

UNIVERSIDADE DE BRASÍLIA

Faculdade do Gama

Fundamentos de Redes de Computadores

Prof. Fernando W. Cruz

Projeto de pesquisa

Lucas Pimentel Quintão 190016663

Victor Rayan Ferreira 190044390

Matheus Calixto Vaz Pinheiro 190055201

Brasília, DF

2023

1. Introdução

A compreensão da arquitetura de aplicações de rede é essencial para profissionais de TI e estudantes na área de redes de computadores. A arquitetura TCP/IP é amplamente utilizada na internet e em redes corporativas, e compreender seus princípios e elementos é fundamental para a implementação eficiente e segura de sistemas de comunicação.

O presente relatório tem como objetivo descrever a metodologia e a solução desenvolvida para o projeto de pesquisa da disciplina de Fundamentos de Redes de Computadores. Para alcançar esse objetivo, foi desenvolvida uma aplicação que oferece salas de bate-papo virtuais, onde os usuários podem ingressar e interagir entre si. O projeto buscou criar uma solução prática e interativa para que pudéssemos explorar e experimentar os conceitos envolvidos na arquitetura TCP/IP, com foco na gerência de diálogo.

Esperamos que este relatório forneça uma visão abrangente sobre o processo de desenvolvimento da solução e que sirva como um recurso valioso para o aprendizado e a compreensão da arquitetura de aplicações de rede.

O repositório com o código fonte da aplicação pode ser encontrado a partir do seguinte link: <https://github.com/matheuscvp/FRC-trab1>

2. Metodologia

A metodologia adotada para o desenvolvimento da solução do projeto foi baseada em uma abordagem colaborativa, com o uso de ferramentas de comunicação online para facilitar a interação entre os membros do grupo. Devido à dificuldade de encontrar horários em que todos os membros pudessem trabalhar juntos, optamos por marcar reuniões pelo Discord, uma plataforma de chat de voz e texto.

Uma vez que definimos a visão geral da solução, optamos por fazer uma divisão de tarefas entre os membros do grupo. Cada membro ficou responsável por uma parte específica do desenvolvimento da aplicação,

levando em consideração as habilidades individuais e os interesses de cada um.

A combinação da abordagem colaborativa, com reuniões pelo Discord e a divisão de tarefas, juntamente com o uso de ferramentas de comunicação online, permitiu que o grupo trabalhasse de maneira eficaz e produtiva, superando os desafios de agenda e garantindo o progresso constante no desenvolvimento da solução.

3. Solução

A seguir, será apresentada uma descrição da solução do código da aplicação em linguagem C para o chatroom. O código implementa um servidor de chat que permite aos clientes se conectarem, criar salas de chat, entrar em salas existentes, trocar mensagens com outros participantes da sala e realizar outras operações relacionadas ao chat.

A solução é baseada no arquivo “código select()” disponível no moodle e em sockets TCP/IP e utiliza a biblioteca de funções de rede e manipulação de arquivos. Também nela é utilizada a chamada de sistema select para a para monitorar vários descritores de arquivo em busca de eventos de leitura, escrita ou excepcionais, de forma síncrona e eficiente.

O código organiza os dados dos clientes e das salas em estruturas de dados definidas (struct User, struct ChatRoom, e struct Client).

```

struct User {
    int socket;
    char username[50];
};

struct ChatRoom {
    int id;
    char name[MAX_ROOM_NAME];
    int participants;
    int maxParticipants;
    struct User users[MAX_USERS_IN_ROOM];
};

struct Client {
    int id;
    int socket;
    int roomID;
    char username[50];
};

```

Figura 1: Definição das estruturas

A implementação do servidor consiste em um loop principal que aguarda a atividade nos sockets, tanto para novas conexões de clientes quanto para mensagens enviadas pelos clientes conectados. O servidor fornece um menu interativo aos clientes, permitindo-lhes listar salas disponíveis, criar novas salas, entrar em salas existentes, sair do chat e realizar outras ações relacionadas às salas. Além disso, os clientes podem trocar mensagens com outros participantes da sala em que estão conectados.

