

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Redes de Computadores 1

Trabalho Prático 1

Guilherme Marx Ferreira Tavares
Professor - Theo Silva Lins

João Monlevade
19 de junho de 2017

Sumário

1	Introdução	2
2	Objetivos	3
3	Implementação	4
4	Apêndice	5

1 Introdução

Desde o início da computação, foi necessária criar uma comunicação entre máquinas diferentes. Para padronizar essa comunicação, foram criados diversos protocolos. Hoje, temos a rede mundial de computadores, onde as máquinas estão todas conectadas por esses protocolos.

2 Objetivos

Esse trabalho tem como objetivo aplicar o conceito de socket em rede de Computadores. Será implementada uma comunicação entre dois sistemas diferentes.

O cliente dessa comunicação irá receber uma sequência de 10 números inteiros e enviá-los ao Servidor. O Servidor deverá ordenar esses números inteiros e retornar ao Cliente.

3 Implementação

Primeiramente, é iniciado um servidor na porta 55555(vide códigos no apêndice). Esse servidor ficará esperando a conexão do Cliente.

No Cliente, primeiramente é criado um Scanner para receber as entradas do teclado que o usuário digitará. Em seguida, inicia-se uma conexão com o Servidor e cria-se os canais de entrada e saída de dados.

Em seguida o cliente envia os dados recebidos pelo Scanner pro Servidor.

Novamente no servidor, esses dados recebidos como string primeiramente devem ser "cortados" em strings menores a fim de transformar esses números recebidos como chars em inteiros. Uma vez passados para inteiros, o comando `Arrays.sort(vetor)` é utilizado pra ordenar esse vetor. O vetor ordenado é passado, então, para uma string e retornado pro Cliente. Com isso, o Servidor encerra seu ciclo de operação.

No Cliente, é recebido esse valor e exibido na tela e, em seguida, é fechada a conexão.

4 Apêndice

Código 1: mainCliente.java

```
1 import java.io.DataInputStream;
2 import java.io.DataOutputStream;
3 import java.net.Socket;
4 import java.util.Scanner;
5
6 public class mainCliente {
7
8     public static void main(String[] args) {
9
10
11         Scanner in = new Scanner(System.in);
12
13         System.out.printf("Informe os 10 numeros na mesma linha separados
14                             por espacos: ");
15         String numero = in.nextLine();
16
17         try {
18             Socket conexao = new Socket("localhost", 55555); // Conectando ao
19                             servidor local na porta 5555
20             System.out.println("Detalhes da conexao: " + conexao.toString());
21
22             DataOutputStream saida = new DataOutputStream(conexao.
23                 getOutputStream()); // Canal de saida
24             DataInputStream entrada = new DataInputStream(conexao.
25                 getInputStream()); // Canal de entrada
26
27
28             System.out.println("Enviando os numeros lidos para o Servidor..."
29                 );
30
31             saida.writeUTF(numero); // Envia o numero lido para o servidor em
32                             formato de STRING codificada em UTF
33
34             System.out.println("Esperando a resposta do Servidor");
35
36             String resposta = entrada.readUTF(); // Recebe o resultado do
37                             Servidor em formato de STRING codificada em UTF
38
39             System.out.println("Vetor ordenado: " + resposta);
40
41             in.close();
42             System.out.println("Encerrando conexao");
43             conexao.close();
44         } catch (Exception e) {
45             e.printStackTrace();
46         }
47     }
48 }
```

Código 2: mainServidor.java

```
1 import java.io.DataInputStream;
```

```

2 import java.io.DataOutputStream;
3 import java.io.IOException;
4 import java.net.ServerSocket;
5 import java.net.Socket;
6 import java.util.Arrays;
7
8 public class mainServidor {
9
10
11     public static void main(String[] args) {
12         // TODO Auto-generated method stub
13
14         System.out.print("Iniciando servidor na porta 55555\n");
15         try {
16
17             ServerSocket servidor = new ServerSocket(55555);
18             Socket novaConexao = servidor.accept();
19
20             DataOutputStream saida = new DataOutputStream(novaConexao.
                getOutputStream()); // Canal de saida
21             DataInputStream entrada = new DataInputStream(novaConexao.
                getInputStream()); // Canal de entrada
22
23             String valores = entrada.readUTF();
24
25             System.out.println("Vetor recebido: " + valores);
26
27             System.out.println("Ordenando o vetor...");
28             String[] n = valores.split(" ");
29
30             int[] numeros = new int[10];
31
32             for(int i=0;i<10;i++)
33             {
34                 numeros[i] = Integer.parseInt(n[i]);
35             }
36
37             Arrays.sort(numeros);
38             System.out.println("Vetor ordenado: ");
39             for(int i =0; i<10;i++)
40             {
41                 System.out.printf(numeros[i] + " ");
42             }
43             System.out.printf("\n");
44             String resultado = new String();
45
46             resultado = String.format(numeros[0] + " " + numeros[1] + " " +
                numeros[2] + " " + numeros[3] + " " + numeros[4] + " " +
                numeros[5] + " " + numeros[6] + " " + numeros[7] + " " +
                numeros[8] + " " + numeros[9]);
47
48             System.out.println("Enviando resultado de volta pro cliente...");
49             saida.writeUTF(resultado);
50
51             System.out.println("Encerrando servidor...");
52             novaConexao.close();
53             servidor.close();

```

```
54
55     } catch (IOException e) {
56         // TODO Auto-generated catch block
57         e.printStackTrace();
58
59     }
60
61
62 }
63
64 }
```
