

<Password> - необязательный, строка. Пароль пользователя, если сетевой диск подключается от пользователя, отличного от текущего.

**Пример:**

```
Set WshNetwork = CreateObject("WScript.Network")  
WshNetwork.MapNetworkDrive "Z:", \\SERVER\Programs
```

**RemoveNetworkDrive**

RemoveNetworkDrive(<Name>,<Force>,<UpdateProfile>)

Отключает сетевой диск.

**Параметры:**

<Name> - строка, локальное имя диска (или сетевое имя, если ресурсу не сопоставлена никакая буква).

<Force> - необязательный, число (булево). Если указано True, отключение будет произведено вне зависимости от того, используется ресурс в настоящий момент или нет.

<UpdateProfile> - необязательный, число (булево). Если указано True, сетевое подключение будет удалено из профиля пользователя.

**Пример:**

```
Set WshNetwork = CreateObject("WScript.Network")  
WshNetwork.RemoveNetworkDrive "Z:"
```

**Свойства:**

**ComputerName**

Строка, имя компьютера.

**Замечание:**

Только чтение.

**Пример:**

```
Set WshNetwork = CreateObject("WScript.Network")  
MsgBox WshNetwork.ComputerName
```

**UserName**

Строка, имя пользователя.

**Замечание:**

Только чтение.

**Пример:**

```
Set WshNetwork = CreateObject("WScript.Network")  
MsgBox WshNetwork.UserName
```

**UserDomain**

Строка, имя домена.

**Замечание:**

Только чтение.

**Пример:**

```
Set WshNetwork = CreateObject("WScript.Network")  
MsgBox WshNetwork.UserDomain
```

## Основные команды ОС Linux для работы с сетевыми ресурсами

## ifconfig

Команда используется для настройки сетевых интерфейсов

Команда **ifconfig** имеет следующий синтаксис:

```
ifconfig [-L] [-m] interface [create] [address_family] [address  
[dest_address]] [parameters] ifconfig interface destroy ifconfig -a [-  
L] [-d] [-m] [-u] [address_family] ifconfig -l [-d] [-u]  
[address_family] ifconfig [-L] [-d] [-m] [-u] [-C]
```

Команда **ifconfig** используется для настройки сетевых интерфейсов. Команда должна использоваться при загрузке системы для настройки адресов каждого сетевого интерфейса, а также может использоваться после загрузки для изменения параметров сетевых интерфейсов. Если команда введена без аргументов, **ifconfig** выдает информацию о состоянии активных интерфейсов. Если в качестве аргумента указан какой-либо интерфейс, то выдается информация только о состоянии этого интерфейса; если указан один аргумент -а, выдается информация о состоянии всех интерфейсов, даже отключенных. Пример:

```
user@desktop$ ifconfig rl0  
rl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500  
    options=8<VLAN_MTU>  
    inet6 fe80::250:22ff:febb:5f1%rl0 prefixlen 64 scopeid 0x3  
    inet 192.168.19.86 netmask 0xffffffff broadcast  
192.168.19.255  
    ether 00:50:22:bb:05:f1  
    media: Ethernet autoselect (100baseTX <full-duplex>)  
    status: active
```

Иначе команда конфигурирует указанный интерфейс. Изменить настройки какого-либо интерфейса может только суперпользователь.

Опции:

<i>интерфейс</i>	– имя интерфейса (например, rl0 в BSD или eth0 в Linux).
<i>up</i>	– вызывает активизацию интерфейса. Задается неявно при присвоении адреса интерфейсу.
<i>down</i>	– вызывает остановку работы драйвера для интерфейса.
<i>[-]arp</i>	– включает или отключает использование протокола ARP для интерфейса.
<i>[-]promisc</i>	– включает или отключает неразборчивый режим (promiscuous mode) работы интерфейса. В этом режиме все проходящие по сети пакеты будут приниматься интерфейсом.
<i>[-]allmulti</i>	– включает или отключает режим <i>all-multicast</i> . В этом режиме все многоадресные (multicast) пакеты в сети будут приниматься интерфейсом.
<i>metric N</i>	– устанавливает метрику интерфейса.
<i>mtu N</i>	– устанавливает максимальный размер пакета (Maximum Transfer Unit - MTU) для интерфейса.
<i>адрес</i>	– IP-адрес, присваиваемый интерфейсу.
<i>netmask адрес</i>	– устанавливает маску сети IP для этого интерфейса. По умолчанию используется обычная маска сети класса А, В или С

