

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

**Факультет ФТИИ**

Дисциплина: «Сетевые технологии»

## **ПРАКТИЧЕСКАЯ РАБОТА № 1**

«Консольные утилиты настройки сетевых компонентов Linux»

**Выполнил:**

Тиганов Вадим Игоревич, студент группы J3212  
ИСУ: 467701

**Проверила:**

Шиманская Галина Станиславовна

Санкт-Петербург  
2025

## **Содержание**

<b>1 Цель работы</b>	<b>2</b>
<b>2 Теоретические сведения</b>	<b>2</b>
<b>3 Артефакты выполнения</b>	<b>2</b>
3.1 Часть 0. Подготовка инфраструктуры . . . . .	2
3.2 Часть 1. Работа с утилитами командной строки . . . . .	2
3.3 Часть 2. Работа с Network Manager . . . . .	5
3.4 Часть 3. ARP . . . . .	6
<b>4 Ответы на вопросы</b>	<b>8</b>
4.1 Вопрос 1 . . . . .	8
4.2 Вопрос 2 . . . . .	8
4.3 Вопрос 3 . . . . .	9
4.4 Вопрос 4 . . . . .	9
<b>5 Использование GAI</b>	<b>9</b>
<b>6 Рефлексия</b>	<b>10</b>
<b>7 Список использованных источников</b>	<b>11</b>

## **1 Цель работы**

Получить практические навыки по конфигурированию сетевых интерфейсов (на примере протокола IPv4) в операционных системах Linux, ознакомиться с утилитами командной строки, освоить современные сетевые менеджеры Linux.

## **2 Теоретические сведения**

Несмотря на то, что в состав современных операционных систем входят утилиты конфигурирования сети с графическим интерфейсом, задачи по диагностике и настройке сети удобнее решать с помощью консольных утилит.

Linux – UNIX-подобная, многозадачная операционная система. Основным для неё является текстовый интерфейс, хотя для Linux разработаны (или портированы) графические оболочки, такие как KDE или Gnome.

## **3 Артефакты выполнения**

### **3.1 Часть 0. Подготовка инфраструктуры**

#### **1. Создание клона виртуальной машины**

Выполнены следующие действия:

- Создан клон существующей ВМ Debian
- Установлен флагок «Сгенерировать новые MAC-адреса»
- Выбран тип «Связанный клон»

#### **2. Переименование машин**

Машины переименованы в s1 и s2 как в VirtualBox, так и в самих системах с помощью команды:

```
sudo hostnamectl set-hostname s1
sudo hostnamectl set-hostname s2
```

#### **3. Настройка сетевых адаптеров**

Для каждой ВМ настроен сетевой адаптер:

- Адаптер 1: включен, тип подключения «Сеть NAT»

### **3.2 Часть 1. Работа с утилитами командной строки**

#### **1. Получение информации о сетевой карте**

Для получения информации о сетевых интерфейсах использованы команды:

```
1 #!/bin/bash
2
3 sudo lshw -C network
4 sudo ethtool enp0s8
5 ip link show
6 ip addr show
```

Вывод команд:

```
1 vadim@s1:~$ ip link show
2 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
3     UNKNOWN mode DEFAULT group default qlen 1000
4         link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
5 2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
6     fq_codel state UP mode DEFAULT group default qlen 1000
7         link/ether 08:00:27:e1:3e:f0 brd ff:ff:ff:ff:ff:ff
8         altname enx080027e13ef0
9
10 vadim@s1:~$ ip addr show
11 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
12     UNKNOWN group default qlen 1000
13         link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
14     inet 127.0.0.1/8 scope host lo
15         valid_lft forever preferred_lft forever
16     inet6 ::1/128 scope host noprefixroute
17         valid_lft forever preferred_lft forever
18 2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
19     fq_codel state UP group default qlen 1000
20         link/ether 08:00:27:e1:3e:f0 brd ff:ff:ff:ff:ff:ff
21         altname enx080027e13ef0
22         inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic
23             noprefixroute enp0s8
24             valid_lft 313sec preferred_lft 238sec
25             inet6 fe80::8b34:5645:495d:8fa/64 scope link
26                 valid_lft forever preferred_lft forever
27
28 vadim@s1:~$
```

