



Universidade Federal do Ceará  
Centro de Tecnologia  
Departamento de Engenharia de Teleinformática  
Redes de Computadores, 2007.2  
Professor: Menotti Machado  
Colaborador: Lius Fontenelle Carneiro

### **Laboratório 3: Programação para Redes – Multithread e Multicast**

Entrega do relatório: 28 de setembro de 2007

Na prática anterior desenvolvemos servidores e clientes simples, mostrando as funcionalidades básicas da programação para redes com sockets. Porém, estávamos limitados a comunicação entre apenas dois nós (um servidor e um cliente). Resolveremos o problema através do uso de multithreads.

Um dos objetivos desta prática de laboratório é aprimorar os programas anteriormente desenvolvidos para obter esta funcionalidade (concorrência). Mantendo a continuidade, abordaremos a parte prática do laboratório com a linguagem Java, lembrando que os alunos continuam livres para escolher o ambiente de desenvolvimento para realizar os exercícios solicitados.

Depois de explicado o conceito de multithread, o servidor de echo baseado em sockets TCP, desenvolvido no Laboratório 2 será reescrito para suportar mais de um cliente ao mesmo tempo. Para tanto, a parte básica sobre paralelismo com threads usando Java será explicada.

A segunda parte da prática trata sobre um ponto muito importante no desenvolvimento de software distribuído atualmente: a otimização da propagação da informação. Dentre várias técnicas existentes, utilizaremos o Multicast para construir um servidor de notificações assíncrono que fará o "data push" para os clientes sempre que necessário.

O uso de threads também será necessário para desenvolver o servidor multicast, que será o servidor de echo multithread modificado, com a funcionalidade adicional de enviar notificações aos clientes, que também terão de ser modificados para suportar o multicast.

#### **Problema proposto**

Para colocar em prática os conceitos vistos sobre multithread, crie um servidor web que aceite múltiplas requisições simples do método GET, leia corretamente a requisição e monte uma resposta adequada ao padrão do protocolo. O servidor deverá responder corretamente a um navegador web comum, como o Firefox.

Para praticar conceitos vistos sobre multicast, modifique o servidor de chat escrito previamente (ou crie um novo) para que ele suporte múltiplas conexões (utilizando multithreads) e envie alguma notificação para os clientes, que também deverão ser modificados para recebê-las.

## **Referências – Multithread**

*Creating a Simple, Multithreaded Chat System with Java*

<http://www.acm.org/crossroads/xrds6-1/ovp61.html>

*A Simple Multithreaded Web Server*

<http://java.sun.com/developer/technicalArticles/Networking/Webserver/>

*QuickServer Framework*

<http://www.quickserver.org/docs/features.html>

## **Referências – Multicast**

*Multicast na Wikipedia*

<http://en.wikipedia.org/wiki/Multicast>

*IGMP (Internet Group Management Protocol)*

<http://www.networksorcery.com/enp/protocol/igmp.htm>

*Descrição dos endereços dos grupos de multicast*

[http://en.wikipedia.org/wiki/Multicast\\_address](http://en.wikipedia.org/wiki/Multicast_address)

*Vídeo sobre multicast da Cisco*

[http://www.cisco.com/en/US/products/ps6552/products\\_ios\\_technology\\_home.html](http://www.cisco.com/en/US/products/ps6552/products_ios_technology_home.html)