Отчёт по лабораторной работе №4

Работа с программными пакетами

Максат Хемраев

Содержание

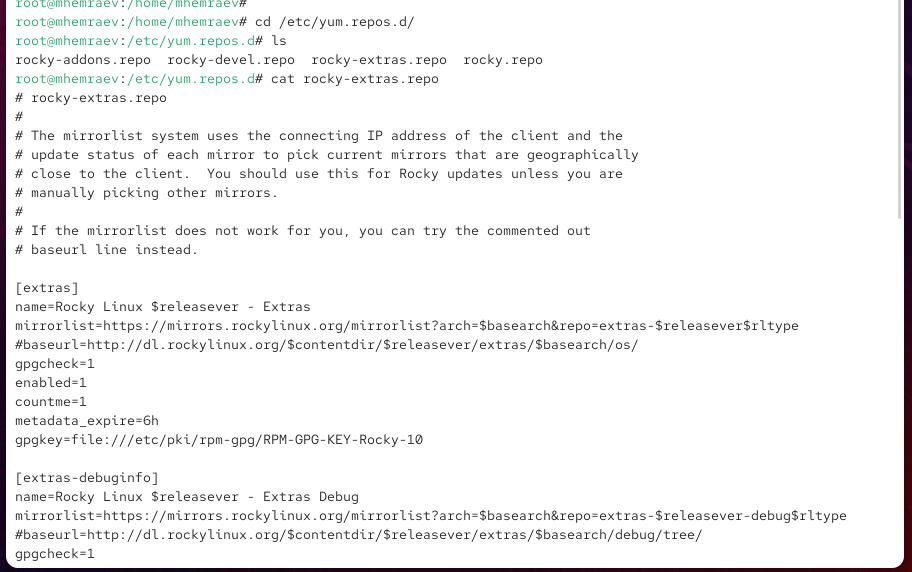
# 1 Цель работы

Получить навыки работы с репозиториями и менеджерами пакетов.

