

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по практической работе №1
по курсу «Компьютерные сети»

Тема: Консольные утилиты настройки сетевых компонентов в ОС
Windows.

Выполнил:

Привалов Кирилл Алексеевич

К3242

Преподаватель:

Харитонов Антон Юрьевич

Санкт-Петербург

2024 г.

Содержание

Основная часть	3
Контрольные вопросы	12

Основная часть

Для выполнения работы потребовалось установить Parallels Desktop, так как операционная система по умолчанию – MacOS. В ходе поиска ISO образа мне удалось найти образ Windows 10 с ARM архитектурой. После установки образа я приступил к выполнению заданий.

Сначала проверил состояние следующих служб:

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

Все компоненты были включены по умолчанию.

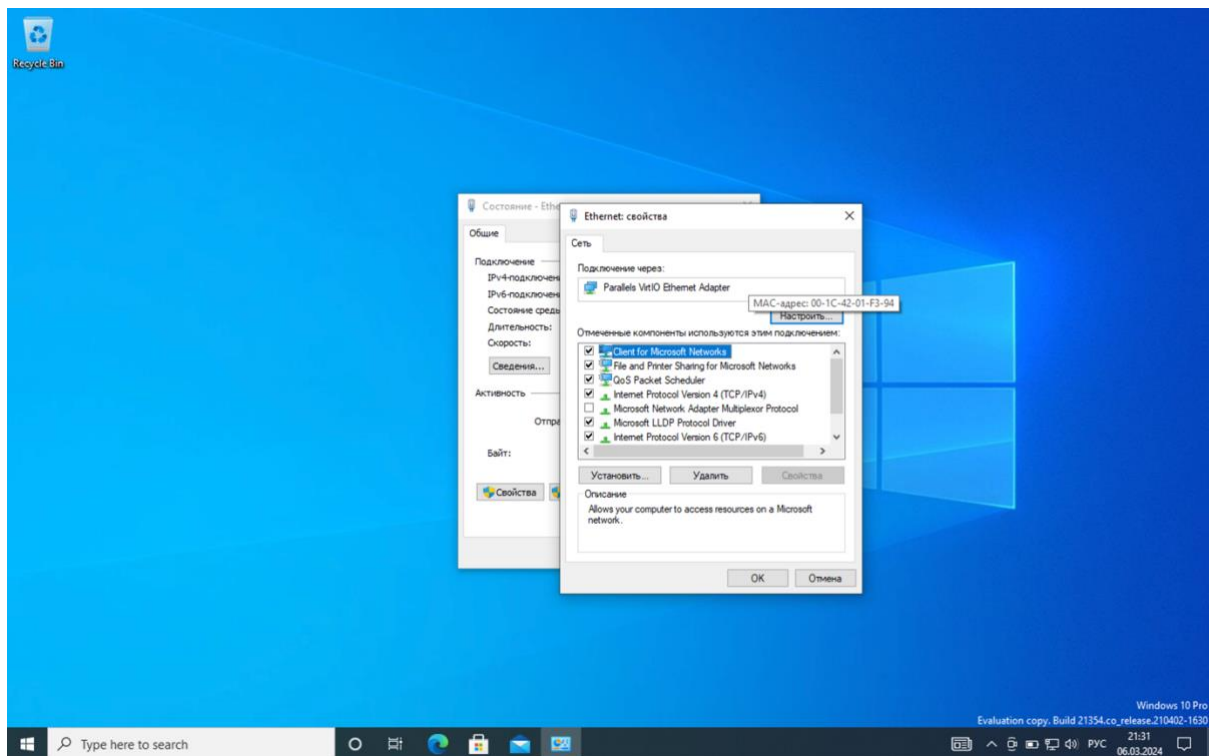


Рисунок 1 – Компоненты подключения Ethernet

Далее я настроил сетевой интерфейс таким образом, чтобы внешние пользователи не могли получить доступ к ресурсам виртуальной машины по протоколу SMB.

