

USHTRIMI 1:**Programimi me Socket i një klienti aplikacioni që komunikon në rrjetë me një server aplikacion nëpërmjet protokollit TCP**

Objektivat e Ushtrimit 1 janë:

- ✓ Të kuptuarit e Socket-ave
- ✓ Të kuptuarit e një programi që funksionon në rrjetë
- ✓ Njohja dhe familjarizimi me klasat e .NET Framework të cilat realizojnë komunikimin në rrjetë me Socket-a.
- ✓ Kodimi i një klient aplikacioni që komunikon në rrjetë.

Kur një aplikacioni i nevojitet të lidhet në rrjetë lokale (intranet) ose në internet , ai aplikacion e përdorë komponentën softuerike të quajtur *Socket*. *Socket* i hap lidhjet ne rrjetë për t'i lejuar aplikacionet që të mund të shkruajnë dhe të lexojnë nëpërmjet rrjetës.

Socket janë një pikë fundore (ang. Endpoint) e komunikimit në mes të dy programeve në rrjetë. Një *Socket* perbehet prej numrit te portit (ang. port number) dhe prej një IP adresë dhe këto dy komponente e identifikojnë në mënyrë unike *Socket*-in. Gjatë shkruarjes së kodëve për programet që komunikojnë në rrjetë, programerët duhet të vendosin se me cilin protokol të komunikimit do të komunikojnë aplikacionet. Kemi dy protokolle të cilat mund ti përdorim, protokollin TCP dhe protokollin UDP. TCP protokollin është protokoll që siguron një kanal të sigurtë të komunikimit në mes klientit dhe serverit ndërsa protokollin UDP nuk siguron nje kanal të tillë mirpo këtu rrjedha e informatave bëhet pa garancionin se ato do të arrijnë në destinacion. Në gjatë këtij ushtrimit do të përdorim protokollin TCP.

Kur një klient aplikacion ekzekutohet ai krijon një proces njëjtë sikurse edhe një server aplikacion. Kështu komunikimi në mes të ketyre dy proceseve bëhet duke dërguar mesazhe në *Socket*-a. dhe përmes tij bëhet komunikimi me procesin tjetër i cili ekzekutohet në *host*-in tjetër duke përdorur protokollin TCP.

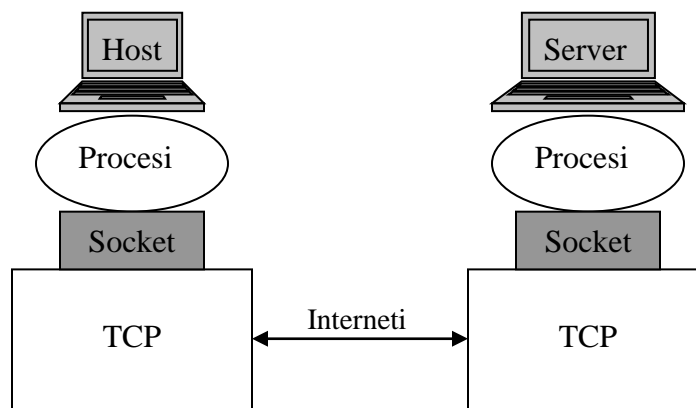


Fig 1. Komunikimi i proceseve nëpërmjet TCP *socket*-ave

Klasat .NET për programimin me Socket-a

.NET Framework ofron një numer të madh klasash që nevoiten për të shkruar programe që komunikojnë ndërmjet vetë në rrjetë. Namespace-at që përdoren në këtë rast janë:

1. System.Net
2. System.Net.Sockets

Në vazhdim do të flasim për klasat që do ti përdorim në këtë ushtrim.

Klasa **IPEndPoint** perfshinë hostin dhe informacionet për portin lokal ose remote që nevoiten për një aplikacion. Duke i kombinuar IP adresen e hostit dhe numrin e portit, klasa IPEndPoint e krijon lidhjen me hostin tjetër në rrjetë.

Klasa **IPAddress** përmban IP adresat e kompjutereve në rrjetë. P.sh. *Public class IPAddress*

Klasa **Socket** përfshin një numër të madh metodash që përdoren për të komunikuar në rrjetë.

Klasa **Encoding** paraqet enkodimin e karakterëve. Enkodimi është një proces i transformimit të një bashkësie karakteresh unicode në një sekuence të bajtave ndërsa Dekodimi është procesi i kundërt. Kjo klasë bën pjesë në bashkësinë e klasave *System.Text*. P.sh: *Encoding ascii = Encoding.ASCII;*

Shembulli: Kodimi i një klient/server aplikacioni në C#

Skenari

1. Klienti lexon një fjali nga hyrja standarde (tastiera) dhe e dërgon këtë fjali në socket për ta dërguar metutje të serveri.
2. Serveri e lexon fjalin nga lidhja që krijon me socket-in.
3. Serveri e konvertton fjalin në shkronja të medha.
4. Serveri e dërgon fjalin e modifikuar në socket për ta dërguar drejt klientit.
5. Klienti e lexon fjalin e modifikuar nga socket-i i saj dhe e shfaq në monitor.

