

**Министерство науки и высшего образования
Российской Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Практическая работа № 1

Консольные утилиты настройки сетевых компонентов Linux

Выполнил студент группы № М3302

Суворин Ярослав Владимирович

**Санкт-Петербург
2024**

Артефакты выполнения:

1. Скрипт часть 1 пункт 2

```
#!/usr/bin/bash
iscontinue=1
while [ $iscontinue -eq 1 ]
do
read argument
if [ "$argument" == "a" ]; then
    ethtool -i enp0s3 | awk 'NR==2'
    ethtool enp0s3 | awk '/Speed:|/Duplex|/Link detected/'
    ethtool -P enp0s3
elif [ "$argument" == "b" ]; then
    ipaddress=$(ip -br a s enp0s3 | awk '{for(i=3;i<=NF;i++){print " ", $i}}')
    echo "IP addresses and masks: $ipaddress"
    gate=$(ip r | head --lines 1 | awk '{print $3}')
    echo "Gateway: $gate"
    dns=$(cat /etc/resolv.conf | awk '/nameserver /{print $2}')
    echo "DNS: $dns"
elif [ "$argument" == "c" ]; then
    ip a add 10.100.0.2/255.255.255.0 dev enp0s3
    ip r del default
    ip route add default via 10.100.0.1
    if ! grep -q "nameserver 8.8.8.8" "/etc/resolv.conf"; then
        echo "nameserver 8.8.8.8" >> /etc/resolv.conf
    fi
elif [ "$argument" == "d" ]; then
    dhclient -r enp0s3
    dhclient enp0s3
elif [ "$argument" == "e" ]; then
    iscontinue=0
fi
done
```

2. Команды части 2 пункт 2

```
[root@localhost myuser]# nmcli con mod enp0s3 ipv4.addresses 10.100.0.2/24
[root@localhost myuser]# nmcli con mod enp0s3 ipv4.gateway 10.100.0.1
```

```
[root@localhost myuser]# nmcli con mod enp0s3 ipv4.dns "8.8.8.8"
```

```
[root@localhost myuser]# nmcli con mod enp0s3 ipv4.method manual
[root@localhost myuser]# nmcli con up enp0s3
Подключение успешно активировано (активный путь D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/12)
```

```
[root@localhost myuser]# ip link add br0 type dummy
```

```
[root@localhost myuser]# ip addr a 10.100.0.3 brd + dev br0
```

```
[root@localhost myuser]# ip link set dev br0 up
```

Проверка связи:

```
[root@localhost myuser]# ping 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=0.097 ms
64 bytes from 10.100.0.3: icmp_seq=2 ttl=64 time=0.105 ms
64 bytes from 10.100.0.3: icmp_seq=3 ttl=64 time=0.055 ms
64 bytes from 10.100.0.3: icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from 10.100.0.3: icmp_seq=5 ttl=64 time=0.055 ms
64 bytes from 10.100.0.3: icmp_seq=6 ttl=64 time=0.053 ms
64 bytes from 10.100.0.3: icmp_seq=7 ttl=64 time=0.067 ms
64 bytes from 10.100.0.3: icmp_seq=8 ttl=64 time=0.057 ms
64 bytes from 10.100.0.3: icmp_seq=9 ttl=64 time=0.052 ms
64 bytes from 10.100.0.3: icmp_seq=10 ttl=64 time=0.080 ms
64 bytes from 10.100.0.3: icmp_seq=11 ttl=64 time=0.065 ms
64 bytes from 10.100.0.3: icmp_seq=12 ttl=64 time=0.054 ms
64 bytes from 10.100.0.3: icmp_seq=13 ttl=64 time=0.063 ms
^C
--- 10.100.0.3 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12266ms
rtt min/avg/max/mdev = 0.050/0.065/0.105/0.017 ms
```

3. Команда и консольный вывод части 2 пункт 4

```
[root@localhost myuser]# ip l show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:00:27:e1:43:c7 brd ff:ff:ff:ff:ff:ff
3: br0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether fa:7e:1a:5a:c0:8a brd ff:ff:ff:ff:ff:ff
```

Определили MAC адрес (fa:7e:1a:c0:8a)

