Criando ambientes virtuais de conversação com uso system call select()
Fundamentos de Redes de Computadores
Professor Fernando William
Universidade de Brasília

Arthur Sena
180030345

Pedro Rodrigues
170062686

Youssef Muhamad
170024334

***Abstract*—O presente projeto tem como objetivo desenvolver uma aplicação para criação de ambientes virtuais de conversação utilizando a chamada de sistema select(). O foco principal do projeto é buscar uma compreensão aprofundada da arquitetura de aplicações de rede (especialmente TCP/IP) que envolvam o gerenciamento de diálogos.**

***Index Terms*—TCP/IP, select, system call.**

1. Introdução

A evolução das tecnologias de comunicação tem impulsionado o desenvolvimento de aplicações de rede cada vez mais interativas e colaborativas. Nesse contexto, este trabalho tem como propósito explorar a criação de ambientes virtuais de conversação por meio da implementação de uma aplicação que utiliza a chamada de sistema select() e a arquitetura TCP/IP.

A arquitetura TCP/IP é amplamente adotada como base para a comunicação em redes de computadores, permitindo a troca de dados de forma confiável e eficiente. A chamada de sistema select() é uma ferramenta poderosa para o desenvolvimento de aplicações de rede, pois permite monitorar vários descritores de arquivo simultaneamente e lidar com eventos de leitura e escrita de forma concorrente.

O objetivo principal deste projeto é explorar a arquitetura de aplicações de rede, especialmente o gerenciamento de diálogos, e entender como o select() pode ser utilizado para criar ambientes virtuais de conversação. Esses ambientes permitem que usuários conectados troquem mensagens e interajam em tempo real, simulando uma experiência de conversação em grupo. Essa experiência nos proporcionou uma base sólida para o desenvolvimento de aplicações de rede mais complexas e interativas.

2. Metodologia

Foram realizados encontros síncronos em plataformas de comunicação online para a implementação do código utilizando a técnica de programação em pareamento. Foram consultados materiais da bibliografia da disciplina para dar suporte ao desenvolvimento, além de discussões com outros grupos. Todo o código foi versionado através do git e publicado na plataforma GitHub.

3. Implementação

O sistema é composto por dois programas principais: server.c e client.c. O programa server.c é responsável por aceitar conexões de clientes, gerenciar mensagens entre clientes e transmitir mensagens para todos os clientes conectados. Por outro lado, o client.c conecta-se ao servidor, envia mensagens ao servidor e recebe mensagens de outros clientes através do servidor.

Ambos os programas, server.c e client.c, utilizam a biblioteca de sockets para estabelecer a comunicação de rede. A comunicação baseia-se no protocolo TCP, que garante a entrega de pacotes na ordem correta e a confiabilidade dos dados transmitidos.

O programa server.c inicia sua execução ouvindo por novas conexões. Quando um novo cliente é conectado, ele é adicionado ao conjunto de sockets do servidor e pode enviar mensagens para o servidor. O servidor utiliza a chamada de sistema select() para monitorar a atividade dos sockets e tratar eventos de leitura e escrita de forma concorrente. Quando o servidor recebe uma mensagem de um cliente, ele transmite essa mensagem para todos os outros clientes conectados, criando um ambiente virtual de conversação em tempo real.

O programa client.c estabelece uma conexão com o servidor por meio da função connect(). Após a conexão ser estabelecida, o cliente entra em um loop onde aguarda a entrada do usuário. Quando o usuário digita uma mensagem, ela é enviada para o servidor por meio da função send(). O cliente também utiliza a função select() para monitorar a entrada do usuário e a chegada de mensagens do servidor.

A solução implementada permite a comunicação bidirecional entre o servidor e os clientes conectados. O servidor atua como um intermediário, recebendo mensagens dos clientes e distribuindo-as para todos os outros clientes. Dessa forma, os participantes do ambiente virtual de conversação podem trocar mensagens em tempo real, simulando uma experiência de chat em grupo.

Como resultado, o sistema de chat desenvolvido foi capaz de suportar múltiplas conexões de clientes e transmitir mensagens entre eles. As mensagens foram entregues na ordem correta e sem perda de dados, graças ao protocolo TCP. A utilização da chamada de sistema select() permitiu o gerenciamento eficiente dos eventos de comunicação, garantindo um ambiente de conversação fluido e responsivo.

Essa solução representa um exemplo básico e funcional de aplicação de rede utilizando a arquitetura TCP/IP e a chamada de sistema select(). A partir desse projeto, é possível aprimorar e expandir as funcionalidades, como implementar recursos adicionais de autenticação de usuários, criptografia de dados e suporte a recursos multimídia, tornando-o mais robusto e adequado a diferentes cenários de aplicação.

4. Conclusões e Resultados

Como resultado, nosso sistema de chat foi capaz de suportar múltiplas conexões de clientes e transmitir mensagens entre elas. As mensagens foram entregues na ordem correta e sem perda de dados.

Nossa equipe concluiu com sucesso o desenvolvimento de um sistema de chat simples. Através deste projeto, adquirimos uma compreensão mais profunda dos princípios de redes de computadores e dos desafios da programação de rede. As habilidades e o conhecimento que adquirimos neste projeto serão valiosos para nossos futuros estudos e carreiras na área de ciência da computação.

O presente projeto teve como objetivo desenvolver um ambiente virtual de conversação utilizando a chamada de sistema select() e a arquitetura de aplicativos de rede, com foco no gerenciamento de diálogos e na compreensão aprofundada de aplicações de rede TCP/IP.

A implementação do sistema de chat obteve resultados positivos. Foi possível estabelecer conexões TCP/IP entre o servidor e os clientes, permitindo a troca de mensagens de forma eficiente e correta. O sistema suportou múltiplas conexões simultâneas e garantiu a entrega das mensagens na ordem correta.

No entanto, é importante destacar que a solução possui algumas limitações. Primeiramente, não foram implementados recursos avançados, como autenticação de usuários (como foi encorajado pelo professor, como uma funcionalidade extra), criptografia de mensagens ou suporte a recursos multimídia. Além disso, o sistema depende da chamada de sistema select(), o que pode limitar sua escalabilidade em situações com muitas conexões ou alto tráfego de dados.

Apesar das limitações, a implementação do ambiente de conversação em rede apresentou um bom desempenho e demonstrou os princípios fundamentais do gerenciamento de diálogos utilizando a chamada de sistema select(). O sistema fornece uma base sólida para futuras melhorias, como a adição de recursos de segurança, aprimoramentos na interface do usuário e suporte a funcionalidades avançadas.

Em conclusão, o projeto contribuiu para o aprofundamento do conhecimento em aplicações de rede TCP/IP e ofereceu uma solução funcional para a criação de um ambiente virtual de conversação. Embora haja espaço para melhorias e expansões, o sistema implementado atendeu aos objetivos propostos e forneceu uma base sólida para o desenvolvimento de soluções mais robustas e escaláveis no futuro.

Bibliografia

[1] Tanenbaum, A., Computer Networks Prentice-Hall, 5a. edition, 2011