

Лабораторная работа №1. Консольные утилиты настройки сетевых  
компонентов Linux

Выполнили студенты группы М3311  
Авсюкевич Анастасия  
Худашов Богдан

**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.

## Артефакты:

### 1. Скрипт Часть 1 п. 2

```
#!/bin/bash

function get_network_info {
    echo "Info about network card:"
    ip link show | grep -E '^[2]:' | awk -F ':' '{print $2}' | while read -r iface; do
        echo "Interface: $iface"
        lshw -class network | grep -E 'product'
        ethtool $iface | grep -E 'Speed|Duplex|Link detected'
        ip a | grep -E 'ether' | sed 's/brd.*//g'
        echo ""
    done
}

function get_ipv4_info {
    echo "Current IPv4 configuration:"
    ip -4 addr show | awk '/^[2]:' /{iface=$2} iface && /inet /{print iface $2}'
    ip route | grep default
    cat /etc/resolv.conf | grep nameserver
    echo ""
}
```

Для п.3 (статическая адресация)

```
function configure_static_scenario {
    ip addr flush dev enp0s3
    ip route flush dev enp0s3
    ip addr add 10.100.0.2/24 dev enp0s3
    ip route add default via 10.100.0.1
    ip a
    ip r
    # echo ""
    echo "nameserver 8.8.8.8" | tee /etc/resolv.conf > /dev/null
}

function configure_dynamic_scenario {
    dhclient -r enp0s3
    ip a
    echo ""
    dhclient enp0s3
    ip a
}
```

## Консольный вывод части 1 п. 2

```
Chose action:
1. Get network info
2. Get IPv4 info
3. Configure static interface
4. Configure dynamic interface
5. Exit
Your choice: 1
Info about network card:
Interface: enp0s3
    product: 82540EM Gigabit Ethernet Controller
    Speed: 1000Mb/s
    Duplex: Full
    Link detected: yes
link/ether 08:00:27:c3:04:a5

Chose action:
1. Get network info
2. Get IPv4 info
3. Configure static interface
4. Configure dynamic interface
5. Exit
Your choice: 2
Current IPv4 configuration:
enp0s3:10.0.2.15/24
default via 10.0.2.2 dev enp0s3
nameserver 192.168.8.1
nameserver 85.21.192.5
nameserver 213.234.192.7
```

```
4. Exit
Your choice: 3
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c3:04:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.2/24 scope global enp0s3
        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
Chose action:
```

```
5. Exit
Your choice: 4
Killed old client process
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c3:04:a5 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fec3:4a5/64 scope link
        valid_lft forever preferred_lft forever

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c3:04:a5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86400sec preferred_lft 86400sec
    inet6 fe80::a00:27ff:fec3:4a5/64 scope link
        valid_lft forever preferred_lft forever
```

## 2. Команды Части 2 п. 2

```
#!/bin/bash

nmcli con modify "enp0s3" ipv4.method manual ip4 10.100.0.2/24 gw4 10.100.0.1 ipv4.dns "8.8.8.8"
nmcli con up "enp0s3"

nmcli connection add type bridge ifname br0 con-name br0 ipv4.method manual ip4 10.100.0.3/24
nmcli connection up "br0"
nmcli con show
ip a
echo ""

ping -c 4 10.100.0.3

mac_addr=$(nmcli d show br0 | grep "GENERAL.HWADDR:")
echo "mac-address: $mac_addr"
```

```
lroot@localhost ~# ./lab1_task2.sh
Подключение успешно активировано (активный путь D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/?)
Подключение «br0» (9a52936f-2e26-4f5e-aa52-8fe07ce589de) успешно добавлено.
Подключение успешно активировано (controller waiting for ports) (активный путь D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/?)
NAME UUID TYPE DEVICE
enp0s3 808a7420-0768-32d1-a9f4-cef142bfa449 ethernet enp0s3
br0 9a52936f-2e26-4f5e-aa52-8fe07ce589de bridge br0
lo 402657e0-47e9-4069-9258-619c2ba2bccc loopback lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9a:a4:ef 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::a00:27ff:fe9a:a4ef/64 scope link tentative noprefixroute
        valid_lft forever preferred_lft forever
5: br0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether d2:52:16:b2:fe:f5 brd ff:ff:ff:ff:ff:ff
    inet 10.100.0.3/24 brd 10.100.0.255 scope global noprefixroute br0
        valid_lft forever preferred_lft forever

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=0.162 ms
64 bytes from 10.100.0.3: icmp_seq=2 ttl=64 time=0.121 ms
64 bytes from 10.100.0.3: icmp_seq=3 ttl=64 time=0.142 ms
64 bytes from 10.100.0.3: icmp_seq=4 ttl=64 time=0.361 ms

--- 10.100.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3036ms
rtt min/avg/max/mdev = 0.121/0.196/0.361/0.096 ms
mac-address: GENERAL.HWADDR: D2:52:16:B2:FE:F5
```

