

Faculdade do Gama
Universidade de Brasília

Fundamentos de Rede de computadores

Professor: Fernando William Cruz

Alunos: Christian Fleury Alencar Siqueira - 190011602

Thiago Siqueira Gomes -1293192391293

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

Introdução.....	1
Metodologia.....	1
Solução.....	1
Conclusão.....	1
Referências.....	1

Introdução

O presente relatório tem como objetivo descrever o trabalho desenvolvido de criação de um cliente e um servidor em TCP, utilizando a linguagem C e a função select. A implementação consiste em um sistema de salas de bate-papo, onde múltiplos clientes podem se conectar ao servidor e interagir com outros clientes dentro de salas específicas.

Metodologia

A metodologia adotada para o desenvolvimento do sistema de bate-papo envolveu a colaboração de dois integrantes por meio da plataforma Discord para se comunicar sobre o projeto. Durante a implementação, foram utilizadas contribuições isoladas assíncronas, permitindo que cada integrante trabalhasse em diferentes aspectos do sistema em momentos convenientes para cada um dos integrantes. Além disso, também foi adotada a prática de pair programming em algumas etapas, permitindo uma colaboração mais direta e em tempo real.

O código base do projeto foi fornecido pelo professor e consistia em um arquivo com a implementação de uma função select. A partir desse ponto de partida, os integrantes do projeto realizaram modificações e adicionaram novas funcionalidades para a criação das aplicações cliente e servidor.

Solução

Foram desenvolvidas duas aplicações em C, um cliente e um servidor, para implementar um sistema de bate-papo. O servidor foi projetado para gerenciar as interações entre os clientes, enquanto o cliente permite que os usuários se conectem e participem das salas de bate-papo.

O servidor possui um conjunto de funções de gerenciamento de diálogo que podem ser consumidas pelo cliente. Essas funções incluem a criação, entrada, saída ou exclusão de salas, bem como a listagem das salas disponíveis e dos clientes presentes em uma sala específica. Cada sala é única por nome.

Comandos disponíveis:

```
[chfleury@fedora salada]$ telnet 127.0.0.0 8000
Trying 127.0.0.0...
Connected to 127.0.0.0.
Escape character is '^]'.
Bem vindo(a), voce esta no saguao.
-----LISTA DE COMANDOS-----.
$setname <nome> para escolher um nome.
$join <nome_da_sala> para entrar numa sala.
$listrooms para listar salas existentes.
$create <nome_da_sala> para criar uma sala.
$listroomclients <id_da_sala> para listar clientes de uma sala.
$delete <nome_da_sala> deleta a sala e manda os participantes pro saguao
$lobby Te faz sair da sala e retornar ao saguao
```

O cliente tem acesso a uma série de comandos que permitem interagir com o servidor. O comando "setname <nome>" permite que o cliente escolha um nome para si mesmo. O comando "join <nome_da_sala>" permite que o cliente entre em uma sala existente. O comando "listrooms" lista as salas disponíveis. O comando "create <nome_da_sala>" cria uma nova sala. O comando "listroomclients <id_da_sala>" lista os clientes presentes em

uma sala específica. E o comando "delete <nome_da_sala>" permite que o cliente delete uma sala, movendo os participantes para o saguão.

Essa solução oferece aos usuários a capacidade de se conectarem e interagirem em um ambiente de bate-papo estruturado. Os comandos fornecidos pelo cliente e as funções implementadas no servidor permitem a criação, gerenciamento e exclusão de salas, bem como a movimentação dos clientes entre as salas e o saguão.

Obs: Por padrão o servidor cria 5 salas iniciais que os clientes podem ingressar sendo essas:

Sala 1, Sala 2, Sala 3, Sala 4 e Sala 5

Conclusão

O trabalho de implementação do cliente e servidor TCP utilizando select() foi concluído com sucesso. O sistema de chat desenvolvido permite a comunicação assíncrona de múltiplos clientes conectados simultaneamente.

A utilização da função select() possibilita um gerenciamento eficiente de múltiplas conexões, garantindo um desempenho adequado mesmo em cenários com muitos clientes ativos.

O trabalho desenvolvido proporcionou um melhor entendimento dos conceitos de comunicação em rede, programação concorrente e gerenciamento de sockets utilizando a linguagem C. Além disso, permitiu explorar a utilização da função select() como uma ferramenta poderosa para lidar com múltiplas conexões em aplicações de rede.

Referências

<https://man7.org/linux/man-pages/man2/select.2.html>