	(что определяется по IP-адресу интерфейса), но можно усановить любое значение.
<i>add</i>	– добавляет адрес IPv6 для интерфейса.
<i>адрес/длина_префикса</i>	
<i>del</i>	– удаляет адрес IPv6 для интерфейса.
<i>адрес/длина_префикса</i>	
<i>irq адрес</i>	– устанавливает аппаратное прерывание, используемое устройством. Не для всех устройств можно динамически менять значение IRQ.
<i>media min</i>	– устанавливает физический порт или тип носителя, используемый устройством. Не для всех устройств можно менять этот параметр, и для разных устройств могут поддерживаться различные значения. Типичные значения типа - 10base2 (коаксиальный кабель Ethernet), 10baseT (витая пара Ethernet 10 Мбит/сек), AUI (внешний передатчик) и т.д. Специальный тип носителя auto можно использовать, чтобы потребовать от драйвера автоматически обпределять тип носителя. Не все драйверы могут это делать.
<i>[-]broadcast [адрес]</i>	– если указан аргумент адрес, задает соответствующий протоколу широковещательный адрес для интерфейса. В противном случае устанавливает (или сбрасывает) флаг IFF_BROADCAST для интерфейса.

*Пример.* изменение IP-адреса интерфейса r10:

```

user@desktop ~ $ ifconfig r10
r10: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=8<VLAN_MTU>
    inet6 fe80::250:22ff:febb:5f1%r10 prefixlen 64 scopeid 0x3
    inet 192.168.19.86 netmask 0xffffffff00 broadcast
192.168.19.255
    ether 00:50:22:bb:05:f1
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active
user@desktop ~ $ ifconfig r10 192.168.0.1
user@desktop ~ $ ifconfig r10
r10: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=8<VLAN_MTU>
    inet6 fe80::250:22ff:febb:5f1%r10 prefixlen 64 scopeid 0x3
    inet 192.168.0.1 netmask 0xffffffff00 broadcast 192.168.19.255
    ether 00:50:22:bb:05:f1
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active

```

## arp

Команда **arp** отображает ARP-таблицу данного хоста. С помощью параметра *-i* можно специфицировать сетевой интерфейс, информация о котором интересует.

```

desktop ~ # arp -i eth0
Address                  HWtype  HWaddress          Flags Mask
Iface
DIMON.mshome.net         ether    00:50:BF:12:8A:9E   C
eth0

```

Таблица с информацией о канальном уровне содержит связь IP- и MAC-адресов. При использовании параметра `-n` IP-адреса не будут заменяться символьными именами хостов.

## route

Эта команда используется для просмотра и изменения таблицы маршрутизации хоста. Для этой команды также работает параметр `-n`, при использовании которого IP-адреса не будут заменяться символьными именами хостов.

Пример обычной таблицы маршрутизации для отдельного компьютера в сети:

```
desktop ~ # route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use
Iface
192.168.5.0      0.0.0.0         255.255.255.0    U        0      0      0
eth1
127.0.0.0        0.0.0.0         255.0.0.0        U        0      0      0
lo
0.0.0.0          192.168.5.254   0.0.0.0          UG       0      0      0
eth1
```

Особый интерес представляет адрес `0.0.0.0`, который соответствует хосту назначения по умолчанию.

Для добавление нового маршрута к определённому хосту используются параметры `add` и `-host`:

```
desktop ~ # route add -host 192.168.0.1 eth0
```

Эта команда создаёт новую строку в таблице маршрутизации, согласно которой все пакеты к узлу `192.168.0.1` должны отправляться в сетевой интерфейс `eth0`.

