

# Телекоммуникационные системы и технологии

## Лабораторная работа №1 Консольные утилиты настройки сетевых компонентов Linux

Выполнил: Птицын Владислав

Группа: М3301

Преподаватель: Береснев А.Д.

Санкт-Петербург, 2024

**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.

## Часть 1

### Скрипт часть 1 п. 2

```
lshw -c network | grep product | sed 's/^[ \t]*///'
ethtool enp0s3 | grep -e Speed -e Duplex -e Link | sed 's/^[ \t]*///'
ip link show enp0s3 | grep ether | awk '{print "mac:", $2}'
ip -4 a show enp0s3 | grep inet | awk '{print "ip/mask:", $2}'
ip route | grep default | awk '{print "gate:", $3}'
cat ../../etc/resolv.conf | grep nameserver | awk '{print "dns:", $2}'

echo choose network configuration scenario:[1/2]
read -r choice
oldip="$(ip -4 a show enp0s3 | grep inet | awk '{print $2}')"
ip a del "$oldip" dev enp0s3
if [[ "$choice" == "1" ]]; then
    ip a add 10.100.0.2/24 dev enp0s3
    ip r del default dev enp0s3
    ip route add default via 10.100.0.1 dev enp0s3
    echo "nameserver 8.8.8.8" > ../../etc/resolv.conf
elif [[ "$choice" == "2" ]]; then
    dhclient -r dev enp0s3
    dhclient enp0s3
else
    echo input error
fi
```

## Вывод скрипта

```
product: 82540EM Gigabit Ethernet Controller
Speed: 1000Mb/s
Duplex: Full
Link detected: yes
mac: 08:00:27:99:36:06
ip/mask: 10.0.2.15/24
gate: 10.0.2.2
dns: 192.168.1.1
```

## Настройка сетевого интерфейса

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:99:36:06 brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.2/24 scope global enp0s3
        | valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe99:3606/64 scope link
        | valid_lft forever preferred_lft forever
```

```
default via 10.100.0.1 dev enp0s3
10.100.0.0/24 dev enp0s3 proto kernel scope link src 10.100.0.2
nameserver 8.8.8.8
```

# Часть 2

## Скрипт часть 2 п. 2

```
nmcli con modify "enp0s3" ipv4.dns 8.8.8.8 ipv4.address 10.100.0.2/24 gw4 10.100.0.1
nmcli con down enp0s3
nmcli con up enp0s3

oldip="$(ip a s dev enp0s3 | grep "inet " | tail -1 | awk '{print $2}')"
ip a del "$oldip" dev enp0s3
nmcli con add con-name bridge type bridge ifname br0 ip4 10.100.0.3
nmcli con up bridge

ip link show dev br0 | grep ether | awk '{print "mac:", $2}'
```

## Настройка сетевого интерфейса

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ef:c0:34 brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.2/24 brd 10.100.0.255 scope global noprefixroute enp0s3
        | valid_lft forever preferred_lft forever
    inet6 fe80::938c:c47a:6caa:6add/64 scope link noprefixroute
        | valid_lft forever preferred_lft forever
14: br0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether ba:9f:4e:dd:cf:bb brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.3/32 scope global noprefixroute br0
        | valid_lft forever preferred_lft forever
```

## Консольный вывод и mac-address

```
Подключение «bridge» (3686fb49-3781-4f1e-b45b-fcf2c62a14e7) успешно добавлено.
Подключение успешно активировано (controller waiting for ports) (активный путь D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/4)
mac: ba:9f:4e:dd:cf:bb
```

## Часть 3

### Скрипт часть 3 п. 3

```
str="network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 10.100.0.4/24
        - 10.100.0.5/24
      routes:
        - to: default
          via: 10.100.0.3"
echo "$str" > /etc/netplan/myconf.yaml

netplan apply
```

