МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**О Т Ч Е Т**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

# «**МЕТОДЫ АДРЕСАЦИИ И ПРОТОКОЛЫ РАЗРЕШЕНИЯ АДРЕСОВ**»

**по курсу:**

**"КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ "**

|  |  |
| --- | --- |
| Выполнил: | Ст. гр. 10701321 Царикевич В.С. |
| Проверил, доцент: | Белова С.В. |

Минск 2022

# ЛАБОРАТОРНАЯ РАБОТА 3

**Цель работы:**

Изучить методы адресации узлов сети и протоколы разрешения адресов. Разработать программу, позволяющую получить все виды адресов узла.

**Постановка задачи:**

1. Изучить основные теоретические вопросы, используя материалы лекций,  
рекомендуемую литературу и методические указания к лабораторной работе:

- методы адресации узлов сети;

- физические адреса;

- числовые адреса;

- доменные имена;

- протоколы разрешения адресов.  
2. Выполнить задания по лабораторной работе. При разработке программ разрешается использовать любой язык программирования и среду разработки.  
3. Ответить на контрольные вопросы.  
4. Подготовить отчет по лабораторной работе.  
5. Выполнить контрольный тест.

**Задания:**

Задание 1. Определение адресов локального узла с помощью утилиты ipconfigОпределить типы адресов локального узла и их значения, используя утилиту ipconfig с ключом /all.  
Задание 2. Разработка программы для определения адресов локального узлаРазработать консольное приложение (аналог утилиты ipconfig), в котором определить и вывести на экран:  
1) имя локального хоста, имя домена, полное доменное имя хоста;  
2) все сетевые интерфейсы локального хоста (тип, описание, имя);  
3) состояние интерфейса (подключен или нет в настоящее время);  
4) для каждого интерфейса:  
- физический адрес и размер физического адреса;  
- IPv4-адрес, маску, размер IPv4-адреса; размер сетевого префикса;  
- IPv6-адрес, размер IPv6-адреса, размер сетевого префикса.  
Адреса выводить на экран в общепринятой форме записи. Для получения адресов использовать классы пространства имен System.Net.NetworkInformation (IPGlobalProperties, NetworkInterface, IPInterfaceProperties (свойство UnicastAddresses), PhysicalAddress, DNS и др.). Сравнить полученные значения адресов с адресами из Задания 1.  
Задание 3. Изучение специальных IP-адресовИспользуя специальные поля класса IPAddress, вывести на экран для адресов IPv4 и IPv6:  
1) адрес петли обратной связи;  
2) широковещательный IP-адрес;  
3) адрес, обозначающий все сетевые интерфейсы данного узла.  
Задание 4. Определение IP-адреса по доменному имениРазработать консольное приложение для получения IP адреса по доменному имени (аналог утилиты nslookup). Получить и вывести на экран для заданного пользователем произвольного DNSимени:  
1) IPv4-адреса;  
2) IPv6-адреса;  
3) Имена-псевдонимы узла (Alias-имена).  
Для получения адресов использовать методы класса Dns пространства имен System.Net.

Задание 5. Изучение протокола ARP1. С помощью утилиты arp просмотреть ARP-таблицу локального узла. Сохранить  
полученную информацию в файле.  
2. Организовать сетевую активность (ping, tracert). Просмотреть таблицу  
преобразования адресов и сравнить ее с ранее полученными результатами. Пояснить  
причины изменений.  
3. Выполнить ping локального DNS-сервера. Определить по таблице arp mac-адрес  
DNS-сервера.  
4. Сделать перерыв в сетевой активности на несколько минут, после которого опять просмотреть arp-таблицу. Пояснить причины изменений (или отсутствия таковых) в таблице arp за время перерыва

**ВНИМАНИЕ! Данное задание выполнять в локальной сети с установленным** **DNS****-****сервером****.** **Н****апример****,** **в локальной сети** **Ф****ИТР****.**