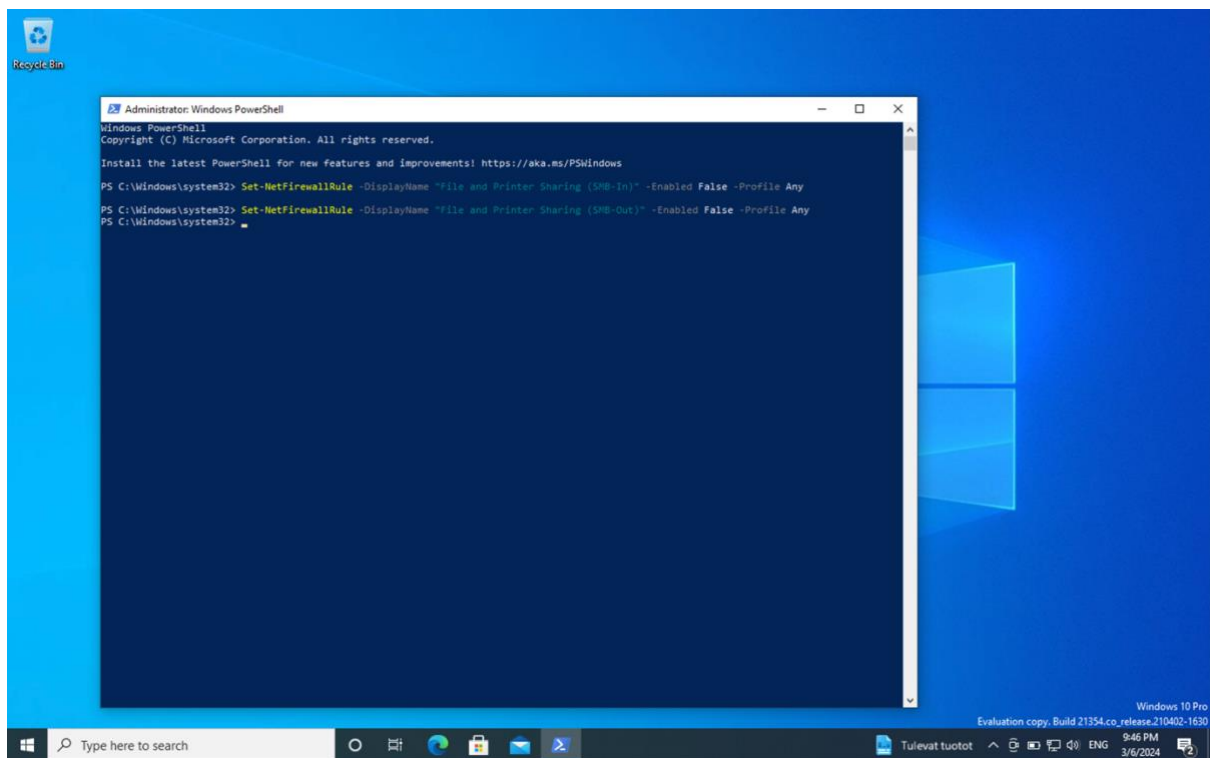


Рисунок 2 - Ограничение доступа к ресурсам хоста по протоколу SMB

Использовал команду *ping*, которая нужна для проверки подключения к другому компьютеру на уровне IP, с разными флагами и аргументами.

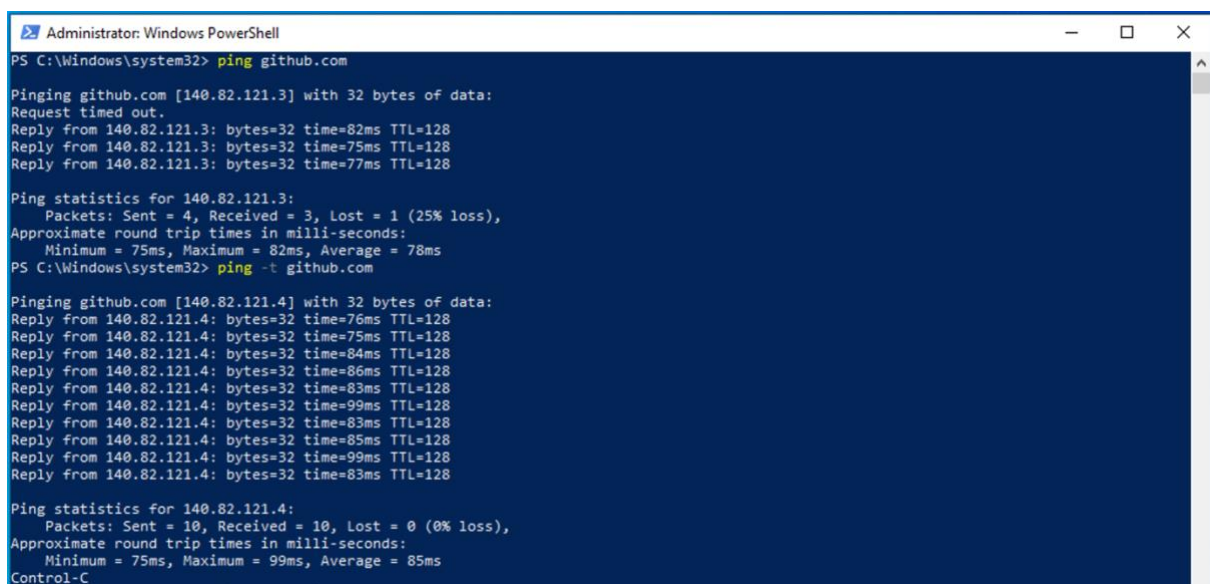
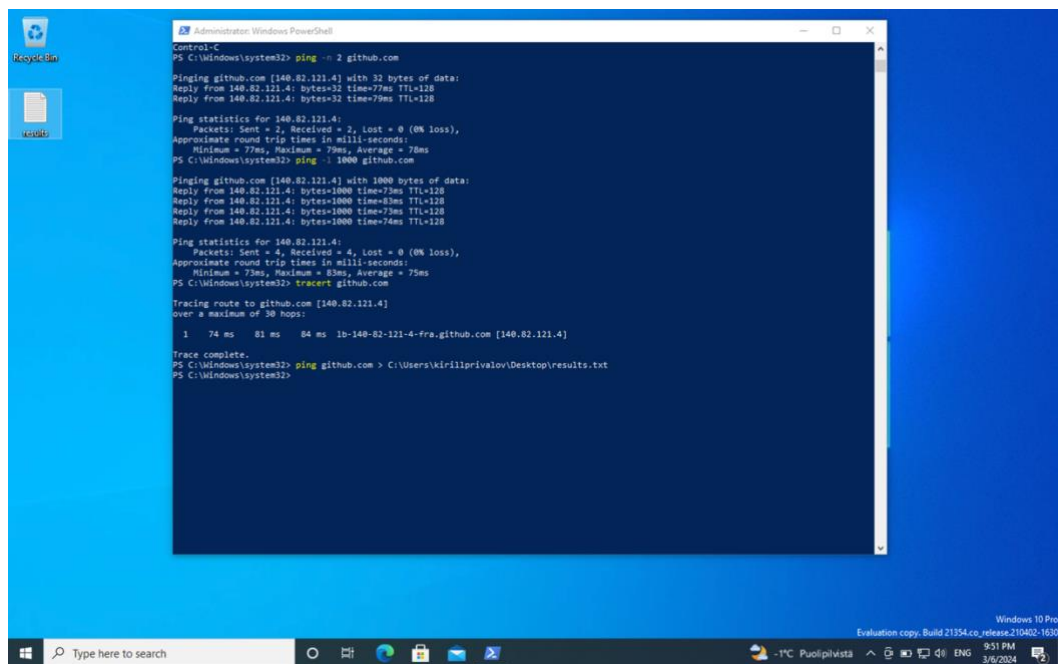


