# Практическая работа № 1.

Консольные утилиты настройки сетевых компонентов в ОС Windows.

# Цель работы:

Получить практические навыки по конфигурированию сети в операционных системах Microsoft Windows, ознакомится с утилитами командной строки, предназначенными для диагностики и настройки сети, разработать исполняемые файлы, конфигурирующие сетевой интерфейс по заданным параметрам, ознакомиться с форматом записи пути до сетевого ресурса UNC.

# Требования:

для выполнения работы необходима установленная ОС Windows 10 или Windows Server. В случае если установленная на ПК операционная систем отличается от Windows, необходимо поставить виртуальную машину ORACLE Virtual Box.

# Краткие теоретические сведения

Несмотря на то, что в состав современных операционных систем входят утилиты конфигурирования сети с графическим интерфейсом, задачи по диагностике и настройке сети удобнее решать с помощью консольных утилит.

В MS Windows к этим утилитам относят:

* **Ipconfig** – утилита отображения конфигурации IP,
* **Ping** – утилита диагностики сетевого соединения,
* **Net** – комплекс утилит для работы с сетью Microsoft,
* **Netsh** – утилита настройки всего стека протоколов MS Windows.

Справку по утилитам командной строки можно получить так:

**command\_name /? ,**

а по команде net так:

**net help имя\_директивы**

Управление сетевыми устройствами в ОС Windows реализовано и через PowerShell – актуальное, расширяемое средство автоматизации от Microsoft с открытым исходным кодом, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев.

PowerShell доступен как в операционных системах серверного класса (например, Windows Server) так и на настольных операционных системах (например, Windows 10). Несмотря на то, что синтаксис и основные концепции PowerShell остаются похожими, есть некоторые различия между их версиями в Windows Server и Windows 10:

1. **Модули и функции**: PowerShell предоставляет различные модули и функции для управления различными аспектами операционной системы и приложений. В Windows Server могут быть доступны дополнительные модули и функции, специфические для серверных функций, таких как управление службами, доменными службами Active Directory, гипервизорами и т.д. В Windows 10 набор модулей и функций может быть более ограничен.

2. **Уровни безопасности**: В Windows Server существует более строгий контроль над разрешениями и безопасностью. PowerShell может использоваться для автоматизации административных задач, и в серверной среде это может подразумевать более жесткий контроль над тем, кто и как использует PowerShell.

3. **Функциональность серверных ролей**: Windows Server обычно используется для выполнения различных серверных ролей, таких как файловые серверы, веб-серверы, контроллеры домена и т.д. В связи с этим, PowerShell в Windows Server может иметь больше функциональности для управления этими серверными ролями.

4. **Управление ресурсами**: В Windows Server управление ресурсами, такими как диски, сетевые адаптеры, память и др., может иметь более расширенные возможности из-за потребностей серверной инфраструктуры.

5. **Версии PowerShell**: В Windows Server и Windows 10 могут быть установлены разные версии PowerShell. Новые версии могут включать в себя дополнительные функции, улучшения производительности и исправления ошибок, поэтому рекомендуется обращаться к официальной документации Microsoft для получения актуальной информации.

Предусмотрены следующие расширения для файлов PowerShell:

.ps1 - файлы скриптов,

.psd1 - файлы данных скриптов,

.psm1 - файлы модулей скриптов,

.ps1xml - файлы конфигурации.

С ОС поставляется интегрированная среда сценариев Windows Powershell ISE – облегченная IDE для PowerShell. Для разработки подходит MS Visual Studio Code.

Вместо утилит используются командлеты – команды PowerShell с объектным интерфейсом.

Для управления сетевым соединением, среди прочих, используются командлеты:

Get-NetIPConfiguration,

New-NetIPAddress,

Set-NetIPAddress ,

Get-NetRoute,

New-NetRoute,

Get-NetAdapterBinding,

Set-DnsClientServerAddress.