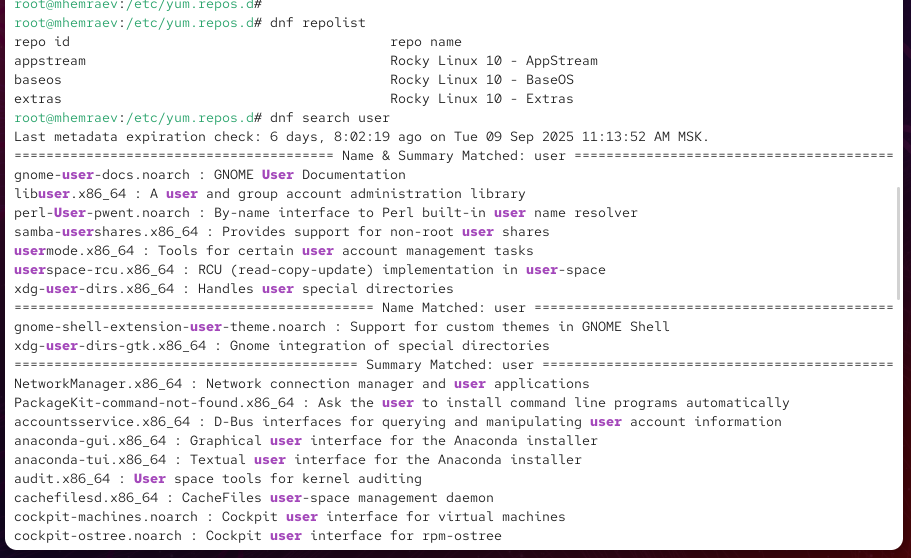
# 2 Отчёт по выполнению работы

## 2.1 Работа с репозиториями

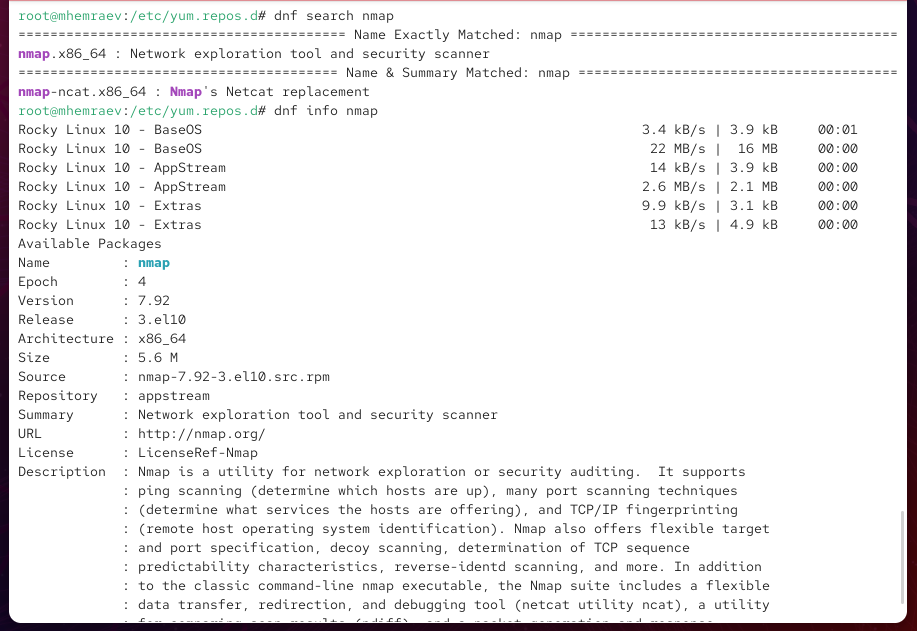
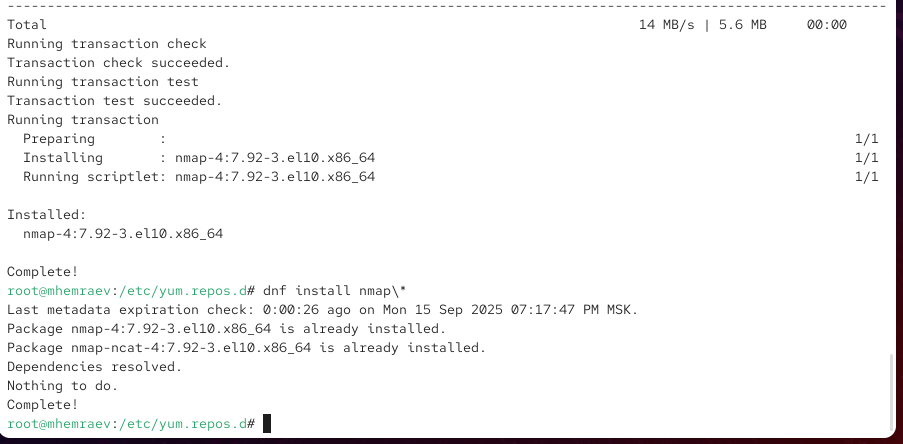
1. Вошёл в систему под пользователем **root** и перешёл в каталог **/etc/yum.repos.d**. Изучил доступные файлы конфигурации репозиториев, среди которых:
   * *rocky-addons.repo*
   * *rocky-devel.repo*
   * *rocky-extras.repo*
   * *rocky.repo*

* Содержимое файла *rocky-extras.repo* подтверждает наличие настроенного зеркала для получения дополнительных пакетов.
* 
* Рис. 1: Файл rocky-extras.repo