Рисунок 3 - Логи команды *ping* с разными аргументами



```
Administrator: Windows PowerShell
Control-C
PS C:\Windows\system32> ping -n 2 github.com

Pinging github.com [140.82.121.4] with 32 bytes of data:
Reply from 140.82.121.4: bytes=32 time=77ms TTL=128
Reply from 140.82.121.4: bytes=32 time=79ms TTL=128

Ping statistics for 140.82.121.4:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 77ms, Maximum = 79ms, Average = 78ms
PS C:\Windows\system32> ping -l 1000 github.com

Pinging github.com [140.82.121.4] with 1000 bytes of data:
Reply from 140.82.121.4: bytes=1000 time=73ms TTL=128
Reply from 140.82.121.4: bytes=1000 time=83ms TTL=128
Reply from 140.82.121.4: bytes=1000 time=73ms TTL=128
Reply from 140.82.121.4: bytes=1000 time=74ms TTL=128

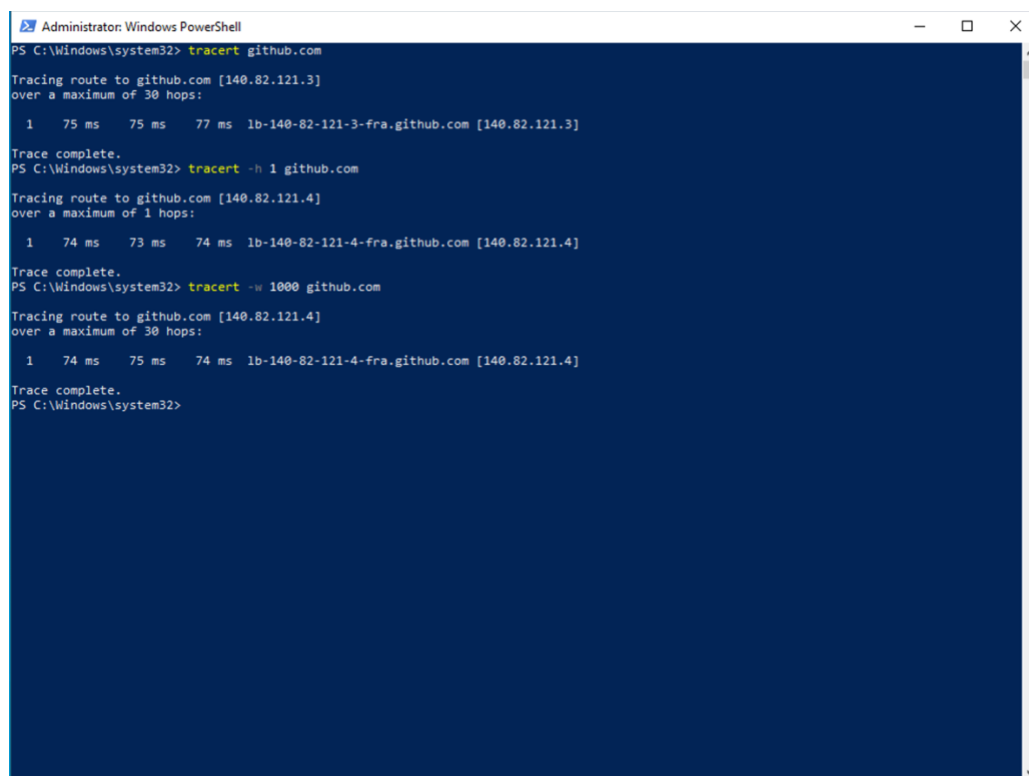
Ping statistics for 140.82.121.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 73ms, Maximum = 83ms, Average = 75ms
PS C:\Windows\system32> tracert github.com

Tracing route to github.com [140.82.121.4]
over a maximum of 30 hops:
  0  74 ms  81 ms  84 ms  1b-140-82-121-4-fra.github.com [140.82.121.4]

Trace complete.
PS C:\Windows\system32> ping github.com > C:\Users\kirillprivalov\Desktop\results.txt
PS C:\Windows\system32>
```

