

# **Primeiro Trabalho**

## **Implementação de um Servidor Web Multithreaded**

### **1. Objetivo**

O objetivo deste trabalho é a implementação de um servidor Web "multithreaded", que deverá ser capaz de processar múltiplas requisições simultâneas de serviços em paralelo. Implementaremos a versão 1.0 do HTTP, definida no RFC 1945, onde mensagens de pedido HTTP separadas são enviadas para cada objeto da página Web. Este servidor deverá manipular múltiplas requisições simultâneas de serviços em paralelo. Isso significa que o servidor Web é "multithreaded". Ele deverá ser capaz de atender pedidos de transferência de arquivos gerados a partir de um browser comum, como o Firefox. Os pedidos serão apresentados em uma porta específica e o servidor Web deverá aceitar conexões não persistentes e simultâneas de um número arbitrário de clientes Web.

### **2. Introdução**

O servidor Web é um dos componentes da aplicação Web e se baseia num programa servidor. Quando esse programa é executado, cria-se o processo servidor que se mantém na "escuta" pelos clientes Web que queiram extrair documentos usando o protocolo HTTP. Por meio de conexões TCP, ele recebe pedidos e envia respostas até que o cliente Web ou o próprio servidor Web decida encerrar a conexão. Embora o protocolo HTTP ofereça muitas opções, pretende-se implementar aqui uma forma simplificada de servidor HTTP.

Além dos livros indicados na bibliografia do curso e da exposição em aulas sobre o protocolo HTTP, a referência bibliográfica [1] oferece uma descrição simplificada que pode facilitar a compreensão e implementação do protocolo. Em [2] e [3] tem-se uma descrição completa dos protocolos HTTP 1.0 e HTTP 1.1. Todavia, devido à extensão e ao nível de detalhamento, não se espera que os alunos estudem esses dois RFCs.

### **3. Execução do Trabalho**

Estude as seções 2.7, 2.8 e 2.9 da referência [4] e execute a Tarefa 1 de programação apresentada na página 138. Para isso, leia o texto: "Tarefa de Programação 1:

Construindo um servidor Web multithreaded", disponível na sub-pasta "Primeiro Trabalho" da pasta "Trabalhos" do canal Geral da equipe Redes no Teams.

Obs.:

- a) Qualquer edição mais recente da referência [4] também poderá ser utilizada.
- b) O trabalho deverá ser executado individualmente. As cópias ou indícios de plágio, se detectadas, receberão nota zero.

**Dica Importante:** "Comece o seu trabalho o mais cedo possível". Os trabalhos sempre parecem mais fáceis de fazer enquanto você ainda não começou.

#### 4. Testes

Uma parte muito importante da programação em geral é saber como testar e depurar os programas. Há muitas maneiras de executar esta etapa do trabalho. Seja criativo(a). Pretende-se saber como o servidor foi testado, e como se convenceu de que ele realmente funciona. Para isso, documente os testes realizados.

#### 5. Entrega do Trabalho

Deverá ocorrer na forma de um relatório sobre a realização do trabalho, incluindo uma enumeração dos testes de funcionamento realizados e os resultados obtidos.

Os arquivos de relatório e do programa servidor Web implementado deverão ser entregues por meio do e-mail [prguardieiro@ufu.br](mailto:prguardieiro@ufu.br) (solicite confirmação de recebimento de seu e-mail).

#### 6. Referências

- [1] Marshall, J.; "HTTP Made Really Easy", disponível no URL:  
<http://www.jmarshall.com/easy/http/>
- [2] Berners-Lee, T.; et al.; "Hypertext Transfer Protocol: HTTP/1.0", RFC 1945.
- [3] Fielding, R.; et al.; "Hypertext Transfer Protocol: HTTP/1.1", RFC 2616.
- [4] Kurose, J. F. and Ross, K. W.; "Redes de Computadores e a Internet – Terceira Edição", Addison-Wesley. Disponível na biblioteca: 681.3.02 INTERN K96cP 5.ed.o
- [5] Kurose, J. F.; Ross, K. W.; "Computer Networking - A Top-Down Approach - Sixth Edition", Pearson, 2013. Disponível na biblioteca: 681.3.02 INTERN K96c 6.ed
- [6] Kurose, J. F.; Ross, K. W.; "Computer Networking - A Top-Down Approach - 8th Edition", Pearson, 2021.