

Trabalho Prático 1 - Documentação do protocolo de transferência de arquivos

Nome : João Pedro Miranda de Faria

Matrícula : 20180722555

Introdução

Neste trabalho prático, temos como objetivo implementar um sistema de transferência de arquivos simples entre um servidor e um cliente utilizando sockets em C. Nosso objetivo é criar uma aplicação que permita a troca de arquivos entre essas duas entidades.

O servidor será capaz de aceitar conexões de vários clientes e receber o conteúdo enviado por essas conexões. Ele irá transformar esses dados recebidos em arquivos e armazená-los no local onde a aplicação está sendo executada. Por sua vez, o cliente terá a capacidade de se conectar ao servidor, selecionar um arquivo local para enviar e receber uma confirmação do servidor de que o arquivo foi recebido com sucesso. Tanto o servidor quanto o cliente terão a opção de encerrar a conexão quando necessário, permitindo uma comunicação flexível e controlada.

Para alcançar esse objetivo, foram desenvolvidas duas aplicações distintas: uma para o lado do servidor e outra para o lado do cliente. Cada uma dessas aplicações possui funcionalidades específicas para realizar a comunicação e a troca de arquivos. Para organizar e agrupar as funcionalidades necessárias para ambas as aplicações, foram criados arquivos auxiliares responsáveis pela parte de conexão via sockets e pela parte de manipulação de arquivos.

Desenvolvimento

- Interface com o usuário

Foi desenvolvida a interface com o usuário conforme proposto pelo exercício, onde o usuário deverá digitar os comandos disponíveis para conseguir fazer os envio dos arquivos. O comando digitado no terminal será transformado em uma ação que determinará a função que será executada no sistema. Vale destacar que, visando melhorar a interface de uso com o usuário final, ao ser inserido um comando desconhecido, será informado que o comando não é válido e nenhuma informação será enviada para o servidor.

- Seleção e tratamento dos arquivos

Para lidar com os arquivos, foi implementado dentro do “comom.c”, todas as funções necessárias para a transferência de arquivos em ambas as aplicações. Utilizamos a interface da própria stdlib para criar funções que irão realizar a leitura do conteúdo do arquivo e salvar o conteúdo recebido pelo servidor em seu local de execução.

Vale destacar que do lado do cliente, após a seleção do arquivo, salvamos a mensagem já formatada para o envio em uma variável global, e a utilizamos quando o cliente insere o comando para envio deste conteúdo. Caso o cliente troque de arquivo, antes de enviar, a variável global é atualizada com o novo conteúdo

- Comunicação entre aplicações

O protocolo de comunicação implementado neste trabalho foi baseado no vídeo de referência, que utilizou funções de comunicação entre servidor e cliente para a troca de mensagens. Essa abordagem foi escolhida por proporcionar suporte tanto para IPV4 quanto para IPV6, garantindo uma comunicação abrangente.

No servidor, adotou-se um modelo baseado em threads, permitindo que múltiplos clientes pudessem ser atendidos simultaneamente. A estrutura foi projetada para iniciar um socket e aguardar conexões na porta especificada. Assim, o servidor fica em um estado de espera, aguardando novas conexões através da função accept. Quando uma conexão é estabelecida, uma nova thread é aberta exclusivamente para lidar com aquele cliente específico, permitindo um atendimento eficiente e independente para cada conexão.

No cliente, ao ser iniciado, ele estabelece uma conexão com o servidor e aguarda o comando do usuário para enviar um arquivo. Após o envio, o cliente espera pela resposta do servidor e exibe a resposta na tela. Caso o cliente envie o comando para encerrar a conexão, é enviada uma mensagem de "kill" para o servidor, encerrando a aplicação de forma adequada.

Essa implementação demonstra uma estrutura bem organizada e funcional, possibilitando uma troca de arquivos entre servidor e cliente. A utilização de threads no servidor permite um atendimento simultâneo de múltiplos clientes. Já no cliente, as interações são realizadas de forma intuitiva, facilitando o envio de arquivos e o recebimento de respostas do servidor.

Desafios

Durante o desenvolvimento das aplicações, alguns desafios e considerações se mostraram relevantes para a construção do projeto. Entre eles, destacam-se os seguintes pontos:

- A linguagem C em um nível relativamente baixo trouxe complexidade adicional em determinados aspectos da aplicação, principalmente relacionados a estruturas de ponteiros e alocação de memória. Esses conceitos requerem um entendimento mais aprofundado e cuidadoso para garantir a correta manipulação dos dados e evitar vazamentos de memória.
- Foi identificado um problema em relação ao uso da função printf para exibir informações no console em tempo de execução do lado do servidor. A saída das informações só era apresentada após a conexão ser encerrada. Para contornar essa questão, optou-se por utilizar a função puts(), que permitiu exibir as informações no console em tempo real durante a execução.
- Complexidade de lidar com strings na linguagem C, o que exigiu a criação de funções específicas para o tratamento de arquivos, envio e leitura de mensagens. Diferentemente de outras linguagens de programação, que possuem recursos

nativos mais simplificados para manipulação de strings, foi necessário elaborar abordagens personalizadas para garantir o correto manuseio dos dados.

Resultado Final

As aplicações desenvolvidas conseguiram com sucesso atender aos requisitos do projeto. Para a verificação do seu comportamento, foram realizados testes, que tiveram como base os comandos de exemplo na descrição do trabalho. O resultado do funcionamento do protocolo de comunicação para transferência de arquivos obteve resultados positivos.

Conclusão

O trabalho em si foi extremamente desafiador, e considero que tive uma aprendizagem prática valiosa dos conceitos que estudamos em sala de aula. Um aspecto particularmente interessante foi retomar o desenvolvimento na linguagem C, pois ela possui peculiaridades que exigem uma atenção especial durante o seu uso. Através desse desafio, pude aprimorar minha compreensão dessas peculiaridades e desenvolver habilidades mais sólidas na programação em C. Superar essas dificuldades foi fundamental para alcançar um projeto funcional e capaz de realizar a transferência de arquivos entre servidor e cliente de forma adequada.