### Настройка сетевого интерфейса

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:99:36:06 brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.4/24 brd 10.100.0.255 scope global enp0s3
        | valid_lft forever preferred_lft forever
    inet 10.100.0.5/24 brd 10.100.0.255 scope global secondary enp0s3
        | valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe99:3606/64 scope link
        | valid_lft forever preferred_lft forever
default via 10.100.0.3 dev enp0s3 proto static
10.100.0.0/24 dev enp0s3 proto kernel scope link src 10.100.0.4
```

## Скрипт часть 3 п. 4

```
for i in $(seq 2 5);  
do  
    ping -c 2 10.100.0.$i  
    echo "  
done  
ip neigh show > test.txt
```

## Таблица arp cache

```
10.100.0.2 dev enp0s3 lladdr 08:00:27:ef:c0:34 REACHABLE  
10.100.0.3 dev enp0s3 lladdr 08:00:27:ef:c0:34 STALE  
10.0.2.3 dev enp0s3 lladdr 08:00:27:5d:9f:10 STALE
```

# Часть 4

## Скрипт часть 4 п.3

```
nmcli con del bond007
nmcli con del enp0s3
nmcli con del enp0s8
nmcli con mod "Проводное подключение 1" con-name enp0s3
nmcli con add type bond con-name bond007 ifname bond007 mode 0
nmcli con add type bond-slave ifname enp0s3 con-name enp0s3 master bond007
nmcli con add type bond-slave ifname enp0s8 con-name enp0s8 master bond007
```

## Вывод информации: ip4, mac-address + /proc/net/bonding/bond007

```
nmcli c s bond007 | grep "4\.ADDRESS" | awk '{print "bond007\n" "ip: " $2}'
ip a show dev bond007 | grep link/ether | awk '{print "mac: " $2 "\n"}'
cat /proc/net/bonding/bond007 | grep -e "Slave Int" -e "Perm" | sed -e 's/Permanent[ ]HW[ ]addr/mac/g' -e 's/Slave[ ]Interface:[ ]//g'
echo ""
cat /proc/net/bonding/bond007
```

bond007  
ip: 10.0.2.15/24  
mac: 08:00:27:ef:c0:34

enp0s3  
mac: 08:00:27:ef:c0:34  
enp0s8  
mac: 08:00:27:07:17:be

Ethernet Channel Bonding Driver: v5.14.0-503.el9.x86\_64

Bonding Mode: load balancing (round-robin)  
MII Status: up  
MII Polling Interval (ms): 100  
Up Delay (ms): 0  
Down Delay (ms): 0  
Peer Notification Delay (ms): 0

Slave Interface: enp0s3  
MII Status: up  
Speed: 1000 Mbps  
Duplex: full  
Link Failure Count: 0  
Permanent HW addr: 08:00:27:ef:c0:34  
Slave queue ID: 0

Slave Interface: enp0s8  
MII Status: up  
Speed: 1000 Mbps  
Duplex: full  
Link Failure Count: 0  
Permanent HW addr: 08:00:27:07:17:be  
Slave queue ID: 0



## Скрипт часть 4 п. 7

```
date +%D\ %H:%M:%S
echo -e "Interface\tRecieve-packets\tTransmit-packets"
cat /proc/net/dev | tail -4 | sed 's/007/07/g' | awk '{print $1 "\t\t" $3 "\t\t" $11}'
```

## Сбор данных

```
09/10/24 19:13:49
Interface    Recieve-packets Transmit-packets
lo:          0          0
enp0s3:      128        109
enp0s8:       19         45
bond07:       90         95
09/10/24 19:13:50
Interface    Recieve-packets Transmit-packets
lo:          0          0
enp0s3:      129        110
enp0s8:       19         45
bond07:       91         96
09/10/24 19:13:51
Interface    Recieve-packets Transmit-packets
lo:          0          0
enp0s3:      130        110
enp0s8:       19         46
bond07:       92         97
```

# Ответы на вопросы и задания

1. Как с помощью команды `ip`:

a. назначить новый IPv4 адрес?

```
ip addr add [IP_адрес/маска] dev [интерфейс]
```

b. назначить новый MAC адрес?