## 3. Команды и консольный вывод Части 2 п.4

```
mac_addr=$(nmcli d show br0 | grep "GENERAL.HWADDR:")
echo "mac-address: $mac_addr"
```

```
mac-address: GENERAL.HWADDR:
```

```
D2:52:16:B2:FE:F5
```

## 4. YAML файл Части 3 п.3

YAML файл конфигурации netplan:

```
network:
  version: 2
  ethernets:
    enp0s3:
      addresses:
        - 10.100.0.4/24
        - 10.100.0.5/24
      routes:
        - to: 0.0.0.0/0
          via: 10.100.0.3
```

Проверка ping:

Debian

```
root@d12:~# ping -c 3 10.100.0.2
PING 10.100.0.2 (10.100.0.2) 56(84) bytes of data.
64 bytes from 10.100.0.2: icmp_seq=1 ttl=64 time=2.00 ms
64 bytes from 10.100.0.2: icmp_seq=2 ttl=64 time=1.78 ms
64 bytes from 10.100.0.2: icmp_seq=3 ttl=64 time=2.45 ms

--- 10.100.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.782/2.076/2.451/0.278 ms
```

CentOs

```
[root@localhost ~]# ping -c 3 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=4.38 ms
64 bytes from 10.100.0.4: icmp_seq=2 ttl=64 time=2.99 ms
64 bytes from 10.100.0.4: icmp_seq=3 ttl=64 time=2.61 ms

--- 10.100.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 2.610/3.327/4.384/0.762 ms
```

## 5. Команды и консольный вывод Части 3 п.5

Таблицы arp кэша:

Debian:

```
root@d12 ~/ $ ip neigh show
10.100.0.3 dev enp0s3 lladdr 08:00:27:9a:a4:ef REACHABLE
10.100.0.2 dev enp0s3 lladdr 08:00:27:9a:a4:ef REACHABLE
```

CentOS:

```
[root@localhost ~]# ip neigh show
10.100.0.5 dev enp0s3 lladdr 08:00:27:c3:04:a5 REACHABLE
10.100.0.4 dev enp0s3 lladdr 08:00:27:c3:04:a5 REACHABLE
10.100.0.1 dev enp0s3 FAILED
```

## 6. Команды и конфигурационные файлы (если таковые использовались) Части 4 п.5

nmcli connection add type bond ifname bond007 con-name bond007 mode balance-rr ipv4.method auto

nmcli connection add type bond-slave ifname enp0s3 master bond007

nmcli connection add type bond-slave ifname enp0s8 master bond007

```
root@d12:~# nmcli con add type bond ifname bond007 con-name bond007 mode balance-rr ipv4.method auto
Подключение «bond007» (828af424-05a8-45e6-b1b3-aea18107096e) успешно добавлено.
root@d12:~# nmcli con add type bond-slave ifname enp0s3 master bond007
Подключение «bond-slave-enp0s3» (c2099b25-3a40-4149-ba10-a37e3c6f49b3) успешно добавлено.
root@d12:~# nmcli con add type bond-slave ifname enp0s8 master bond007
Подключение «bond-slave-enp0s8» (3857a6a1-2add-4906-baf4-4e8b1ae4a35c) успешно добавлено.
```

nmcli connection up bond007

nmcli connection up bond-slave-enp0s3

nmcli connection up bond-slave-enp0s8

Альтернативный вариант:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: true
  bonds:
    bond007:
      dhcp4: true
      interfaces:
        - enp0s3
        - enp0s8
      parameters:
        mode: balance-rr
        mii-monitor-interval: 1
```

## 7. Консольный вывод Части 4 п.5