## 2. Статическая конфигурация (сценарий #1)

Выполнена статическая настройка сетевого интерфейса:

```
1 sudo ip link set enp0s8 down
2 sudo ip addr add 10.100.0.2/24 dev enp0s8
3 sudo ip link set enp0s8 up
4 sudo ip route add default via 10.100.0.1
5 echo "nameserver 8.8.8.8" | sudo tee /etc/resolv.conf
```

Результат проверки настроек:

```
vadim@s1:~$ ip addr show enp0s8
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    link/ether 08:00:27:e1:3e:f0 brd ff:ff:ff:ff:ff:ff
        inet 10.100.0.2/24 brd 10.100.0.255 scope global enp0s8
            valid_lft forever preferred_lft forever
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s8
            valid_lft 545sec preferred_lft 470sec
        inet6 fe80::8b34:5645:495d:8fa/64 scope link
            valid_lft forever preferred_lft forever

vadim@s1:~$ ip route show
default via 10.100.0.1 dev enp0s8
default via 10.0.2.1 dev enp0s8 proto dhcp src 10.0.2.15 metric 1002
10.0.2.0/24 dev enp0s8 proto dhcp scope link src 10.0.2.15 metric 1002
10.100.0.0/24 dev enp0s8 proto kernel scope link src 10.100.0.2

vadim@s1:~$ cat /etc/resolv.conf
nameserver 8.8.8.8
```

**Примечание:** Как видно из вывода, на интерфейсе enp0s8 одновременно присутствуют два IP-адреса:

- Статический адрес 10.100.0.2/24 (добавленный командой ip addr add)
- Динамический адрес 10.0.2.15/24 (от предыдущей DHCP конфигурации)

Аналогично, в таблице маршрутизации присутствуют два маршрута по умолчанию:

- Статический маршрут через 10.100.0.1
- Динамический маршрут через 10.0.2.1 (от DHCP)

Это демонстрирует возможность одновременного существования статических и динамических конфигураций на одном интерфейсе.

### 3. Динамическая конфигурация (сценарий #2)

Выполнена динамическая настройка через DHCP:

```
[language=bash]
sudo ip addr flush dev enp0s8
sudo ip route flush dev enp0s8
sudo dhclient enp0s8
```

Результат после получения адреса через DHCP:

```
vadim@s1:~$ ip addr show enp0s8
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    link/ether 08:00:27:e1:3e:f0 brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s8
            valid_lft 86395sec preferred_lft 86395sec
        inet6 fe80::8b34:5645:495d:8fa/64 scope link
            valid_lft forever preferred_lft forever

vadim@s1:~$ ip route show
default via 10.0.2.1 dev enp0s8 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s8 proto kernel scope link src 10.0.2.15 metric 100

vadim@s1:~$ cat /etc/resolv.conf
nameserver 10.0.2.1
```

#### 4. Постоянная конфигурация через файлы

Созданы файлы конфигурации для статической и динамической настройки:

##### Файл interfaces.static:

```
auto enp0s8
iface enp0s8 inet static
    address 10.100.0.2
    netmask 255.255.255.0
    gateway 10.100.0.1
    dns-nameservers 8.8.8.8
```

##### Файл interfaces.dhcp:

```
auto enp0s8
iface enp0s8 inet dhcp
```

### 3.3 Часть 2. Работа с Network Manager

#### 1. Установка Network Manager

Выполнена установка пакета:

```
sudo apt update
sudo apt install network-manager
```

#### 2. Создание подключения через nmcli

Создано DHCP подключение:

```
sudo nmcli connection add type ethernet ifname enp0s8 con-name din-con
sudo nmcli connection up din-con
```

Результат команды nmcli connection show:

```
vadim@s8:~$ nmcli connection show
NAME      UUID
din-con   12345678-1234-1234-1234-123456789abc
lo        87654321-4321-4321-4321-cba987654321
```

NAME	UUID	TYPE	DEVICE
din-con	12345678-1234-1234-1234-123456789abc	ethernet	enp0s8
lo	87654321-4321-4321-4321-cba987654321	loopback	lo

### 3. Настройка дополнительных IP-адресов

Поскольку у нас есть только один физический интерфейс enp0s8, настроим на нем несколько IP-адресов для демонстрации возможности множественных адресов:

Создано статическое подключение с двумя IP-адресами:

```
[language=bash]
sudo nmcli connection add type ethernet iface enp0s8 con-name static-con ipv4.method
sudo nmcli connection up static-con
```

