретном репозитории.

Для чего это нужно:

- Проверка доступности репозитория.
- Получение информации о том, включён ли репозиторий.
- Узнать, сколько пакетов доступно для обновления.

Проверим идентификаторы репозиториев:

```
$ sudo dnf repolist
```

Ответ:

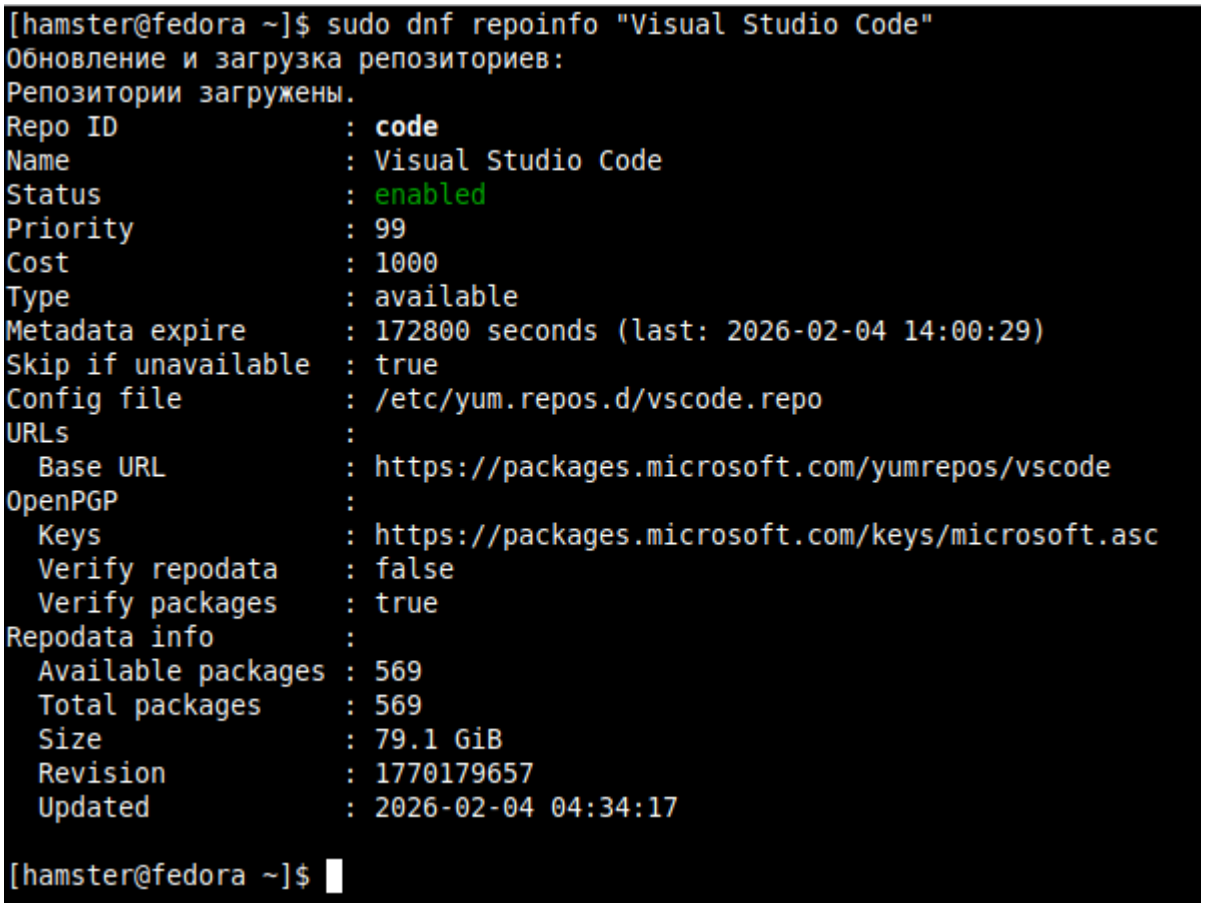


Выберем **repo_name**, пускай это будет Visual Studio Code.