```
root@d12 ~] $ cat /proc/net/bonding/bond007
Ethernet Channel Bonding Driver: v6.1.0-25-amd64

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: enp0s8
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 08:00:27:f9:a6:bd
Slave queue ID: 0

Slave Interface: enp0s3
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 08:00:27:c3:04:a5
Slave queue ID: 0
```

```

root@d12:~# cat /proc/net/dev
Inter-| Receive
face |bytes  packets errs drop fifo frame compressed multicast|bytes  packets errs drop fifo colls carrier compressed
lo:   0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
enp0s3: 1828    15      0      0      0      0      0      0      2480    29      0      0      0      0      0      0      0
enp0s8: 2360     4      0      0      0      0      0      0      3768    38      0      0      0      0      0      0      0
bond007: 1180     2      0      0      0      0      0      0      1932    20      0      0      0      0      0      0      0
root@d12:~#

```

## 8. Скрипт Части 4 п.7

```

#!/bin/bash

echo "Current time: $(date)"

grep "bond007" /proc/net/dev | while read -r line; do
    interface=$(echo $line | awk '{print $1}')
    receive=$(echo $line | awk '{print $3}')
    transmit=$(echo $line | awk '{print $11}')
    echo "Interface name: $interface, Receive-packets: $receive, Transmit-packets: $transmit"
done

```

Вывод:

```

root@d12 ~/] $ ./spy.sh
Current time: Cp 09 окт 2024 18:48:00 MSK
Interface | Receive-Packets | Transmit-packets
lo | 210 | 210
enp0s3 | 99 | 175
enp0s8 | 75 | 417
bond007 | 163 | 540
root@d12 ~/] $ ./spy.sh
Current time: Cp 09 окт 2024 18:48:02 MSK
Interface | Receive-Packets | Transmit-packets
lo | 210 | 210
enp0s3 | 100 | 176
enp0s8 | 77 | 418
bond007 | 166 | 542
root@d12 ~/] $ ./spy.sh
Current time: Cp 09 окт 2024 18:48:04 MSK
Interface | Receive-Packets | Transmit-packets
lo | 210 | 210
enp0s3 | 100 | 177
enp0s8 | 78 | 419
bond007 | 168 | 544

```

По полученным результатам видно, что отправленные bond007 пакеты распределены равномерно между обоими интерфейсами.

## Вопросы и задания:

1. Как с помощью команды ip:

a. назначить новый IPv4 адрес?

```
sudo ip addr add 10.0.2.100/255.255.255.0 dev enp0s3
```

```
$ ip a/addr add [ip-address] dev [net interface]
```

b. назначить новый MAC адрес?

```
sudo ip link set dev enp0s3 address AA:BB:CC:DD:EE:FF
```

```
$ sudo ip link set dev vlan1 down - выключаем интерфейс
$ sudo ip link set dev vlan1 address 12:e1:7d:f8:a3:e5 - назначаем новый MAC
адрес
$ sudo ip link set dev vlan1 up - включаем интерфейс
```

с. назначить новый gateway?

```
sudo ip route add default via 192.168.1.1
$ ip r/route add default via [gateway] dev [net interface]
```

d. вывести информацию arp кэше?

```
ip neigh show / arp -n
ip n show
```

е. очистить arp кэш?

```
sudo ip neigh del dev enp0s3 192.168.0.105
ip neigh flush / sudo ip neigh flush all
ip n del hostname (имя хоста) dev name (название устройства)
```

f. включить интерфейс?

```
ip link set dev enp0s3 up
$ sudo ip link set dev [net interface] up
```

g. выключить интерфейс?