**Задание 6. Работа с** **UNC****-****именами**  
Разработать консольное приложение, которое выполняет следующие действия:  
1. Создает файл на удаленном компьютере.  
2. Записывает в файл вашу фамилию и номер группы.  
**Для выполнения задания использовать** **UNC****-****имена файлов.** **ВНИМАНИЕ! Для выполнения задания необходимо наличие локальной сети и  
общей папки на удаленном компьютере.**

**Листинг программы:**

**Файл КCиC Лаба3 V3.cs**

**using** System;

using System.Net;

using System.Net.NetworkInformation;

using System.Net.Sockets;

namespace Addresses // Вывод адресов всех селевых интерфейсов локального компьтера

{

class Program

{

static void Main()

{

try

{

IPGlobalProperties computerProperties = IPGlobalProperties.GetIPGlobalProperties();

Console.WriteLine("Имя компьютера: {0}", computerProperties.HostName);

Console.WriteLine("Имя домена: {0}", computerProperties.DomainName);

// Получаем все сетевые интерфейсы локальното компьютера

NetworkInterface[] adapters = NetworkInterface.GetAllNetworkInterfaces(); //массив указател

if (adapters == null || adapters.Length < 1)

{

Console.WriteLine("Сетевые адаптера не найдены");

return;

}

Console.WriteLine("Количество сетевых инперфейсов: {0}\n", adapters.Length);

// Цикл по интерфейсам

foreach (NetworkInterface adapter in adapters)

{

//Console.WriteLine(String.Empty.PadLeft(50, '='));

Console.WriteLine("Имя сетевого адаптера . . . . . . : {0}", adapter.Name);

Console.WriteLine("Тип сетевого интерфейса . . . . . : {0}", adapter.NetworkInterfaceType);

Console.WriteLine("Описание интерфейса . . . . . . . : {0}", adapter.Description);

//Получение и вывод физического адреса

PhysicalAddress physicalAddress = adapter.GetPhysicalAddress();

byte[] bytes = physicalAddress.GetAddressBytes();

Console.Write("Физический адрес. . . . . . . . . : ");

for (int i = 0; i < bytes.Length; i++)

{

string temp = bytes[i].ToString("X2");

Console.Write(temp);

if (i != bytes.Length - 1)

{ Console.Write("-"); }

}

Console.WriteLine();

Console.WriteLine("Размер физического адреса. . . . .: {0} байт", bytes.Length);

// Получение и вывод IP-адресов

// Получаем объект, описывающий конфигурацию сетевого интерфейса

IPInterfaceProperties adapterProperties = adapter.GetIPProperties();

// Получаем unicast-адреса, назначенные текущему интерфейсу

UnicastIPAddressInformationCollection unicastСollection =

adapterProperties.UnicastAddresses;

if (unicastСollection.Count > 0)

{

foreach (UnicastIPAddressInformation unicastAddr in unicastСollection)

{

// выводим IPV4

if (unicastAddr.Address.AddressFamily == AddressFamily.InterNetwork)

{

Console.WriteLine("IRV4 адрес. . . . . . . . . . . . : {0}", unicastAddr.Address.ToString());

Console.WriteLine("Маска. . . . . . . . . . . . . . .: " + unicastAddr.IPv4Mask);

Console.WriteLine("Размер IPv4-адреса. . . . . . . . : {0} байт",unicastAddr.Address.GetAddressBytes().Length);

Console.WriteLine("Размер сетевого префикса. . . . . : " + unicastAddr.PrefixLength + " бит");

// Добавить вывод длины IPV4 адреса

}

// Выводим IPv6 адреса

if (unicastAddr.Address.AddressFamily == AddressFamily.InterNetworkV6)

{

Console.WriteLine("IPv6 адрес. . . . . . . . . . . . : {0}", unicastAddr.Address);

Console.WriteLine("Размер IPv6-адреса. . . . . . . . : {0} байт",unicastAddr.Address.GetAddressBytes().Length);

Console.WriteLine("Размер сетевого префикса. . . . . : " + unicastAddr.PrefixLength + " бит");

//Добавлть вывод длины IPv6 адреса

}

} //foreach

} //if

Console.WriteLine();

} //foreach

} //try

catch (Exception ex)