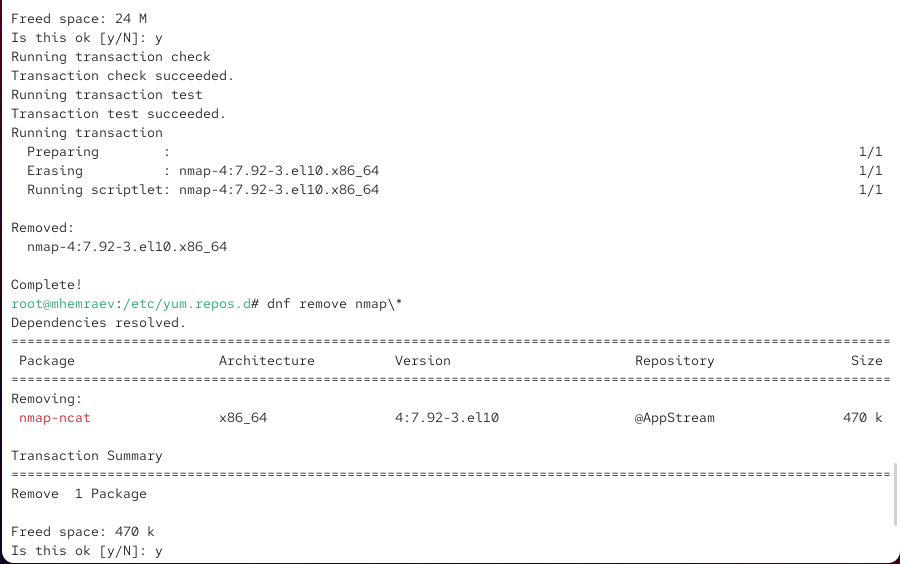
1. Вывел список доступных репозиториев с помощью команды **dnf repolist**. Система определила активные репозитории:
   * *AppStream* — дополнительные пакеты приложений,
   * *BaseOS* — базовые компоненты системы,
   * *Extras* — дополнительные пакеты и утилиты.

* 
* Рис. 2: Список репозиториев

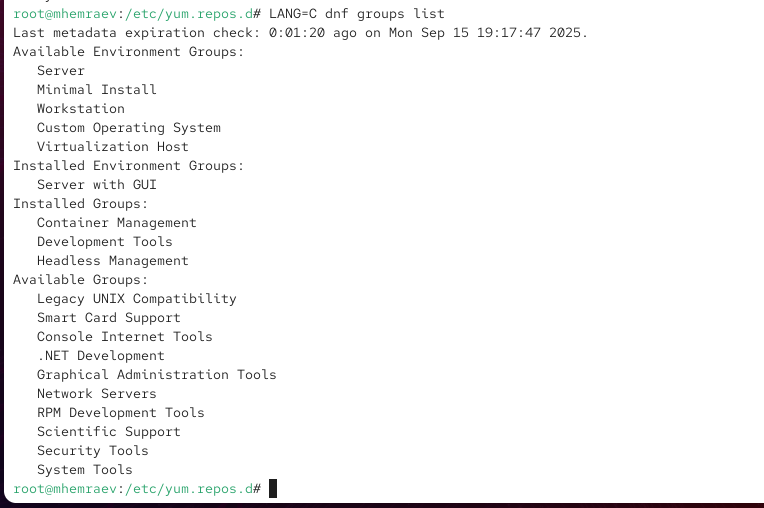
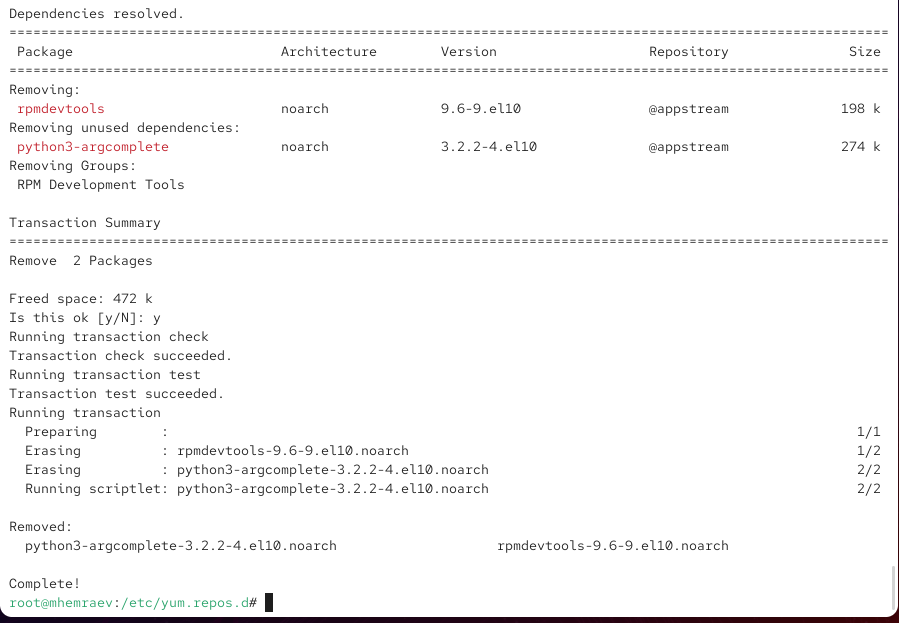
1. Выполнил поиск пакетов, содержащих слово **user** в названии или описании. Были найдены библиотеки и утилиты, связанные с управлением пользователями и пользовательскими каталогами (например, *libuser*, *usermode*, *xdg-user-dirs*).
2. Провёл поиск и изучение пакета **nmap**. Получил информацию о версии, размере, назначении и репозитории, из которого он доступен (*AppStream*).

