

TRABALHO PRÁTICO 2 EM REDES DE COMPUTADORES

**Departamento de Ciência da Computação
Instituto de Ciência Exatas
Universidade Federal de Minas Gerais
Belo Horizonte - MG - Brasil
Novembro de 2023**

Gustavo Freitas Cunha - 2020054498
gustavocunha@dcc.ufmg.br

1. INTRODUÇÃO

O projeto consiste em um servidor que gerencia um blog virtual. É possível que múltiplos clientes se conectem simultaneamente ao servidor para publicar conteúdo e interagir com os tópicos. Cada cliente é identificado por um ID único, e os tópicos são criados pelos clientes à medida que eles publicam em um tópico. O servidor notifica os clientes inscritos quando novas publicações são feitas em tópicos que eles acompanham.

2. DESAFIOS E SOLUÇÕES

2.1 Comunicação entre Servidor e Cliente

Um dos principais desafios do projeto foi estabelecer uma comunicação eficiente entre o servidor e os clientes. A comunicação entre processos ocorre por meio de sockets, mas garantir que as mensagens fossem transmitidas corretamente e no formato adequado foi um desafio.

Para resolver esse desafio, foi definido um protocolo de comunicação específico, onde as mensagens são serializadas em estruturas de dados (structs) e enviadas por meio da função send. Os clientes e o servidor usam estruturas de dados compartilhadas (por exemplo, BlogOperation) para trocar informações. O uso de uma estrutura comum garantiu que ambas as partes entendessem o formato das mensagens.

2.2 Gerenciamento de Conexões

Gerenciar múltiplas conexões de clientes ao servidor é complexo, especialmente quando clientes se conectam e desconectam de forma dinâmica.

Foi implementada uma abordagem baseada em threads para lidar com múltiplos clientes. Cada novo cliente que se conecta ao servidor é atendido por uma nova thread. Isso permite que o servidor atenda a múltiplos clientes simultaneamente. Além disso, um vetor de estruturas foi criado para armazenar informações sobre cada cliente, como seu ID e status de conexão. Esse vetor é acessado de forma segura com o uso de mutexes para evitar condições de corrida.

2.3 Inscrições em Tópicos

Permitir que os clientes se inscrevam em tópicos específicos também foi desafiador. O servidor precisa manter um registro das inscrições e notificar os clientes adequadamente.

O servidor mantém uma estrutura de dados para rastrear as inscrições de clientes em tópicos. Quando um cliente se inscreve em um tópico, o servidor verifica se o tópico já existe e inscreve o cliente. Para notificar os clientes inscritos sobre novas publicações, o servidor percorre a lista de inscrições e envia mensagens para os clientes relevantes. Isso é feito de forma segura usando mutexes para evitar problemas de concorrência.

2.4 Tratamento de Comandos do Cliente

O tratamento de comandos dos clientes, como publicar em um tópico, listar tópicos disponíveis, se inscrever e desinscrever em tópicos e sair do sistema, requer a análise e processamento correto dos comandos.

O servidor e o cliente têm um conjunto de comandos específicos definidos. Cada comando é processado de acordo com seu tipo. Por exemplo, ao receber um comando "publish", o servidor cria um novo tópico se necessário e notifica os inscritos. O uso de estruturas de dados compartilhadas facilita o tratamento consistente dos comandos e o envio de respostas aos clientes.

2.5 Notificações de Novas Publicações

Notificar os clientes sobre novas publicações em tópicos que eles acompanham requer um mecanismo eficiente de notificação em tempo real.

A solução adotada é que o cliente mantenha um thread separada para receber notificações de novas publicações. Quando uma nova publicação é

feita em um tópico, o servidor envia notificações para todos os clientes inscritos neste tópico. Isso permite que os clientes recebam as notificações em tempo real, mantendo a eficiência da comunicação.

3. CONCLUSÃO

O desenvolvimento deste trabalho apresentou desafios relacionados à comunicação, gerenciamento de conexões, inscrição em tópicos e notificação de novas publicações. As soluções adotadas, como o uso de threads para gerenciamento de conexões e notificações em tempo real, bem como a definição de um protocolo de comunicação consistente, permitiram, com algum esforço, que o projeto fosse implementado. O resultado é um sistema que permite aos clientes interagirem com tópicos de forma eficaz e em tempo real.