

Лабораторная работа «Настройка сети в Linux системах»

Утилита `ping` - это очень простой инструмент для диагностики сети. Она позволяет проверить доступен удаленный хост или нет и все. Для этого утилита проверяет, может ли хост отвечать на сетевые запросы с помощью протокола ICMP.

КОМАНДА PING

Теперь нам осталось немного попрактиковаться. Но перед тем, как мы перейдем к практике давайте рассмотрим синтаксис команды и ее опции. Синтаксис довольно прост:

`$ ping опции адрес_узла`

Формат команды `ping` очень прост. В качестве адреса узла можно передавать как `ip` адрес, так и доменное имя. Опции настраивают поведение утилиты. Рассмотрим основные из них:

- `-4` - использовать только `ipv4` (по умолчанию);
- `-6` - использовать только `ipv6`;
- `-A` - адаптивный режим, время между отправками пакета адаптируется к времени передачи и приема пакета, но не меньше чем 200мс;
- `-b` - разрешить `ping` широковещательного адреса;
- `-c` - количество пакетов, которые нужно отправить;
- `-D` - выводить время в виде UNIX timestamp;
- `-f` - режим флуда, в этом режиме пакеты передаются без задержек, может использоваться для совершения DoS атак на отдельные узлы. Количество точек, которые выводит утилита обозначает количество потерянных пакетов;
- `-i` - интервал в секундах между отправкой пакетов;
- `-I` - использовать этот сетевой интерфейс для отправки пакетов;
- `-l` - режим перегрузки, отправляется очень много пакетов и система не следит за ответными пакетами;
- `-n` - не получать домены для `ip` адресов;
- `-r` - игнорировать таблицы маршрутизации и отправить пакет на указанный интерфейс;
- `-s` - размер одного пакета;

- -t - установить TTL вручную;
- -v - более подробный вывод.

Примеры Ping-запросов:

```
administrator@G501-1:~$ ping 10.1.30.203
PING 10.1.30.203 (10.1.30.203) 56(84) bytes of data.
64 bytes from 10.1.30.203: icmp_seq=1 ttl=128 time=0.334 ms
64 bytes from 10.1.30.203: icmp_seq=2 ttl=128 time=0.151 ms
64 bytes from 10.1.30.203: icmp_seq=3 ttl=128 time=0.157 ms
64 bytes from 10.1.30.203: icmp_seq=4 ttl=128 time=0.160 ms
64 bytes from 10.1.30.203: icmp_seq=5 ttl=128 time=0.161 ms
64 bytes from 10.1.30.203: icmp_seq=6 ttl=128 time=0.149 ms
64 bytes from 10.1.30.203: icmp_seq=7 ttl=128 time=0.159 ms
64 bytes from 10.1.30.203: icmp_seq=8 ttl=128 time=0.154 ms
64 bytes from 10.1.30.203: icmp_seq=9 ttl=128 time=0.152 ms
^C
--- 10.1.30.203 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8178ms
rtt min/avg/max/mdev = 0.149/0.175/0.334/0.056 ms
administrator@G501-1:~$
```

```
administrator@G501-1:/etc$ ping -c 4 www.google.com
PING www.google.com (173.194.73.147) 56(84) bytes of data.
64 bytes from lq-in-f147.1e100.net (173.194.73.147): icmp_seq=1 ttl=109 time=24.9 ms
64 bytes from lq-in-f147.1e100.net (173.194.73.147): icmp_seq=2 ttl=109 time=24.9 ms
64 bytes from lq-in-f147.1e100.net (173.194.73.147): icmp_seq=3 ttl=109 time=24.8 ms
64 bytes from lq-in-f147.1e100.net (173.194.73.147): icmp_seq=4 ttl=109 time=24.6 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 24.632/24.832/24.943/0.123 ms
administrator@G501-1:/etc$
```

```
administrator@G501-1:/etc$ ping -4 www.google.com
PING www.google.com (173.194.73.99) 56(84) bytes of data.
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=1 ttl=109 time=19.2 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=2 ttl=109 time=19.0 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=3 ttl=109 time=19.1 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=4 ttl=109 time=18.9 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=5 ttl=109 time=19.3 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=6 ttl=109 time=19.1 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=7 ttl=109 time=19.3 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=8 ttl=109 time=19.1 ms
64 bytes from lq-in-f99.1e100.net (173.194.73.99): icmp_seq=9 ttl=109 time=19.1 ms
^C
--- www.google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8010ms
rtt min/avg/max/mdev = 18.935/19.099/19.283/0.109 ms
administrator@G501-1:/etc$
```

