

Universidade Federal de Alagoas
Instituto de Computação
Redes de Computadores
2020.1 – PLE

Alunos: João Vitor Santos Tavares
Rick Martim Lino dos Santos

Relatório

- **Descrição**

A aplicação implementada nesse projeto foi uma sala de chat tomando como base a arquitetura cliente – servidor sobre o protocolo de transmissão TCP, fundamentando-se ligeiramente no modelo de aplicações como o Telegram e WhatsApp, que se utilizam do protocolo XMPP.

Nesta aplicação não utilizamos o protocolo XMPP, mas usamos um modelo de comunicação JSON para enviar um dicionário Python do cliente para o servidor e vice-versa.

- **Desenvolvimento**

- 1. Utilização do JSON**

Estamos implementando basicamente uma sala de chat, assim sendo, precisamos enviar informações para outros clientes, entre elas, temos o nome do destinatário e sua mensagem, que são as que usamos no código, contudo também poderíamos ter o horário de envio, data, entre outras informações.

Para impedir o envio de várias mensagens e proporcionar mais tráfego desnecessário na rede enviamos um JSON, cujo formato e tipo de arquivo

permite converter um objeto em Python em uma estrutura simples e de fácil interpretação.

2. Threading

O encadeamento nesta aplicação é essencial, pois permite que o servidor continue atendendo novas conexões enquanto mantém as antigas ativas, sem a necessidade de renova-las, ou seja, ele mantém múltiplas conexões.

Há uma função no código para aceitar as novas conexões dentro de um loop principal e sempre que uma nova conexão é gerada a `start_new_thread` é a responsável por iniciar uma thread a partir de uma função que se responsabilizará por tratar as solicitações do cliente.

No cliente usamos thread para esperar por novas mensagens sem a necessidade de usar o loop principal do programa.

Por fim, implementamos um sistema de log de erros e histórico de mensagens, que consiste em escrever quais erros ou mensagens o cliente / servidor envia e/ou recebe e as exceções que podem vir a ocorrer durante a sua execução.

• Dificuldades encontradas

- Como identificar os destinatários da mensagem sem gerar muito tráfego na rede. Solucionado com JSON.
- Como conseguir realizar conexões simultâneas sem parar de aceitar novas conexões. Solucionado com uso de Thread.
- Envio de mensagens para todos os clientes na ordem em que chegam. Solucionado com uma função simples de “broadcasting”.
- Como fazer o cliente enviar mensagens, mas recebe-las a qualquer momento. Solucionado também com o uso de Thread.
- Servidor em standby por muito tempo. Solucionado com uso da biblioteca Time.