Справку можно получить через командлет

**Get-Help** (например **Get-Help New-NetRoute -Detailed**).

Список всех командлетов через командлет

**Get-Command** (например **Get-Command -Noun NetAdapter** или Get-Command -Name \*Help\*).

# Порядок выполнения работы

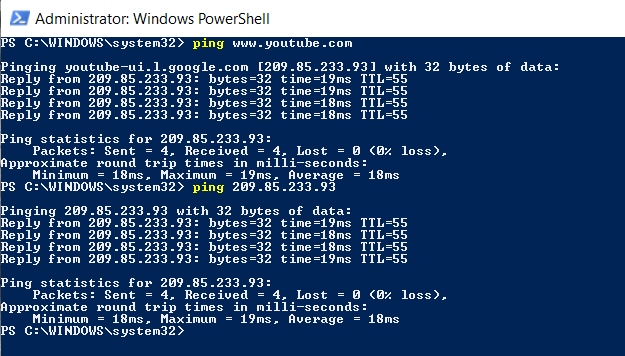
1. Запустить ОС Windows (или запустить виртуальную машину и авторизоваться в системе под администраторской учётной записью, используя заданное преподавателем имя пользователя и пароль). Проверить, активны ли следующие пункты в свойствах используемого сетевого подключения:

* **Клиент для сетей Microsoft**
* **Служба доступа к файлам и принтерам Microsoft**
* **Протокол TCP/IP**.

Определить назначение этих компонентов. Выяснить, за что отвечает каждый компонент.

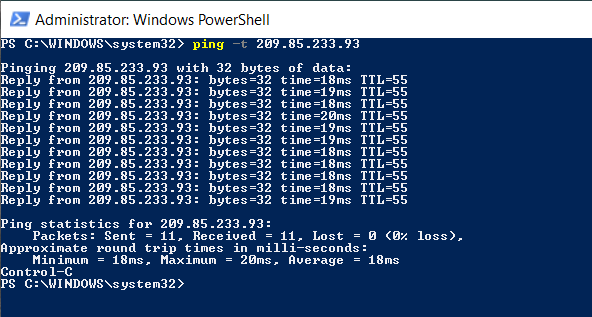
1. Используя знания, полученные в **пункте 1**, настроить сетевой интерфейс таким образом, чтобы внешние пользователи не могли получить доступ к ресурсам компьютера по протоколу SMB. Применять можно только параметры соединения.
2. Разобраться в назначении параметров и ключей утилиты ping. Команда ping в Microsoft Windows используется для проверки доступности устройства или хоста в сети, а также для измерения времени, которое требуется для отправки и получения данных от этого устройства. Выполните следующие действия:
   1. Проверка доступности удаленного хоста:

ping [хост]

Необходимо заменить [хост] на IP-адрес или доменное имя удаленного хоста, который вы хотите проверить. Команда ping отправит несколько ICMP-запросов этому хосту и выведет результаты, включая время задержки (ping) в миллисекундах. 

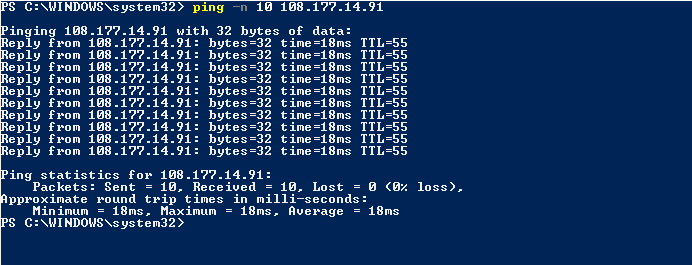
* 1. Запуск бесконечной проверки доступности:

ping -t [хост]

Эта команда будет продолжать отправлять ICMP-запросы на указанный хост бесконечно, пока вы не прервете выполнение команды (нажмите Ctrl+C). 