Узнаем про него информацию:

```
$ sudo dnf repoinfo "Visual Studio Code"
```

Ответ:



Как откатить изменения — транзакции.

Теория откатов.

Все операции **DNF** / **DNF5** могут быть отменены откатом, что обеспечивает больше безопасности при установке или обновлении пакетов.

Утилита **DNF** / **DNF5** является более современным и функциональным инструментом по сравнению с предыдущей утилитой управления пакетами **YUM**.

Для начала обновим систему:

```
$ sudo dnf upgrade --refresh
```

Далее проверим систему на ошибки:

```
$ sudo dnf check
```

Ответ: *пустая строка.*

Как видим **DNF** / **DNF5** выдал пустую строку, это значит проблем не нашел.

Проверяет локальный **packagedb** и выдает информацию о любых найденных проблемах.

Далее посмотрим историю:

```
$ sudo dnf history list
```

Команда **history** позволяет пользователю просматривать события прошлых транзакций и действовать в соответствии с этой информацией.

По умолчанию выводит весь список, также можно искать по номерам или пакетам.

Далее можно посмотреть историю отдельной транзакции:

```
$ sudo dnf history info
```

По умолчанию выводит последнюю, также можно искать по номеру или по произведенным действиям, установленным пакетам.

Здесь следует обратить внимание на строку:

```
Status : Ok
```

Вот мы и пришли собственно к откатам:

```
$ sudo history rollback №(package-name)
```

Эта команда позволяет отменить все транзакции, выполненные после указанной транзакции.

Здесь и далее используем номера транзакций (можно делать как удобнее), если невозможно отменить некоторые транзакции из-за текущего состояния **RPMDB** (**Код возврата**), он не отменит ни одну транзакцию.

Также есть и другие варианты использования команды **history**.

Повторить указанную транзакцию:

```
$ sudo history redo №(package-name)
```

Отменить указанную транзакцию:

```
$ sudo dnf history undo №(package-name)
```

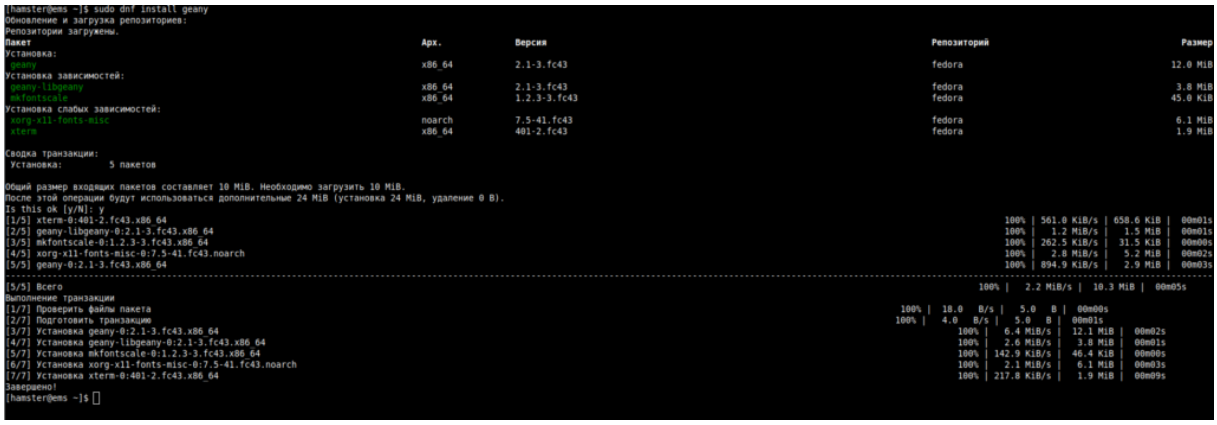
Еще следует отметить, что откаты работают с пакетами установленными при помощи **DNF / DNF5** и **PackageKit** (**Gnome-Software, Dnfdragora**). Флатпаки управляются отдельно.

Практическое применение отката.

