

Zadaci za samostalni rad

1. (UDP vs TCP) You will connect to the daytime server at the National Institute for Standards and Technology (NIST) and ask it for the current time. However, this time you'll use UDP instead of TCP. Recall that the daytime server listens on port 13, and that the server sends the time in a human-readable format and closes the connection.

Pogledati sličan zadatak:

<http://poincare.matf.bg.ac.rs/~jelenagr/mreze/cas04/5%20Server%20sockets/primer2/>

HINT

```
import java.io.*;

import java.net.*;

public class DaytimeUDPClient {

    private final static int PORT = 13;

    private static final String HOSTNAME = "time.nist.gov";

    public static void main(String[] args) {

        try (DatagramSocket socket = new DatagramSocket(0)) {

            socket.setSoTimeout(10000);

            InetAddress host = InetAddress.getByName(HOSTNAME);

            DatagramPacket request = new DatagramPacket(new byte[1], 1, host , PORT);

            DatagramPacket response = new DatagramPacket(new byte[1024], 1024);

            socket.send(request);    socket.receive(response);

            String result = new String(response.getData(), 0, response.getLength(), "US-ASCII");

            System.out.println(result);  }

        catch (IOException ex) {

            ex.printStackTrace();

        }

    }

}
```

POZIV SLICAN KAO I KAD SE KORISTI TCP
\$ java DaytimeUDPClient

56375 13-04-11 19:55:22 50 0 0 843.6 UTC(NIST) *

2. IMPLEMENTACIJA UDP daytime protocol server

A UDP server follows almost the same pattern as a UDP client, except that you usually receive before sending and don't choose an anonymous port to bind to. Unlike TCP, there's no separate `DatagramServerSocket` class.

For example, let's implement a daytime server over UDP. Begin by opening a datagram socket on a well-known port. For daytime, this port is 13:

```
DatagramSocket socket = new DatagramSocket(13);
```

As with TCP sockets, on Unix systems (including Linux and Mac OS X) you need to be running as root in order to bind to a port below 1024. **You can either use `sudo` to run the program or simply change the port to something 1024 or higher.** Next, create a packet into which to receive a request. You need to supply both a byte array in which to store incoming data, the offset into the array, and the number of bytes to store.

3. IMPLEMENTIRAJTE UDP port Scanner (Look for local UDP ports)

Pokretanje programa

```
% java UDPPortScanner
```

There is a server on port 2049.

There is a server on port 32768.

There is a server on port 32770.

There is a server on port 32771.

Pojasnenje: The first port, 2049, is an NFS server. The high-numbered ports in the 30,000 range are Remote Procedure Call (RPC) services. Along with RPC, common protocols that use UDP include NFS, TFTP, and FSP. It's much harder to scan UDP ports on a remote system than to scan

A port scanner that looks for UDP ports in use on the local host. It decides that the port is in use if the `DatagramSocket` constructor throws an exception.

As written, it looks at ports from 1024 and up to avoid Unix's requirement that it run as root to bind to ports below 1024.

You can easily extend it to check ports below 1024, however, if you have root access or are running it on Windows.

```
import java.net.*;
```

```
public class UDPPortScanner {
```

```
public static void main(String[] args) {  
    for (int port = 1024; port <= 65535; port++) {  
        try {           // the next line will fail and drop into the catch block if  
            // there is already a server running on port i  
            DatagramSocket server = new DatagramSocket(port);  
            server.close();  
        }  
        catch (SocketException ex) {  
            System.out.println("There is a server on port " + port + ".");  
        }  
    }  
}
```