* 
* Рис. 3: Информация о пакете nmap
  + Команда **dnf install nmap** устанавливает только основной пакет.
  + Команда **dnf install nmap\*** дополнительно устанавливает связанные компоненты (*nmap-ncat*).
* 
* Рис. 4: Установка пакета nmap

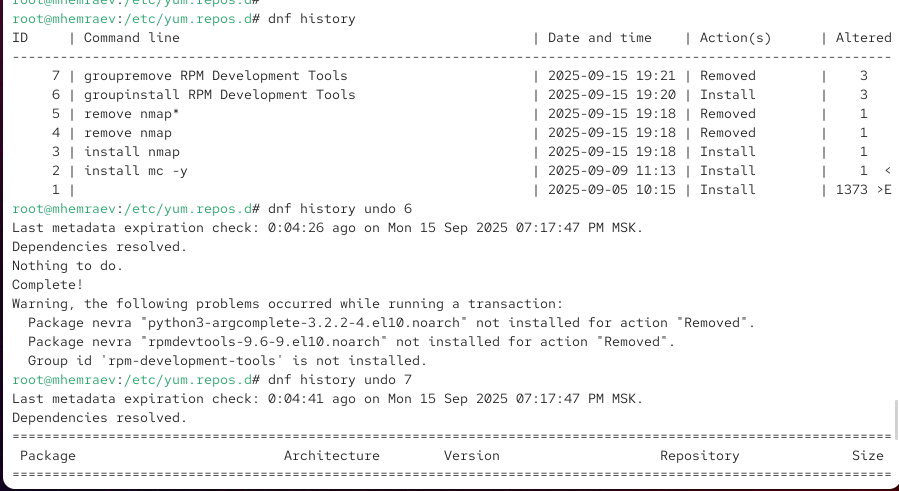
1. Удалил установленный пакет **nmap** вместе с зависимостями, используя команды:
   * dnf remove nmap — удаляет основной пакет,
   * dnf remove nmap\\* — удаляет все пакеты, связанные с nmap (в том числе *nmap-ncat*).

* 
* Рис. 5: Удаление пакета nmap

1. Получил список групп пакетов. Среди доступных — *Development Tools*, *Security Tools*, *RPM Development Tools* и др.

