

Relatório Projeto Redes de Computadores A

Projeto: Chat

**Luiz Vinícius dos Santos Ruoso RA: 18233486,
Victor Felipe dos Santos RA: 18117820,
Victor Luiz Fraga Soldera RA: 18045674.**

1

1. Introdução

Neste projeto foi proposto que fizéssemos um chat na linguagem de programação C. Desta forma, teve-se de se utilizar conceitos referentes a sistemas operacionais, como comunicação entre processo, dentro deste conceito, utilizou-se de sockets TCP e threads para realizar-se a comunicação entre o cliente e o servidor.

2. Desenvolvimento

2.1. Ambiente utilizado

Para a realização deste projeto utilizou-se o Subsistema Windows para Linux (WSL) que possibilitou a utilização de diversas funções do compilador GCC no sistema Windows, como por exemplo a flag `-pthread`, que adiciona suporte ao multithreading utilizando a biblioteca de threads do sistema POSIX. Para o desenvolvimento do código houve a utilização da IDE (Ambiente de Desenvolvimento Integrado) Visual Studio Code.

2.2. Funcionamento

Para o desenvolvimento, escolheu-se o protocolo de comunicação TCP por realizar o handshaking. Ou seja, espera-se pela mensagem, e, se ela não chegar, realiza-se o envio novamente. Desta forma, há a garantia de que a mensagem chegará ao destinatário sem percas durante o envio. Além disso, escolheu-se a comunicação entre cliente e servidor, porque desta há o controle de envios e recebimento de mensagens e arquivos, de forma que, sem o servidor, o usuário não possui acesso ou controle a estes comandos.

Para o envio de mensagens, decidiu-se o envio de structs, por conterem diversos tipos e facilitarem o acesso ao conteúdo desejado. Todas as structs utilizadas são explicitadas abaixo:

```
struct {  
    char content[1024];  
    int nBytes;  
}  
  
typedef charContentOps;
```

```

struct {
    int operation;
    charContentOps username;
    charContentOps message;
    int myPortNumber;
    charContentOps userDestiny;

} typedef msg;

struct {
    int statusCode;
    int operation;
    msg payload;

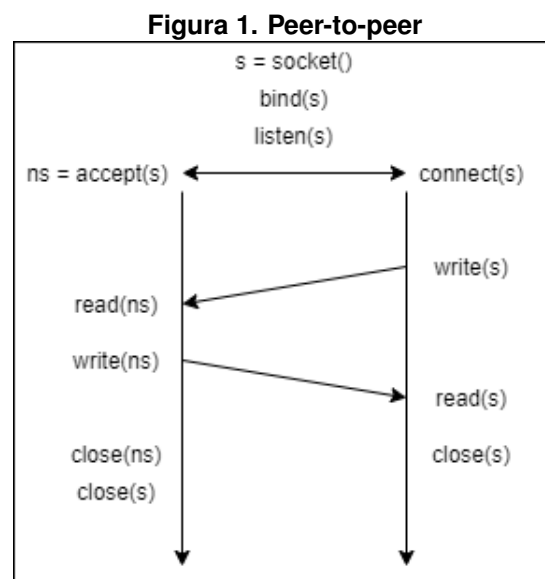
} typedef serverResponse;

struct {
    int * socket;
    char username[1024];

}typedef user;

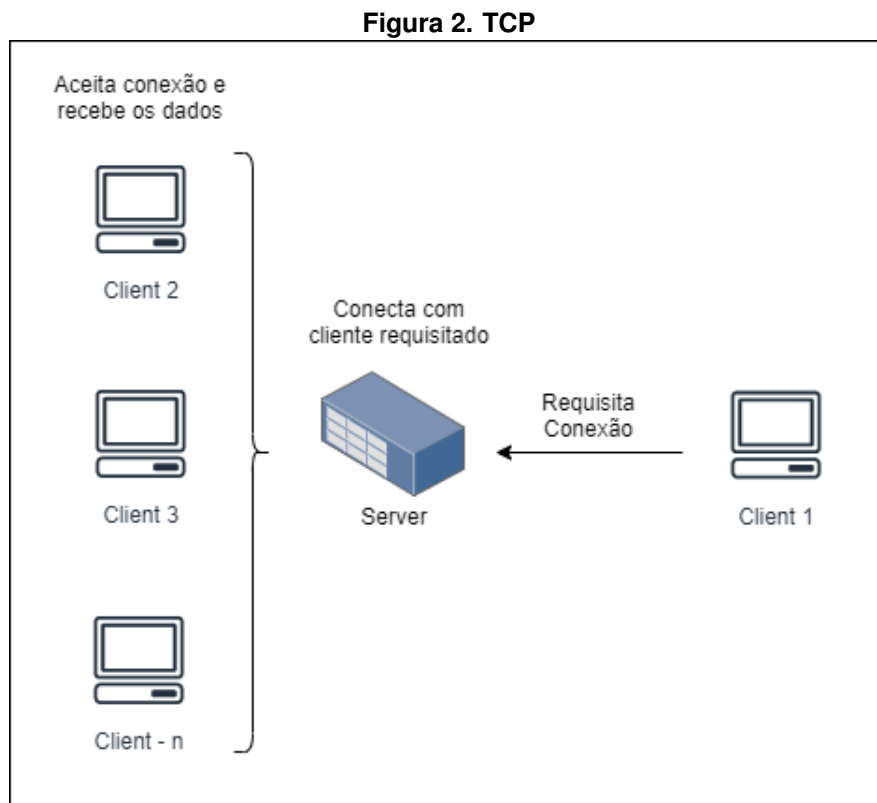
```

