

Redes de Comunicação

Engenharia Informática • 2024

Relatório • Turmas online

Trabalho realizado por:

Daniel Pereira - 2021237092

Eduardo Marques - 2022231584

Índice

Objetivo do projeto.....	2
Funcionamento do código.....	2
Funcionalidades dos clientes.....	3
Conclusão.....	3

Objetivo do projeto

O objetivo deste trabalho prático é implementar um sistema de turmas online para difusão de conteúdos com o uso dos protocolos UDP e TCP, bem como das comunicações IP *multicast*.

Neste projeto existem três tipos de agentes, sendo eles os clientes: alunos e professores que se comunicam através do protocolo TCP e o administrador que comunica com o servidor por UDP através do comando *netcat*. O programa servidor é encarregado de verificar e aceitar as tentativas de *login* dos agentes e servir os seus pedidos.

Funcionamento do código

Para começar, o código dispõe de importantes bibliotecas que são cruciais no desenvolvimento do trabalho prático como por exemplo `stdlib.h`, `stdio.h`, `string.h`, `sys/socket.h`. O programa começa por ler as portas TCP e UDP e o nome do ficheiro que armazena os dados dos clientes através dos argumentos da linha de comandos. Além disso, o servidor garante a criação de processos filhos através da função *fork()* que lidam com as conexões de ambos os protocolos TCP e UDP e é responsável por gerir múltiplas conexões simultâneas.

A função *process_doc()* faz a leitura correta dos dados do ficheiro de texto e guarda-os numa struct do tipo *Cliente*, que por sua vez é guardada num array que contém todos os clientes. Na função principal do código, *main()*, é ainda criada a memória partilhada que possibilita o uso e alteração dos dados dos clientes e das turmas por parte de diferentes processos de forma simultânea e eficiente.

A função *connection_tcp()* aguarda as conexões de clientes via TCP e quando estes realizam o *login* através do *username* e *password*, se forem alunos ou professores, a função *process_client()* é chamada e o servidor fica responsável por servir os seus pedidos e garantir uma resposta adequada consoante as funcionalidades de cada cliente.

Por outro lado, a função *connection_udp()* garante a comunicação por UDP com o servidor através do comando *netcat* e permite aos administradores realizarem as suas tarefas.

A função *escreve_doc()* é responsável por manter as informações do ficheiro de texto referentes aos clientes atualizadas para garantir a continuidade dos dados dos clientes sempre que o servidor é fechado.

Em relação às turmas, as comunicações são efetuadas via *multicast*. Para este efeito, o servidor envia os endereços multicast necessários à receção dos conteúdos numa determinada turma aos alunos e aos professores sempre que ocorre criação e subscrição de uma turma. As turmas são guardadas num array e contém um nome, um endereço multicast e um tamanho máximo. No que toca à

subscrição nas turmas, existe um array para cada cliente que guarda as informações sobre as mesmas.

O servidor contém outras funções muito importantes como a *verifica_nome()* que garante que não haja nomes repetidos para facilitar a parte de eliminação de clientes, a *gerar_multicast()* que gera um endereço *multicast* de forma *random* dentro da *class D* de endereços recorrendo à função *verifica_multicast()* que impossibilita o uso de endereços iguais para turmas diferentes. As funções *imprimir_clientes()*, *imprimir_turmas()* e *imprimir_subscribed()*, imprimem a lista dos clientes, a lista de turmas disponíveis e as turmas nas quais os clientes estão inscritos, respetivamente.

Funcionalidades dos clientes

Os agentes dispõem de diferentes funcionalidades suportadas pelo servidor, isto é, o administrador tem a possibilidade de adicionar um utilizador (ADD_USER), eliminar um utilizador (DEL), listar os utilizadores (LIST) e desligar o servidor (QUIT_SERVER). Os alunos e professores podem listar as turmas disponíveis (LIST_CLASSES), obter informações sobre as turmas nas quais já estão inscritos (LIST_SUBSCRIBED) e inscrição numa turma (SUBSCRIBE_CLASS). Os professores conseguem ainda criar turmas (CREATE_CLASS) e enviar conteúdos para as mesmas (SEND). O login é feito por todos através do comando LOGIN.

Conclusão

Para concluir, o código desenvolvido em C permite a criação de um servidor que dispõe de várias funcionalidades e procura responder aos pedidos dos clientes através da troca de mensagens entre os mesmos. O uso dos protocolos TCP e UDP garante a partição e divisão das diferentes funções, o que garante a eficiência do programa.