УСТАНОВКА IFCONFIG

С появлением утилиты `ip`, `ifconfig` была удалена из многих дистрибутивов. Но в большинстве она все еще поставляется по умолчанию. Установка `ifconfig` в Ubuntu выполняется командой:

```
sudo apt install net-tools
```

В Arch Linux:

```
sudo pacman -S net-tools
```

СИНТАКСИС IFCONFIG

Утилита `ifconfig` имеет очень простой синтаксис при вызове без параметров, она возвратит список подключенных к системе сетевых интерфейсов и их характеристики, например, `ip` адрес, адрес шлюза, размер пакета, частоту для беспроводных сетей и другие параметры. Если указать интерфейс и нужные команды, то можно изменять различные настройки интерфейса. Вот сам синтаксис:

\$ ifconfig опции интерфейс команда параметры адрес

Интерфейс - это сетевой интерфейс, о котором вы хотите посмотреть информацию или изменить его параметры. Все это можно сделать с помощью команд. Давайте рассмотрим основные из них:

- **up** - включить интерфейс;
- **down** - выключить интерфейс;
- **(-)arp** - включить или выключить использование протокола ARP для интерфейса;
- **(-)promisc** - включить или выключить неразборчивый режим для интерфейса;
- **(-)allmulti** - включить или выключить режим multicast;
- **metric** - изменить параметр `metric`;
- **mtu** - изменить максимальный размер пакета;
- **netmask** - установить маску сети;

- **add** - добавить ip адрес для интерфейса;
- **del** - удалить ip адрес интерфейса;
- **media** - установить тип внешнего протокола;
- **[-]broadcast** - установить широковещательный адрес или отключить эту функцию;
- **hw** - установить MAC адрес для интерфейса;
- **txqueuelen** - размер очереди интерфейса;

Параметры и адрес необязательны и зависят от используемой команды. А опции влияют на поведение утилиты более глобально. Опций всего несколько, рассмотрим их:

- **-a** - применять команду ко всем интерфейсам, например, полезно, если вы хотите отключить или включить все сетевые интерфейсы;
- **-s** - вывести краткий список интерфейсов.

Обратите внимание, что для использования утилиты вам понадобятся права суперпользователя. А теперь давайте рассмотрим несколько примеров.

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ IFCONFIG

Просмотр текущих параметров сети, для этого в консоли от суперпользователя выполним команду:

sudo ifconfig и sudo ifconfig -a

```
administrator@G501-1:/etc$ ifconfig
enpls0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.30.197 netmask 255.255.255.0 broadcast 10.1.30.255
    inet6 fe80::8ced:75cc:c59d:df76 prefixlen 64 scopeid 0x20<link>
    ether 20:cf:30:95:cf:dc txqueuelen 1000 (Ethernet)
    RX packets 72351 bytes 61784126 (61.7 MB)
    RX errors 0 dropped 387 overruns 0 frame 0
    TX packets 61293 bytes 8246072 (8.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 2 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4424 bytes 375637 (375.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4424 bytes 375637 (375.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

administrator@G501-1:/etc$ ifconfig -a
enpls0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.30.197 netmask 255.255.255.0 broadcast 10.1.30.255
    inet6 fe80::8ced:75cc:c59d:df76 prefixlen 64 scopeid 0x20<link>
    ether 20:cf:30:95:cf:dc txqueuelen 1000 (Ethernet)
    RX packets 72351 bytes 61784126 (61.7 MB)
    RX errors 0 dropped 387 overruns 0 frame 0
    TX packets 61295 bytes 8246220 (8.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 2 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4424 bytes 375637 (375.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4424 bytes 375637 (375.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 1c:7e:e5:10:2c:f7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

administrator@G501-1:/etc$
```