Например установим текстовый редактор **Geany**:

```
$ sudo dnf install geany
```

Ответ:



Далее посмотрим историю:

```
$ sudo dnf history list
```

Ответ:



Далее можно снова посмотреть историю отдельной транзакции:

```
$ sudo dnf history info
```

Ответ: Status : **Ok**

```
[hamster@ems ~]$ sudo dnf history info
Transaction ID : 65
Begin time     : 2026-02-02 14:29:38
Begin rpmdb    : ed4ae8b023588532bc61538e785692835f51f90526caeaa634687092a3965c7e
End time       : 2026-02-02 14:29:54
End rpmdb      : 16c339b1ffd492c9f30f0f5b17d6854db4d007743a78a915dd0393015d1c2f3a
User           : 1000 hamster <hamster>
Status         : Ok
Releasever     : 43
Description    : dnf install geany
Comment        :
Packages altered:
  Action  Package                                    Reason          Repository
  Install geany-0:2.1-3.fc43.x86_64                  User            fedora
  Install geany-libgeany-0:2.1-3.fc43.x86_64         Dependency      fedora
  Install xterm-0:401-2.fc43.x86_64                  Weak Dependency fedora
  Install xorg-x11-fonts-misc-0:7.5-41.fc43.noarch   Weak Dependency fedora
  Install mkfontscale-0:1.2.3-3.fc43.x86_64          Dependency      fedora

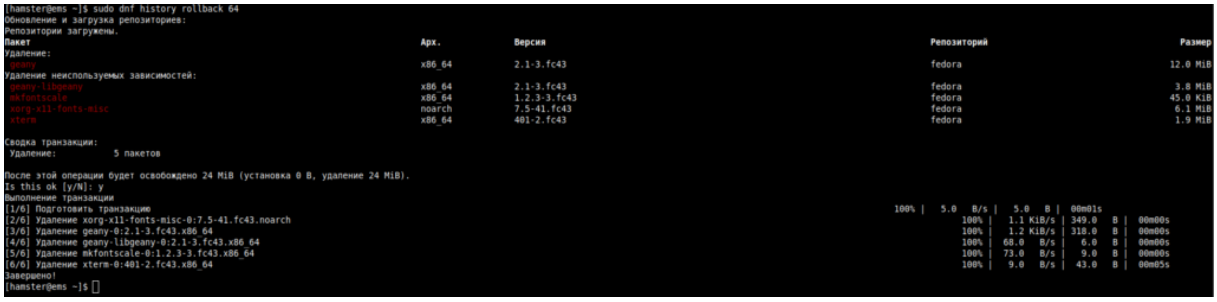
[hamster@ems ~]$
```

Далее откатим установку текстового редактора **Geany**.

Это подразумевает возврат/откат до **транзакции 64** в **history**, так как **транзакция 65** была – это установка **Geany**.

```
$ sudo dnf history rollback 64
```

Ответ:

