

Rafhael Rodrigues Cunha

Marcelo Maia Lopes

*Redes e Sistemas Distribuídos: Trabalho
Prático II*

Alegrete - RS, Brasil

20 de Maio de 2012

Rafhael Rodrigues Cunha

Marcelo Maia Lopes

*Redes e Sistemas Distribuídos: Trabalho
Prático II*

UNIVERSIDADE FEDERAL DO PAMPA
ENGENHARIA DA SOFTWARE

Alegrete - RS, Brasil

20 de Maio de 2012

Sumário

1	Introdução	p. 3
2	Manual do Usuário	p. 4
2.1	Instação	p. 4
2.2	Linguagem Utilizada	p. 4
	Referências	p. 5

1 Introdução

Esse Documento tem por objetivo auxiliar o usuário para rodar a aplicação cliente/servidor UDP que desenvolvemos.

2 *Manual do Usuário*

2.1 Instalação

Para utilizar o cliente servidor UDP que desenvolvemos é necessário pegar a pasta do projeto que vai estar contida no zip no qual vai ser enviado como entrega final do Trabalho. Logo após isso é necessário vincular o mesmo a uma IDE de programação que suporte a linguagem java (de preferência o Netbeans) e abrir duas classes: Após abrir a classe TCPServer (o nome da classe é irrelevante estamos utilizando o protocolo UDP para implementar a mesma), faremos a seguinte alteração:

- `DatagramSocket serverSocket = new DatagramSocket(2222);` Esse 2222 entre parênteses refere-se a porta na qual eu gostaria que meu server roda-se. Caso essa não seja a sua escolhida, e só trocar ali.
- Após essa pequena alteração de porta, e só rodar a classe do servidor.

Para realmente começarmos a utilizar o sistema precisamos configurar nossa classe ClienteUDP Para isso na seguinte linha iremos colocar o ip do nosso server:

- `InetAddress IPAddress = InetAddress.getByName("201.66.204.109");` Lembrando que se deseja testar em sua própria máquina substitua o que está entre aspas por localhost.

Após, vamos alterar a seguinte linha:

- `new DatagramPacket(sendData, sendData.length, IPAddress, 2222);` Onde o 2222 é a porta na qual efetuei comunicação com o meu servidor (lembro que configuramos isso na hora de rodar o servidor.)

Após essas configurações, rodamos a classe ClienteUDP, informamos o número de requisições que a mesma deseja fazer ao servidor e logo recebemos como resposta na tela a impressão da que demorou menos tempo para ser concluída.

2.2 Linguagem Utilizada

A linguagem utilizada de apoio para desenvolvimento da solução do trabalho proposto foi o java, porque além de ele ser multiplataforma, ou seja, rodar em diversos sistemas operacionais ele também possui muitos tratamentos na parte de sockets que é uma classe utilizada na implementação da solução que facilita na parte de comunicação entre um cliente e outro. [1]

Referências

- [1] Oracle. Lesson: All about sockets. Disponível em: <http://docs.oracle.com/javase/tutorial/networking/sockets/>. Acessado em 06/05/2012.