Em relação às funcionalidades implementadas, a aplicação permite com que o usuário liste as salas existentes, crie uma sala, entre em uma sala, saia de uma sala, saia do chat e liste os usuários presentes em uma sala. Dessa forma, o usuário, por meio de um menu, seleciona a ação que deseja executar.

```

Connected to localhost.
Escape character is '^]'.
Connected to the chatroom.
/1 - Listar Salas
/2 - Criar Sala
/3 - Entrar em Sala
/4 - Sair do Chat
/6 - Sair da sala
/7 - Listar users da sala

```

Figura 2: Menu de funcionalidades

Estas funcionalidades foram implementadas em funções separadas para que dessa forma o código fosse modularizado e se tornasse mais fácil de ser lido e compreendido.

```
void send_menu(int socket);
void list_rooms(int socket, struct ChatRoom rooms[], int totalRooms);
void create_room(int socket, struct ChatRoom rooms[], int* totalRooms);
void join_room(int socket, struct ChatRoom rooms[], int totalRooms, struct Client clients[], int* totalClients);
void leave_room(int socket, struct ChatRoom rooms[], int totalRooms, struct Client clients[], int totalClients);
void disconnect_client(int socket, struct ChatRoom rooms[], int totalRooms, struct Client clients[], int totalClients);
void list_users(int socket, struct ChatRoom rooms[], int totalRooms, int roomID);
```

Figura 3: Assinatura das funções

Para o caso de enviar mensagens, basta que o usuário esteja em uma sala e digite o conteúdo de sua mensagem no console, então o sistema envia a mensagem para todos os outros participantes da sala de chat do usuário. Conforme podemos ver pelo trecho de código abaixo que realiza este processo.

```
// Enviar mensagem para todos os outros participantes da sala
for (int j = 0; j < totalClients; j++)
{
    if (clients[j].roomID == clients[i].roomID && clients[j].socket != sd)
    {
        char message[BUFFER_SIZE];
        strcat(message, buffer);
        strcat(message, "\n");
        send(clients[j].socket, message, strlen(message), 0);
    }
}
```

Figura 4: Código envio de mensagens

O código-fonte completo e comentado dessa solução de chatroom em linguagem C estará disponível no repositório da nossa equipe. Ao acessar o repositório, os interessados poderão analisar de forma mais detalhada cada trecho do código, compreender as implementações específicas de cada função e explorar as estruturas de dados utilizadas.

4. Extra - OAuth2.0

OAuth 2.0 é um protocolo de autorização que permite que aplicativos de terceiros obtenham acesso limitado a recursos em nome de um usuário, sem a necessidade de compartilhar suas credenciais de login. Em outras palavras, o OAuth 2.0 é usado para delegar o acesso a informações protegidas, como dados pessoais ou serviços online, de forma segura e controlada.

Com base nisso, com o objetivo de implementar este protocolo em nosso projeto, primeiro fomos atrás de bibliotecas da linguagem C que nos auxiliassem fornecendo uma API para facilitar a comunicação entre nosso servidor e o servidor externo responsável pelas informações que queríamos acessar. Porém, após algumas pesquisas não encontramos bibliotecas devidamente documentadas para C que assumem este papel.

A partir disso, decidimos utilizar a libcurl para fazer a comunicação necessária com os servidores externos. Então, antes de implementar o código, fizemos uma lista dos passos que deveriam ser realizados para esta implementação:

- a. Configurar as credenciais de OAuth2.0: optamos por utilizar o google cloud para ser nosso provedor de autenticação OAuth2.0 e nele configuramos nossos Client ID e Client Secret.
- b. Integrar a biblioteca ao seu projeto: para isso era necessário que no login usuário as requisições referentes ao OAuth2.0 fossem realizadas.
- c. Implementar o fluxo de autorização: isso envolve redirecionar o usuário para a página de autenticação do provedor OAuth2.0, onde ele será solicitado a fazer login e autorizar o acesso do aplicativo.
- d. Obter o token de acesso: após o usuário autorizar o acesso, o provedor OAuth2.0 redireciona o usuário de volta para o chatroom, junto com um código de autorização.
- e. Implementar a lógica de autenticação: utilizar o token de acesso para autenticar o usuário no chatroom.