* 1. Ограничение числа запросов:

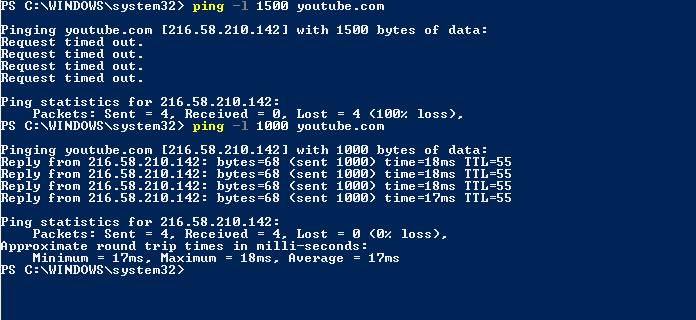
ping -n [количество запросов] [хост]

Эта команда отправит указанное количество ICMP-запросов на хост и затем завершится. Например, ping -n 5 google.com отправит 5 запросов на google.com. 

* 1. Изменение размера пакетов:

ping -l [размер] [хост]

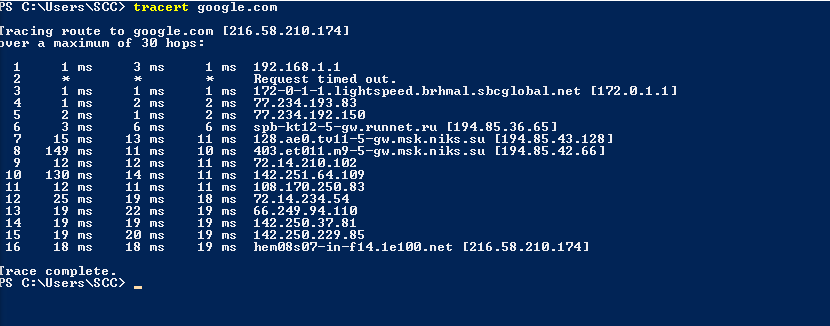
Эта команда позволяет вам отправить пакеты определенного размера в байтах. Например, ping -l 1000 google.com отправит пакеты размером 1000 байт.



* 1. Определение маршрута к хосту:

tracert [хост]

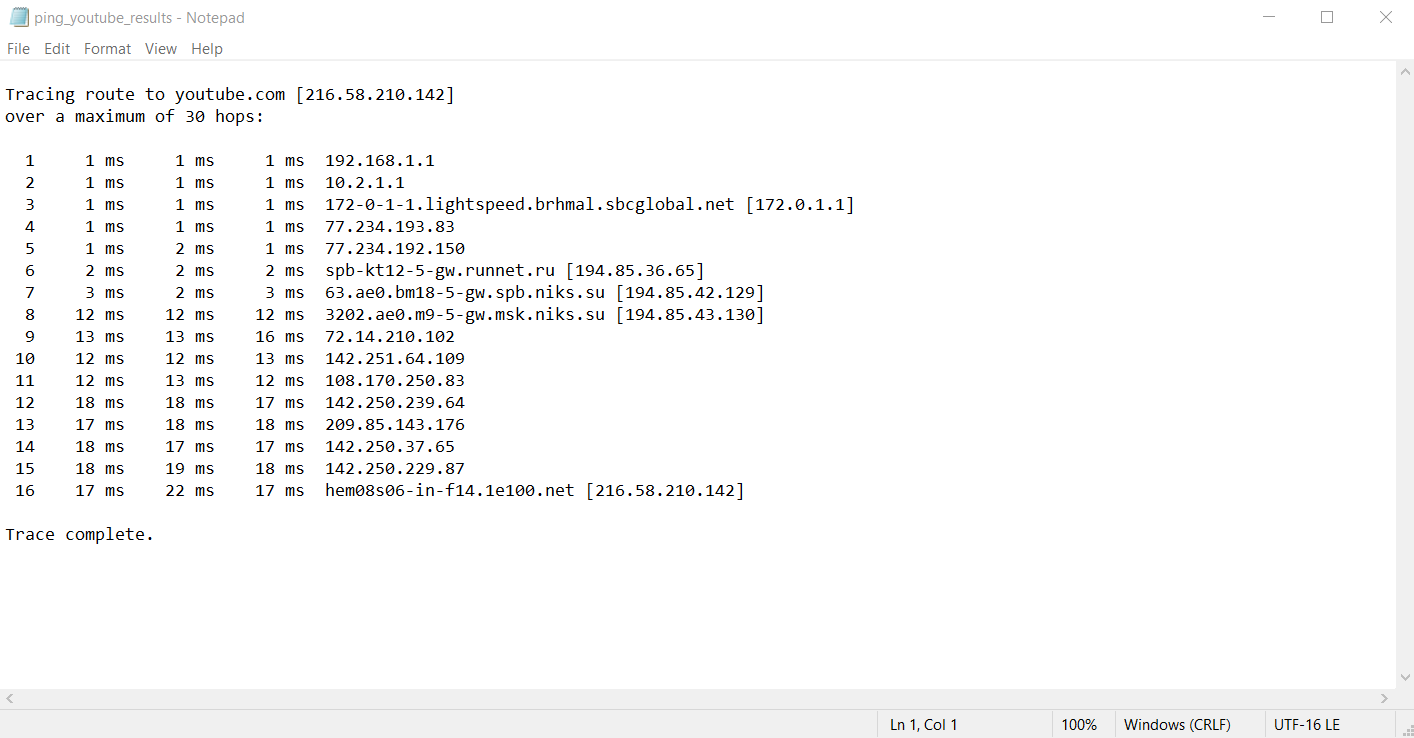
Команда tracert используется для отслеживания маршрута пакетов к указанному хосту и отображает список промежуточных узлов, через которые проходят пакеты.



