



Escolha de Ciências e Tecnologia
Departamento de Informática
Licenciatura em Engenharia Informática
Unidade Curricular Redes de Computadores
Ano letivo 2020/2021

Relatório do Trabalho Final

Docentes

Professor Luís Rato

Professor Pedro Patinho

Discente

José Santos nº 43017

Évora, fevereiro de 2021

Introdução

No âmbito da unidade curricular de Redes de Computadores, lecionada pelo Professor Luís Rato e pelo Professor Pedro Patinho, foi solicitado que os alunos desenvolvessem um cliente e um servidor assente nas diretrizes fornecidas no enunciado do presente trabalho, onde o servidor servia como um chat de conversação entre clientes.

Neste sentido, foi iniciado o processo de desenvolvimento do mesmo, usando como base, para o servidor e para o cliente, exemplos dados pelos docentes na plataforma Moodle.

Posto isto, foi criada uma Struct cliente, com o objetivo de guardar informações sobre o cliente, tais como:

int sock, para guardar o valor da socket associada ao cliente.

bool operador, permite saber se o cliente é ou não um operador.

char chat_pref[10], guarda qual o canal preferido do cliente.

char current_chat[10], guarda o canal atual onde está o cliente.

char username[10], guarda o username do cliente.

char password[10], guarda a password do cliente.

bool online, permite saber se o cliente está ou não online.

Para além disso, foi criada também uma função auxiliar **codificarMensagem**, que tem como argumentos um array de caracteres, que contém uma mensagem enviada pelo cliente, outro array de caracteres vazio, onde se coloca a mensagem sem o comando, e o tamanho do array com a mensagem. E, ainda, um array de Clientes e de canais.

Programa

Inicialmente, no servidor, é criada uma socket, faz-se bind e listen, e, mais tarde, um ciclo while onde se procuram conexões com o servidor. Quando esta é aceite, o novo cliente é adicionado ao array, e espera-se pela mensagem deste cliente.

Com esta mensagem recebida é realizada uma das seguintes operações consoante o comando, NICK, MSSG, JOIN, LIST e WHOS.

O comando NICK começa por averiguar se o nome que o cliente pretende utilizar é válido, e se outro cliente já está a utilizar o mesmo nome, se for possível mudar de nome, o cliente é atualizado.

O comando MSSG começa por verificar se o cliente já tem um nome. Se já tiver, envia a mensagem, se esta tiver tamanho inferior a 512 bytes, a todos os clientes que estejam no mesmo canal.

O comando JOIN verifica também se o cliente já tem um nome, e, se sim, muda o cliente para o canal escolhido, se esse existir.

O comando LIST, mais uma vez, verifica se o cliente já tem um nome, e mostra todos os canais que existem.

Por fim, o comando WHOS, se o cliente já tiver um nome, mostra todos os clientes que estão no mesmo canal.

Para todos estes comandos, são enviadas as devidas mensagens do servidor.

No cliente é apenas feita a conexão ao servidor, e depois um ciclo while, que espera por uma mensagem, envia ao servidor e depois efetua o output da resposta.

Conclusão

Embora não tenha implementado todas as funcionalidades descritas no enunciado, creio que neste trabalho foi possível aplicar os conteúdos lecionados nas aulas práticas ao longo do semestre.

Deparei-me com dificuldades ao tentar criar diversas funções devido a problemas com pointers, e por consequente, o trabalho foi implementado na main, o que o tornou mais ilegível e confuso. Outro problema que enfrentei foi referente aos scan's e print's do cliente, no sentido que um cliente apenas recebe as mensagens do servidor, após enviar uma mensagem.

Apesar de ter sido um trabalho desafiante, gostei de o realizar, e sinto que as funcionalidades implementadas funcionam como esperado.