Рисунок 4 - Логи команды ping с разными аргументами

Дальнейшим шагом стало использование утилиты *tracert*, которая нужна для диагностики сети. Написал в консоли несколько способов использования утилиты с разными флагами.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> tracert github.com

Tracing route to github.com [140.82.121.3]
over a maximum of 30 hops:
  0  75 ms  75 ms  77 ms  1b-140-82-121-3-fra.github.com [140.82.121.3]

Trace complete.
PS C:\Windows\system32> tracert -h 1 github.com

Tracing route to github.com [140.82.121.4]
over a maximum of 1 hops:
  0  74 ms  73 ms  74 ms  1b-140-82-121-4-fra.github.com [140.82.121.4]

Trace complete.
PS C:\Windows\system32> tracert -w 1000 github.com

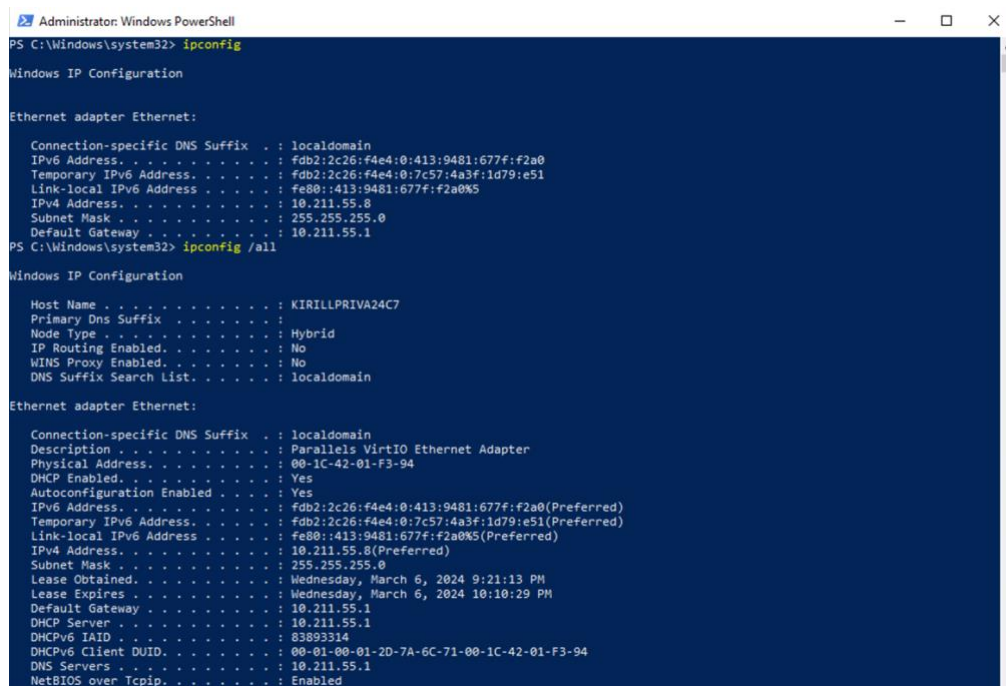
Tracing route to github.com [140.82.121.4]
over a maximum of 30 hops:
  0  74 ms  75 ms  74 ms  1b-140-82-121-4-fra.github.com [140.82.121.4]