Далее мы научимся останавливать, запускать, перезапускать сетевые интерфейсы. И первой командой будет команда которая остановит сетевой интерфейс eth0:

```
sudo ifconfig eth0 down
```

Раз мы с вами его остановили, то надо научиться его и обратно подключать:

```
sudo ifconfig eth0 up
```

Следующая команда остановит все, без исключения сетевые интерфейсы на вашем ПК:

```
sudo /etc/init.d/networking stop
```

Следующая опять же запустит, только теперь все сразу:

```
sudo /etc/init.d/networking start
```

Ну и осталось только научиться перезапускать их:

```
sudo /etc/init.d/networking restart
```

```

administrator@G501-1:/etc$ sudo ifconfig wlp3s0 down
administrator@G501-1:/etc$ sudo ifconfig wlp3s0 hw ether 1a:2b:3c:4d:5e:6f
administrator@G501-1:/etc$ sudo ifconfig wlp3s0 up
administrator@G501-1:/etc$ ifconfig -a
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.30.201 netmask 255.255.255.0 broadcast 10.1.30.255
    inet6 fe80::8ced:75cc:c59d:df76 prefixlen 64 scopeid 0x20<link>
    ether 20:cf:30:95:cf:dc txqueuelen 1000 (Ethernet)
    RX packets 70361 bytes 61183323 (61.1 MB)
    RX errors 0 dropped 354 overruns 0 frame 0
    TX packets 60021 bytes 8061875 (8.0 MB)
    TX errors 0 dropped 0 overruns 0 carrier 2 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 3839 bytes 329665 (329.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3839 bytes 329665 (329.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 1a:2b:3c:4d:5e:6f txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

administrator@G501-1:/etc$ █

```

Полезные опции dhclient

-q — не выводить данные в консоли и в лог, кроме ошибок;

-1 — отправить запрос только один раз; в случае ошибки будет exit 2;

-r — освободить текущий адрес;

-lf <lease-file> — файл базы данных аренды; если не указан будет использован файл по-умолчанию /var/lib/dhclient/dhclient.leases;

-pf <pid-file> — **PID**-файл процесса; если не указан — будет использован /var/run/dhclient.pid;

-cf <config-file> — файл конфигурации dhcp-клиента; если не указан — будет использоваться /etc/dhcp/dhclient.conf (в **CentOS** его всё-же надо создавать вручную);

-s <server> — указать сервер **DHCP** для отправки запроса; если не указан — запрос отправляет по всей сети 255.255.255.255;

-I <dhcp-client-identifier> — указание dhcp-идентификатора клиента;

-H <host-name> — указание опции host-name в запросе к **DHCP**-серверу; строка host-name должна содержать только префикс имени хоста клиента, к которому сервер добавит ddns или dns имя для получения полного **FQDN**-имени; нельзя использовать с опцией -F;

-F <fqdn.fqdn> — указать опцию fqdn.fqdn для отправки серверу; нельзя использовать с опцией -H; опция fqdn.fqdn определяет полное имя хоста клиента, которое сервер будет использовать для обновления **DDNS**;

-R <option>[,<option>...] — указать список опций, которые клиент хочет получить от сервера; список по умолчанию содержит такие опции:

subnet-mask, broadcast-address, time-offset, routers, domain-name, domain-name-servers, host-name, nis-domain, nis-servers, ntp-servers, interface-mtu

опция -R не добавляет список запросов к запросу по-умолчанию, а перезаписывает его;

-timeout <timeout> — указать *timeout* запроса вручную;

-v — подробный режим;

Полный список опций dhclient можно посмотреть [тут>>> \(русс\)](#), [тут>>> \(eng\)](#) и [тут>>> \(eng\)](#).

