CHƯƠNG 2: BÀI TẬP 4

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class VarTcpSrvr
{
  private static int SendData(Socket s, byte[] data)
  {
   int total = 0;
   int size = data.Length;
   int dataleft = size;
   int sent;
   byte[] datasize = new byte[4];
   datasize = BitConverter.GetBytes(size);
   sent = s.Send(datasize);
   while (total < size)
   {
     sent = s.Send(data, total, dataleft, SocketFlags.None);
     total += sent;
     dataleft -= sent;
   }
   return total;
  }
  private static byte[] ReceiveVarData(Socket s)
  {
   int total = 0;
   int recv;
   byte[] datasize = new byte[4];
   recv = s.Receive(datasize, 0, 4, 0);
   int size = BitConverter.ToInt32(datasize, 0);
   int dataleft = size;
   byte[] data = new byte[size];
   while(total < size)
   {
     recv = s.Receive(data, total, dataleft, 0);
     if (recv == 0)
     {
      data = Encoding.ASCII.GetBytes("exit ");
      break;
     }
     total += recv;
     dataleft -= recv;
   }
   return data;
  }
  public static void Main()
  {
   byte[] data = new byte[1024];
   IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 9050);
   Socket newsock = new Socket(AddressFamily.InterNetwork,
           SocketType.Stream, ProtocolType.Tcp);
   newsock.Bind(ipep);
   newsock.Listen(10);
```

```
   Console.WriteLine("Waiting for a client...");
   Socket client = newsock.Accept();
   IPEndPoint newclient = (IPEndPoint)client.RemoteEndPoint;
   Console.WriteLine("Connected with {0} at port {1}",
           newclient.Address, newclient.Port);
   string welcome = "Welcome to my test server";
   data = Encoding.ASCII.GetBytes(welcome);
   int sent = SendVarData(client, data);
   for (int i = 0; i < 5; i++)
   {
     data = ReceiveVarData(client);
     Console.WriteLine(Encoding.ASCII.GetString(data));
   }
   Console.WriteLine("Disconnected from {0}", newclient.Address);
   client.Close();
   newsock.Close();
  }
}
```

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class VarTcpClient
{
  private static int SendVarData(Socket s, byte[] data)
  {
   int total = 0;
   int size = data.Length;
   int dataleft = size;
   int sent;
   byte[] datasize = new byte[4];
   datasize = BitConverter.GetBytes(size);
   sent = s.Send(datasize);
   while (total < size)
   {
     sent = s.Send(data, total, dataleft, SocketFlags.None);
     total += sent;
     dataleft -= sent;
   }
   return total;
  }
  private static byte[] ReceiveVarData(Socket s)
  {
   int total = 0;
   int recv;
   byte[] datasize = new byte[4];
   recv = s.Receive(datasize, 0, 4, 0);
   int size = BitConverter.ToInt32(datasize, 0);
   int dataleft = size;
   byte[] data = new byte[size];
   while(total < size)
```

```
    {
      recv = s.Receive(data, total, dataleft, 0);
      if (recv == 0)
      {
       data = Encoding.ASCII.GetBytes("exit ");
       break;
      }
      total += recv;
      dataleft -= recv;
    }
    return data;
  }
  public static void Main()
  {
   byte[] data = new byte[1024];
   int sent;
   IPEndPoint ipep = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);
   Socket server = new Socket(AddressFamily.InterNetwork,
           SocketType.Stream, ProtocolType.Tcp);
   try
   {
     server.Connect(ipep);
   } catch (SocketException e)
   {
     Console.WriteLine("Unable to connect to server.");
     Console.WriteLine(e.ToString());
     return;
   }
   data = ReceiveVarData(server);
   string stringData = Encoding.ASCII.GetString(data);
   Console.WriteLine(stringData);
   string message1 = "This is the first test";
   string message2 = "A short test";
   string message3 = "This string is an even longer test. The quick brown Â_
fox jumps over the lazy dog.";
   string message4 = "a";
   string message5 = "The last test";
   sent = SendVarData(server, Encoding.ASCII.GetBytes(message1));
   sent = SendVarData(server, Encoding.ASCII.GetBytes(message2));
   sent = SendVarData(server, Encoding.ASCII.GetBytes(message3));
   sent = SendVarData(server, Encoding.ASCII.GetBytes(message4));
   sent = SendVarData(server, Encoding.ASCII.GetBytes(message5));
   Console.WriteLine("Disconnecting from server...");
   server.Shutdown(SocketShutdown.Both);
   server.Close();
  }
}
```