UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS

RELATÓRIO DE DESENVOLVIMENTO DO CLIENTE-SERVIDOR

Eliziane Silva de Lima - 403940

Lara Roque Sartori - 514839

Natam Leão Ferreira - 516205

SUMÁRIO

1. INTRODUÇÃO

Neste relatório, será compartilhada a experiência no desenvolvimento de um servidor e cliente em Python. O objetivo era criar um servidor capaz de lidar com múltiplos clientes, registrar logs, permitir a transferência de arquivos e desenvolver um cliente que pudesse se comunicar com o servidor, enviar solicitações e receber respostas. Embora tenham surgido desafios ao longo do caminho, a conclusão é motivo de satisfação.

2. DESAFIOS ENFRENTADOS

2.1 Desenvolvimento do Servidor

Estrutura Inicial: o primeiro passo envolveu a definição da estrutura inicial do servidor. Embora houvesse experiência em programação Python, criar um servidor era uma tarefa nova. A pesquisa inicial sobre como criar sockets e configurar um servidor foi fundamental.

Transferência de Arquivos: a principal dificuldade foi encontrada na função de transferência de arquivos. Problemas relacionados ao tamanho do buffer usado para enviar dados surgiram. A primeira versão do código enfrentou problemas de desempenho ao lidar com arquivos maiores. Para melhorar isso, o tamanho do buffer foi ajustado.

Funções de Transferência e Recebimento de Arquivos: a criação das funções de transferência e recebimento de arquivos foi desafiadora. Foram necessárias várias iterações para garantir que os arquivos fossem transmitidos com sucesso entre o servidor e o cliente.

Estruturação das Trocas de Dados: a estruturação das trocas de dados entre o cliente e o servidor também foi um ponto crítico. Garantir que as solicitações e respostas fossem interpretadas corretamente exigiu um planejamento cuidadoso.

Tratamento de Erros: garantir que o servidor fosse robusto foi um desafio constante. O tratamento de erros foi implementado em várias partes do código para lidar com exceções, como conexões perdidas e erros na transferência de dados. Isso exigiu uma abordagem cuidadosa para evitar travamentos ou falhas inesperadas.

Checagens no Servidor: no servidor, foram implementadas checagens para garantir a integridade e a segurança das operações. Isso incluiu a verificação de solicitações válidas dos clientes, bem como o controle de erros durante a transferência de arquivos.

2.2 Desenvolvimento do Cliente

Conexão Inicial: o início do desenvolvimento envolveu a definição da lógica para a conexão inicial do cliente com o servidor. Entender como criar um socket e estabelecer uma conexão com o servidor foi um desafio inicial superado com pesquisa e prática.

Envio de Solicitações: implementar a lógica para enviar solicitações ao servidor foi uma parte fundamental do cliente. Foi necessário garantir que as solicitações fossem enviadas corretamente e que o cliente pudesse lidar com diferentes tipos de respostas.

Recebimento de Arquivos: uma das partes mais desafiadoras foi a implementação do recebimento de arquivos do servidor. Isso envolveu a lógica para receber o tamanho do arquivo e os dados do arquivo em si. Houve dificuldades iniciais relacionadas ao tamanho do buffer usado para receber os dados.

Checagens no Cliente: no cliente, as checagens incluíram a validação de respostas do servidor para garantir que as operações fossem executadas com sucesso e a verificação cuidadosa dos dados recebidos durante o recebimento de arquivos.

3. SOLUÇÕES ADOTADAS

3.1 Desenvolvimento do Servidor

Ajuste do Buffer: para resolver o problema da transferência de arquivos, o tamanho do buffer usado na comunicação entre o servidor e os clientes foi ajustado. Isso resultou em uma melhoria significativa no desempenho, permitindo a transferência suave de arquivos grandes.

Registros de Logs: foi implementado um sistema de registro de logs para rastrear as interações dos clientes com o servidor. Isso não apenas fornece informações úteis para depuração, mas também auxilia na monitorização da atividade do servidor.

Tratamento de Erros: garantiu-se que o código do servidor estivesse preparado para lidar com diversas situações de erro. Isso incluiu o tratamento de exceções ao enviar dados e fechar conexões, garantindo que o servidor continuasse funcionando mesmo quando ocorriam problemas.

Checagens no Servidor: no servidor, as checagens incluíram a validação de solicitações dos clientes para evitar operações não autorizadas e a verificação cuidadosa dos dados recebidos durante a transferência de arquivos.

3.2 Desenvolvimento do Cliente

Lógica de Conexão: para lidar com a conexão inicial, foram seguidas as diretrizes de criação de sockets e estabelecimento de conexões. Foi implementado tratamento de erros para lidar com problemas de conexão, garantindo uma conexão estável com o servidor.

Envio de Solicitações: foram criadas funções para enviar solicitações ao servidor, garantindo que os dados fossem codificados corretamente antes de serem enviados. Isso possibilitou uma comunicação eficaz entre o cliente e o servidor.

Recebimento de Arquivos: para superar as dificuldades relacionadas ao tamanho do buffer ao receber arquivos, o tamanho do buffer foi ajustado para garantir uma transferência de arquivos mais eficiente. Também foi implementado tratamento de erros para lidar com problemas durante o recebimento de arquivos.

Checagens no Cliente: no cliente, as checagens incluíram a validação de respostas do servidor para garantir que as operações fossem executadas com sucesso e a verificação cuidadosa dos dados recebidos durante o recebimento de arquivos.

4. CONCLUSÃO

O desenvolvimento deste servidor e cliente foi uma experiência desafiadora, porém gratificante. Aprendeu-se a lidar com a comunicação de rede, o tratamento de erros, a concorrência e a estruturação das trocas de dados. Isso proporcionou uma compreensão mais profunda do desenvolvimento de software. Embora tenham ocorrido dificuldades ao longo do caminho, o resultado é motivo de satisfação.