Пример сброса и установки ip-адреса через DHCP:

```
administrator@G501-1:/etc$ sudo dhclient -r enp1s0
Killed old client process
administrator@G501-1:/etc$ sudo dhclient -v enp1s0
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp1s0/20:cf:30:95:cf:dc
Sending on   LPF/enp1s0/20:cf:30:95:cf:dc
Sending on   Socket/fallback
DHCPDISCOVER on enp1s0 to 255.255.255.255 port 67 interval 3 (xid=0x90fb3e5f)
DHCPOFFER of 10.1.30.197 from 10.1.30.2
DHCPREQUEST for 10.1.30.197 on enp1s0 to 255.255.255.255 port 67 (xid=0x5f3efb90)
DHCPACK of 10.1.30.197 from 10.1.30.2 (xid=0x90fb3e5f)
cmp: EOF on /tmp/tmp.hskdgjX00T which is empty
bound to 10.1.30.197 -- renewal in 8531 seconds.
administrator@G501-1:/etc$ ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.30.197 netmask 255.255.255.0 broadcast 10.1.30.255
    inet6 fe80::8ced:75cc:c59d:df76 prefixlen 64 scopeid 0x20<link>
    ether 20:cf:30:95:cf:dc txqueuelen 1000 (Ethernet)
    RX packets 71933 bytes 61620665 (61.6 MB)
    RX errors 0 dropped 387 overruns 0 frame 0
    TX packets 60938 bytes 8207363 (8.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 2 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4350 bytes 369452 (369.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4350 bytes 369452 (369.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 1c:7e:e5:10:2c:f7 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

administrator@G501-1:/etc$
```

Команда IP

Утилита довольно многофункциональная, поэтому и синтаксис её вызова непростой:

\$ ip [опции] объект команда [параметры]

Опции - это глобальные настройки, которые сказываются на работе всей утилиты независимо от других аргументов, их указывать необязательно.

- **объект** - это тип данных, с которым надо будет работать, например:
адреса, устройства, таблица arp, таблица маршрутизации и так далее;
- **команды** - какое-либо действие с объектом;
- **параметры** - само собой, командам иногда нужно передавать параметры, они передаются в этом пункте.

Доступные параметры:

- **-v, -Version** - только вывод информации об утилите и ее версии.
- **-h, -human** - выводить данные в удобном для человека виде.
- **-s, -stats** - включает вывод статистической информации.
- **-d, -details** - показывать ещё больше подробностей.
- **-f, -family** - позволяет указать протокол, с которым нужно работать, если протокол не указан, то берется на основе параметров команды. Опция должна принимать одно из значений: bridge, dnet, inet, inet6, ipx или link.
По умолчанию используется inet, link - означает отсутствие протокола.
- **-o, -oneline** - выводить каждую запись с новой строки.
- **-r, -resolve** - определять имена хостов с помощью DNS.
- **-a, -all** - применить команду ко всем объектам.
- **-c, -color** - позволяет настроить цветной, доступные значения: auto, always и never.

- **-br, -brief** - выводить только базовую информацию для удобства чтения.
- **-4** - короткая запись для -f inet.
- **-6** - короткая запись для -f inet-f inet6.
- **-B** - короткая запись для -f inet-f bridge.
- **-0** - короткая запись для -f inet -f link.

Объекты:

- **address** или **a** - сетевые адреса.
- **link** или **l** - физическое сетевое устройство.
- **neighbour** или **neigh** - просмотр и управление ARP.
- **route** или **r** - управление маршрутизацией.
- **rule** или **ru** - правила маршрутизации.
- **tunnel** или **t** - настройка туннелирования.