```
ip link set dev enp0s3 down
$ sudo ip link set dev [net interface] down
```

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

```
sudo nmcli con mod "имя_вашего_соединения"
ipv4.addresses 192.168.1.100/24
sudo nmcli con mod "имя_вашего_соединения"
ipv4.gateway 192.168.1.1
sudo nmcli con mod "имя_вашего_соединения"
ipv4.method manual
```



```
sudo nmcli con up "имя_вашего_соединения"
```

`nmcli connection modify <interface name> ipv4.method manual ip4 <ip_address/mask> gw4/ipv4.gateway <gateway address>` - команда, которая меняет параметры интерфейса (ipv4, можно аналогично настроить ipv6)

Если нам нужно настроить новый сетевой интерфейс:

```
nmcli connection add type <type> ifname <ifname> con-name <con_name>
ipv4.method      manual      <ip_address/mask>      gw4/ipv4.gateway
<gateway_address>
```

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

```
network:
```

```
  version: 2
```

```
  ethernets:
```

```
    eth0:
```

```
    dhcp4: no
```

```
    addresses:
```

```
    - 192.168.1.100/24
```

```
    gateway4: 192.168.1.1
```

```
    nameservers:
```

```
    addresses:
```

```
      - 8.8.8.8
```

```
      - 8.8.4.4
```

```
sudo netplan apply
```

Все это происходит в yaml файле. Для получения статического адреса необходимо отключить: `dhcp4: no`. Для назначения адреса далее в `addresses` необходимо указать адрес. Маска указывается через `"/`:

```
addresses:
```

```
  - 10.100.0.4/24
```

Настройка default gateway:

```
routes:
```

- to: default
- via: 10.100.0.3

4. Какие режимы bonding стандартно существуют в Linux? Опишите их назначение, возможности по отказоустойчивости и необходимость поддержки со стороны оборудования.

- mode=0 (balance-rr): Передача пакетов по очереди через все интерфейсы.
- mode=1 (active-backup): Один интерфейс активен, остальные в резерве.
- mode=2 (balance-xor): Пакеты распределяются по интерфейсам на основе MAC-адресов.
- mode=3 (broadcast): Все пакеты отправляются через все интерфейсы.
- mode=4 (802.3ad): Динамическое распределение нагрузки (требует поддержки со стороны коммутатора).
- mode=5 (balance-tlb): Автоматическое распределение нагрузки с учетом текущей загрузки.
- mode=6 (balance-alb): Автоматическое распределение нагрузки с учетом текущей загрузки и возможности приема.

- balance-rr: пакеты отправляются по очереди, как Round-robin. Режим обеспечивает балансировку нагрузки и отказоустойчивость
- active-backup: работает только один интерфейс, при его отказе активируется другой, обеспечивая стабильный доступ
- balance-xor: задействуются все интерфейсы. Одна и та же карта передает пакеты одним и тем же получателям. Рассчитывается по формуле: «MAC адрес источника» XOR «MAC адрес назначения») по модулю «число интерфейсов».
- broadcast: передаются все пакеты на все карты
- 802.3ad: задействуются все сетевые карты. Требует специальной настройки коммутатора, реализует стандарт IEEE. Это обеспечивает отказоустойчивость
- balance-tlb: балансировка нагрузки. Пакеты распределяются в зависимости от загруженности всех сетевых карт. Входящий трафик в обычном режиме
- balance-alb: balance-tlb + балансировка входящего трафика

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

- Half Duplex: Передача данных возможна в одном направлении в любой момент времени (например, радиостанции).

- Full Duplex: Передача данных возможна одновременно в обоих направлениях (например, телефонные разговоры).

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

Это может пригодиться, когда есть необходимость запустить несколько веб-сайтов с уникальными IP адресами на одном сервере. Также можно использовать разные IP-адреса для тестирования или для разных типов трафика (например, внутренний и внешний). Если использовать IP-адреса IPv4 и IPv6, то можно достичь поддержки разных протоколов.

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

Виртуальные интерфейсы позволяют:

- Создавать изолированные среды для тестирования или разработки.
- Разделять трафик между различными службами или пользователями.
- Упрощать управление сетевыми настройками и маршрутизацией без добавления физического оборудования.

Виртуальные интерфейсы позволяют нескольким сетям использовать один физический интерфейс => снижается потребность в дополнительном оборудовании. Также виртуальные интерфейсы могут использоваться для создания изолированных сетевых сегментов, что повышает безопасность и управляемость. С помощью виртуальных интерфейсов легче управлять сетевыми настройками, например, изменять конфигурацию или добавлять новые сети без необходимости физического вмешательства. В целом, виртуальные интерфейсы делают сетевые решения более гибкими.