

**USHTRIMI 2:****Programimi me Socket i një server aplikacioni që komunikon në rrjetë me klient aplikacione nëpërmjet protokolit TCP**

Objektivat e Ushtrimit 2 janë:

- ✓ Çka është serveri
- ✓ Të kuptuarit e arkitekturës klient/server
- ✓ Kodimi i një server aplikacioni që komunikon në rrjetë.

Në ushtrimin paraprak ne folëm dhe diskutuam për Socket-at. Pam se çka janë ato dhe se si mundësojnë komunikimin në rrjetë.

Serveri njëjtë sikurse edhe kompjuterët e tjerë është një makinë procesorike e fuqishme. Detyrat e kësaj makinë janë që të shërbejnë klientët. Si punon arkitektura klient/server? Klienti e bënë kërkesën dhe kështu kërkesa dërgohet për tek serveri, pastaj serveri e merr atë kërkesë dhe e proceson. Pas procesimit të kërkesës serveri përsëri e kthen përgjigjen tek klienti. Në figurën e mëposhtme është paraqitur një arkitekturë e thjeshtë klient/server.

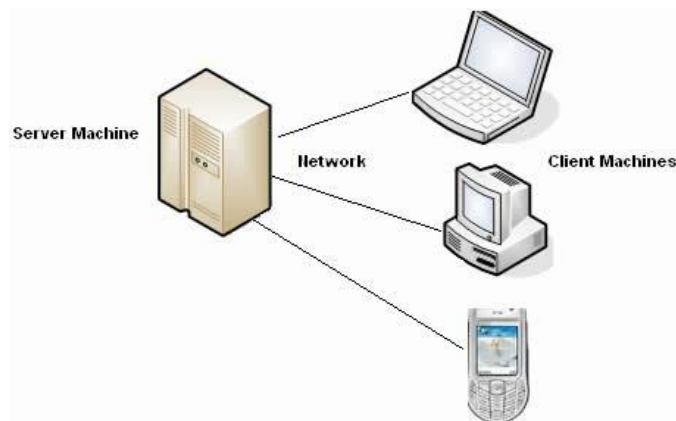


Fig 1. Arkitektura Klient/Server

**Klasat .NET për programimin me Socket-a**

.NET Framework ofron një numër të madhe klasash që nevojiten 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 nevojiten për një aplikacion. Duke i kombinuar IP adresën e hostit dhe numrin e portit, klasa **IPEndPoint** e krijon lidhjen me hostin tjetër në rrjetë.

Klasa **IPAddress** përmban IP adresat e kompjuterëve në rrjetë. Psh. *Public class IPAddress*

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

Klasa **Enkoding** paraqet enkodimin e karaktereve. Enkodimi është një proces i transformimit të një bashkësie karakteresh unicode në një sekuenca 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 server aplikacioni në C#

#### Skenari

1. Serveri e lexon fjalinë nga lidhja që krijon me klientin nëpërmjet socket-it.
2. Serveri e konvertton fjalinë në shkronja të mëdha.
3. Serveri e dërgon fjalinë e modifikuar në socket për ta dërguar drejt klientit.