{ Console.WriteLine("Ошибка: " + ex.ToString()); }

finally

{ Console.ReadLine(); }

}

}

}

**Файл Lab3T4**

**using** System;

using System.Net;

using System.Net.NetworkInformation;

using System.Net.Sockets;

using System.Security.Cryptography;

namespace Lab3T4

{

class Program

{

[Obsolete]

static void Main(string[] args)

{

try

{

string dnsName;

Console.Write("Введите доменное имя: ");

dnsName = Console.ReadLine().Trim(new char[] { ' ', '\n' });

IPAddress[] ipAddr = Dns.GetHostByName(dnsName).AddressList;

// не выводит IPv6 адреса

foreach (IPAddress ip in ipAddr)

{

if (ip.AddressFamily == AddressFamily.InterNetwork)

{

Console.WriteLine($"\tIPv4 адрес: {ip}");

}

if (ip.AddressFamily == AddressFamily.InterNetworkV6)

{

Console.WriteLine($"\tIPv6 адрес: {ip}");

}

}

Console.WriteLine();

string[] aliases = Dns.GetHostByName(dnsName).Aliases;

Console.WriteLine($"Имя хоста: {Dns.GetHostByName(dnsName).HostName} \n");

if (aliases.Length != 0)

{

Console.WriteLine("Aliases: ");

foreach (string alias in aliases) { Console.WriteLine(alias); }

}

else

Console.WriteLine("Имен-псевдонимов нет!");

}

catch (Exception ex)

{

Console.WriteLine("Ошибка" + ex.ToString());

}

finally

{

Console.ReadLine();

}

}

}

}

**Файл Lab3T6**

using System;

using System.IO;

namespace Lab3T6

{

class Program6

{

static void Main(string[] args)

{

string path = @"\\DESKTOP-4DN7JDM\Users\HP\Desktop\Txt\txt.txt";

using(StreamWriter sw = new StreamWriter(path, false))

{

sw.WriteLine("Антон Садовский 10701321");

