Aula prática 2

Monitoria InfraCom 2020.3

Josenildo Vicente (jva)

Objetivo

- Programação Sockets em Java
  - RTT
  - TCP
  - UDP

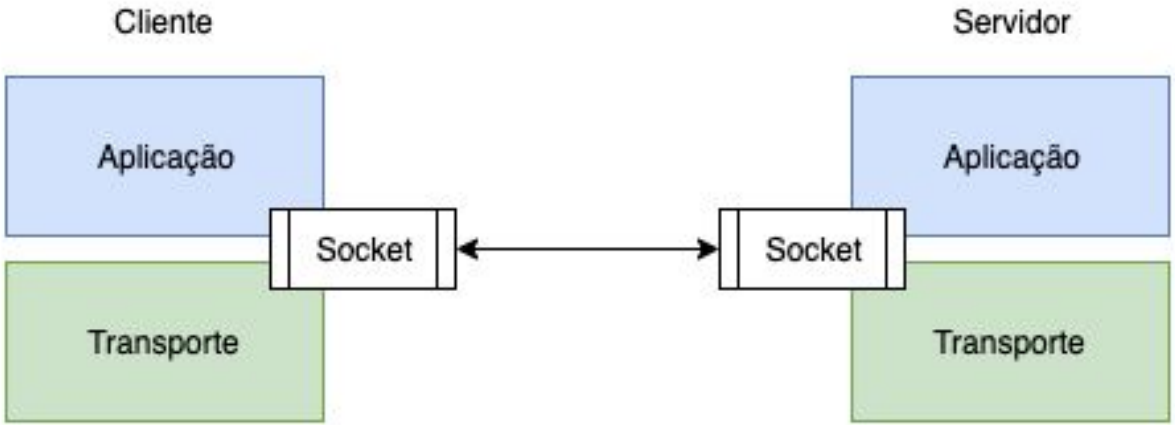
Eclipse

Download -> https://www.eclipse.org/downloads/

Socket

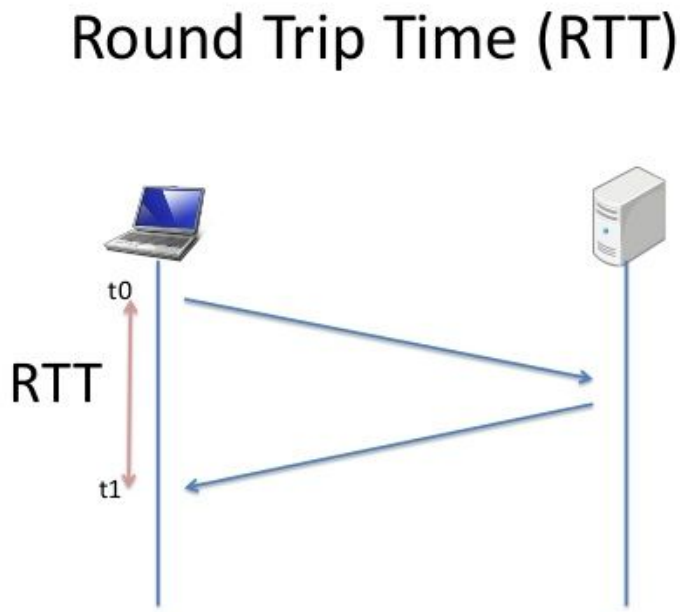
Faz a ligação entre a camada de aplicação e a camada de transporte, é a interface que fica entre as duas camadas.

É tipo uma porta em que os processos enviam/recebem mensagens para/dos sockets.



RTT

É o tempo gasto entre o envio de uma mensagem e a sua resposta.



TCP

Protocolo de controle de transmissão

Protocolo com entrega confiável e ordenada, controle de congestionamento, controle de fluxo, estabelecimento de conexão.

UDP

Protocolo de datagrama do usuário

Protocolo com entrega não confiável e não ordenada, extensão sem ‘gorduras’ do ‘melhor esforço’ do IP.

TCP - Servidor

```java
public class servidor {

    public static void main(String[] args) {
        int porta = 3001;

        try {
            ServerSocket tmpsocket = new ServerSocket(porta);
            System.out.println("Aguardando cliente");
            Socket socket = tmpsocket.accept();

            InputStreamReader entrada = new InputStreamReader(socket.getInputStream());
            BufferedReader le = new BufferedReader(entrada);
            String resposta = le.readLine();
            System.out.println("Cliente: " + resposta);
            socket.close();

        } catch (BindException e) {
            System.out.println("Endere\uFFFDo em uso");
        } catch (Exception e) {
            System.out.println("Erro " + e);
        }

    }

}
```

TCP - Cliente

```java
public class Cliente {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int porta = 3001;
        String endereco = "localhost";

        System.out.print("Digite uma mensagem: ");
        Scanner in = new Scanner(System.in);
        String mensagem = in.nextLine();

        try {
            Socket socket = new Socket(endereco, porta);
            DataOutputStream saida = new DataOutputStream(socket.getOutputStream());
            saida.write(mensagem.getBytes());

            System.out.println("Mensagem enviada.");
            socket.close();

        } catch (ConnectException e) {
            System.out.println("N□o foi poss□vel chegar ao destinat□rio");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```

https://docs.oracle.com/javase/9/docs/api/java/net/ServerSocket.html

https://docs.oracle.com/javase/9/docs/api/java/net/Socket.html

https://docs.oracle.com/javase/9/docs/api/java/io/DataInputStream.html

https://docs.oracle.com/javase/9/docs/api/java/io/DataOutputStream.html

UDP (RTT) - Servidor

```java
public class Servidor {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub

        DatagramSocket serverSocket = new DatagramSocket(5000);
        byte[] receberDados = new byte[1];
        byte[] enviarDados;
        InetAddress ipCliente;
        int porta;
        while(true) {
            DatagramPacket receberPacote = new DatagramPacket(receberDados, receberDados.length);
            serverSocket.receive(receberPacote);
            ipCliente = receberPacote.getAddress();
            porta = receberPacote.getPort();
            enviarDados = ("2").getBytes();
            DatagramPacket enviarPacotes = new DatagramPacket(enviarDados, enviarDados.length,ipCliente, porta);
            serverSocket.send(enviarPacotes);
        }

    }

}
```

UDP (RTT) - Cliente

```java
public class Cliente {

    public static void main(String[] args) throws IOException{
        // TODO Auto-generated method stub
        int porta = 5000;
        String endereco = "localhost";

        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress ipServidor = InetAddress.getByName(endereco);
        byte[] enviarDados;
        enviarDados = ("1").getBytes();
        DatagramPacket enviarPacote = new DatagramPacket(enviarDados, enviarDados.length, ipServidor, porta);

        long tempInicial = System.nanoTime();
        clientSocket.send(enviarPacote);
        byte[] receberDados = new byte[1];
        DatagramPacket receberPacote = new DatagramPacket(receberDados, receberDados.length);
        clientSocket.receive(receberPacote);
        System.out.println("RTT: " + (System.nanoTime() - tempInicial)/1000 );
        clientSocket.close();
    }

}
```

https://docs.oracle.com/javase/9/docs/api/java/net/DatagramSocket.html

https://docs.oracle.com/javase/9/docs/api/java/net/DatagramPacket.html

https://docs.oracle.com/javase/9/docs/api/java/net/InetAddress.html

https://docs.oracle.com/javase/9/docs/api/overview-summary.html

Atividade

Modifique o sistema UDP para que o cliente envie mensagem do teclado para o servidor e o servidor imprima a mensagem no console.

entregar pelo classroom

Até: 02/10 23:59