Trace complete.
PS C:\Windows\system32>
```

Рисунок 5 – Логи команды tracert

Дополнительные утилиты для работы с сетями, которые я рассмотрел в рамках работы:

- *ipconfig* позволяет определять, какие значения конфигурации были получены с помощью ДНСП, АРІРА или другой службы ІР-конфигурирования либо заданы администратором вручную.
- *net* предназначена для управления сетевыми настройками и службами компьютера.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : localdomain
    IPv6 Address. . . . . : fdb2:2c26:f4e4:0:413:9481:677f:f2a0
    Temporary IPv6 Address. . . . . : fdb2:2c26:f4e4:0:7c57:4a3f:1d79:e51
    Link-local IPv6 Address . . . . . : fe80::413:9481:677f:f2a0%5
    IPv4 Address. . . . . : 10.211.55.8
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.211.55.1

PS C:\Windows\system32> ipconfig /all

Windows IP Configuration

    Host Name . . . . . : KIRILLPRIVA24C7
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    DNS Suffix Search List. . . . . : localdomain

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : localdomain
    Description . . . . . : Parallels VirtIO Ethernet Adapter
    Physical Address. . . . . : 00-1C-42-01-F3-94
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IPv6 Address. . . . . : fdb2:2c26:f4e4:0:413:9481:677f:f2a0(Preferred)
    Temporary IPv6 Address. . . . . : fdb2:2c26:f4e4:0:7c57:4a3f:1d79:e51(Preferred)
    Link-local IPv6 Address . . . . . : fe80::413:9481:677f:f2a0%5(Preferred)
    IPv4 Address. . . . . : 10.211.55.8(Preferred)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained. . . . . : Wednesday, March 6, 2024 9:21:13 PM
    Lease Expires . . . . . : Wednesday, March 6, 2024 10:10:29 PM
    Default Gateway . . . . . : 10.211.55.1
    DHCP Server . . . . . : 10.211.55.1
    DHCPv6 IAID . . . . . : 83893314
    DHCPv6 Client DUID. . . . . : 00-01-00-01-2D-7A-6C-71-00-1C-42-01-F3-94
    DNS Servers . . . . . : 10.211.55.1
    NetBIOS over Tcpip. . . . . : Enabled
```

Рисунок 6 - Логи утилиты ipconfig

```
Administrator: Windows PowerShell
PS C:\Windows\system32> ipconfig /displaydns

Windows IP Configuration

ocsp.digicert.com
-----
Record Name . . . . . : ocsp.digicert.com
Record Type . . . . . : 5
Time To Live . . . . . : 2937
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : ocsp.edge.digicert.com

Record Name . . . . . : ocsp.edge.digicert.com
Record Type . . . . . : 5
Time To Live . . . . . : 2937
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : fp2e7a.wpc.2be4.phicdn.net

Record Name . . . . . : fp2e7a.wpc.2be4.phicdn.net
Record Type . . . . . : 5
Time To Live . . . . . : 2937
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : fp2e7a.wpc.phicdn.net

Record Name . . . . . : fp2e7a.wpc.phicdn.net
Record Type . . . . . : 1
Time To Live . . . . . : 2937
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . : 192.229.221.95

fp-vs-nocache.azureedge.net
-----
Record Name . . . . . : fp-vs-nocache.azureedge.net
Record Type . . . . . : 5
Time To Live . . . . . : 212
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : fp-vs-nocache.ec.azureedge.net

Record Name . . . . . : fp-vs-nocache.ec.azureedge.net
Record Type . . . . . : 5
Time To Live . . . . . : 212
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : cs9.wpc.v0cdn.net
```

Рисунок 7 - Логи утилиты ipconfig

```
Administrator: Windows PowerShell
PS C:\Windows\system32> ipconfig /allcompartments

Windows IP Configuration

=====
Network Information for Compartment 1 (ACTIVE)
=====

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : localdomain
IPv6 Address. . . . . : fdb2:2c26:f4e4:0:413:9481:677f:f2a0
Temporary IPv6 Address. . . . . : fdb2:2c26:f4e4:0:7c57:4a3f:1d79:e51
Link-local IPv6 Address . . . . . : fe80::413:9481:677f:f2a0%5
IPv4 Address. . . . . : 10.211.55.8
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.211.55.1

PS C:\Windows\system32>
```