Также можно добавлять шлюзы для отправки пакетов в определённую сеть или к хосту:

```
desktop ~ # route add -net 192.168.1.0 gw 192.168.0.5
```

Таким образом, все пакеты для сети `192.168.1.0` будут направляться на узел `192.168.0.5`.

Аналогично, маршруты удаляются параметром `del` с указанием всей информации о маршруте:

```
desktop ~ # route del default gw 192.168.0.1
```

Эта команда удаляет маршрут по умолчанию через хост `192.168.0.1`.

## ping

Команда используется для отправки пакетов ICMP ECHO\_REQUEST сетевым хостам.

Команда **ping** имеет следующий синтаксис:

```
ping [-AaDdfnoQqRrv] [-c число_пакетов] [-i секунд] [-l preload] [-M
mask | time] [-m ttl] [-P policy] [-p pattern] [-S src_addr] [-s
packetsize] [-t timeout] [-z tos] host ping [-AaDdfLnoQqRrv] [-c
число_пакетов] [-I iface] [-i секунд] [-l preload] [-M mask | time] [-m
ttl] [-P policy] [-p pattern] [-S src_addr] [-s packetsize] [-T ttl] [-
t timeout] [-z tos] mcast-group
```

Команда **ping** использует датаграмму ECHO\_REQUEST протокола ICMP, чтобы вызвать ответ ICMP ECHO\_RESPONSE указанного хоста или сетевого шлюза. Если хост отвечает, **ping** выдает сообщение, что хост включен (хост is alive), в стандартный выходной поток.

Для проверки наличия хоста в сети достаточно ввести команду **ping** с аргументом - именем или адресом хоста:

```
user@desktop$ ping yandex.ru
64 bytes from 213.180.204.11: icmp_seq=0 ttl=48 time=5.659 ms
64 bytes from 213.180.204.11: icmp_seq=1 ttl=48 time=5.404 ms
64 bytes from 213.180.204.11: icmp_seq=2 ttl=48 time=4.889 ms
^C
--- yandex.ru ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 4.889/5.317/5.659/0.320 ms
```

Для отправки определенного числа пакетов необходимо указать опцию *-c число\_пакетов*. Для установки интервала между отправкой пакетов используется опция *-i секунд*.

## traceroute

Команда **traceroute** служит для отладки сетевых соединений посредством построения маршрута следования пакетов к хосту назначения. Для этой команды также работает параметр *-n*, при использовании которого IP-адреса не будут заменяться символьными именами хостов.

Пример следования пакетов до хоста ya.ru:

```
desktop ~ # traceroute ya.ru
traceroute to ya.ru (213.180.204.8), 64 hops max, 40 byte packets
 1  195.91.230.65 (195.91.230.65)  0.890 ms  1.907 ms  0.809 ms
 2  cs7206.rinet.ru (195.54.192.28)  0.895 ms  0.769 ms  0.605 ms
 3  ix2-m9.yandex.net (193.232.244.93)  1.855 ms  1.519 ms  2.95 ms
 4  c3-vlan4.yandex.net (213.180.210.146)  3.412 ms  2.698 ms  2.654 ms
 5  ya.ru (213.180.204.8)  2.336 ms  2.612 ms  3.482 ms
```

## netstat

Команда используется для показа состояния сети.

Команда **netstat** имеет следующий синтаксис:

```
netstat [-AaLnSW] [-f protocol_family | -p protocol] [-M core] [-N
system]
```

Команда **netstat** показывает содержимое различных структур данных, связанных с сетью, в различных форматах в зависимости от указанных опций. *Первая форма* команды показывает список активных сокетов (sockets) для каждого протокола. *Вторая форма* выбирает одну из нескольких других сетевых структур данных. *Третья форма* показывает динамическую статистику пересылки пакетов по сконфигурированным сетевым интерфейсам; аргумент интервал задает, сколько секунд собирается информация между последовательными показами.

