

Trabalho final de Transmissão de dados Desenvolvimento de um sistema de chat online

Abdullah Zaiter - 15/0089392

Departamento de ciência de computação, Universidade de Brasília, Brasília

Repositorio git de desenvolvimento

Video do funcionamento

Abstract— This document shows the step-by-step, results and comments about the final project of Data Transmission 2-2018.

Keywords— Servers, Data transmission, chat, multi-threading.

Resumo— Este documento apresenta os passo a passo, explicação e resultados do trabalho final da disciplina transmissão de dados 2-2018

Palavras chave— Servidores, transmissão de dados, bate papo, programação paralela.

I. OBJETIVOS

Objetivo desse trabalho é fortalecer os conceitos aprendidos na disciplina como comunicação por protocolo TCP, Sockets e transmissão de dados no geral desenvolvendo um sistema de chat que suporte a criação de salas de conversa e envio, recebimento de dados simultâneo de todos os usuários. Assim como ter ideia melhor do funcionamento dos aplicativos de bate papo e métodos de segurança implementando salas privadas.

II. MÓDULOS UTILIZADOS

Utilizou-se a linguagem de programação Python versão 2.7 pois a mesma possui muitos módulos prontos que possam facilitar a implementação do desafio a resolver.

A. Módulos prontos:

- **socket**[1]: implementação das funções básicas de comunicação por socket que lidam com o sistema operacional para realizar as trocas de dados desejadas.
- **threading**[2]: para resolver o problema de atender múltiplos clientes ao mesmo tempo para envio e recebimento, fazendo assim um Thread novo para atender as aquisições de cada cliente simultaneamente.
- **logging**[3]: facilita a geração de logs em um determinado código, usada para gerar o log do servidor
- **os.path**[4]: funções básicas de diretório no Linux, foi usada para checar se algum arquivo existe na pasta atual ou não.

B. Módulos criados:

- **chatroom**: Contem uma classe que representa a sala de bate papo com os métodos relacionados a ela e algumas funções de uso comum entre todas as salas..
- **user**: Contem uma classe que representa o usuário incluindo todas as informações necessárias.
- **commonfuncts**: conjunto de funções uteis utilizadas em todos os módulos.

O código está anexado junto com este relatório, encontram-se no mesmo comentários explicando todos as funções e métodos utilizados.