Në figurën e mëposhtme (Fig. 2) po e bëjme paraqitjen grafike të këtij skenari.

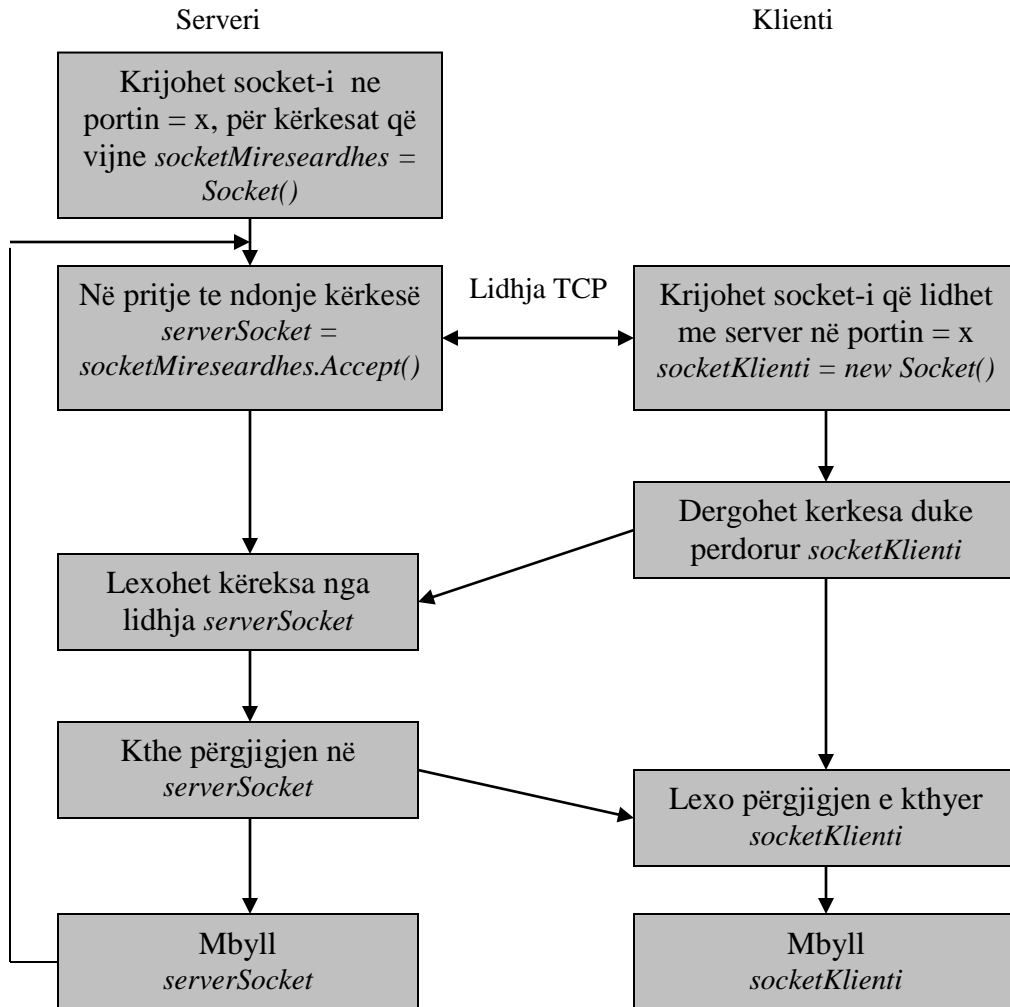


Fig. 2. Klient – Server aplikacioni duke perdorur TCP

Kodimi i aplikacionit

Më poshtë po e paraqesim zgjidhjen e algoritmit të dhënë me lartë. Në këtë rast do ti shkruajmë dy klasa: klasen e klientit dhe klasen e serverit.

TCPKlienti.cs

Pjesa e kodit për pjesën e klientit :

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Net.Sockets;
using System.Net;

namespace TCPKlientiServeri
{
    class Klienti
    {
        static void Main(string[] args)
    }
}
  
```

```
{
    //Hederi
    Console.WriteLine("=====");
    Console.WriteLine("UNIVERSITETI I PRISHTINES");
    Console.WriteLine("Fakulteti i Inxhinierise Elektrike dhe
Kompjuterike");
    Console.WriteLine("Departament i Kompjuterikes");
    Console.WriteLine("Lenda: Rrjetat Kompjuterike");
    Console.WriteLine("USHTRIMI 1");
    Console.WriteLine("=====\n");