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

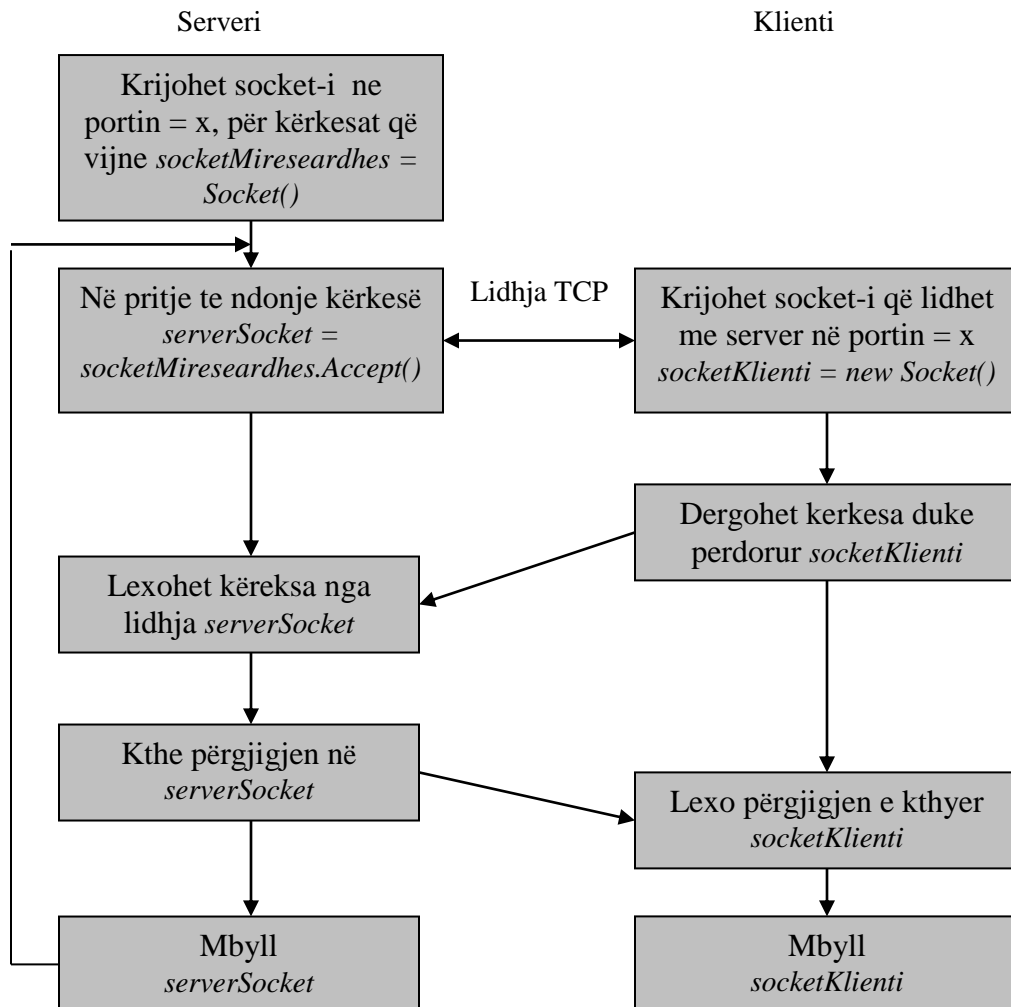


Fig. 2. Klient – Server aplikacioni duke përdorur TCP

**Kodimi i pjesës së serverit**

Më poshtë po e paraqesim zgjidhjen e algoritmit të dhënë me lartë.

TCPServeri.cs

Pjesa e kodit për serverin :

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net.Sockets;
using System.IO;
using System.Threading;
using System.Net;

namespace server
{
    class Program
    {
        static void Main(string[] args)
        {

            Console.WriteLine("=====");
            Console.WriteLine("UNIVERSITETI I PRISHTINES");
            Console.WriteLine("FIEK");
            Console.WriteLine("Departament i Kompjuterikes");
            Console.WriteLine("Lenda: Rrjetat Kompjuterike");
            Console.WriteLine("USHTRIMI 3");
            Console.WriteLine("=====\\n");

            IPEndPoint ip = new IPEndPoint(IPAddress.Any, 9999);
            Socket socketMireseardhes = new
            Socket(AddressFamily.InterNetwork, SocketType.Stream,
            ProtocolType.Tcp);

            socketMireseardhes.Bind (ip);
            socketMireseardhes.Listen(10);
            Console.WriteLine("Duke pritur për ndonjë klient...");
            Socket serverSocket = socketMireseardhes.Accept();
            IPEndPoint clientep =
            (IPEndPoint) serverSocket.RemoteEndPoint;
            Console.WriteLine("Serveri u lidh me hostin: {0} ne portin
            {1}", clientep.Address, clientep.Port);

            byte[] edhenaP = new byte[1024];
            int gjatesia = serverSocket.Receive(edhenaP);
            string eardhura = Encoding.ASCII.GetString(edhenaP, 0,
            gjatesia);
            Console.WriteLine(eardhura);

            string welcome = eardhura.ToUpper();
            byte[] edhenaD = new byte[1024];
            edhenaD = Encoding.ASCII.GetBytes(welcome);
            serverSocket.Send(edhenaD, edhenaD.Length,
            SocketFlags.None);
```

```
        Console.WriteLine("Lidhja u shkeput nga {0}",
        clientep.Address);
        serverSocket.Close();
        socketMireseardhes.Close();

        Console.WriteLine("\n\nShtypeni tastin 'Enter' per te
        perfunduar punen");
        Console.ReadLine();
    }
}
```

### Pershkrimi I kodit:

```
IPEndPoint ip = new IPEndPoint(IPAddress.Any, 9999);
```

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

```
Socket socketMireseardhes = 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.

```
socketMireseardhes.Bind (ip);
```

Metoda Bind e lidh socketin me një *ipendpoint* local dhe gjithmonë thirret para metodës Listen.

```
socketMireseardhes.Listen(10);
```

Kjo metod e vendos socketin në gjendje për të ndëgjuar. Numri 10 tregon numrin e lidhjeve që presin për tu pranuar.

```
Socket serverSocket = socketMireseardhes.Accept();
```