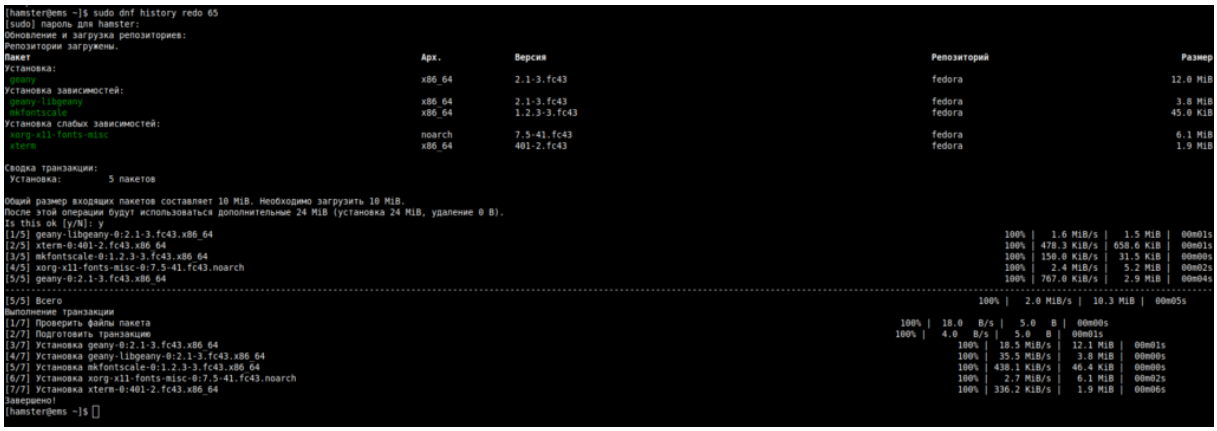


Как видим мы откатили все изменения после **транзакции 65**.

Повторить указанную транзакцию:

```
$ sudo dnf history redo 65
```

Ответ:

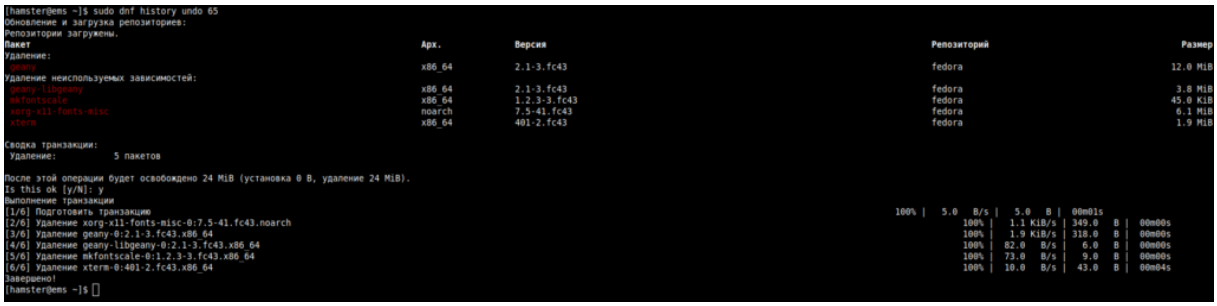


Как видим мы опять установили текстовый редактор **Geany**.

Отменить транзакцию установки:

```
$ sudo dnf history undo 65
```

Ответ:



Как видим мы опять удалили **Geany**.

Проверим!

Показать все пакеты установленные пользователем и найти в списке текстовый реактор **Geany**:

```
$ sudo dnf repoquery --userinstalled | grep geany
```

Ответ: *пустой ответ.*

Списки пакетов.

Есть несколько команд для получения списка установленных пакетов, включая те, которые были установлены вручную пользователем.

Список установленных и доступных пакетов.

Внимание! В зависимости от состояния вашей системы и подключения к

Интернету выполнение этой команды может занять некоторое время. Поскольку она извлекает метаданные с сервера.

Команда предоставляет список установленных и доступных пакетов:

```
$ sudo dnf list
```

Список доступных пакетов.

Если вам нужен более конкретный список пакетов, вы можете использовать available или installed, чтобы отфильтровать список пакетов:

```
$ sudo dnf list available
```

Список всех установленных пакетов.

Чтобы просмотреть список установленных, используйте приведенную ниже команду:

```
$ sudo dnf list installed
```

Компактный вид с прокруткой результатов:

```
$ sudo dnf list installed | more
```

Список пакетов, установленных вручную (без зависимостей).

Эта команда покажет все пакеты, которые были установлены вручную, исключая те, что были добавлены как зависимости.

Чтобы получить список пакетов, установленных вручную пользователем, можно использовать следующую команду:

```
$ sudo dnf repoquery --userinstalled
```

Список пакетов, установленных вручную (включая зависимости).

```
$ sudo dnf repoquery --installed
```

Список пакетов, установленных вручную, в формате, удобном для повторной установки.

```
$ sudo dnf repoquery --userinstalled --qf "%{name}"
```

Ответ:



Использование утилиты rpm.

Можно использовать старую-добрую утилиту **rpm** для получения списка установленных пакетов.

Все установленные пакеты по алфавиту.