    IPEndPoint ip = new IPEndPoint
(IPAddress.Parse("127.0.0.1"), 9999);

    Socket socketKlienti = new
Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);

    try
    {
        socketKlienti.Connect(ip);
    }
    catch (SocketException se)
    {
        Console.WriteLine("E pamundur per tu lidhur me
serverin." + se.Message);
        return;
    }
    Console.Write("Shtypni tekstin me shkronja te vogla: ");
    string input = Console.ReadLine();
    socketKlienti.Send(Encoding.ASCII.GetBytes(input));
    byte[] data = new byte[1024];
    int receivedDataLength = socketKlienti.Receive(data);
    string stringEdhena = Encoding.ASCII.GetString(data, 0,
receivedDataLength);
    Console.WriteLine("Rezultati i kthyer nga serveri: " +
stringEdhena);
    Console.WriteLine("\n\nShtypni tastin 'Enter' per te
perfunduar punen");
    Console.ReadLine();
    socketKlienti.Shutdown(SocketShutdown.Both);
    socketKlienti.Close();
}
}
```

Pershkrimi I kodit:

```
IPEndPoint ip = new IPEndPoint (IPAddress.Parse("127.0.0.1"), 9999);
```

Në këtë pjesë të kodit bëhet krijimi i një pikë fundore (ang. Endpoint) e cila përmban një ip adresë dhe një port numer.

```
Socket socketKlienti = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
```

Në këtë pjesë të kodit bëhet krijimi i një socket-i dhe inicializimi i tij. Siç po shihet protokolli i cili është përdorur për ta krijuar këtë socket është TCP.

```
socketKlienti.Connect(ip);
```

Këtu bëhet vendosja e lidhjes me hostin e largët (ang. remote host) e cila lidhje bëhet në pikën e fundme të cekur me lart pra “ip”.

```
socketKlienti.Send(Encoding.ASCII.GetBytes(input));
```

Në këtë pjesë të kodit bëhet dërgimi i mesazhit nëpërmjet socket-it *socketKlientit*

```
int receivedDataLength = socketKlienti.Receive(data);
```

Këtu bëhet leximi i gjatësisë së mesazhit (te dhenës) që vijnë nga serveri

```
string stringEdhena = Encoding.ASCII.GetString(data, 0, receivedDataLength);
```

Në këtë rresht të kodit bëhet ruajtja e rezultatit të kthyer në një string variabel, në rastin tonë në ‘stringEdhena’.

```
socketKlienti.Shutdown(SocketShutdown.Both);  
socketKlienti.Close();
```

Dhe në fund me keta dy rreshta të kodit bëhet ndërprerja e dërgimit/pranimi nëpërmjet socketKlientit gjegjësisht mbyllja e socketKlient lidhjes.

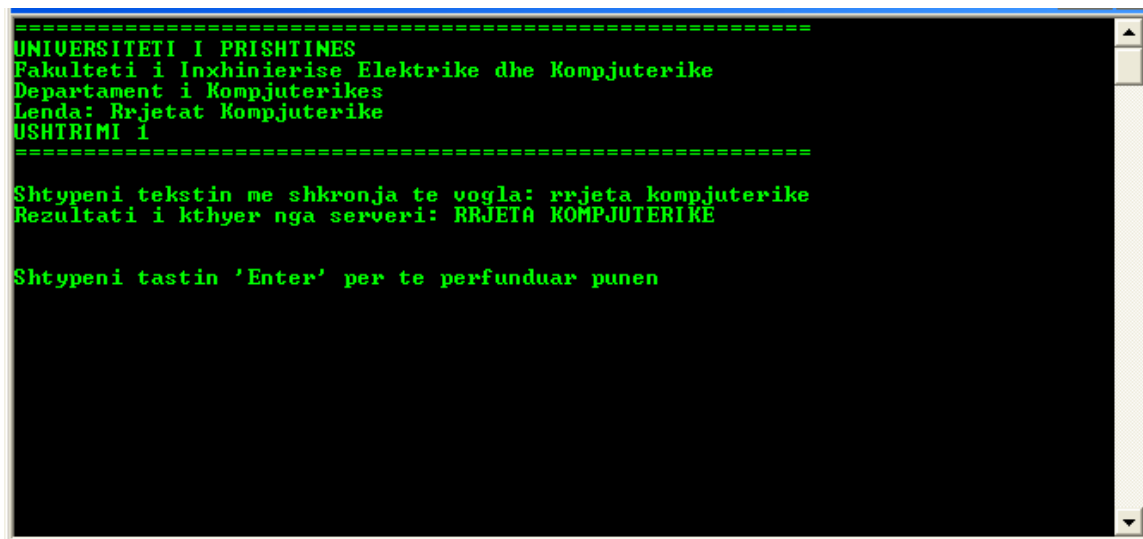


Fig. 3: Rezultati i shfaqur në program