Redes de computadores

Trabalho prático 2

Diego Tomaz

Sistema de múltiplas conexões

1. Introdução

O objetivo deste trabalho é implementar um serviço de **chat** multi usuários baseado em socket **TCP/IP**, compatível com **IPv4** e **IPv6**. O código permite que até 15 clientes se conectem ao servidor e possam se comunicar por mensagens privadas ou de broadcast.

Fundamentalmente, um cliente tenta se conectar ao servidor e - a partir da aceitação do mesmo seguido pela atribuição de um ID - esse cliente passa a fazer parte da comunidade e pode receber ou enviar mensagens.

O servidor utiliza o conceito de múltiplas **threads** para aceitar e lidar com vários usuários simultaneamente (Limitado a 15). Quando uma nova conexão é estabelecida com o servidor, uma nova thread é criada para tratar essa conexão e cada thread é responsável por receber as mensagens do cliente, interpretar o conteúdo da mensagem e tomar as ações apropriadas com base no tipo de mensagem recebida, nas quais serão detalhadas adiante.

2. Protocolo

As definições das especificidades se deram, majoritariamente, pelas especificações do documento do trabalho. No entanto, conforme as conversas no fórum da disciplina se desenvolveram, alguns conceitos foram tomados como requisito/característica do protocolo como um todo, por exemplo sobre o tamanho da mensagem que trafega na rede, que foi discutido como sendo 2048 bytes no total.

Dito isso, o dado que trafega na rede possui o seguinte formato:

ID da mensagem (10 bytes)	ID do remetente (10 bytes)	ID do destinatário (10 bytes)	**Mensagem (2018 bytes)**

Ou seja, temos um protocolo relativamente simples mas que é funcional para esta finalidade, dado que possui todos os campos necessários para um serviço de chat: tipo da mensagem, ID do remetente, ID do destinatário e o payload/mensagem.

Como supracitado, o serviço de chat implementado permite que o usuário envie mensagens em broadcast ou privadas. Nesse sentido, mais especificamente temos como **tipos de mensagens** que trafegam na rede os seguintes IDs:

1	requisição para adicionar um novo usuário ao chat
2	requisição para remover um usuário do chat
4	resposta que contém a lista de usuários ativos no chat
6	por padrão, broadcast, ou seja, uma mensagem enviada para todos os usuários do chat. Se preenchido o ID do destinatário, torna-se uma mensagem privada
7	resposta de erro sobre limite de usuários ou de usuário desconhecido
8	resposta a uma requisição de remoção

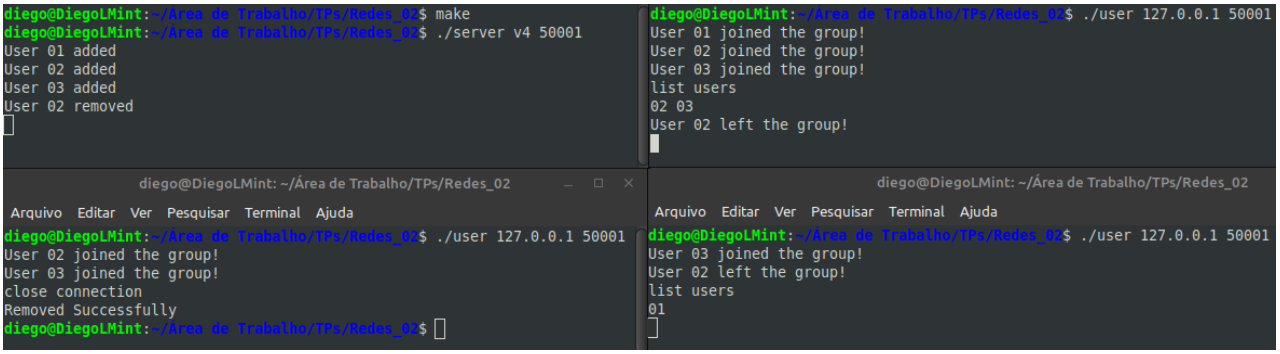
3. Ambiente, execução e testes

O desenvolvimento do trabalho se deu no Linux Mint, utilizando a ferramenta Visual Studio Code.

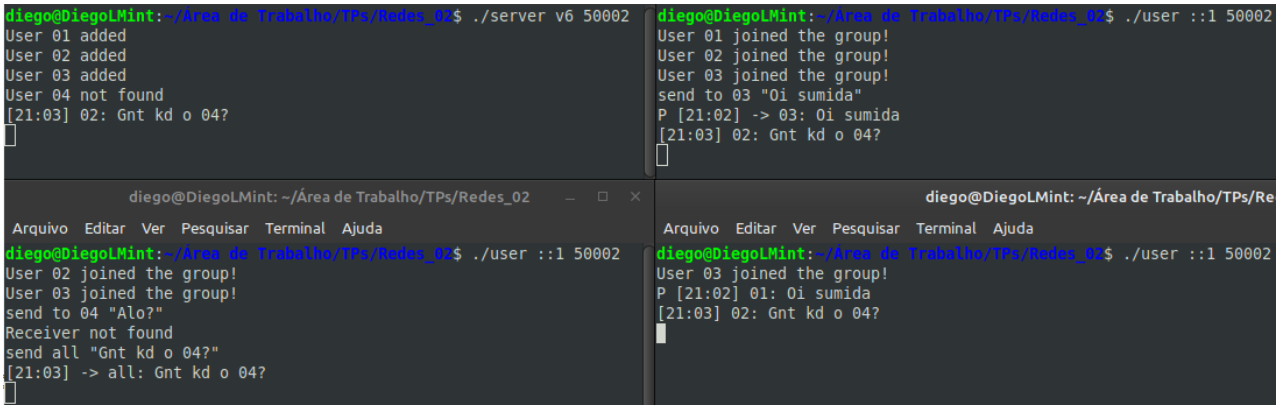
Para executar o projeto, basta executar ***make*** na raiz do (onde se encontra o *Makefile*) e, em seguida, abrir o terminal. O **Makefile** foi configurado para não imprimir nada na tela de modo a manter o terminal "limpo" e exibir única e exclusivamente os retornos do programa.

Ademais, ao digitar ***make clean*** são excluídos os arquivos gerados pela última compilação.

A solução desenvolvida atende a todos os requisitos estabelecidos. Para essa avaliação, utilizei as tabelas de output esperados que constam ao final da especificação do trabalho para efetuar os **testes**, conforme abaixo.



CENÁRIO 1: Abertura e encerramento de conexão e listagem de usuários. Executado em **IPv4**.



CENÁRIO 2: Abertura e encerramento de conexão, conversação privada e broadcast. Executado em **IPv6**.

4. Desafios

O desenvolvimento de um sistema de multi clientes trouxe dificuldades e desafios dado a necessidade de usar recursos que levam a esse resultado, que no caso foi escolhido trabalhar com **threads**. O uso de threads envolve o conhecimento da disciplina de Sistemas Operacionais e, portanto, esse trabalho teve um forte aspecto **multidisciplinar**, o que considero algo **positivo** ao passo que foi possível aprofundar conhecimentos em ambos assuntos simultaneamente.

5. Conclusão

A construção do trabalho foi desafiadora e também muito agregadora, dado que pude colocar em prática o que foi visto em sala de aula. Ademais, a intensa movimentação no fórum da disciplina foi de grande ajuda dado que a maioria dos tópicos que causavam dúvidas na especificação foram discutidos e, portanto, deixaram de ser uma dúvida e pudemos desenvolver o trabalho como um todo.