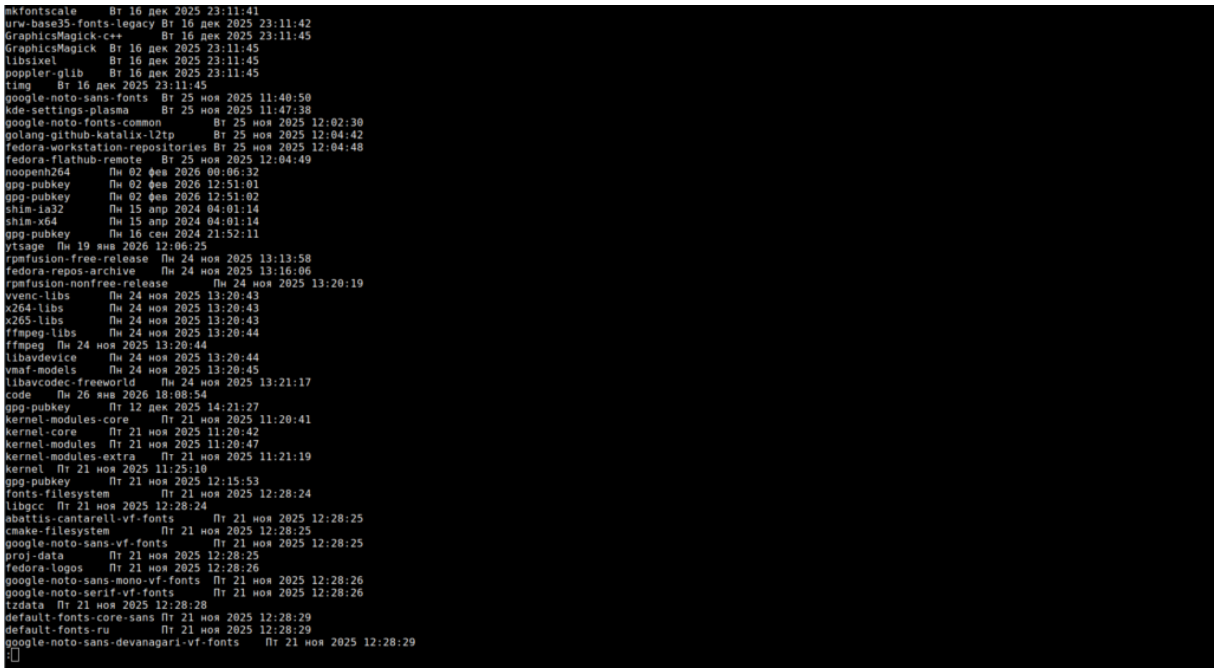
```
$ sudo rpm -qa | sort | less
```

Пакеты, установленные вручную.

Вывести пакеты с датой их установки в систему:

```
$ sudo rpm -qa --qf "%{NAME}\t%{INSTALLTIME:date}\n" | sort -k2 |
grep -v "dependency" | less
```

Ответ:



Траблшутинг.

Удаление нежелательных служб.

Помимо удаления ненужных служб, вы также можете удалить все пакеты, которые больше не нужны. Неиспользуемые пакеты могут занимать место на диске и потенциально создавать уязвимости в системе безопасности.

Чтобы узнать, какой пакет предоставляет ту или иную услугу, используйте:

Чтобы удалить пакет и связанные с ним файлы, используйте:

$ sudo sudo dnf remove <package_name>

При отключении и удалении служб необходимо следовать лучшим практикам, чтобы ваша система оставалась безопасной и функциональной.

Например, удалим из системы **Avahi** – это сервис, который используется для создания сетей с нулевой конфигурацией, позволяя устройствам обнаруживать и взаимодействовать друг с другом в одной сети.

Узнаем какой пакет связан этой службой:

$ sudo rpm -qf /usr/lib/systemd/system/avahi-daemon.service

Ответ:

```
hamster@fedora:~$ rpm -qf /usr/lib/systemd/system/avahi-daemon.service
avahi-0.9~rc2-6.fc43.x86_64
hamster@fedora:~$
```

Удалим пакет из системы:

```
$ sudo dnf remove avahi-0.9~rc2-6.fc43.x86_64
```

Если в системе DNF / DNF5 есть ошибки.