III. MAQUINAS DE ESTADOS IMPLEMENTADAS

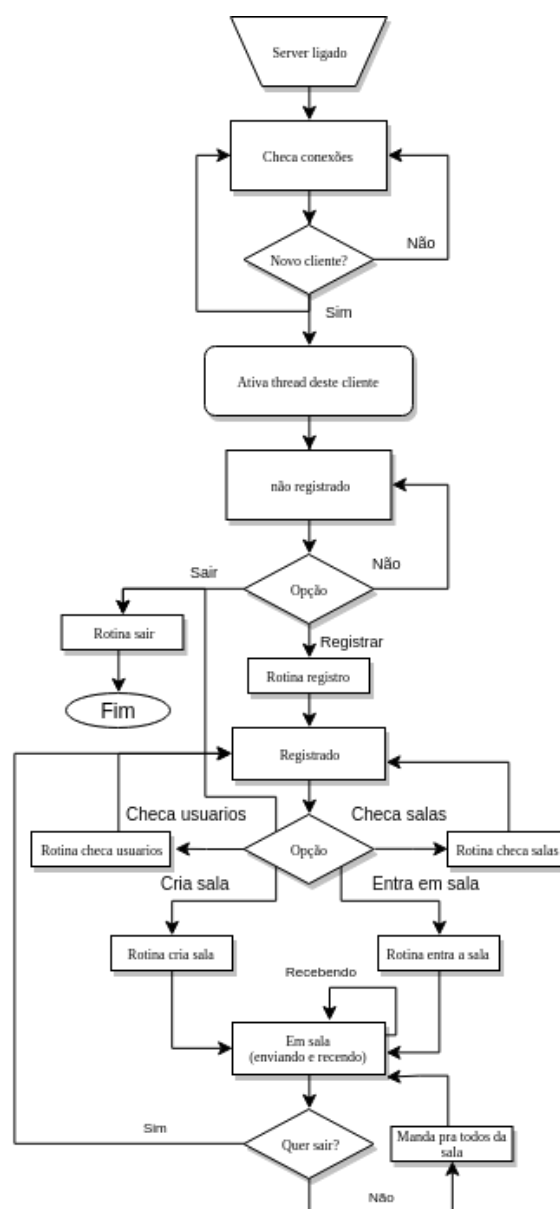


Fig. 1. Máquina de estados do servidor

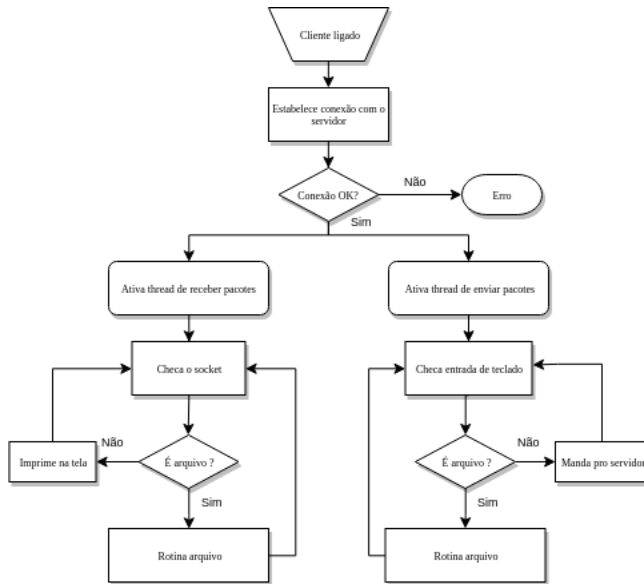


Fig. 2. Máquina de estados do cliente

IV. DIFICULDADES ENCONTRADAS

- 1- sincronização entre threads que atuam em cima de memórias compartilhadas, foi resolvido eliminando os *Deadlocks* (partes do código que possam travar ele por um tempo).
- 2- o código e o problema estavam ficando muito complexos até o momento que começou-se a utilizar os conceitos de orientação a objetos de uma maneira que tornou o código das salas extremamente mais simples.
- 3- no inicial, se o código do servidor fosse interrompido, teria que fechar o terminal e limpar as conexões para poder utilizar a mesma porta do código interrompido pois ela continuava aberta após a interrupção do código, isso foi resolvendo capturando a interrupção por teclado em todos os loops e fechando a comunicação socket antes de fechar o código.

V. CONCLUSÕES

Foi possível realizar o que foi pedido na especificação no trabalho, concluiu-se que sistemas que demandam processamento em paralelo se tornam mais complexos e eles estão muito presentes na nossa vida do dia a dia como um simples aplicativo de bate papo no celular. Também obteve-se um entendimento melhor do funcionamento dos processos em cima da camada de aplicações e como a mesma facilita a comunicação entre os processos, isso era o esperado considerando os conceitos aprendidos na matéria. Um link do funcionamento do trabalho encontra-se clicável a baixo do título deste documento.

REFERENCIAS

- [1] Documentação do módulo socket de Python<<https://docs.python.org/3/library/socket.html>>. Acesso em: 23 nov. 2018.
- [2] Documentação do módulo threading de Python<<https://docs.python.org/2/library/threading.html>>. Acesso em: 23 nov. 2018.

- [3] Documentação do módulo logging de Python<<https://docs.python.org/2/library/logging.html>>. Acesso em: 23 nov. 2018.
- [4] Documentação do módulo socket de Python<<https://docs.python.org/2/library/os.html>>. Acesso em: 23 nov. 2018.