Опции:

- `-a` — показывать состояние всех сокетов; обычно сокет, используемый серверными процессами, не показывается.
- `-A` — показывать адреса любых управляющих блоков протокола, связанных с сокетами; используется для отладки.
- `-i` — показывать состояние автоматически сконфигурированных (auto-configured) интерфейсов. Интерфейсы, статически сконфигурированные в системе, но не найденные во время загрузки, не показываются.
- `-n` — показывать сетевые адреса как числа. **netstat** обычно показывает адреса как символы. Эту опцию можно использовать с любым форматом показа.
- `-r` — показать таблицы маршрутизации. При использовании с опцией `-s`, показывает статистику маршрутизации.
- `-s` — показать статистическую информацию по протоколам. При использовании с опцией `-r`, показывает статистику маршрутизации.
- `-f` — ограничить показ статистики или адресов управляющих блоков только указанным семейством\_адресов, в качестве которого можно указывать:  
`inet` Для семейства адресов AF\_INET  
`unix` Для семейства адресов AF\_UNIX
- `-I интерфейс` — выделить информацию об указанном интерфейсе в отдельный столбец; по умолчанию (для третьей формы команды) используется интерфейс с наибольшим объемом переданной информации с момента последней перезагрузки системы. В качестве интерфейса можно указывать любой из интерфейсов, перечисленных в файле конфигурации системы, например, `eml1` или `lo0`.
- `-p имя_протокола` — Ограничить показ статистики или адресов управляющих блоков только протоколом с указанным именем\_протокола, например, `tcp`.

*Пример.* показ таблицы маршрутизации:

```
user@desktop ~$ netstat -r
Routing tables
Internet:
Destination      Gateway           Flags    Refs      Use  Netif
Expire
default          19-101.local     UGS             0 1373769   rl0
localhost        localhost        UH              1   290     lo0
192.168.0         link#1           UC              0    0      dc0
192.168.19        link#3           UC              0    0      rl0
19-86.local       localhost        UGHS            0    0      lo0
```

```
19-101.local      00:0d:bc:e4:27:bf  UHLW      1          0      r10
116
```

```
Internet6:
Destination      Gateway          Flags      Netif Expire
localhost prov.ru localhost prov.ru UH         lo0
fe80::%dc0       link#1          UC         dc0
fe80::2a0:ccff:fe3 00:a0:cc:3d:1f:bd UHL        lo0
fe80::%r10       link#3          UC         r10
fe80::250:22ff:feb 00:50:22:bb:05:f1 UHL        lo0
fe80::%lo0       fe80::1%lo0     U          lo0
fe80::1%lo0      link#5          UHL        lo0
ff01::          localhost prov.ru U          lo0
ff02::%dc0       link#1          UC         dc0
ff02::%r10       link#3          UC         r10
ff02::%lo0       localhost prov.ru UC         lo0
```

## host

Команда **host** служит для получения доменной информации о хосте: IP-адрес, MX-записи и другой информации, связанной с данным символьным именем. Имя хоста указывается в качестве аргумента команды.

Пример работы команды:

```
user@desktop ~$ host yandex.ru
yandex.ru has address 213.180.204.11
yandex.ru mail is handled by 10 mx2.yandex.ru.
yandex.ru mail is handled by 0 mx1.yandex.ru.
```

Вторым аргументом можно указать DNS-сервер, который будет использоваться при получении этой информации:

```
user@desktop ~$ host yandex.ru ns1.aiya.ru
Using domain server:
Name: ns1.aiya.ru
Address: 85.142.20.152#53
Aliases:

yandex.ru has address 213.180.204.11
Using domain server:
Name: ns1.aiya.ru
Address: 85.142.20.152#53
Aliases:

Using domain server:
Name: ns1.aiya.ru
Address: 85.142.20.152#53
Aliases:

yandex.ru mail is handled by 0 mx1.yandex.ru.
yandex.ru mail is handled by 10 mx2.yandex.ru.
```

## smbclient

Для просмотра ресурсов сети Microsoft используется программа **smbclient**. Допустим, вы хотите подключиться к общему каталогу `share` компьютера `nt_wsl`. При этом допустим, что ваше имя пользователя `user` и пароль `123456`. В этом случае использование команды **smbclient** выглядит следующим образом:

\$ smbclient //nt\_wsl/share -U user%123456

Если пароль не нужен, то указывается только имя пользователя без знака процента.