Обычно такая проблема возникает, когда обновление пакетов завершается аварийно: закрытие консоли или отключение электроэнергии.

Проверим систему на ошибки:

```
$ sudo dnf check
```

Ответ: *в идеале пустая строка.*

Как видим **DNF / DNF5** выдал пустую строку, это значит проблем не нашел.

Проверяет локальный **packagedb** и выдает информацию о любых найденных проблемах.

Вариант, когда ошибки-таки есть.

Ответ:

```
systemd-sysusers-0:258.3-2.fc43.x86_64
 duplicate with "systemd-sysusers-0:258.4-1.fc43.x86_64"
systemd-sysusers-0:258.4-1.fc43.x86_64
 duplicate with "systemd-sysusers-0:258.3-2.fc43.x86_64"
systemd-udev-0:258.3-2.fc43.x86_64
 duplicate with "systemd-udev-0:258.4-1.fc43.x86_64"
systemd-udev-0:258.4-1.fc43.x86_64
 duplicate with "systemd-udev-0:258.3-2.fc43.x86_64"
Проверьте обнаруженную(-ые) 108 проблему(-мы) в 106 пакете(-ах)
[hamster@ems ~]$
```

Как видно у нас есть дубликаты пакетов.

Как удалить дубликаты пакетов.

При обновлении **DNF / DNF5** ругается на дубликаты пакетов.

Установим утилиту **package-cleanup**:

```
$ sudo dnf install dnf-utils
```

Удалим дубликаты и повреждённые пакеты:

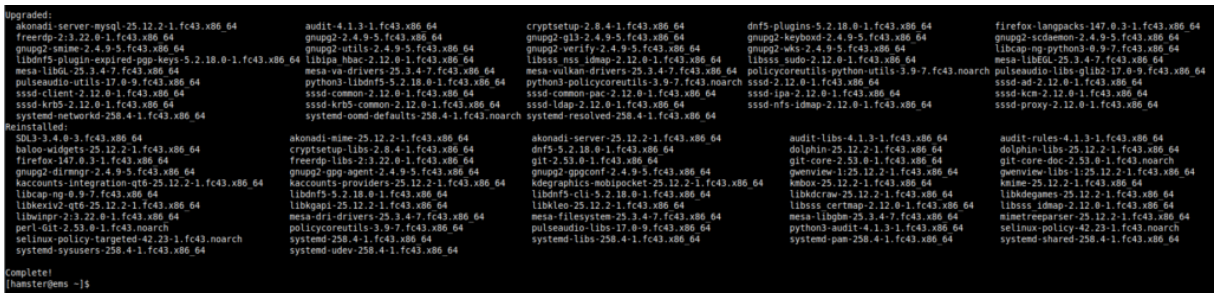
```
$ sudo package-cleanup --cleandupes
```

Ответ: *в этом примере всё хорошо.*

```
[hamster@fedora ~]$ sudo package-cleanup --cleandupes
Last metadata expiration check: 0:05:29 ago on Ср 04 фев 2026 19:34:16.
No duplicated packages found for removal.
Dependencies resolved.
Nothing to do.
Complete!
[hamster@fedora ~]$
```

Если были бы дубликаты, то **DNF / DNF5** исправил бы и переустановил всё что посчитал неисправным.

Ответ:



База RPM оказалась повреждена. Как восстановить?

Для запуска пересборки базы данных **RPM** следует выполнить:

```
$ sudo rpm --rebuilddb
```

Настоятельно рекомендуется сделать резервную копию каталога /var/lib/rpm перед этим действием.

Оригиналы источников информации.

1. losst.pro «Как откатить изменения в Fedora.»
2. comss.ru «Оптимизация DNF5 в Fedora 43: рабочие параметры и реальные улучшения скорости.»
3. debugpoint.com «Essential DNF Commands for Linux Users [With Examples].»
4. dnf.readthedocs.io «DNF Command Reference.»