Në këtë pjesë të kodit bëhet pranimi I soketit mireseardhës nga ana e socketit të serverit ndërsa

```
IPEndPoint clientep = (IPEndPoint)serverSocket.RemoteEndPoint;
```

në këtë pjesë bëhet krijimi I një *IpEndPointi* në server dhe lidhja e tij me *Ipendpointin* e klientit

```
byte[] edhenaP = new byte[1024];
int gjatesia = serverSocket.Receive(edhenaP);
string eardhura = Encoding.ASCII.GetString(edhenaP, 0, gjatesia);
        Console.WriteLine(eardhura);
```

Kodi me lartë është pjesa ku e dhëna që dërgohet nga klienti pranohet si *byte* vektor dhe pastaj enkodohet dhe ruhet në një string variabël me të cilën pastaj manipulohet në server ndërsa pjesa e kodit mëposhtë

```
string welcome = eardhura.ToUpper();
byte[] edhenaD = new byte[1024];
edhenaD = Encoding.ASCII.GetBytes(welcome);
serverSocket.Send(edhenaD, edhenaD.Length, SocketFlags.None);
```

tregon se e dhëna që është pranuar nga klienti e që në rastin tonë është e dhënë e tipit string kthehet në shkronja të mëdha dhe dërgohet përsëri për tek klienti.

```
serverSocket.Close();
socketMireseardhes.Close();
```

Dhe në fund me keta dy rreshta të kodit bëhet ndërprerja e dërgimit/pranimit nëpërmes *serverSocket* gjegjësisht mbyllja e *serverSocket* lidhjes dhe *socketMireseardhes* lidhjes.

Për ta testuar detyrën së pari duhet ta ekzekutoni server (shih figuren 3)

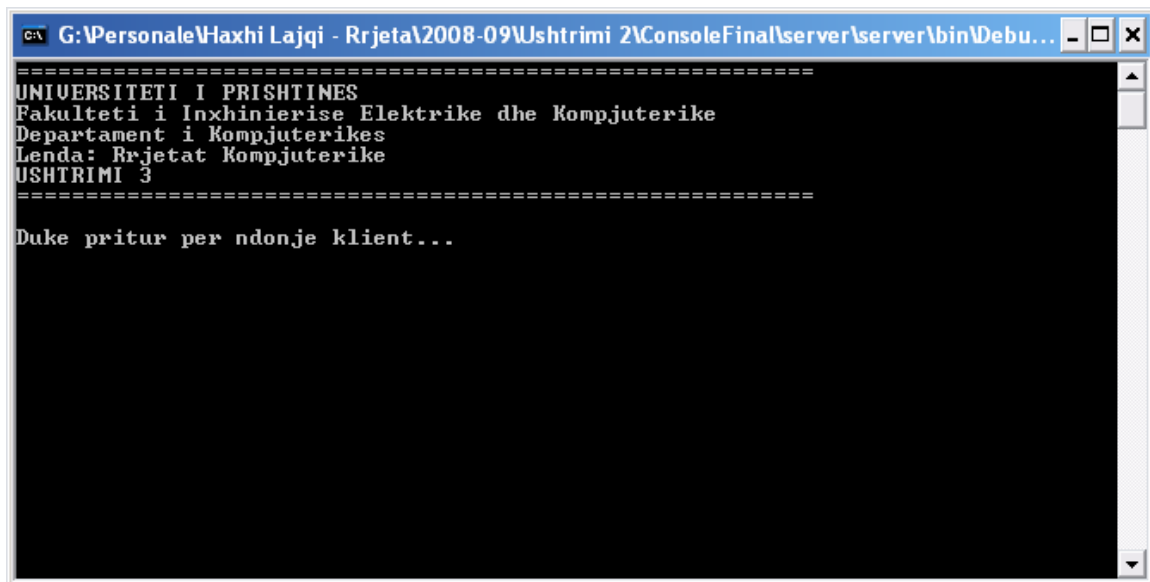


Fig. 3: Rezultati i shfaqur ne program

Pastaj pasi të ekzekutohet serveri në ekran na shfaqet teksti i shkruar si më lartë. Tutje ekzekutojmë klientin dhe në server do të shohim të dhënat që janë shfaqur mëposhtë:

```
G:\Personale\Haxhi Lajqi - Rrjeta\2008-09\Ushtrimi 2\ConsoleFinal\server\server\bin\Debu...
=====
UNIVERSITETI I PRISHTINES
Fakulteti i Inzhinierise Elektrike dhe Kompjuterike
Departament i Kompjuterikes
Lenda: Rrjetat Kompjuterike
USHTRIMI 3
=====
Duke pritur per ndonje klient...
Serveri u lidh me hostin: 127.0.0.1 ne portin 4815
tungjatjeta
Lidhja u shkeput nga 127.0.0.1

Shtypeni tastin 'Enter' per te perfunduar punen
_
```

Fig. 4: Rezultati pasi është ekzekutuar klienti dhe është dërguar një mesazh.