



**UNIVERSIDADE FEDERAL DO PARANÁ**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DISCIPLINA DE REDES 1 2022.2**

**TRABALHO 1: MENSAGEIRO LOCAL**

**Aluno:** Ricardo Farias - GRR20196525

**Professor:** Eduardo Todt

**Curitiba**  
**2023**

Estou enviando 2 arquivos, o **projeto-redes-1.tar.gz** e o **tentativa\_de\_arrumar.tar.gz**. O **projeto-redes-1.tar.gz** é o que foi apresentado no laboratório, foi usado por ser mais estável e executar corretamente o que deveria ser feito, embora possua algumas falhas no cenário real, que seria um computador ligado por cabo de rede. Em localhost, através da interface loopback tudo funciona perfeitamente. Já o **tentativa\_de\_arrumar.tar.gz** foi o que eu disse que estava tentando arrumar, esse está mais completo, no entanto, instável, mas possui a implementação das **janelas deslizantes** e verificação de erros através de **checksum**. Novamente o **checksum** foi utilizado por ser mais simples e mais rápido de implementar, agilizando a entrega do projeto.

### **Arquivo: client.c**

Implementa um cliente de rede para um sistema de bate-papo (chat) e transferência de arquivos. O código inclui bibliotecas de socket e de manipulação de arquivos e define constantes e estruturas que são usadas ao longo do programa.

O código cria uma conexão com um servidor de rede usando a função socket e a função connect. A partir disso, o programa apresenta um menu interativo para o usuário escolher entre as opções de chat de mensagens ou envio de arquivos.

Se o usuário escolhe a opção de chat, o programa envia um código ao servidor para notificá-lo e aguarda uma resposta para iniciar o chat. Durante o chat, o programa obtém a hora do sistema e permite que o usuário envie mensagens de texto ao servidor, que são exibidas na tela e transmitidas ao servidor. O usuário pode encerrar o chat digitando 'q'. O sistema de chat simula a técnica **para-e-espera**.

Se o usuário escolhe a opção de envio de arquivos, o programa apresenta um prompt para o usuário inserir o caminho do arquivo que deseja enviar. Em seguida, o programa abre o arquivo e envia seu conteúdo para o servidor. O programa também informa o servidor sobre o nome do arquivo e seu tamanho.

O programa usa diversas funções auxiliares definidas em um arquivo de cabeçalho rickwebtools.h, que inclui funções para tratar entrada do usuário, limpar a tela, comparar strings e obter a hora atual do sistema.

### **Arquivo: server.c**

Implementa um servidor que permite ao usuário escolher entre as opções de chat e recebimento de arquivos. Ele aguarda conexões de clientes e, após a conexão ser estabelecida, o servidor solicita ao cliente que insira seu nome. Em seguida, exibe um menu com as opções de chat e recebimento de arquivos.

Se o usuário selecionar a opção de chat, o servidor aguarda uma solicitação do cliente para iniciar a comunicação. Caso receba a solicitação, o servidor pergunta ao usuário se deseja aceitar a solicitação de chat ou não. Se o usuário aceitar, o servidor envia uma mensagem para o cliente informando que a solicitação foi aceita. Caso contrário, envia uma mensagem para o cliente informando que a solicitação foi negada.

Se o usuário selecionar a opção de recebimento de arquivos, o servidor aguarda que o cliente inicie a transferência do arquivo. O servidor então recebe o arquivo enviado pelo cliente e armazena em um buffer de tamanho máximo definido pela constante MAX\_BYTES.