Лабораторная работа №0

Стек протоколов TCP/IP. Передача данных по сети средствами стека протоколов TCP/IP

1. Теоретические сведения

Работа с сокетами в .NET

Для работы с соектами в .NET разработан класс Socket, который расположен в пространстве имен System.Net.Sockets.Класс Socket обеспечивает широкий набор методов и свойств для сетевых взаимодействий. Класс Socket позволяет выполнять как синхронную, так и асинхронную передачу данных с использованием любого из коммуникационных протоколов, имеющихся в перечислении ProtocolType.

Класс Socket придерживается шаблона имен платформы .NET Framework для асинхронных методов. Например, синхронный метод Receive соответствует асинхронным методам BeginReceive и EndReceive. Если приложению при его исполнении требуется только один поток, воспользуйтесь приведенными ниже методами, которые разработаны для работы в синхронном режиме.

Если используется протокол, ориентированный на установление соединения, такой как протокол TCP, сервер должен выполнять прослушивание подключений, используя метод Listen. Метод Accept обрабатывает любые входящие запросы на подключение и возвращает объект Socket, который может использоваться для передачи данных с удаленного узла. Используйте этот возвращенный объект Socket для вызова метода Send или Receive. Вызовите метод Bind, прежде чем производить обращение к методу Listen, если необходимо указать локальный IP-адрес или номер порта. Используйте нулевое значение для номера порта, если требуется, чтобы свободный порт был назначен основным поставщиком услуг. Если требуется произвести подключение к прослушивающему узлу, вызовите метод Connect. Для обмена данными вызовите метод Send или Receive.

Чтобы выполнить передачи с использованием отдельных потоков во время исполнения, воспользуйтесь следующими методами, предложенными для работы в асинхронном режиме.

Если применяется протокол, ориентированный на установление соединения, такой как протокол TCP, используйте методы Socket, BeginConnect и EndConnect для подключения к прослушивающему узлу. Для асинхронного обмена данными воспользуйтесь методами BeginSend и EndSend или методами BeginReceive и EndReceive. Входящие запросы на подключение могут быть обработаны с помощью методов BeginAccept и EndAccept.

Если на сокете выполняется несколько асинхронных операций, они не обязательно должны завершаться в том же порядке, в котором эти операции запускаются.

Когда прием и отправка данных завершены, используйте метод Shutdown для того, чтобы отключить объект Socket. После вызова метода Shutdown обратитесь к методу Close, чтобы освободить все связанные с объектом Socket ресурсы.

Класс Socket позволяет выполнить настройку объекта Socket с использованием метода SetSocketOption. Извлеките эти параметры, используя метод GetSocketOption

Пример работы с сокетами на языке C#

```
using System;
using System.Text;
using System.IO;
using System.Net;
using System.Net.Sockets;
public class GetSocket
{
   private static Socket ConnectSocket(string server, int port)
   {
      Socket s = null;
      IPHostEntry hostEntry = null;
```

```
    // Get host related information.
    hostEntry = Dns.GetHostEntry(server);
    // Loop through the AddressList to obtain the supported ddressFamily.
    //Thisis to avoid
    // an exception that occurs when the host IP Address is not
    //compatible with the address family
    // (typical in the IPv6 case).
    foreach (IPAddress address in hostEntry.AddressList)
    {
       IPEndPoint ipe = new IPEndPoint(address, port);
       Socket tempSocket =
       new Socket(ipe.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
       tempSocket.Connect(ipe);
       if (tempSocket.Connected)
       {
          s = tempSocket;
          break;
       }
       else
       {
          continue;
       }
    }
    return s;
}
// This method requests the home page content for the specified server.
private static string SocketSendReceive(string server, int port)
{
    string request = "GET / HTTP/1.1\r\nHost: " + server +
    "\r\nConnection: Close\r\n\r\n";
    Byte[] bytesSent = Encoding.ASCII.GetBytes(request);
    Byte[] bytesReceived = new Byte[256];
    // Create a socket connection with the specified server and port.
    Socket s = ConnectSocket(server, port);
    if (s == null)
       return ("Connection failed");
    // Send request to the server.
    s.Send(bytesSent, bytesSent.Length, 0);
```

```
        // Receive the server home page content. 94
        int bytes = 0;
        string page = "Default HTML page on " + server + ":\r\n";
        // The following will block until page is transmitted.
        do
        {
            bytes = s.Receive(bytesReceived, bytesReceived.Length, 0);
            page = page + Encoding.ASCII.GetString(bytesReceived, 0, bytes);
        }
        while (bytes > 0);
        return page;
    }
    public static void Main(string[] args)
    {
        string host;
        int port = 80;
        if (args.Length == 0)
            // If no server name is passed as argument to this program,
            // use the current host name as the default.
            host = Dns.GetHostName();
        else
            host = args[0];
        string result = SocketSendReceive(host, port);
        Console.WriteLine(result);
    }
}
```

2. Задание на лабораторную работу.

1. Разработать программное обеспечение, реализующее передачу данных между компьютерами на уровне стека протоколов TCP/IP средствами ОС Windows, ОС Linux, .Net.

2. Используя возможности стека протоколов TCP/IP организовать распределѐнную обработку информации не менее чем на 3 компьютерах для решения конкретной прикладной задачи (Табл. 0.1)

3. Решение задачи осуществить в ОС Windows, ОС Linux и dot.Net. Для претендующих на оценки 9-10 обеспечить кроссплатформенное взаимодействие.

4. Сравнить время нахождения решения на нескольких компьютерах с временем решения задачи на одном компьютере. 5. Сравнить время нахождения решений в разных ОС и платформах.

Отчѐт должен содержать:

1. Блок-схему алгоритма решения поставленной задачи
2. Распечатку листингов программы
3. Распечатку внешнего вида окон программы
4. Распечатку результатов работы
5. Сравнительный анализ

Результаты моделирования обязательно должны быть продемонстрированы на компьютере.

Вариант	Задание	Платформа реализации	
1	Определить ранг квадратной матрицы размерности N	ОС Linux	ОС Windows
2	Вычислить обратную матрицу матрицы размерностью N методом Гаусса	ОС Linux	ОС Windows
3	Найти алгебраические дополнения к элементам матрицы размерности N	ОС Linux	ОС Windows
4	Найти все собственные значения квадратной матрицы размерности N	ОС Linux	ОС Windows
5	Найти матрицу смежности (расстояний) для M векторов размерности N	ОС Linux	ОС Windows
6	Найти результат возведение квадратной матрицы размерности N в степень M	ОС Linux	ОС Windows
7	Каждый элемент в прямоугольной матрице NxM заменить средним значением из его окрестности радиусом R-элементов	ОС Linux	ОС Windows
8	Определить ранг квадратной матрицы размерности N	ОС Linux	dot.Net
9	Вычислить обратную матрицу матрицы размерностью N методом Гаусса	ОС Linux	dot.Net
10	Найти алгебраические дополнения к элементам матрицы размерности N	ОС Linux	dot.Net
11	Найти все собственные значения квадратной матрицы размерности N	ОС Linux	dot.Net
12	Найти матрицу смежности (расстояний) для M векторов размерности N	ОС Linux	dot.Net
13	Найти результат возведение квадратной матрицы размерности N в степень M	ОС Linux	dot.Net
14	Каждый элемент в прямоугольной матрице NxM заменить средним значением из его окрестности радиусом R-элементов	ОС Linux	dot.Net
15	Каждый элемент в квадратной матрице NxN заменить расстоянием между векторами, сформированными из элементов столбца и строки.	ОС Linux	ОС Windows