Примеры использования IP:

Просмотр IP адресов:

Чтобы посмотреть все IP адреса, связанные с сетевыми интерфейсами используйте такую команду:

`ip a`

Или:

`ip addr show`

Пример использования утилиты `ip a` приведен на рисунке

```
adm501@G501-1: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
adm501@G501-1:~$ ip a
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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 20:cf:30:95:cf:dc brd ff:ff:ff:ff:ff:ff
    inet 10.1.30.198/24 brd 10.1.30.255 scope global dynamic noprefixroute enp1s0
        valid_lft 21180sec preferred_lft 21180sec
    inet6 fe80::8ced:75cc:c59d:df76/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 1c:7e:e5:10:2c:f7 brd ff:ff:ff:ff:ff:ff
    inet 10.5.247.43/22 brd 10.5.247.255 scope global dynamic noprefixroute wlp3s0
        valid_lft 18728sec preferred_lft 18728sec
    inet6 fe80::3703:dc7:7fa3:5bc0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
adm501@G501-1:~$
```

Для просмотра информации в кратком виде используйте опцию `-br`:

`ip -br a show`

```
adm501@G501-1:~$ ip -br a show
lo                UNKNOWN      127.0.0.1/8  ::1/128
enp1s0            UP            10.1.30.198/24  fe80::8ced:75cc:c59d:df76/64
wlp3s0            UP            10.5.247.43/22  fe80::3703:dc7:7fa3:5bc0/64
adm501@G501-1:~$
```

Можно посмотреть IP адреса только по определённому сетевому интерфейсу, например: `enp1s0`:

`ip a show enp1s0`

Или:

`ip a show dev enp1s0`

```
adm501@G501-1:~$ ip a show enpls0
2: enpls0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 20:cf:30:95:cf:dc brd ff:ff:ff:ff:ff:ff
    inet 10.1.30.198/24 brd 10.1.30.255 scope global dynamic noprefixroute enpls0
        valid_lft 20781sec preferred_lft 20781sec
    inet6 fe80::8ced:75cc:c59d:df76/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
adm501@G501-1:~$
```

Добавление IP адреса

Чтобы присвоить IP адрес для устройства нужно использовать команду add. Её общий синтаксис такой:

\$ ip addr add IP_адрес/маска dev интерфейс

Например, давайте присвоим тому же интерфейсу enpls3 IP адрес 10.0.2.100 с маской подсети 255.255.255.0:

`ip addr add 10.0.2.100/255.255.255.0 dev enpls3`

```
adm501@G501-1:~$ sudo ip addr add 10.1.30.201/24 dev enpls0
RTNETLINK answers: File exists
adm501@G501-1:~$ ip a show enpls0
2: enpls0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 20:cf:30:95:cf:dc brd ff:ff:ff:ff:ff:ff
    inet 10.1.30.198/24 brd 10.1.30.255 scope global dynamic noprefixroute enpls0
        valid_lft 20257sec preferred_lft 20257sec
    inet 10.1.30.201/24 scope global secondary enpls0
        valid_lft forever preferred_lft forever
    inet6 fe80::8ced:75cc:c59d:df76/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
adm501@G501-1:~$
```

Удаление IP адреса

Чтобы удалить IP адрес из интерфейса надо использовать команду del. Синтаксис её очень похож на предыдущую команду.

```
ip addr del 10.0.2.100/255.255.255.0 dev enp0s3
```

Можно удалять IP адреса по одному или удалить все сразу с помощью команды flush:

```
ip a flush
```

Список интерфейсов

Чтобы посмотреть список сетевых интерфейсов используйте объект link:

```
ip l
```

Или:

```
ip link show
```

```
adm501@G501-1:~$ ip l
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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
   DEFAULT group default qlen 1000
    link/ether 20:cf:30:95:cf:dc brd ff:ff:ff:ff:ff:ff
3: wlp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode
   DORMANT group default qlen 1000
    link/ether 1c:7e:e5:10:2c:f7 brd ff:ff:ff:ff:ff:ff
adm501@G501-1:~$
```

Включение или выключение интерфейсов

Для решения этой задачи тоже используется объект link, но с командой set.