Рисунок 8 – Логи утилиты ipconfig

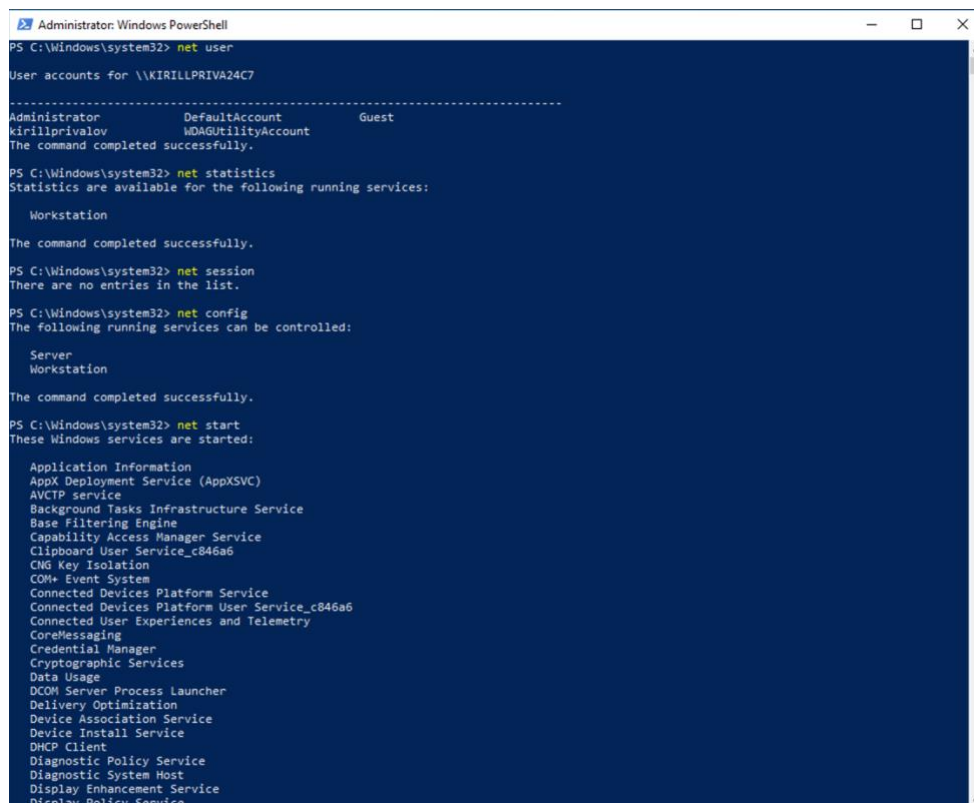
A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window has a dark blue background with white text. The command prompt shows the following sequence of commands and their outputs:
1. `PS C:\Windows\system32> net user`
Output: "User accounts for \\KIRILLPRIVA24C7", followed by a table of users: Administrator (DefaultAccount), kirillprivalov (WDAGUtilityAccount), and Guest. The command completed successfully.
2. `PS C:\Windows\system32> net statistics`
Output: "Statistics are available for the following running services:", followed by a list of services: Workstation. The command completed successfully.
3. `PS C:\Windows\system32> net session`
Output: "There are no entries in the list."
4. `PS C:\Windows\system32> net config`
Output: "The following running services can be controlled:", followed by a list of services: Server, Workstation. The command completed successfully.
5. `PS C:\Windows\system32> net start`
Output: "These Windows services are started:", followed by a long list of services including Application Information, AppX Deployment Service, AVCTP service, Background Tasks Infrastructure Service, Base Filtering Engine, Capability Access Manager Service, Clipboard User Service, CNG Key Isolation, COM+ Event System, Connected Devices Platform Service, Connected Devices Platform User Service, Connected User Experiences and Telemetry, CoreMessaging, Credential Manager, Cryptographic Services, Data Usage, DCOM Server Process Launcher, Delivery Optimization, Device Association Service, Device Install Service, DHCP Client, Diagnostic Policy Service, Diagnostic System Host, Display Enhancement Service, and Display Policy Service.

Рисунок 9 – Логи утилиты net

Программа *netsh* — это служебная программа командной строки, которая позволяет настраивать и отображать состояние различных сетевых ролей коммуникационных серверов и компонентов. С использованием этой утилиты я написал скрипт .bat, с помощью которого можно настроить IP вручную или автоматически:

```
@echo off
set /p choice=Choose method of configuration (1 - DHCP, 2 - manual):

rem Trim leading and trailing spaces
set "choice=%choice: =%"

if "%choice%"=="1" (
echo Configuring with DHCP...
netsh interface ip set address name="Ethernet" source=dhcp
netsh interface ip set dns name="Ethernet" source=dhcp
echo Configuration completed.
) else if "%choice%"=="2" (
set /p ip=Enter IP-address:
set /p mask=Enter subnet mask:
set /p gateway=Enter Default Gateway:
set /p dns=Enter DNS-Server:
goto setup_manual
) else (
echo Incorrect choice.
)

:setup_manual
netsh interface ip set address "Ethernet" static %ip% %mask% %gateway%
netsh interface ip set dns "Ethernet" static %dns%

echo Configuration completed.
pause
```

