

Redes de Computadores
Laboratório II
Prof. Jó Ueyama

Nome: Bruna Zamith
Número USP: 11383109

1) Instale e compile os programas Java TCPClient e UDPClient em um host e os programas TCPServer e UDPServer num outro host.

a) Suponha que você rode o TCPClient antes de rodar o TCPServer. O que acontece? Por que?

A conexão é recusada ("Connection Refused"). Porque o servidor não está ativo então não é possível completar a requisição.

b) Suponha que você rode o UDPClient antes de rodar o UDPServer. O que acontece? Por que?

Não é mostrada nenhuma mensagem de erro. Porque no protocolo UDP o pacote é enviado, tenta chegar ao servidor e percebe que a porta de destino não está recebendo nada, mas não aponta nenhum erro propriamente dito. Isso porque ele não dá garantia de entrega de pacote.

c) O que acontece se você utilizar números de portas diferentes para o lado do cliente e do servidor?

São apontados os mesmos erros. Não vai funcionar.

2) Suponha que em UDPClient.java, a linha

DatagramSocket clientSocket = new DatagramSocket();

Seja substituída por

DatagramSocket clientSocket = new DatagramSocket(5432);

a) Será necessário mudar UDPServer.java?

Sim. Para o mesmo valor de porta.

b) Quais são os números de portas agora para os sockets em UDPClient e do UDPServer?

5432

c) Quais eram esses números antes dessa mudança?

9876

3) Tente executar o TCPServer na máquina do colega ao lado. Para isso, informe-se do endereço IP da máquina onde o seu servidor TCPServer será executado. Para verificar o endereço IP, você poderá usar o aplicativo ifconfig(Linux) ou o ipconfig(Windows).

TCPServer.java permaneceu igual.

```
import java.io.*;
```

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

```
class TCPClient
```

```

{
    public static void main(String argv[]) throws Exception
    {
        String sentence;
        String modifiedSentence;
        BufferedReader inFromUser = new BufferedReader( new
InputStreamReader(System.in));
        /* Modifiquei a partir daqui */
        Socket clientSocket = new Socket("192.168.182.153", 6788);
        /* Fim das modificações */
        DataOutputStream outToServer = new
DataOutputStream(clientSocket.getOutputStream());
        BufferedReader inFromServer = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
        System.out.println("Informe uma cadeia a ser mofificada: ");
        sentence = inFromUser.readLine();
        outToServer.writeBytes(sentence + '\n');
        modifiedSentence = inFromServer.readLine();
        System.out.println("FROM SERVER: " + modifiedSentence);
        clientSocket.close();
    }
}

```

4) Agora implemente um servidor que recebe dois valores a e b de um cliente usando o mesmo esqueleto do código provido TCPClient e TCPServer. O servidor efetua a soma de a e b e devolve a soma ao servidor. Ambos o cliente e o servidor usam sockets TCP para se comunicarem. O número de porta fica a critério do desenvolvedor e pode ser um número aleatório como 8000.

TCPClient.java:

```

import java.io.*;
import java.net.*;

```

```

class TCPClient

```

```

{
    public static void main(String argv[]) throws Exception
    {
        String sentence;
        String modifiedSentence;
        BufferedReader inFromUser = new BufferedReader( new
InputStreamReader(System.in));
        Socket clientSocket = new Socket("localhost", 6788);
        DataOutputStream outToServer = new
DataOutputStream(clientSocket.getOutputStream());
        BufferedReader inFromServer = new BufferedReader(new

```

```

InputStreamReader(clientSocket.getInputStream()));
    /* Modifiquei a partir daqui */
    System.out.println("Informe inteiros a e b no formato a,b: ");
    /* Fim das modificações */
    sentence = inFromUser.readLine();
    outToServer.writeBytes(sentence + '\n');
    modifiedSentence = inFromServer.readLine();
    System.out.println("FROM SERVER: " + modifiedSentence);
    clientSocket.close();
}
}

```

TCPServer.java:

```

import java.io.*;
import java.net.*;

```

```

class TCPServer

```

```

{
    public static void main(String argv[]) throws Exception {
        String clientSentence;
        String capitalizedSentence;
        ServerSocket welcomeSocket = new ServerSocket(6788);

        while(true) {
            Socket connectionSocket = welcomeSocket.accept();
            BufferedReader inFromClient =
                new
InputStreamReader(connectionSocket.getInputStream());
                DataOutputStream outToClient =
DataOutputStream(connectionSocket.getOutputStream());
            clientSentence = inFromClient.readLine();
            /* Modifiquei a partir daqui */
            String[] parts = clientSentence.split(",");
            Integer a = Integer.parseInt(parts[0]);
            Integer b = Integer.parseInt(parts[1]);
            Integer sum = a+b;
            System.out.println("Sum: " + sum);
            outToClient.writeBytes(Integer.toString(sum)+'\n');
            /* Fim das modificações */
        }
    }
}

```