* 1. Сохранение результатов в файл:

ping [хост] > [путь\_к\_файлу]

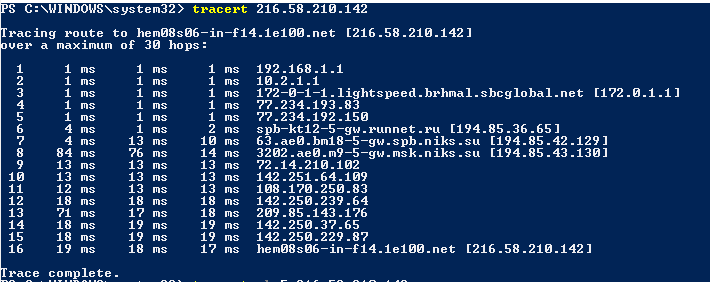
Вы можете перенаправить вывод команды ping в файл, чтобы сохранить результаты для дальнейшего анализа. Например, ping google.com > C:\ping\_results.txt сохранит результаты в файле ping\_results.txt на диске C: 



1. Разобраться в назначении параметров и ключей утилиты tracert.   
   Команда tracert (или traceroute в некоторых других операционных системах) в Microsoft Windows используется для отслеживания маршрута, который сетевой пакет проходит от вашего компьютера к удаленному хосту. Это может помочь в выявлении сетевых проблем, таких как маршрутизация или проблемы с узлами на пути. Необходимо выполнить следующие задания по использованию команды tracert:
   1. Отслеживание маршрута к удаленному хосту:

tracert [хост]

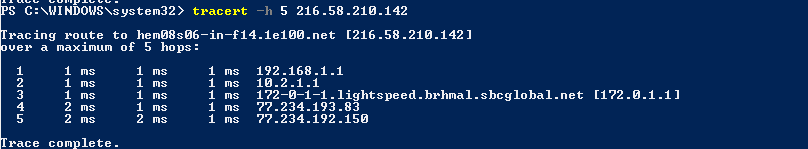
Замените [хост] на IP-адрес или доменное имя удаленного хоста, к которому вы хотите проследить маршрут. tracert выполнит последовательность запросов к хосту и отобразит список узлов (маршрут), через которые прошли запросы.



* 1. Изменение максимального количества прыжков (хопов):

tracert -h [число] [хост]

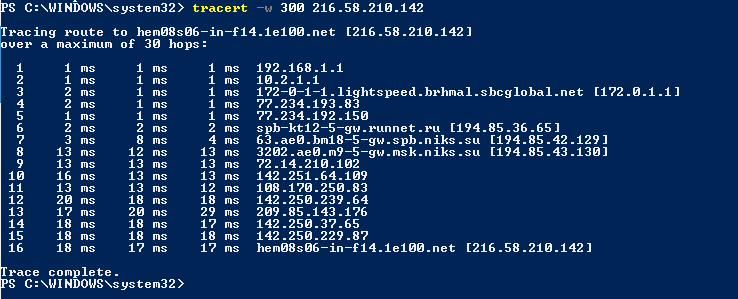
Эта команда позволяет установить максимальное количество хопов (узлов) в маршруте. Например, tracert -h 30 google.com позволит проследить маршрут с максимум 30 хопами.