}

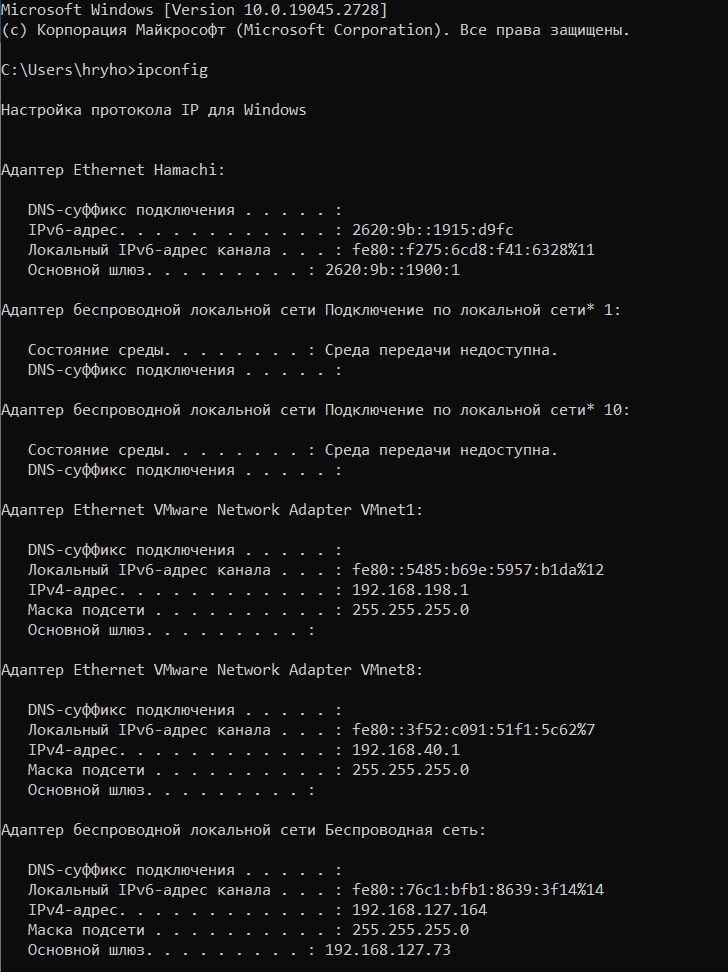
}

}

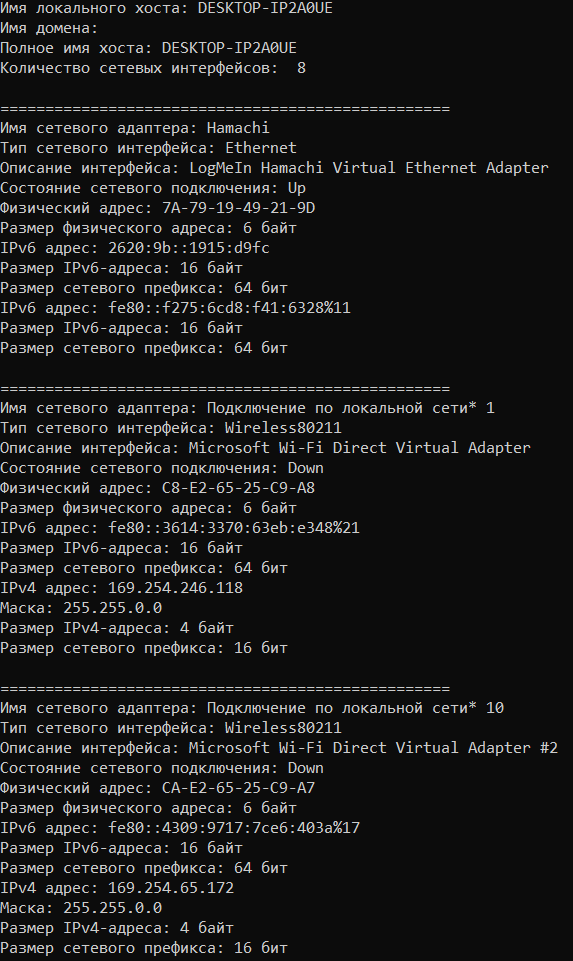
}

**Результат выполнения программы:**

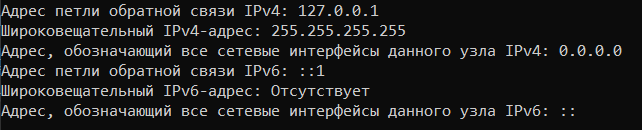
Результат выполнения задания №1



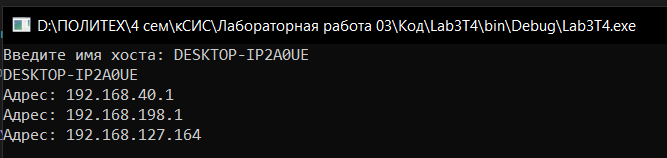
Результат выполнения задания №2



Результат выполнения заданий №3



Результат выполнения задания №4



Результат выполнения задания №5

**Контрольные вопросы:**

1. **Какие требования предъявляются к адресу узла сети?**

Адрес должен уникально идентифицировать компьютер в сети любого масштаба.

Схема назначения адресов должна сводить к минимуму ручной труд администратора и вероятность дублирования адресов.

Адрес должен иметь иерархическую структуру, удобную для построения больших сетей. В крупных сетях отсутствие иерархии адресов может привести к большим издержкам - конечным узлам и коммуникационному оборудованию придется оперировать с таблицами адресов, состоящими из тысяч записей.

Адрес должен быть удобен для пользователей сети, т.е. должен иметь символьное представление.

Адрес должен иметь по возможности компактное представление, чтобы не перегружать память коммуникационной аппаратуры - сетевых адаптеров, маршрутизаторов и т. П

1. **Что такое адресное пространство? Приведите пример плоского и иерархического адресного пространства.**

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется адресным пространством.

