



Chat textual via terminal

Aplicação interativa de serviço de chat conectado e seguro através de um socket TCP

- - Enzo Nunes Sedenho - 13671810
- - Vincenzo D'Arezzo Zilio - 13671790
- - Lucas de Souza Brandão - 13695021
- - Miller Matheus Lima Anacleto Rocha - 13727954

- Sobre a Aplicação

Uma breve descrição do serviço implementado.

- Aplicações de chat textual e seu contexto em Redes de Computadores

O projeto consiste na implementação de um chat que permita a comunicação e troca de mensagens entre processos distintos. Dessa forma, a aplicação aborda diversos tópicos de redes de computadores (os quais serão mais detalhados em seções futuras) visando implementar um chat de mensagens seguro com a utilização do protocolo TCP em uma rede híbrida.

- Comunicação Híbrida TCP

A implementação é pautada na arquitetura **Peer-to-Peer (P2P)** e na **Cliente-Servidor**, em que duas aplicações estabelecem conexão entre si sem uma hierarquização tão clara, como em aplicações Cliente-Servidor Puras. Nesse contexto, a classe **"Server"** será responsável por configurar a interface de socket através da qual

ocorre a comunicação, enquanto a classe "**Client**" conecta-se ao canal a partir de um novo objeto, endereçado ao mesmo destino IP + Porta. Novamente, a aplicação se diferencia da versão pura de Cliente-Servidor pois a aplicação de chat se encontra distribuída entre os dois processos, de modo que ambos implementem dentro de si os mecanismos necessários para garantir o próprio envio de mensagens.

- Interface de chat

O programa deve ser executado em dois terminais distintos e conta com dois modos de iniciação: "Cliente" e "Servidor". O usuário deve selecionar um dos modos para então acessar a interface correspondente.

Tal interface conta com mensagens que exibem o status da conexão ao decorrer da execução do programa, ou seja, informam quando o servidor aguarda uma conexão, quando a conexão é estabelecida e quando a mesma é encerrada. Ao longo da utilização do programa, as mensagens são escritas no próprio terminal, cujo qual mantém um histórico das mensagens enviadas e recebidas por cada usuário.

- Sobre a implementação

A aplicação foi desenvolvida utilizando a linguagem de programação C++ e suas bibliotecas padrões disponibilizadas.

- Estrutura da Aplicação

A aplicação segue o paradigma de Orientação a Objetos, dividindo ambas as partes da comunicação em duas classes: **Server** e **Client**.

Essas classes são invocadas em um programa principal que associa cada uma delas a um terminal.

- Implementação do Socket

Como ressaltado na seção explicativa, a conexão ponto a ponto entre o remetente e o destinatário de cada mensagem é a essência da implementação de uma aplicação de *chat*.

Nesse sentido, parte do desafio é implementar essa conexão utilizando um **protocolo de transporte**, no caso, **TCP**. Esse processo de conexão se encontra encapsulado em um **Socket**, funcionalidade de comunicação oferecida pelo sistema operacional, através da qual pode ser aberto e configurado um canal entre dois entes.

Para a implementação do canal na comunicação, foram utilizadas as bibliotecas:

- **arpa/inet.h**: ferramentas de manipulação de endereço de rede;

- **sys/socket.h**: disponibiliza o serviço de sistema de configuração e manipulação de sockets.

Em cada classe, é inicializado um socket sobre o endereço de *LoopBack* da interface de rede, sobre a porta 3000. Para isso, é necessário configurar o socket para aceitar o compartilhamento de endereços e de portas por mais de um processo ou thread.

- **INADDR_LOOPBACK** : constante que indica o endereço de loopback da máquina;
- **SO_REUSEADDR** e **SO_REUSEPORT** : constante que indica a permissão do compartilhamento de portas e endereços ip pelos sockets.

O fluxo para a validação de uma conexão TCP através do *socket* é representado no código como:

- Cria e configuração do *socket*
- Associação do socket a um endereço e abertura do mesmo
- Testa o fluxo de dados através de uma escuta
- inicializa o processo do *Handshake* - esperando por uma conexão

● Procedimento de Chat: estruturando a conversação

Do ponto de vista do terminal onde é implementada a configuração do canal (Servidor):

- lê mensagem do socket ;
- se houver mensagem, imprimi-la na saída padrão;
- lê mensagem da entrada padrão ;
- se houver mensagem, enviá-la ao destinatário ;

Do ponto de vista do terminal onde se conecta ao canal (Cliente):

- lê mensagem da entrada padrão ;
- se houver mensagem, enviá-la ao destinatário ;
- lê mensagem do socket ;
- se houver mensagem, imprimi-la na saída padrão;

● Infraestrutura de execução

O processo de compilação, *linkagem* e execução da aplicação é regido pela ferramenta **CMake**, possibilitando a automatização e exportabilidade de todo o processo.

Informações sobre as diretivas de compilação podem ser encontradas dentro do arquivo **CMake.txt** e instruções sobre como realizar esse processo encontram-se no arquivo **README.md**

- Sobre a utilização

Os comandos de execução assim como as instruções de utilização se encontram no arquivo **README.md**.

- Executando o chat

Para executar a interface de chat, será necessário abrir dois terminais e, em cada um deles, executar o programa.

- No primeiro terminal iniciado, deve ser escolhida a opção de servidor;
- No segundo terminal, deve ser escolhida a opção de cliente;
- Após isso, é aberta a interface de chat, em que cada terminal tem uma oportunidade de mandar uma mensagem.

- Finalizando o chat

Para finalizar o chat, basta enviar "**! [exit]**" pelo canal, a partir de qualquer um dos terminais. Esse comando é necessário para finalizar o canal corretamente.

- Sobre os resultados

- Chat em execução:

```
lucas@lucas-M1502IA:~/Documents/2024.1/redes/chat_redes/build-dir$ ./chat_redes
Escolha o modo de operação:
1. Servidor
2. Cliente
2
Conexão estabelecida com sucesso!
[Eu]: 
```

```
lucas@lucas-M1502IA:~/Documents/2024.1/redes/chat_redes/build-dir$ ./chat_redes
Escolha o modo de operação:
1. Servidor
2. Cliente
1
Iniciando Comunicação no endereço: esperando conexão na porta [3000]
CONEXÃO ESTABELECIDA - Iniciando Chat :D !
```

```
lucas@lucas-M1502IA:~/Documents/2024.1/redes/chat_redes/build-dir$ ./chat_redes
Escolha o modo de operação:
1. Servidor
2. Cliente
2
Conexão estabelecida com sucesso!
[Eu]: ola
[Correspondente]: oi
[Eu]: ![exit]
ENCERRANDO CONEXÃO CLIENTE
lucas@lucas-M1502IA:~/Documents/2024.1/redes/chat_redes/build-dir$
```

```
lucas@lucas-M1502IA:~/Documents/2024.1/redes/chat_redes/build-dir$ ./chat_redes
Escolha o modo de operação:
1. Servidor
2. Cliente
1
Iniciando Comunicação no endereço: esperando conexão na porta [3000]
CONEXÃO ESTABELECIDA - Iniciando Chat :D !

[Correspondente]: ola
[Eu]: oi
[Correspondente]: ![exit]
Conexão encerrada pelo cliente.
ENCERRANDO CONEXÃO SERVIDOR
lucas@lucas-M1502IA:~/Documents/2024.1/redes/chat_redes/build-dir$
```