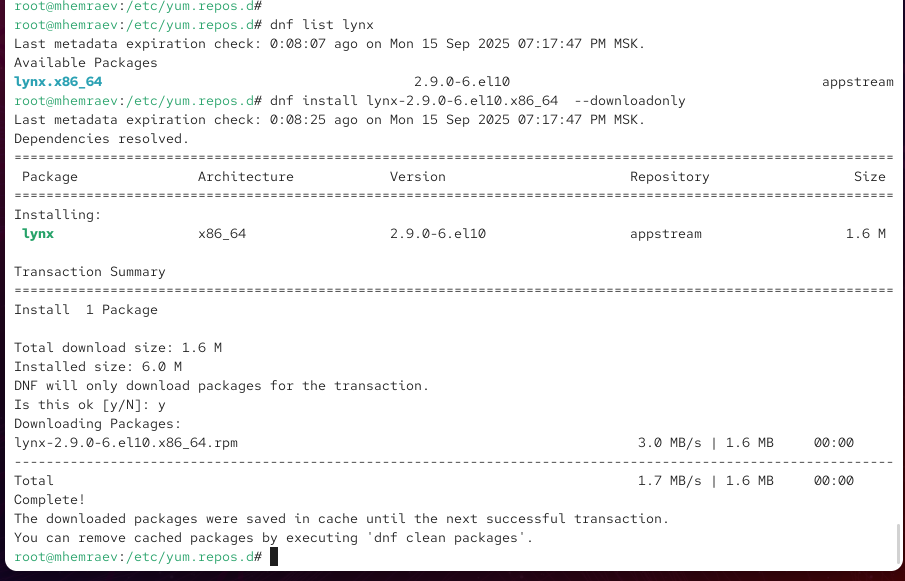
* 
* Рис. 6: Список групп пакетов
* Изучил описание группы **RPM Development Tools**, включающей утилиты для сборки RPM-пакетов (*rpm-build*, *redhat-rpm-config* и др.).
* 
* Рис. 7: Информация о группе RPM Development Tools
* Установил группу командой **dnf groupinstall “RPM Development Tools”**, после чего в систему добавились пакеты *rpmdevtools* и *python3-argcomplete*.
* Для удаления группы использовал команду **dnf groupremove “RPM Development Tools”**.
* 
* Рис. 8: Удаление группы RPM Development Tools

1. Изучил историю операций DNF при помощи команды **dnf history**. Система сохранила последовательность установки и удаления пакетов и групп.

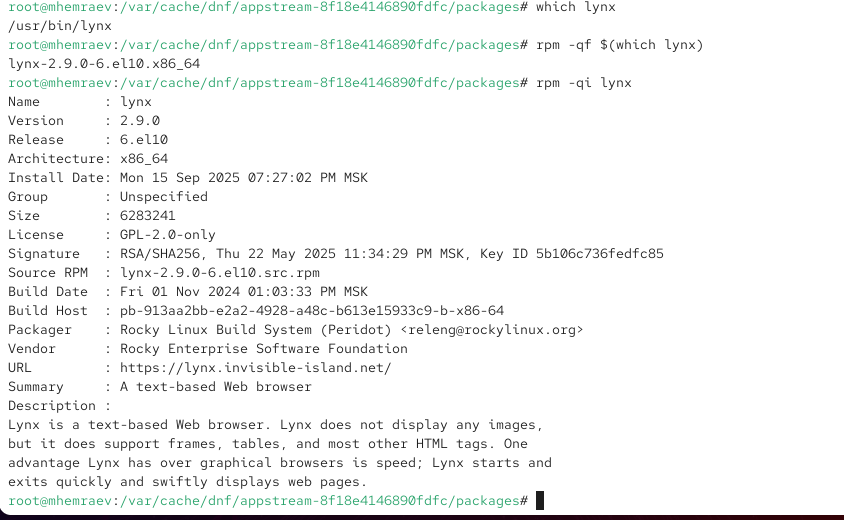
* 
* Рис. 9: История DNF
* Отменил шестое действие из истории командой **dnf history undo 6**, а также проверил результат выполнения команды **dnf history undo 7**.

