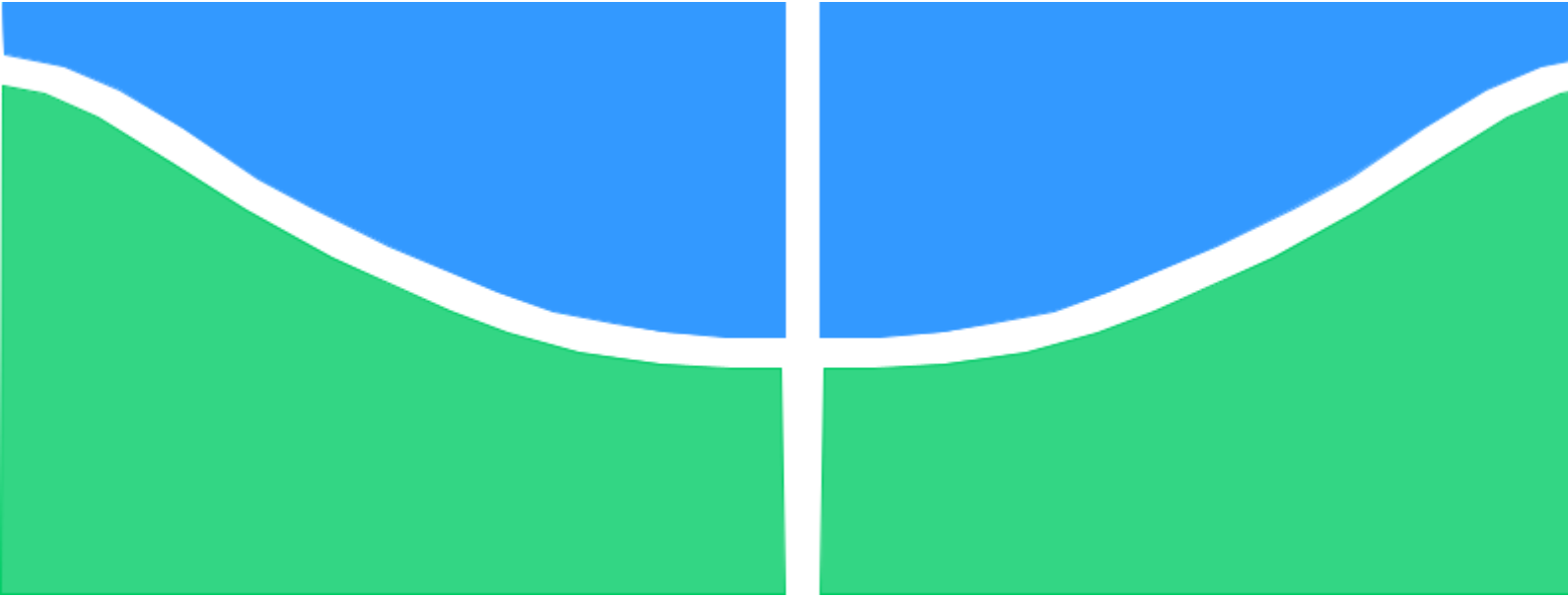

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia de Software

Projeto de pesquisa - Criando ambientes virtuais de conversação com uso system call select()

Disciplina: Fundamentos de Redes de Computadores, T01

Autor(es): André Macedo 190102390
Diógenes Dantas 190105267
Professor: Fernando W. Cruz

Brasília, DF, 2023

1. INTRODUÇÃO

Esse documento explica como foi realizado o projeto de pesquisa da disciplina de Fundamentos de Redes de Computadores do professor Fernando W. Cruz.

Descrição do Problema: Implementar um código em C que atenda os seguintes requisitos.

- A criação de salas virtuais de bate-papo com nome da sala e limite de participantes
- Listar participantes de uma determinada sala
- Permitir ingresso de clientes, com um identificador, em uma sala existente, de acordo com o limite admitido para a sala
- Saída de clientes de uma sala em que estava participando
- Diálogo entre os clientes das salas

Objetivos: Este projeto tem como objeto, permitir que o aluno compreenda a arquitetura de aplicações de rede (segundo arquitetura TCP/IP) que envolvam envolvam gerência de diálogo. Para isso, devem construir uma aplicação que disponibiliza salas de bate-papo virtuais, nas quais os clientes podem ingressar e interagir.