```
ip link set dev [интерфейс] address [mac_адрес]
```

c. назначить новый gateway?

```
ip route add [подсеть/маска] via [шлюз]
```

d. вывести информацию arp кэше?

```
ip n s
```

e. очистить arp кэш?

```
ip n f dev [интерфейс]
```

f. включить интерфейс?

```
ip link set dev [интерфейс] up
```

g. выключить интерфейс?

```
ip link set dev [интерфейс] down
```

2. Как с помощью `nmcli` назначить на интерфейс статический

IP адрес, маску и настроить default gateway?

```
nmcli connection add con-name "[соединение]" ifname [интерфейс]
```

```
ipv4.method manual type [тип_подключения] ip4 [IP_адрес] gw4 [gateway]
```

3. Как с помощью netplan назначить на интерфейс статический IP адрес, маску и настроить default gateway?

```
network:
  version: [версия]
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - [IP_адрес]
      routes:
        - to: default
          | via: [шлюз]
```

4. Какие режимы bonding стандартно существуют в Linux?

Опишите их назначение, возможности по отказоустойчивости и необходимость поддержки со стороны оборудования.

Тип	Отказоустойчивость	Балансировка	Описание
Round Robin	-	+	Циклический режим.
Active Backup	+	-	Один из интерфейсов активен. Если активный интерфейс выходит из строя (link down и т.д.), другой интерфейс заменяет активный
XOR [exclusive OR]	+	+	Передачи распределяются между интерфейсами на основе формулы ((MAC-адрес источника) XOR (MAC-адрес получателя)) % число интерфейсов. Один и тот же интерфейс работает с определённым получателем.
Broadcast	+	-	Широковещательный режим. Все пакеты отправляются на все интерфейсы.

Dynamic Link Aggregation	+	+	IEEE 802.3ad, получить логическое сетевое устройство, обладающее суммарной пропускной способностью всех его адаптеров, с помощью агрегации.
Transmit Load Balancing (TLB)	+	+	Входящий трафик приходит только на активный интерфейс, а исходящий распределяется в зависимости от текущей загрузки каждого интерфейса.
Adaptive Load Balancing (ALB)	+	+	Обеспечивает балансировку нагрузки как исходящего, так и входящего трафика.

Bonding mode	Configuration on the switch
0 - balance-rk	Требуется включить статический EtherChannel
1 - active-backup	Настройка коммутатора не требуется.
2 - balance-xor	Требуется включить статический EtherChannel
3 - broadcast	Требуется включить статический EtherChannel
4 - 802.3ad	Требуется включить согласованный с LACP эфирный канал.
5 - balance-tlb	Настройка коммутатора не требуется.
6 - balance-alb	Не требует настройки коммутатора, но интерфейсы должны уметь изменять MAC.

5. Какие существуют и чем отличаются режимы работы адаптера (duplex)?

**Полудуплексный режим (Half-duplex)** Устройство, установленное в этот режим, или отправляет или получает

**Полнодуплексном режиме (Full-duplex)** - он может отправлять и получать пакеты одновременно.

**Simplex** – данные только отправляются и не факт, что могут быть прочитанны

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

Можно использовать, когда мы устанавливаем несколько сайтов на виртуальных веб-хостинг на одном интерфейсе, или поддерживаете несколько соединений к сети, каждое из которых обслуживает разные цели, или хотим имитировать физическое присутствие некоего оборудования на площадке, фактически висящего в другой отдаленной сети

7. Какой, по-вашему, практический смысл в возможности создания виртуальных интерфейсов?

Иногда, и даже достаточно часто, хотелось бы иметь сетевой интерфейс, который мог бы оперировать с трафиком любого другого интерфейса, но каким-то образом дополнительно «окрашивать» этот трафик. Такое может понадобиться для дополнительного анализа, или контроля трафика, или его шифрования или Он может пригодиться при создании виртуальных машин или контейнеров, нужных для корректной работы и взаимодействия программного обеспечения, которое было создано как сетевое, но запускается на локальном ПК.