4. YAML файл части 3 пункт 3

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

5. Команды и консольный вывод части 3 пункт 5

```
root@d12:/etc/netplan# ping 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=0.061 ms
64 bytes from 10.100.0.4: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 10.100.0.4: icmp_seq=3 ttl=64 time=0.033 ms
64 bytes from 10.100.0.4: icmp_seq=4 ttl=64 time=0.037 ms
64 bytes from 10.100.0.4: icmp_seq=5 ttl=64 time=0.032 ms
64 bytes from 10.100.0.4: icmp_seq=6 ttl=64 time=0.037 ms
^C
--- 10.100.0.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5109ms
rtt min/avg/max/mdev = 0.032/0.038/0.061/0.010 ms
root@d12:/etc/netplan# ping 10.100.0.5
PING 10.100.0.5 (10.100.0.5) 56(84) bytes of data.
64 bytes from 10.100.0.5: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 10.100.0.5: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 10.100.0.5: icmp_seq=3 ttl=64 time=0.032 ms
64 bytes from 10.100.0.5: icmp_seq=4 ttl=64 time=0.032 ms
64 bytes from 10.100.0.5: icmp_seq=5 ttl=64 time=0.033 ms
64 bytes from 10.100.0.5: icmp_seq=6 ttl=64 time=0.032 ms
^C
--- 10.100.0.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5092ms
rtt min/avg/max/mdev = 0.027/0.031/0.033/0.002 ms
root@d12:/etc/netplan# cat /proc/net/arp
IP address      HW type    Flags      HW address            Mask        Device
root@d12:/etc/netplan# ping 10.100.0.2
PING 10.100.0.2 (10.100.0.2) 56(84) bytes of data.
From 10.100.0.4 icmp_seq=1 Destination Host Unreachable
From 10.100.0.4 icmp_seq=2 Destination Host Unreachable
From 10.100.0.4 icmp_seq=3 Destination Host Unreachable
^C
--- 10.100.0.2 ping statistics ---
6 packets transmitted, 0 received, +3 errors, 100% packet loss, time 5071ms
pipe 4
root@d12:/etc/netplan# ping 10.100.0.3
PING 10.100.0.3 (10.100.0.3) 56(84) bytes of data.
From 10.100.0.4 icmp_seq=1 Destination Host Unreachable
From 10.100.0.4 icmp_seq=2 Destination Host Unreachable
From 10.100.0.4 icmp_seq=3 Destination Host Unreachable
^C
--- 10.100.0.3 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4056ms
pipe 4
root@d12:/etc/netplan# cat /proc/net/arp
IP address      HW type    Flags      HW address            Mask        Device
10.100.0.3      0x1        0x0        00:00:00:00:00:00     *           enp0s3
10.100.0.2      0x1        0x0        00:00:00:00:00:00     *           enp0s3
```

6. Команды части 4 пункт 5

```
[root@localhost ~]# cat /proc/net/bonding/bond007
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:e1:43:c7
Slave queue ID: 0

Slave Interface: enp0s8
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 08:00:27:1a:a0:55
Slave queue ID: 0
```

7. Скрипт части 4 пункт 7

```
#!/usr/bin/bash
date
cat /proc/net/dev
```

```
[root@localhost myuser]# ./s.sh
Вс 22 сен 2024 21:38:35 MSK
Inter-| Receive
face |bytes   packets errs drop fifo frame compressed multicast|bytes   packets errs drop fifo colls carrier compressed
lo:   16128    192    0    0    0    0          0          0 16128    192    0    0    0    0    0    0
enp0s3: 438421   4395    0    0    0    0          0          0 221815  2242    0    0    0    0    0    0
enp0s8: 23607    195    0    0    0    0          0          0 236916  2420    0    0    0    0    0    0
bond007: 428943  4365    0    0    0    0          0          0 430093  4394    0    0    0    0    0    0

[root@localhost myuser]# ./s.sh
Вс 22 сен 2024 21:38:38 MSK
Inter-| Receive
face |bytes   packets errs drop fifo frame compressed multicast|bytes   packets errs drop fifo colls carrier compressed
lo:   16128    192    0    0    0    0          0          0 16128    192    0    0    0    0    0    0
enp0s3: 463313   4649    0    0    0    0          0          0 234359  2370    0    0    0    0    0    0
enp0s8: 23607    195    0    0    0    0          0          0 249558  2549    0    0    0    0    0    0
bond007: 453835  4619    0    0    0    0          0          0 455279  4651    0    0    0    0    0    0

[root@localhost myuser]# ./s.sh
Вс 22 сен 2024 21:38:42 MSK
Inter-| Receive
face |bytes   packets errs drop fifo frame compressed multicast|bytes   packets errs drop fifo colls carrier compressed
lo:   16128    192    0    0    0    0          0          0 16128    192    0    0    0    0    0    0
enp0s3: 488597   4907    0    0    0    0          0          0 247001  2499    0    0    0    0    0    0
enp0s8: 23607    195    0    0    0    0          0          0 262200  2678    0    0    0    0    0    0
bond007: 479119  4877    0    0    0    0          0          0 480563  4909    0    0    0    0    0    0
```