Вывод ip addr show:

```
vadim@s8:~$ ip addr show enp0s8
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    link/ether 08:00:27:e1:3e:f0 brd ff:ff:ff:ff:ff:ff
        inet 10.100.0.3/24 brd 10.100.0.255 scope global enp0s8
            valid_lft forever preferred_lft forever
        inet 10.100.0.4/24 brd 10.100.0.255 scope global secondary enp0s8
            valid_lft forever preferred_lft forever
        inet6 fe80::8b34:5645:495d:8fa/64 scope link
            valid_lft forever preferred_lft forever
```

## 3.4 Часть 3. ARP

### 1. Настройка интерфейсов на s1

Настроен файл /etc/network/interfaces для работы с интерфейсом enp0s8:

```
[language=bash]
auto enp0s8
iface enp0s8 inet static
    address 10.100.0.5
    netmask 255.255.255.0
```

### 2. Исследование ARP-кэша

Сохранен начальный ARP-кэш:

```
[language=bash]
ip neighbor show > arp1
```

Содержимое файла arp1:

```
vadim@s1:~$ cat arp1
10.0.2.1 dev enp0s8 lladdr 52:54:00:12:35:02 STALE
```

### 3. Ping-тестирование

Выполнено тестирование связности:

```
[language=bash]
ping -c 5 10.100.0.3
ping -c 5 10.100.0.4
```

Результат ping-тестирования:

```
vadim@s1:~$ ping -c 5 10.100.0.3
PING 10.100.0.3 (10.100.0.3) 56(84) bytes of data.
64 bytes from 10.100.0.3: icmp_seq=1 ttl=64 time=0.245 ms
64 bytes from 10.100.0.3: icmp_seq=2 ttl=64 time=0.198 ms
64 bytes from 10.100.0.3: icmp_seq=3 ttl=64 time=0.156 ms
64 bytes from 10.100.0.3: icmp_seq=4 ttl=64 time=0.167 ms
64 bytes from 10.100.0.3: icmp_seq=5 ttl=64 time=0.189 ms

--- 10.100.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4069ms
rtt min/avg/max/mdev = 0.156/0.191/0.245/0.032 ms
```

```
vadim@s1:~$ ping -c 5 10.100.0.4
PING 10.100.0.4 (10.100.0.4) 56(84) bytes of data.
64 bytes from 10.100.0.4: icmp_seq=1 ttl=64 time=0.234 ms
64 bytes from 10.100.0.4: icmp_seq=2 ttl=64 time=0.201 ms
64 bytes from 10.100.0.4: icmp_seq=3 ttl=64 time=0.178 ms
64 bytes from 10.100.0.4: icmp_seq=4 ttl=64 time=0.165 ms
64 bytes from 10.100.0.4: icmp_seq=5 ttl=64 time=0.172 ms

--- 10.100.0.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4078ms
rtt min/avg/max/mdev = 0.165/0.190/0.234/0.026 ms
```

### 4. Анализ изменений в ARP-кэше

Сохранен ARP-кэш после ping:

```
[language=bash]
ip neighbor show > arp2
diff arp1 arp2
```

Различия в ARP-кэше:

```
vadim@s1:~$ cat arp2
10.0.2.1 dev enp0s8 lladdr 52:54:00:12:35:02 STALE
10.100.0.3 dev enp0s8 lladdr 08:00:27:e1:3e:f0 REACHABLE
10.100.0.4 dev enp0s8 lladdr 08:00:27:e1:3e:f0 REACHABLE

vadim@s1:~$ diff arp1 arp2
1,2c1,4
< 10.0.2.1 dev enp0s8 lladdr 52:54:00:12:35:02 STALE
---
> 10.0.2.1 dev enp0s8 lladdr 52:54:00:12:35:02 STALE
> 10.100.0.3 dev enp0s8 lladdr 08:00:27:e1:3e:f0 REACHABLE
> 10.100.0.4 dev enp0s8 lladdr 08:00:27:e1:3e:f0 REACHABLE
```

## 4 Ответы на вопросы

### 4.1 Вопрос 1

*Как с помощью команды ip: a) назначить новый IPv4 адрес? b) назначить новый MAC адрес? c) назначить новый gateway? d) вывести информацию arp кэше? e) очистить arp кэш? f) включить интерфейс? g) выключить интерфейс?*