Синтаксис её такой:

\$ ip link set dev интерфейс действие

В качестве действия можно использовать **up** или **down**.

```

adm501@G501-1:~$ sudo ip link set dev enp1s0 down
adm501@G501-1:~$ ip a show enp1s0
2: enp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 20:cf:30:95:cf:dc brd ff:ff:ff:ff:ff:ff
    inet 10.1.30.198/24 brd 10.1.30.255 scope global dynamic noprefixroute enp1s0
        valid_lft 19860sec preferred_lft 19860sec
    inet 10.1.30.201/24 scope global secondary enp1s0
        valid_lft forever preferred_lft forever
adm501@G501-1:~$

```

Настройка MTU

Параметр MTU означает размер одного пакета, передаваемого по сети. Этот размер можно изменить с помощью команды set. Например, увеличим MTU для enp0s3 до 4000 тысяч байт:

```
ip link set mtu 4000 dev enp0s3
```

```

adm501@G501-1:~$ sudo ip link set mtu 4000 dev enp1s0
adm501@G501-1:~$ ip a show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 4000 qdisc fq_codel state UP group default qlen 1000
    link/ether 20:cf:30:95:cf:dc brd ff:ff:ff:ff:ff:ff
    inet 10.1.30.198/24 brd 10.1.30.255 scope global dynamic noprefixroute enp1s0
        valid_lft 19645sec preferred_lft 19645sec
    inet6 fe80::8ced:75cc:c59d:df76/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
adm501@G501-1:~$

```

Настройка MAC адреса

Адрес MAC - это физический адрес, который используется для определения какому устройству надо передать сетевой пакет в локальной сети. Прежде чем настраивать MAC адрес ваше устройство надо отключить:

```
sudo ip link set dev enp1s0 down
```

Затем можно установить адрес:


```
sudo ip link set dev enp1s0 address AA:BB:CC:DD:EE:FF
```

А потом включить интерфейс обратно:

```
sudo ip link set dev enp1s0 up
```

```
adm501@G501-1:~$ sudo ip link set dev enp1s0 down
adm501@G501-1:~$ sudo ip link set dev enp1s0 address AA:BB:CC:DD:EE:FF
adm501@G501-1:~$ ip a show enp1s0
2: enp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
adm501@G501-1:~$ sudo ip link set dev enp1s0 up
adm501@G501-1:~$ ip a show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
    inet 10.1.30.161/24 brd 10.1.30.255 scope global dynamic noprefixroute enp1s0
        valid_lft 21596sec preferred_lft 21596sec
    inet6 fe80::8ced:75cc:c59d:df76/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
adm501@G501-1:~$
```

Добавление записи в таблицу ARP

Обычно записи в эту таблицу попадают автоматически, но вы можете добавить их и вручную. Для этого используйте команду add объекта neigh:

```
sudo ip neigh add 192.168.0.105 lladdr b0:be:76:43:21:41 dev enp1s3
```

```
adm501@G501-1:~$ ip n
10.5.244.1 dev wlp3s0 lladdr 84:c9:b2:93:bb:ba STALE
10.1.30.2 dev wlp3s0 lladdr 00:16:3e:51:04:49 STALE
10.1.30.1 dev wlp3s0 lladdr 84:c9:b2:93:b4:ce STALE
adm501@G501-1:~$ sudo ip neigh add 192.168.0.105 lladdr b0:be:76:43:21:41 dev enp1s0
adm501@G501-1:~$
adm501@G501-1:~$ ip n
10.5.244.1 dev wlp3s0 lladdr 84:c9:b2:93:bb:ba STALE
10.1.30.2 dev wlp3s0 lladdr 00:16:3e:51:04:49 STALE
10.1.30.1 dev wlp3s0 lladdr 84:c9:b2:93:b4:ce STALE
10.1.30.2 dev enp1s0 lladdr 00:16:3e:51:04:49 STALE
10.1.30.1 dev enp1s0 lladdr 84:c9:b2:93:b4:ce REACHABLE
192.168.0.105 dev enp1s0 lladdr b0:be:76:43:21:41 PERMANENT
adm501@G501-1:~$
```


Очистка таблицы ARP

Вы можете удалять IP адреса по одному с помощью команды del:

```
sudo ip neigh del dev enp0s3 192.168.0.105
```

В ходе лабораторной работы были изучены команды ifconfig, ping, dhcpclient.