## 2. Задание на лабораторную работу.

Для выполнения лабораторной работы необходимо:

1. Изучить основные команды ОС Windows и ОС Linux для работы с сетевыми ресурсами
2. Научиться определять IP адрес компьютера и сетевые настройки в ОС Windows и ОС Linux
3. Научиться проверять наличие соединения с удалённым узлом. Научиться определять по имени компьютера его IP-адрес
4. Научиться осуществлять мониторинг использования сети и анализ сетевого взаимодействия в ОС Windows и ОС Linux
5. Научиться осуществлять подключение и отключение сетевых дисков с использованием командного процессора и серверов сценариев

## 3. Индивидуальные задания.

Необходимо написать файл сценариев в ОС Windows и ОС Linux, осуществляющий решение задачи согласно варианта. Результаты решения сохранить в текстовый файл

№ варианта	Условие задачи
1	Определить IP-адрес компьютеров, с заданной маской имени.
2	Посчитать количество компьютеров, видимых с данного компьютера.
3	Определить компьютер в сети, скорость взаимодействия с которым наибольшая.
4	Определить компьютер в сети, до которого самый длинный маршрут
5	Определить IP-адреса всех компьютеров, связь с которыми осуществляется через указанный шлюз
6	Определить компьютеры, имеющие более одного IP-адреса
7	Найти компьютер в сети, скорость взаимодействия с которым наименьшая
8	Сформировать список всех доступных сетевых ресурсов в заданном сегменте
9	Определить IP-адреса всех доступных DHCP-серверов
10	Подключить все доступные сетевые ресурсы из указанного списка компьютеров
11	Определить IP-адреса всех доступных DNS-серверов
12	Определить IP-адреса всех доступных WINS-серверов
13	Определить MAC-адреса компьютеров, из указанного списка (задан ip адрес)
14	Определить компьютер в сети, скорость взаимодействия с которым



	наименьшая.
15	Определить MAC-адреса всех доступных DHCP-серверов
16	Определить MAC-адреса всех доступных DNS-серверов
17	Определить MAC-адреса всех доступных WINS-серверов
18	Сформировать список имен компьютеров в заданном сегменте
19	Определить ip адреса компьютеров установивших подключения с данным компьютером
20	Подключить все доступные сетевые ресурсы со всех компьютеров, установивших подключения с данным компьютером
21	Определить самый короткий участок на пути к указанному узлу
22	Определить IP-адреса всех компьютеров, связь с которыми осуществляется через шлюз по умолчанию
23	Вывести список доступных сетевых ресурсов со всех компьютеров, установивших подключения с данным компьютером
24	Посчитать количество компьютеров, видимых через шлюз по умолчанию.
25	Определить количество маршрутизаторов на пути к указанному узлу
26	Вывести ip адреса маршрутизаторов на пути к указанному узлу, отсортированных по возрастанию времени задержки.
27	Определить скорости доступа к компьютерам из списка по ip адресам
28	Определить имя домена в котором находится данный компьютер
29	Определить самый длинный участок на пути к указанному узлу
30	Определить компьютеры, не имеющие имен

Отчёт должен содержать описание всех изученных команд и подробное описание действий для выполнения п.1-5, листинги решения индивидуального задания и результаты проведённой верификации. Результаты выполнения лабораторной работы должны быть **обязательно** продемонстрированы на компьютере.

## 4. Контрольные вопросы

1. Команды ОС Windows и ОС Linux для работы с сетевыми ресурсами. Основные параметры.
2. Проверка наличия соединения с удалённым узлом.
3. Программные средства мониторинга и анализа использования сети в ОС Windows и ОС Linux
4. Основные команды WSH и Bash для работы с сетевыми ресурсами