## 2.2 Использование rpm

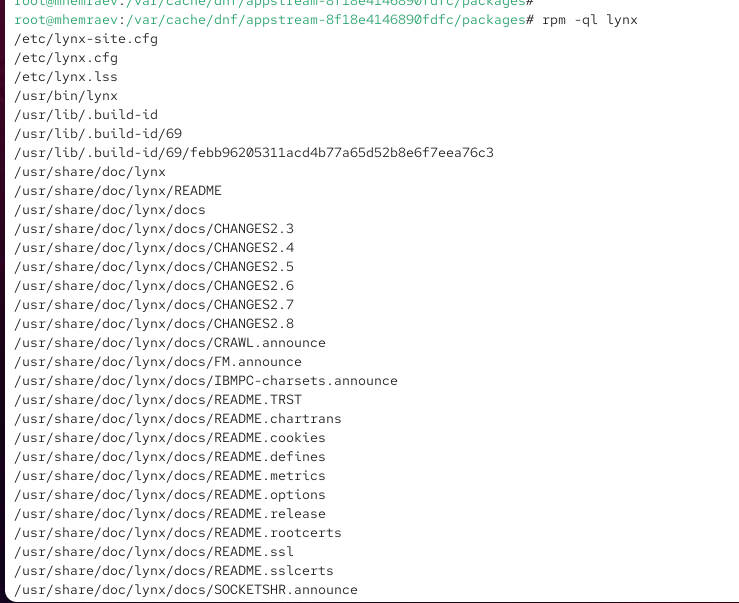
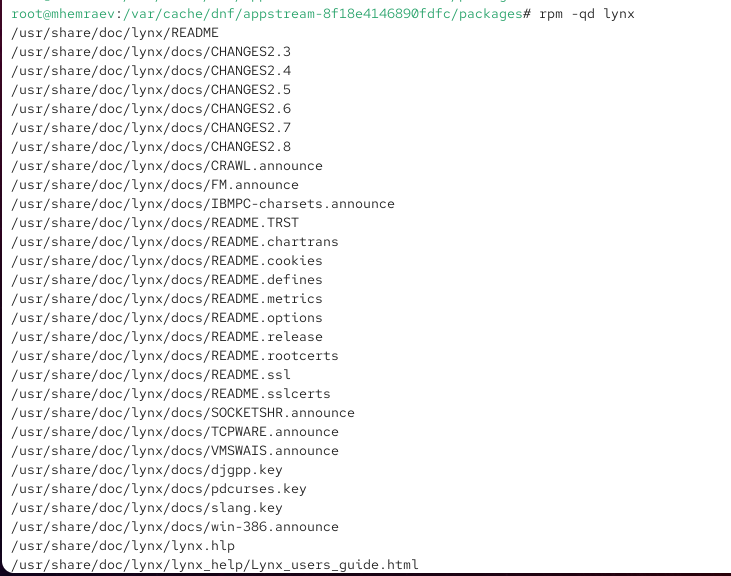
1. Скачал rpm-пакет **lynx** с помощью команд dnf list lynx и dnf install lynx --downloadonly.  
   Пакет был загружен в кэш DNF.

* 
* Рис. 10: Загрузка пакета lynx

1. Определил каталог, куда был помещён загруженный пакет — **/var/cache/dnf/appstream…/packages**.
2. Перешёл в данный каталог и установил пакет командой rpm -Uhv lynx-2.9.0-6.el10.x86\_64.rpm.  
   После установки пакет стал доступен в системе.
3. Проверил расположение исполняемого файла с помощью команды which lynx.  
   Результат — **/usr/bin/lynx**.

* 
* Рис. 11: Исполняемый файл lynx

1. С помощью команды rpm -qf $(which lynx) убедился, что бинарный файл принадлежит пакету lynx.  
   Дополнительно получил сведения о пакете через команду rpm -qi lynx.  
   Информация содержит версию (2.9.0), архитектуру (x86\_64), лицензию (GPL-2.0), URL и краткое описание.
2. Получил список всех файлов пакета с помощью rpm -ql lynx.  
   Среди них: бинарные файлы, документация, служебные файлы.

* Для просмотра документации выполнил rpm -qd lynx.  
  Были найдены файлы руководств и справочные материалы, включая **Lynx\_users\_guide.html**.
*   
  

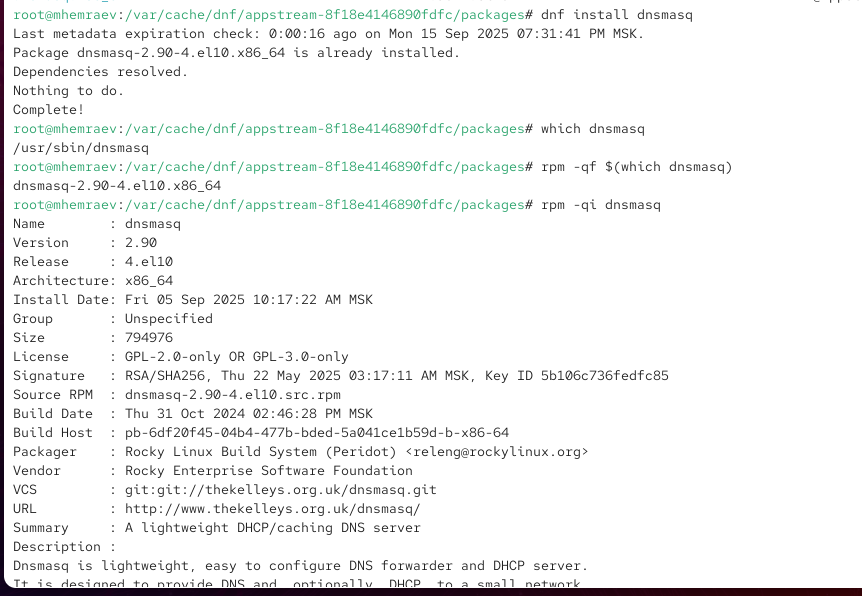
1. Вывел список конфигурационных файлов пакета с помощью команды rpm -qc lynx.  
   Результат — файлы /etc/lynx.cfg, /etc/lynx-site.cfg, /etc/lynx.lss.