Tendo em vista todos estes passos que deveriam ser realizados, configuramos nossas credenciais no google cloud e começamos a implementar as lógicas no código. Porém, por uma questão de falta de tempo

disponível para a equipe trabalhar até a data limite do projeto, não conseguimos fazer com que essa implementação funcionasse por completo. O esboço dela pode ser encontrado na branch OAuth2.0 do repositório.

5. Conclusão

Neste relatório, apresentamos uma solução de chatroom implementada em linguagem C, com o objetivo de fornecer uma plataforma de comunicação eficiente para usuários. Durante o desenvolvimento do projeto, obtivemos resultados satisfatórios, alcançando os principais objetivos propostos pelo roteiro do projeto. No entanto, também identificamos algumas limitações que podem ser aprimoradas em futuras iterações do projeto.

O chatroom foi implementado com sucesso, oferecendo recursos como registro de usuários, criação de salas de chat, envio e recebimento de mensagens, listagem de salas disponíveis e listagem de usuários na sala. Através dos testes realizados, verificamos que o sistema é capaz de lidar com múltiplos usuários simultaneamente, garantindo uma comunicação fluida e em tempo real.

Embora tenhamos alcançado os resultados desejados, reconhecemos que nossa solução possui algumas limitações que podem ser aprimoradas em trabalhos futuros. Alguns pontos que podem ser considerados são: desenvolver uma interface gráfica mais intuitiva e amigável, implementação mecanismos de autenticação/autorização e criptografia, personalização das salas de chat e outros.

Manifestações dos Membros do Grupo

Lucas Pimentel

Durante minha participação no projeto, pude contribuir para o desenvolvimento da aplicação de chatroom, para a elaboração do relatório e dos slides e para as pesquisas e estudos do protocolo OAuth2.0. A compreensão e aplicação das APIs do socket TCP e do comando select foram fundamentais para o sucesso da solução. Através dessas APIs, pude

aprender sobre comunicação em rede, troca de informações e lidar com a entrada e saída de dados em tempo real. A partir disso, acredito que pela minha participação uma nota 9 seria condizente com a minha participação.

Victor Rayan

Com este projeto, pude aprofundar meus conhecimentos sobre a arquitetura TCP/IP. Além disso, consegui aplicar na prática como um socket pode receber múltiplas conexões simultaneamente, sem precisar utilizar threads, através do uso da chamada de sistema `select()`. Isso proporcionou uma abordagem mais eficiente e escalável para lidar com as demandas de múltiplos clientes em uma aplicação de bate-papo virtual baseada na arquitetura TCP/IP.

Foi muito importante o estudo sobre OAuth2, uma tecnologia nova para mim. Vimos a dificuldade que é implementar, mas conseguimos entender a sua importância na segurança.

Toda a equipe se empenhou de maneira semelhante, e conseguimos dividir bem as responsabilidades entre cada um. Portanto, uma nota justa para o meu esforço seria 9 pontos.

Matheus Calixto Vaz Pinheiro

Realizando este projeto de pesquisa tive muito proveito em aprender a como fazer uma aplicação básica de chatroom em tempo real e em como a arquitetura TCP/IP funciona na prática, assim como o uso do `select()` que possibilita escutar vários clientes sem a necessidade de uso de uma thread para cada, fazendo com que a aplicação se tornasse mais escalável. Também aprendi mais sobre o OAuth2.0 e em como poderíamos implementá-lo em nossa aplicação, mesmo que ao final de tudo não tenhamos conseguido realizar sua implementação completa com uma boa usabilidade, foi muito benéfico este estudo e seus resultados. Com base em minha contribuição para o projeto e estudo feito em conjunto acredito que minha nota deveria ser de 9 pontos.

Referências

O que é TCP/IP?. Tecmundo, 29 maio 2012. Disponível em:
https://www.tecmundo.com.br/o-que-e/780-o-que-e-tcp-ip-.htm#google_vignette

select(2). Linux manual page. Disponível em:
<https://man7.org/linux/man-pages/man2/select.2.html>

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