Para a comunicação peer-to-peer, foi realizado o desenvolvimento conforme a imagem abaixo:



Na figura acima, conecta-se, um usuário diretamente ao outro. De forma que os clientes funcionam também como servidor, e cada um dos clientes podem escrever e/ou ler conteúdo dos sockets existentes.

Para o protocolo TCP, realizou-se a implementação de um servidor como intermediário, conforme a imagem abaixo:



Desta forma, toda troca de informações entre clientes é tratada pelo servidor, de forma a garantir o envio e recebimento das mensagens.

2.3. Testes realizados

Durante a realização do projeto, o problema mais encontrado foi o de ter de sempre limpar o valor de variáveis durante envios e recebimentos. Desta forma, durante as trocas de informações entre o cliente e o servidor teve-se de garantir que as mensagens trocadas não fossem alteradas, desta forma, para strings, utilizamos a função *strncpy()* para garantir que o tamanho da string seja sempre fixo. E, para garantir que as variáveis não tivessem conteúdo nenhum na atribuição de valores, usou-se a função *bzero()*.

Para o teste de envio de mensagens entre clientes passando pelo servidor, houve o envio de diversas mensagens entre os clientes, de forma, que toda vez que houvesse o envio de mensagens, imprimia-se a mensagem do lado do servidor e do lado do destinatário, confirmando, assim que as mensagens são entregues e tratadas de maneira correta.

Além disso, houve a utilização de mutex, durante o cadastro e envio de mensagens, para evitar problemas de starvation e deadlock.

Para os testes envolvendo arquivos, realizou-se a verificação de escrita e criação de forma manual dentro do diretório especificado. Para a leitura, imprimiasse o conteúdo no terminal para verificação. Para garantir a segurança e confiabilidade dos dados escritos, fechasse os arquivos após a leitura e escrita.

3. Conclusão

Durante o trabalho utilizamos majoritariamente informações que foram obtidas através dos experimentos anteriores, e também de livros e documentações. Vários componentes dos experimentos foram aproveitados, e melhorados. Por fim, obtivemos sucesso na realização do projeto, de forma que, todas as funcionalidades implementadas realizam suas respectivas funções de acordo com a descrição do que foi proposto.

4. Referências

Documentação de funções

Redes de Computadores e a Internet: Uma Abordagem Top-Down