* 
* Рис. 12: Конфигурационные файлы lynx

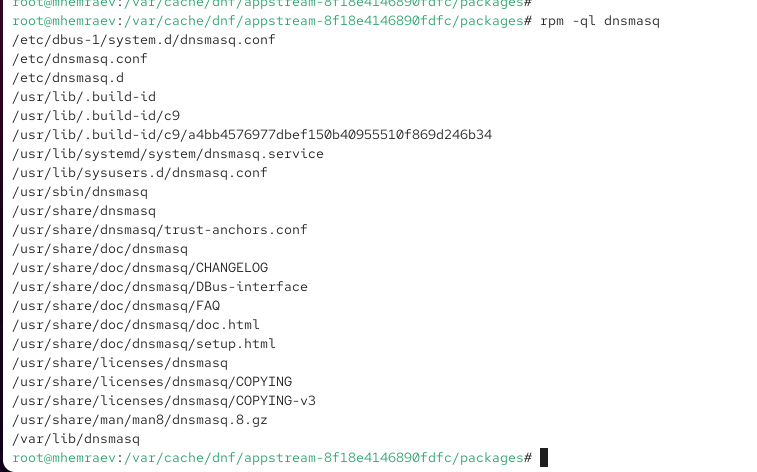
1. Проверил наличие скриптов, выполняемых при установке, с помощью rpm -q --scripts lynx.  
   В данном случае специфических скриптов не было. Скрипты, если они есть, выполняют дополнительные действия при установке, удалении или обновлении пакета (например, настройка сервисов).
2. Под своей учётной записью запустил текстовый браузер **lynx** и убедился в его работоспособности.
3. Вернувшись в терминал **root**, удалил пакет командой rpm -e lynx.  
   После удаления пакет исчез из списка установленных.

## 2.3 Установка и изучение пакета dnsmasq

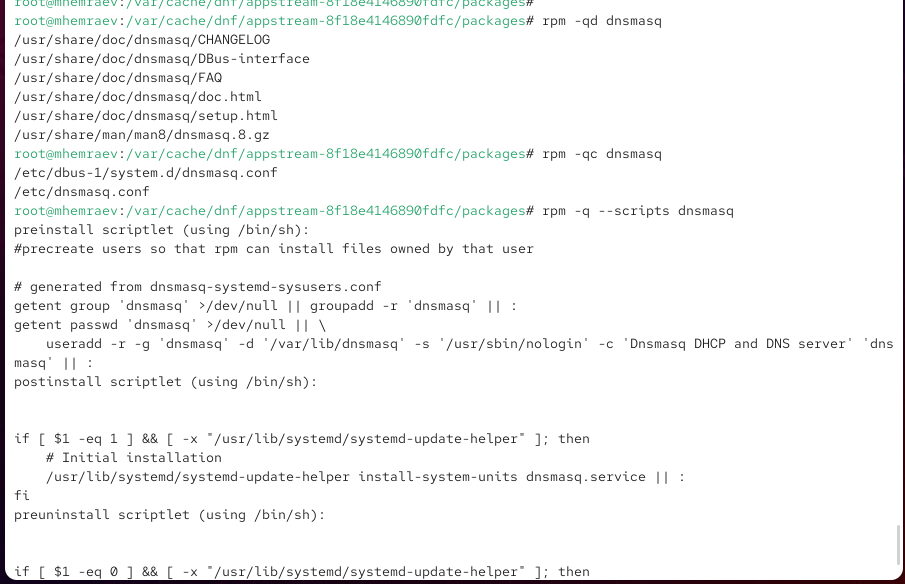
1. Установил пакет **dnsmasq** с помощью команд dnf list dnsmasq и dnf install dnsmasq.  
   Проверил расположение исполняемого файла — он находится в каталоге **/usr/sbin/dnsmasq**.

