Documentação do Protocolo de Comunicação Cliente-Servidor

Introdução

Este documento detalha o protocolo de comunicação estabelecido entre um cliente e um servidor por meio de sockets em Python, visando a transferência de comandos e arquivos. O projeto tem como objetivo criar uma comunicação eficiente via TCP/IPv4, permitindo que o cliente realize operações no servidor, tais como listar arquivos, remover, fazer upload e download.

Comandos Suportados

1. Listagem de Arquivos

Cliente: 1 lista_de_arquivos
Servidor: Resposta com a lista de arquivos do cliente ou do servidor

Foram implementadas duas funcionalidades de listagem de arquivos: uma para o cliente listar seus próprios arquivos e outra para o cliente visualizar os arquivos presentes no servidor. A distinção facilita a compreensão e teste das funções de download e upload.

2. Remoção de Arquivos

Cliente: 2 nome_do_arquivo
Servidor: Resposta indicando se a remoção foi bem-sucedida

O cliente pode solicitar a remoção de um arquivo no servidor, enviando o comando "2 nome_do_arquivo". O servidor responde informando se a operação foi realizada com sucesso ou se ocorreu algum erro.

3. Upload de Arquivos

Cliente: 3 nome_do_arquivo
Servidor: Resposta indicando se o arquivo foi recebido com sucesso

O cliente inicia o upload de um arquivo para o servidor enviando o comando "3 nome_do_arquivo". O servidor responde informando se o arquivo foi recebido com sucesso ou se houve algum erro durante a transferência.

4. Download de Arquivos

Cliente: 4 nome_do_arquivo
Servidor: Envia o arquivo para o cliente

O cliente solicita o download de um arquivo do servidor enviando o comando "4 nome_do_arquivo". O servidor envia o arquivo para o cliente, que o salva em seu diretório local.

Troca de Mensagens

A comunicação entre cliente e servidor utiliza mensagens codificadas em UTF-8. Os comandos são transmitidos como strings, enquanto o conteúdo dos arquivos é enviado como bytes. Ao iniciar o servidor, ele aguarda a conexão do cliente e, em seguida, espera por mensagens. Quando uma mensagem é recebida, ela é decodificada e dividida em uma lista de strings usando o método .split(). A análise das strings permite a execução da função correspondente, seguida pela codificação e envio da resposta ao cliente.

Validação e Testes

Para validar o algoritmo, foram realizados testes extensivos de cada função no cliente. Modificações foram feitas conforme as respostas obtidas, garantindo que as operações fossem satisfatórias. Os testes foram executados no terminal do cliente, analisando os resultados e ajustando o código conforme necessário. Essa abordagem permitiu um refinamento contínuo do código para garantir seu funcionamento correto.

Este documento fornece uma visão geral do protocolo de comunicação, suas funcionalidades e a lógica por trás das operações cliente-servidor, facilitando a compreensão e manutenção do sistema.