1. **Какие методы адресации используются в компьютерных сетях? Приведите примеры адресов каждого типа.**

Наибольшее распространение получили три схемы адресации узлов (три типа адресов):

- локальные адреса; 00-80-48-EB-7E-60 - числовые составные адреса; 194.85.135.75

- символьные адреса или имена. [www.youtube.com](http://www.youtube.com/)

1. **Как можно классифицировать адреса по количеству адресуемых сетевых интерфейсов?**

По количеству адресуемых сетевых интерфейсов адреса можно классифицировать следующим образом:

- **Одноадресный тип или уникальный адрес (unicast)** используется для идентификации отдельных интерфейсов (физический интерфейс между компьютером и сетью) конечного узла или маршрутизатора; позволяет пересылать сообщения в одну точку (на один конечный узел сети).

- **Групповой адрес (multicast)** идентифицирует сразу несколько интерфейсов, данные доставляются каждому из интерфейсов, входящих в группу; позволяет пересылать сообщения группе произвольно расположенных в Internet узлов.

- **Широковещательный адрес (broadcast)** используется для доставки данных всем узлам подсети;

- **Адрес произвольной рассылки (anycast)** задает группу интерфейсов, но данные должны быть доставлены не всем, а одному члену группы, как правило, «ближайшему» (новый тип адреса, определен в протоколе IPv6). Назначается только интерфейсам маршрутизатора.

1. **Приведите пример протокола разрешения адресов, использующего централизованный подход, распределенный подход.**

Пример использования распределенного подхода – **протокол разрешения адресов ARP (Address Resolution Protocol)**, используемый стеком TCP/IP для преобразования IPадреса в аппаратный адрес.

1. **Что такое локальный адрес? Какая форма записи используется для MAC-адресов? Какой аппаратный адрес используется для широковещательной передачи?**

**Локальный адрес** – такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, являющейся элементом составной сети. В разных подсетях допустимы различные сетевые технологии, следовательно, различные протоколы. Аппаратный адрес принято записывать в 16-ричном виде, разделяя байты с помощью “-“. Например: 11-АО-17-3D-ВС-01 - MAC-адрес сетевого адаптера Ethernet. Для широковещательной передачи используют FF:FF:FF:FF:FF:FF.

1. **Какая форма записи используется для IPv4, IPv6? Сколько места в памяти они занимают?**

Для удобства чтения в технической литературе и прикладных программах IPv4-адреса представляются в виде 4-х десятичных чисел, разделенных точками. Каждое из чисел соответствует одному октету (8 битам) и может иметь значения от 0 до 255. Этот формат называется точечно-десятичным (Decimal-Pant Notation).

Например: 10010001.00001010.00100010.00000011 145.10.34.3

IPv6-адреса занимают 128-бит или 16 байт. IPv6-адреса принято записывать в 16-ричном виде. Их делят на 8 блоков по четыре шестнадцатиричных цифры в каждом. Каждый блок отделяется двоеточием. Пример полного IPv6-адреса: 2001:00В8:3FA9:0000:0000:0000:0003:9С5А

1. **Приведите примеры символьных адресов.**

[www.youtube.com](http://www.youtube.com/)

1. **Какие протоколы разрешения имен могут использоваться в сетях Windows?**

Cети Windows включают три системы разрешения имен: DNS, LLMNR (Link Local Multicast Name Resolution) и NetBIOS.

1. **Какова структура доменного имени? Какие типы доменов верхнего уровня вы знаете?**

**Доменное имя** представляет собой перечень имен доменов, разделенных точками. Начинается доменное имя с имени конечного узла или типа ресурса. Последним указывается домен верхнего уровня. Например: www.google.com или 216-5.povt.fitr.bntu.by Существует два основных типа доменов верхнего уровня.

■ Организационные (родовые) домены. Имя такого домена указывает основную функцию или род деятельности организаций в DNS-домене.

■ Географические домены. Эти домены именуются с использованием кодов страны и региона из двух символов согласно стандарту 3166 Международной организации по стандартизации ISO, например .uk (Великобритания) или .it (Италия).

**Вывод:**

При выполнении данной лабораторной работы были изучены методы адресации узлов сети и протоколы разрешения адресов. Также были разработаны программы для получения всех видов узлов.

0 102 204