**Trabalho de Redes sobre Comunicação P2P e P2\***

**Rebeca Teófilo Siqueira e Natália Sales Aragão**

Ciência da Computação – Universidade Estadual do Ceará (UECE)  
Fortaleza – CE – Brasil

Disciplina de Redes de Computadores – UECE – Professor Rafael Lopes Gomes

***Resumo.*** *Este trabalho descreve a implementação de um programa que atua como um mensageiro de texto (estilo Whatsapp) que troca mensagens usando Sockets, onde as funcionalidades de comunicação entre 2 processos e de um processo para vários devem ser consideradas.*

**1. Introdução**

Quando falamos sobre comunicação entre processos, estamos nos referindo a como dois programas independentes podem trocar informações e interagir um com o outro. O TCP (Transmission Control Protocol) desempenha um papel crucial na comunicação entre processos em redes de computadores, pois é um protocolo da camada de transporte que fornece uma comunicação confiável e orientada à conexão entre processos em diferentes máquinas em uma rede.

Para possibilitar essa comunicação, usamos uma técnica chamada "sockets". Um socket é basicamente uma interface de comunicação que permite que processos se conectem e enviem mensagens um para o outro através de uma rede ou mesmo em um mesmo computador.

Ao usar sockets, um dos processos assume o papel de "servidor", enquanto o outro é o de "cliente". O servidor fica "ouvido" em uma porta específica do computador, aguardando a chegada de conexões dos clientes. Quando um cliente deseja se comunicar com o servidor, ele estabelece uma conexão por meio de um socket.

Uma vez estabelecida a conexão, os dois processos podem trocar mensagens entre si. Eles podem enviar e receber dados na forma de caracteres de texto, o que permite a comunicação por meio de mensagens escritas, assim como ocorre em aplicativos de mensagens como o WhatsApp.

Essa comunicação bidirecional permite que os processos troquem informações em tempo real, sejam elas mensagens ou arquivos. Ambos os processos podem enviar mensagens a qualquer momento, e as mensagens são entregues ao destinatário através da conexão estabelecida.

A comunicação entre dois processos usando sockets é amplamente utilizada em aplicações distribuídas, redes de computadores e sistemas cliente/servidor. É uma forma flexível e escalável de permitir a troca de informações entre programas independentes, abrindo caminho para a construção de diversos tipos de aplicativos e serviços que requerem interação entre processos separados.

Em algumas situações, é necessário que um processo envie informações para vários processos simultaneamente. Isso ocorre quando desejamos transmitir uma mensagem para vários destinatários, como enviar uma notificação para vários usuários em um sistema distribuído.

Existem várias abordagens para realizar a comunicação de um processo para vários. A que usamos neste trabalho foi Filas de mensagens (Message Queues). Nessa abordagem, um processo envia mensagens para uma fila de mensagens compartilhadas. Vários processos podem estar aguardando na fila para receber as mensagens. O processo remetente coloca as mensagens na fila, e os processos destinatários as retiram da fila, processando-as uma por vez.

**2. Bibliotecas**

A biblioteca socket do Python é amplamente utilizada para implementar a comunicação entre processos em redes. Ela fornece uma interface de programação para a criação e manipulação de sockets, permitindo a troca de dados entre diferentes processos, seja em uma mesma máquina ou em máquinas distintas em uma rede.

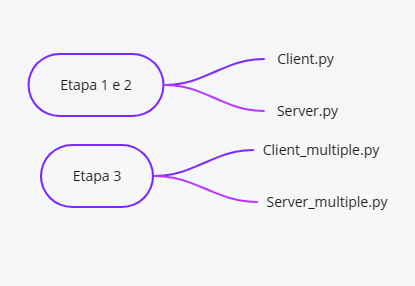
A biblioteca socket do Python suporta os principais tipos de sockets, como sockets TCP/IP e UDP. Isso oferece flexibilidade para implementar diferentes protocolos de comunicação, de acordo com as necessidades do aplicativo.

Nesse contexto, a biblioteca socket tem como funcionalidades importantes a criação de sockets, a conexão, o envio e recebimento de dados, além de encerrar a conexão e tratar erros. Com o conhecimento da biblioteca socket, pode-se implementar facilmente a comunicação entre processos em Python, seja para criar um servidor de chat, compartilhar dados entre aplicativos ou desenvolver sistemas distribuídos complexos.

A biblioteca os do Python é principalmente utilizada para funcionalidades relacionadas ao sistema operacional, como manipulação de arquivos, diretórios, processos e variáveis de ambiente.

Nesse contexto, a biblioteca os possui algumas funcionalidades que podem ser relevantes em um contexto de comunicação entre processos incluem a criação de processos, identificação de processos e o controle de processos.

**3. Estruturação do Código**



**4. Execução do Programa**

Etapas para executar os códigos:

Para executar a funcionalidade 1 do projeto (enviar mensagem de texto):

1) executar no pycharm com o comando do terminal: py server.py

2) executar no pycharm com o comando do terminal: py client.py

3) enviar mensagem do terminal do client

4) enviar mensagem do terminal do server

Para executar a funcionalidade 2 do projeto (trocar arquivos):

1) executar no pycharm com o comando do terminal: py server.py

2) executar no pycharm com o comando do terminal: py client.py

3) enviar o seguinte comando: /send\_file <nome\_do\_arquivo> (Ex: /send\_file oi.txt)

4) o código irá receber e ler o arquivo.

Para executar a funcionalidade 3 do projeto (trocar arquivos):

1) executar no pycharm com o comando do terminal: py server\_multiple.py

2) executar no pycharm com o comando do terminal: py client\_multiple.py) enviar mensagem do client e ela será recebida pelo server, o client receberá um aviso de que a mensagem foi recebida

3) Para enviar de outros clients, basta fechar o terminal, criar outro, e executar o comando 2 novamente, e o server irá receber como se fosse um novo client (uma nova conexão).

**7. References**

Ross, Keith W., and Kurose, James F.. Computer Networking: A Top-down Approach. Alemanha, Pearson, 2013.

CODIFIKE. Como fazer um chat usando Python, <https://youtu.be/PjiDem422OI>. Youtube, 2021.

Felipe Dasr. Como enviar e receber arquivos com socket em Python, <https://youtu.be/j4Drn47pc3o>. Youtube, 2021.

Felipe Dasr. Múltiplas conexões com socket em Python, <https://youtu.be/QyJBrS1c1s8>. Youtube, 2022.