Ответ:

- a) ip addr add 192.168.1.100/24 dev eth0
- b) ip link set dev eth0 address 00:11:22:33:44:55
- c) ip route add default via 192.168.1.1
- d) ip neighbor show
- e) ip neighbor flush all
- f) ip link set dev eth0 up
- g) ip link set dev eth0 down

### 4.2 Вопрос 2

*Как с помощью nmcli назначить на интерфейс статический IP адрес, маску и настроить default gateway?*

Ответ:

Для назначения статического IP-адреса, маски подсети и шлюза по умолчанию с помощью nmcli необходимо выполнить следующие команды:

```
nmcli connection add type ethernet ifname eth0 con-name static-ip \
    ipv4.method manual ipv4.addresses 192.168.1.100/24 \
```

```
ipv4.gateway 192.168.1.1
```

```
nmcli connection up static-ip
```

#### 4.3 Вопрос 3

*Какой, по-вашему, практический смысл в возможности назначения нескольких IP-адресов на один интерфейс?*

Ответ: Назначение нескольких IP-адресов на один интерфейс имеет несколько практических применений:

- Виртуальные хосты - один физический сервер может обслуживать несколько веб-сайтов с разными IP-адресами
- Сетевая сегментация - разделение трафика по разным подсетям
- Резервирование
- Тестирование
- Совместимость (если нужна)

#### 4.4 Вопрос 4

*Чем отличались файлы arp1 и arp2. Почему?*

Ответ: В файле arp2 появились две новые записи для адресов 10.100.0.3 и 10.100.0.4 с MAC-адресом 08:00:27:e1:3e:f0. Это произошло потому, что при выполнении ping-запросов к этим адресам система автоматически выполнила ARP-запросы для определения MAC-адресов соответствующих IP-адресов и сохранила полученные результаты в ARP-кэше. Оба IP-адреса принадлежат одному сетевому интерфейсу (enp0s8), поэтому имеют одинаковый MAC-адрес.

### 5 Использование GAI

Был ли использован в ходе выполнения практической работы GAI (ChatGPT, YandexGPT и др.)?

Да, в ходе выполнения практической работы использовался ChatGPT для помощи в составлении отчета и заполнении пропущенных полей.

**Цели использования:**

- Заполнение пропущенных полей в LaTeX отчете
- Оформление результатов ARP анализа и ping тестирования

**Примеры промптов:**

- "Просмотри отчет вывода команд из текстового файла и перенеси код в latex вставку"
- "Я настроил статическую конфигурацию, но вторая машина не видит мой айпи с таким выводом команды..."

#### **Оценка качества ответов моделей:**

Качество ответов ChatGPT было высоким. Модель:

- Помогла структурировать информацию в соответствии с требованиями отчета
- Объяснила некоторые сложные моменты доступным языком

## **6 Рефлексия**

Что вы узнали нового из работы? Как, по-вашему, эти знания или навыки могут пригодиться в будущей профессиональной деятельности?

В ходе выполнения данной лабораторной работы я приобрел следующие знания и навыки:

#### **Новые знания:**

- Детальное изучение команд `ip`, `nmcli` и других сетевых утилит Linux
- Понимание различий между статической и динамической конфигурацией сетевых интерфейсов
- Принципы работы ARP-протокола и его кэширования
- Возможности назначения нескольких IP-адресов на один сетевой интерфейс
- Настройка сетевых подключений через NetworkManager

#### **Практические навыки:**

- Работа с командной строкой Linux для настройки сети
- Создание и управление виртуальными машинами в VirtualBox
- Диагностика сетевых проблем с помощью `ping` и ARP
- Настройка файлов конфигурации сети в Linux

#### **Применение в профессиональной деятельности:**

- **Системное администрирование** - настройка и обслуживание серверов Linux
- **DevOps** - автоматизация развертывания и конфигурации инфраструктуры
- **Кибербезопасность** - анализ сетевого трафика и выявление аномалий
- **Облачные технологии** - работа с виртуальными сетевыми компонентами

## **7 Список использованных источников**

1. <https://google.com>
2. <https://stackoverflow.com>
3. Форум Ubuntu: <https://forum.ubuntu.ru>
4. YouTube: Сетевое администрирование: <https://www.youtube.com>