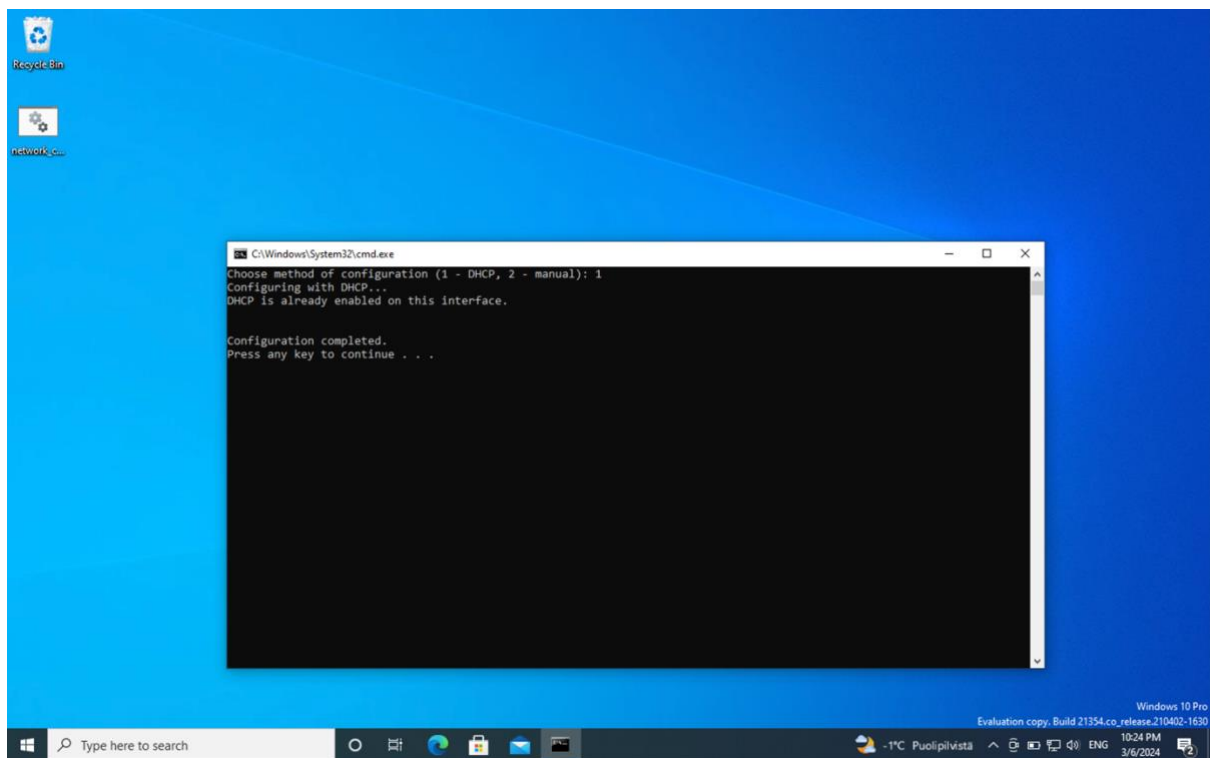



Рисунок 10 - Консольное приложение для настройки IP

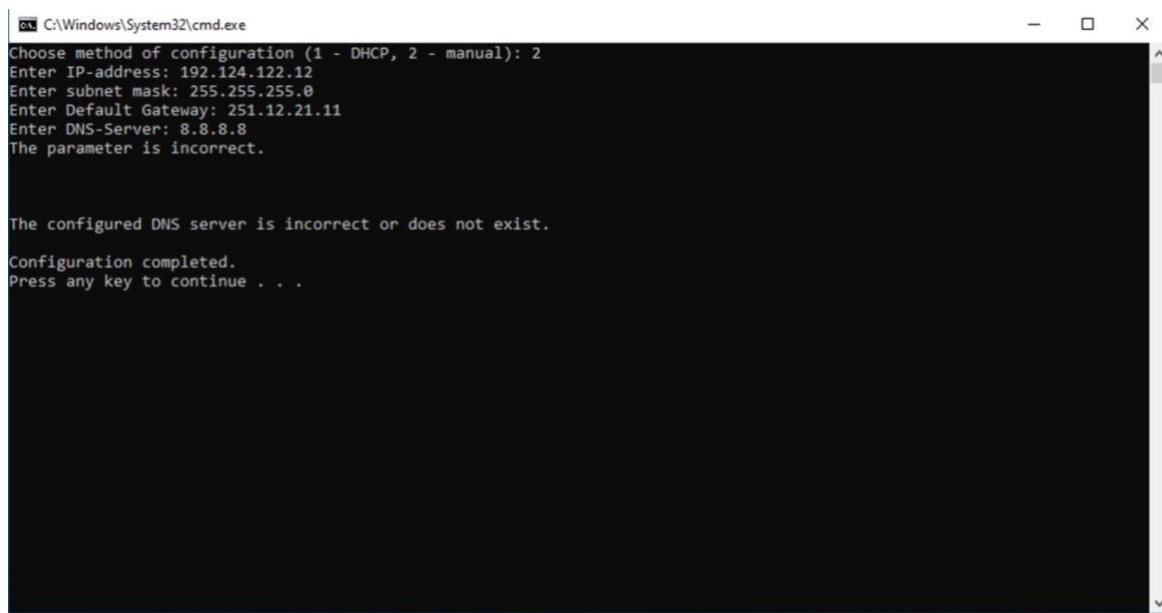


Рисунок 11 – Ручная настройка IP с помощью скрипта

A screenshot of a Windows command prompt window. The title bar shows 'C:\Windows\System32\cmd.exe'. The command prompt displays the following text:
Choose method of configuration (1 - DHCP, 2 - manual): 1
Configuring with DHCP...