* 1. Изменение времени ожидания для каждого хопа:

tracert -w [время\_ожидания] [хост]

Установите время ожидания (в миллисекундах) для каждого хопа. Это будет полезно, если вы хотите увеличить или уменьшить тайм-аут при ожидании ответа от каждого узла.



1. Аналогичным образом самостоятельно разобраться с следующими утилитами и выполнить 4-5 заданий с каждой из них:

* ipconfig
* net с директивами use, view, stop, start, share, config, session, user, statistics, localgroup

1. С помощью утилиты **netsh** создать командные файл для интерпретатора **CMD.exe**, с помощью которого можно было бы, задав параметры запуска скрипта или в диалоге с пользователем, настраивать выбранный сетевой интерфейс двумя способами:

* получение всех настроек через **DHCP-сервер** (автоматически) (**IP**, **mask**, **gateway**, **DNS**)
* ввод всех настроек **вручную** (статически)

Примечание: в качестве сетевых настроек использовать любые статические адреса IPv4 по желанию.

1. Выполнить аналогичное задание с помощью PowerShell. При этом добавьте ключ запуска или пункт меню скрипта, позволяющий узнать:
   1. Модель сетевой карты
   2. Наличие физического подключения (линка)
   3. Скорость и режим работы адаптера (speed, duplex)

# Содержание отчета

Требуется подготовить отчет в формате DOC\DOCX или PDF, а также файлы скриптов. Отчет содержит титульный лист, полную последовательность выполнения (тексты скриптов) и ответы на вопросы и задания.

Отчет выслать в течение 4-х недель на адрес [akharitonov@itmo.ru](mailto:akharitonov@itmo.ru). Если отчет будет выслан позже, то защита будет с понижением оценки.

В теме письма: №группы ФИО (латинскими буквами) №работы (например: 5555 Fedor Sumkin 3)

# Вопросы и задания

1. Как с помощью графической оболочки Windows можно запретить доступ через определенный сетевой интерфейс к ресурсам используемого компьютера? Как можно запретить используемому компьютеру доступ к ресурсам других компьютеров в сети Microsoft?
2. Опишите назначение команды net с директивами use, view, stop, start, share, config, session, user, statistics, localgroup. Приведите примеры.
3. Как с помощью командной строки в Windows узнать адрес **DNS**, на который настроен ваш компьютер?
4. Зачем нужна команда **net use**? Как с помощью этой утилиты подключить на локальный диск **R:** папку **TEST** на компьютере **SRV** (приведите командную строку)?
5. Как в Windows из PowerShell переименовать сетевое соединение?
6. Какие существуют и чем отличаются режимы работы адаптера (duplex) ?

# Понятийный минимум по работе

1. IP адрес
2. Gate
3. Адрес DNS
4. DHCP клиент
5. DHCP сервер
6. Режим работы (duplex)
7. Windows
   1. Ipconfig (получение информации, управление лицензиями dhcp)
   2. Netsh (установка адреса, dns, приемы работы)
   3. Командлеты \*-NetAdapter, \*- NetIPAddress, \*--NetRoute, \*-DnsClientServerAddress (\* - это доступные Verbs - Get, Set и т.д.).
   4. Route (получение информации)

# Материалы для работы

<https://niuitmo-my.sharepoint.com/:u:/g/personal/112280_niuitmo_ru/EZGhDvtc-exPk0CuFWiNaygBxytjT5gtzrYugcF5tQaClw?e=amcuk0>

- Обновленный (во всех смыслах) windows srver 2012 R2. Вкручены компоненты MS C++ 2019, последние Update Pack, Wireshark и VisulaStudio Code.

<https://www.virtualbox.org/>

* среда виртуализации ORACLE Virtual Box