* 
* Рис. 13: Установка и определение расположения dnsmasq

1. С помощью команды rpm -qf $(which dnsmasq) подтвердил, что исполняемый файл принадлежит пакету **dnsmasq**.  
   Далее, командой rpm -qi dnsmasq получил полную информацию о пакете: версия (2.90), архитектура (x86\_64), размер (794976 байт), лицензия (GPL-2.0-only), а также назначение — **легковесный DNS-, DHCP- и TFTP-сервер**.
2. Вывел список всех файлов пакета с помощью команды rpm -ql dnsmasq.  
   Среди них присутствуют:
   * бинарные файлы в **/usr/sbin**,
   * системные юниты для systemd,
   * документация в **/usr/share/doc/dnsmasq**.

* Для вывода только файлов документации использовал rpm -qd dnsmasq. В список входят **FAQ**, **setup.html**, **COPYING** и другие справочные материалы.
* 
* Рис. 14: Файлы пакета dnsmasq

1. С помощью команды rpm -qc dnsmasq получил список конфигурационных файлов пакета.  
   Основные:
   * **/etc/dnsmasq.conf**
   * **/etc/dnsmasq.d/**
2. Изучил скрипты, выполняемые при установке пакета (rpm -q --scripts dnsmasq).
   * *preinstall* — создаёт системного пользователя и группу **dnsmasq** для корректной работы сервиса.
   * *postinstall* — регистрирует службу в systemd.
   * *preuninstall* — удаляет службу при деинсталляции.

* Эти скрипты автоматизируют настройку окружения для пакета.
* 
* Рис. 15: Скрипты пакета dnsmasq

1. В завершение удалил пакет с помощью команды rpm -e dnsmasq.

# 3 Контрольные вопросы

### 3.0.1 1. Какая команда позволяет вам искать пакет rpm, содержащий файл useradd?

* rpm -qf $(which useradd) — определяет, к какому пакету относится файл.
* Если нужно найти через базу: dnf provides \*/useradd.

### 3.0.2 2. Какие команды вам нужно использовать, чтобы показать имя группы dnf, которая содержит инструменты безопасности и показать, что находится в этой группе?

* dnf group list — список всех доступных групп.
* dnf group info "Security Tools" — информация о содержимом группы *Security Tools*.

### 3.0.3 3. Какая команда позволяет вам установить rpm, который вы загрузили из Интернета и который не находится в репозиториях?

* rpm -Uvh имя\_пакета.rpm — установка загруженного пакета.
* Альтернативный вариант через dnf: dnf install ./имя\_пакета.rpm.

### 3.0.4 4. Вы хотите убедиться, что пакет rpm, который вы загрузили, не содержит никакого опасного кода сценария. Какая команда позволяет это сделать?

* rpm -qp --scripts имя\_пакета.rpm — показывает скрипты, выполняемые при установке и удалении пакета.

### 3.0.5 5. Какая команда показывает всю документацию в rpm?

* rpm -qd имя\_пакета — выводит список всех файлов документации пакета.

### 3.0.6 6. Какая команда показывает, какому пакету rpm принадлежит файл?

* rpm -qf /путь/к/файлу — определяет пакет, которому принадлежит данный файл.

# 4 Заключение

В ходе работы были изучены команды **dnf** и **rpm** для установки, удаления и анализа пакетов, а также работы с репозиториями и групповыми установками. Полученные навыки позволяют администрировать программное обеспечение в Linux, контролировать зависимости и проверять безопасность устанавливаемых пакетов.