Configuration completed.
The parameter is incorrect.

Рисунок 12 - Автоматическая настройка с помощью скрипта

Написал аналогичный скрипт для использования в Powershell.

```

$choice = Read-Host "Choose configuration method (1 - DHCP, 2 - manual, 3 - network card
model, 4 - link availability, 5 - speed, duplex)"

switch ($choice) {
    "1" {
        Write-Host "Configuring with DHCP..."
        Write-Host "Configuration completed."
    }
    "2" {
    }
    "3" {
        $networkAdapters = Get-WmiObject win32_NetworkAdapter | where-Object { $_.AdapterType
-likes "*Ethernet*" }
        foreach ($adapter in $networkAdapters) {
            Write-Host "Model: $($adapter.Description)"
        }
    }
    "4" {
        $networkAdapters = Get-WmiObject win32_NetworkAdapter | where-Object { $_.AdapterType
-likes "*Ethernet*" }
        foreach ($adapter in $networkAdapters) {
            $linkStatus = $adapter.NetConnectionStatus -eq 2
            if ($linkStatus) {
                Write-Host "Network adapter $($adapter.Description) is connected."
            } else {
                Write-Host "Network adapter $($adapter.Description) is disconnected."
            }
        }
    }
    "5" {
        $networkAdapters = Get-WmiObject win32_NetworkAdapter | where-Object { $_.AdapterType
-likes "*Ethernet*" }
        if ($networkAdapters) {
            foreach ($adapter in $networkAdapters) {
                $networkParams = Get-WmiObject win32_NetworkAdapterConfiguration | where-
object { $_.Description -eq $adapter.Description }

                if ($networkParams) {
                    $speed = $networkParams.Speed
                    $duplex = $networkParams.FullDuplex
                    if ($speed -and $duplex) {
                        Write-Host "Network adapter $($adapter.Description) Speed: $($speed /
1MB) Mbps, Duplex: $($duplex)"
                    } else {
                        Write-Host "Network adapter $($adapter.Description) does not provide
speed and duplex information."
                    }
                } else {
                    Write-Host "Network adapter $($adapter.Description) configuration
information is not available."
                }
            }
        } else {
            Write-Host "No Ethernet network adapters found."
        }
    }
    default {
        Write-Host "Incorrect choice."
    }
}

```

Контрольные вопросы

1. Чтобы запретить доступ через GUI используемого компьютера, необходимо открыть «Параметры» => «Сеть и Интернет» => «Изменение параметров адаптера» => «Свойства» => «Доступ». Чтобы запретить доступ к ресурсам других компьютеров в сети Microsoft необходимо открыть «Сеть и Интернет» => «Сеть Windows» => «Расширенные параметры сети».
2. Команда *net* нужна для управления сетевыми ресурсами и настройками. Некоторые директивы команды:
 - *use* – подключение к сетевому ресурсу – *net use Z: \\server\share*
 - *view* – просмотр доступных ресурсов – *net view*
 - *stop* и *start* – остановка или запуск службы – *net stop*
 - *share* – управление общим доступом к ресурсам – *net share sharename=drive:path*
 - *config* – просмотр и настройка конфигураций – *net config*
 - *session* – просмотр информации о сеансах – *net session*
 - *user* – управление пользователями и их правами – *net user username password /add*
 - *statistics* – просмотр статистики сетевого использования – *net statistics*
 - *localgroup* – управление локальными группами пользователей – *net localgroup Groupname user /add*
3. Чтобы узнать DNS адрес, на которой настроен компьютер, необходимо использовать команду *ipconfig*.
4. Команда *net use* используется для установки или удаления соединений с общими ресурсами компьютера или устройства: *net use R: \\SRV\TEST*.

5. Чтобы переименовать сетевое соединение Windows можно воспользоваться командой *Rename-NetAdapter*.
6. Режимы работы адаптера (duplex) определяют способ обмена данными между сетевыми устройствами. Режимы:
 - Полудуплексный (Half Duplex): Устройство может либо отправлять, либо принимать данные в определенный момент времени.
 - Полнодуплексный (Full Duplex): Устройство может одновременно отправлять и принимать данные.

Режим полнодуплексного соединения предпочтителен, так как он позволяет более эффективно использовать пропускную способность сети, в отличие от полудуплексного режима, где устройство должно переключаться между режимами отправки и приема данных.