2. METODOLOGIA

A plataforma **GitHub** foi escolhida como o repositório de código fonte do projeto, bem como para o gerenciamento do projeto e alocação do arquivo da apresentação e do relatório.

As reuniões a respeito do projeto foram realizadas às Segundas e Quartas-Feiras, após as aulas de Fundamentos de Redes de Computadores. A cada encontro era discutido informações pertinentes ao projeto e os passos que seriam necessários para atingir o objetivo final do projeto

A gravação do vídeo explicando o projeto foi feito pelo Discord e o relatório escrito no Google Docs.

3. DESCRIÇÃO DA SOLUÇÃO

A solução implementada neste código é baseada em um servidor de chat que permite aos clientes se conectarem a diferentes salas de chat. Cada cliente pode entrar em uma sala específica usando o comando "/join <sala>" e sair da sala usando o comando "/exit".

A lógica por trás da solução é a seguinte:

1. O servidor cria um socket para ouvir conexões de clientes.
2. Configurações de socket são feitas para permitir a reutilização do endereço e porta do socket.
3. O servidor vincula o socket a um endereço IP e porta específicos.
4. O servidor entra em um loop infinito aguardando conexões e mensagens dos clientes.
5 .Dentro do loop, o servidor usa a função select para verificar quais descritores de arquivo estão prontos para leitura.
6. Se o descritor de arquivo pronto for o socket do servidor, isso significa que há uma nova conexão de cliente. O servidor aceita a conexão usando a função accept, envia uma mensagem de boas-vindas para o cliente e adiciona o novo descritor de arquivo à lista de todos os sockets.
7. Se o descritor de arquivo pronto for um cliente existente, isso significa que o cliente enviou uma mensagem. O servidor lê a mensagem usando a função recv e a encaminha para a função handle_request.
8. A função handle_request verifica se a mensagem começa com o caractere "/". Se sim, é um comando. Atualmente, suporta apenas os comandos "/join" e "/exit". O comando "/join" permite que o cliente entre em uma sala específica, enquanto o comando "/exit" faz o cliente sair da sala atual.
9. Se a mensagem não for um comando, ela é enviada para a função send_msg_to_room, que envia a mensagem para todos os clientes na mesma sala do remetente, exceto o próprio remetente.

Essa solução permite que os clientes se comuniquem em diferentes salas de chat, garantindo que apenas os clientes na mesma sala recebam as

mensagens enviadas. Ela estabelece uma comunicação bidirecional entre o servidor e os clientes, permitindo um bate-papo interativo em tempo real.

4. CONCLUSÃO

Sobre o projeto: aprendeu-se sobre bibliotecas de sockets para criar um servidor TCP/IP e lidar com a comunicação entre servidor e clientes. Foi aprendido a respeito da criação de sockets, configuração de opções do socket, vinculação de endereços IP e portas, bem como aceitação de conexões e troca de dados. Além disso, aprendeu-se também como rastrear e controlar as associações de clientes e salas, garantindo que as mensagens sejam entregues apenas aos destinatários corretos.

Melhorias: Uma melhoria para o bate-papo seria exibir o nome dos participantes antes de suas mensagens e notificar quando alguém sair da sala.

Sobre como participaram: o projeto foi feito em dupla, com divisão e gerenciamento de projeto através do GitHuB

Autoavaliação: o grupo acredita que o projeto atingiu os requisitos pedidos pelo professor, porém poderia ter melhorias em relação ao limite de participantes e aspectos ligados a interface. Acredita-se em uma nota em torno de 8.5 e 9.5. Relacionados à participação da dupla, acredita-se que o grupo conseguiu trabalhar bem. O aluno André Macedo teve uma função importante no raciocínio da lógica do chat TCP/IP e no desenvolvimento do código em C. O aluno Diógenes teve uma função importante na organização do código, e na estrutura do relatório. O grupo acredita que as notas individuais pode ser assim: André: entre 9 e 10, Diógenes: entre 9 e 10.

5. REFERÊNCIAS

Materiais disponíveis no aprender da Disciplina

Código base do Professor disponível no aprender

https://www.tutorialspoint.com/unix_system_calls/_newselect.htm