1. Как с помощью команды ip:

a. назначить новый IPv4 адрес?

ip a add <ip address/mask> dev <interface>

b. назначить новый MAC адрес?

ip link set dev <device> down

ip link set dev <device> address <mac address>

ip link set dev <device> up

c. назначить новый gateway?

ip route add <ip address/mask> via <gateway>

ip route add <ip address/mask> dev <device>

ip route add <ip address/mask> gw <gateway>

ip route add <ip address/mask> via <gateway>

ip route add default gw <gateway>

ip route add default via <gateway>

d. вывести информацию arp кэше?

ip neigh show

e. очистить arp кэш?

ip -s -s neigh flush all

f. включить интерфейс?

ip link set dev <device> up

g. выключить интерфейс?

ip link set dev <device> down

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

nmcli con mod <device> ipv4.addresses <ip address/mask>

nmcli con mod <device> ipv4.gateway <gateway>

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

YAML файл:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    <device>:
      addresses:
        -<ip address/mask>
      routes:
        -to: <ip address>
          via: <gateway>
```

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

I. **balance-rr — (round-robin)**

1. Режим циклического выбора активного интерфейса для трафика. Пакеты последовательно передаются и принимаются через каждый интерфейс один за другим
2. отказ нескольких интерфейсов к падению всего bonding не приведёт; балансировка нагрузки
3. специальные коммутаторы не требуются

II. **active-backup**

1. В этом режиме активен только один интерфейс, остальные находятся в режиме горячей замены. Если активный интерфейс выходит из строя, его заменяет резервный.
2. Есть backup в случае отказа активного интерфейса; минус для нагрузки
3. специальные коммутаторы не требуются

III. **balance-xor**

1. Один и тот же интерфейс работает с определённым получателем. Передача пакетов распределяется между интерфейсами на основе формулы ((MAC-адрес источника) XOR (MAC-адрес получателя)) % число интерфейсов.
2. Если за определённый промежуток времени соединение не было восстановлено, то происходит активация следующего интерфейса из связи.
3. специальные коммутаторы не требуются

IV. **broadcast**

1. Трафик идет через все интерфейсы одновременно.
2. Отказоустойчивый
3. специальные коммутаторы не требуются

V. **LACP (802.3ad)**

1. В группу объединяются одинаковые по скорости и режиму интерфейсы. Все физические интерфейсы используются одновременно в соответствии со спецификацией IEEE 802.3ad.
2. Для того чтобы получать информацию о скорости, а также о дуплексе на каждом сетевом интерфейсе, требует поддержки ethtool в драйвере; Требуется специальную настройку на коммутаторе, а также поддержки стандарта IEEE 802.3ad

VI. **balance-tlb**

1. Адаптивная балансировка нагрузки. Входящий трафик получается только активным интерфейсом, исходящий — распределяется в зависимости от текущей загрузки каждого интерфейса.
2. Если карта выходит из строя, то другая сетевая карта берёт себе MAC адрес вышедшей из строя карты.
3. Дополнительная настройка коммутаторов не требуется
4. Для получения информации о скорости загрузки на сетевых интерфейсах требует поддержки ethtool в драйвере.

VII. **balance-alb**

1. Политика адаптивной балансировки нагрузки. Включает в себя политику balance-tlb плюс осуществляет балансировку входящего трафика.
2. Балансировка входящего трафика достигается путём ARP переговоров.
3. Дополнительная настройка коммутаторов не требуется
4. Для получения информации о скорости загрузки на сетевых интерфейсах требуется поддержка ethtool в драйвере; требует поддержки в драйвере замены MAC адреса на включенном устройстве

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

- Симплексный режим – В этом типе режима данные могут передаваться только в одном направлении.
- Полудуплексный режим «Half Duplex» - данные передаются в обоих направлениях, но делает это по очереди.
- Полнодуплексный режим «Full Duplex» - позволяет устройству отправлять или получать пакеты одновременно.

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

- Запуск нескольких веб сайтов на одном сервере
- Если основной IP недоступен, перенаправить на дополнительный

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

Объединение нескольких аппаратных интерфейсов, организация VPN-туннелей.