

# Redes de Computadores

Trabalho Final 2020/2021

Trabalho realizado por: João Raposo nº 45234

Pedro Anjos nº 45558

Rafael Santos nº 45813

Índice

1. Introdução
   1. Problema apresentado
2. Funções
3. Server
4. Comportamento do Server e do Cliente
5. Na comunicação em tempo real 1-1
6. Na comunicação em tempo real 1-todos
7. Na comunicação em diferido 1-1
8. No jogo do galo

1. Introdução
   1. Problema apresentado

A empresa UltraSecure, Lda. pretende utilizar um sistema de comunicação para os membros da sua equipa, mas não pode confiar em software de terceiros. Assim sendo, precisa de implementar “in-house” um software deste tipo, garantindo que não há possibilidade de falhas de segurança nesta aplicação.

Para ajudar a UltraSecure na implementação deste software deste, deve implementar as seguintes componentes:

1. Sistema de “chat”:

O sistema de chat funciona num paradigma cliente/servidor, podendo os clientes comunicar uns com os outros nos seguintes termos:

* Comunicação em tempo real 1-1: um cliente envia mensagens para outro cliente, que as recebe em tempo real;
* Comunicação em tempo real 1-todos: um cliente envia mensagens para um canal comum, e todos os outros clientes recebem as mensagens em tempo real;
* Comunicação em diferido 1-1: um cliente envia mensagens para um outro cliente que não está online, e este receberá as mensagens mais tarde, quando se ligar ao servidor.

1. Sistema de transferência de ficheiros:

Aliado ao sistema de chat, podem existir “mensagens especiais” que não representam texto, mas dados em formato binário que constituem ficheiros a ser transmitidos entre clientes. Tirando partido da infraestrutura de chat, podem ser transmitidos ficheiros nos três métodos referidos anteriormente: 1-1 e 1-todos em tempo real e 1-1 em diferido.

1. Sistemas de jogos com um jogo de exemplo:

Neste trabalho, o jogo escolhido e desenvolvido foi o jogo do galo.

1. Funções

void print\_ttt(char ttt[]) - Função responsável pelo output do tabuleiro após uma jogada por parte de um dos jogadores

int try\_play(char ttt[], int spot, char mark) - Função que dispõe ao oponente a possibilidade de realizar uma jogada

int check\_win(char ttt[]) - Função que verifica a possibilidade de vitória de um dos clientes

char get\_move(char ttt[]) - Função que impõe a possibilidade de realizar uma jogada. Caso uma posição se encontre ocupada, o código impõe a escolha de outra posição livre

bool checkFile(char nome[],char pass[],FILE \*file) - Esta função booleana verifica a possibilidade de um cliente se conectar à rede. A função tem o ficheiro já aberto na função main como argumento, e a mesma lê-o. Caso o char getWord possua corretamente os dados de um utilizador, o mesmo retorna a lógica verdadeira e é autorizado a sua ligação. Em caso contrário, retorna a lógica falsa e o utilizador terá de introduzir novamente os dados

bool verificaSeUtilizadorConectado(char nome[],FILE \*file) - Após a escolha do chat, a função verifica novamente a possibilidade de um utilizador se conectar com o servidor e ter a possibilidade de iniciar o chat. Com o ficheiro passado como argumento, se não existirem diferenciação entre os dados introduzidos e o ficheiro, é retornado a lógica verdadeira

1. O servidor conecta-se e aguarda na função select(), ou por uma nova conexão, ou por alguma mensagem que necessita ser enviada de um cliente para outro.
2. O servidor utiliza um ficheiro que contêm todos os utilizadores online e o seu respetivo socket, quer seja para ter conhecimento do socket para o qual irá enviar uma mensagem , quer seja para verificar se um utilizador encontra-se online ou offline. O servidor ao receber uma mensagem por parte do cliente , vai verificar: primeiro se essa mensagem é um alerta a indicar que o cliente se desconectou .Caso isso aconteça o servidor vai atualizar o ficheiro que contêm todos os utilizadores online e apaga o cliente que se desconectou; Um outro caso ocorre quando é enviada pelo cliente a mensagem "1-1 em tempo real" , isto significa que é para enviar uma mensagem no 1-1 tempo real para outro utilizador, onde o servidor recebe o nome da pessoa para quem quer enviar e procura dentro do ficheiro de utilizadores online o respetivo socket. o servidor recebe também a mensagem e envia essa mensagem para o socket correspondente.Quando a mensagem inicial é "aut\_time", significa que existe uma nova conexão de um cliente e o servidor vai inserir o username e o socket correspondente ao ficheiro de utilizadores online , o servidor também verifica se existe mensagens pendentes aquando este estava offline , ou seja, outro utilizador ligou-se ao chat 1-diferido e o utilizador para o qual enviou as mensagens encontrava-se offline. Outro caso ainda ocorre quando a mensagem inicial contêm 1-1 em diferido , neste caso o servidor recebe o username para quem será enviada a mensagem , tal como a mensagem , e estas são guardadas numa struct "mensagemDiferido" que têm como atributos "char nome[20] que guarda o nome , e "char mensagem[256] que guarda a mensagem que depois será enviada. No ultimo caso pertencente ao chat , se a primeira mensagem enviada pelo utilizador é "1-todos" , o servidor recebe a mensagem e envia-a para todos os utilizadores conectados. Por ultimo , caso não seja nenhuma das opções descritas anteriormente , o servidor assume que a mensagem recebida pertence ao jogo do galo , ou seja, o que é enviado para o servidor é a posição onde o cliente joga e o servidor envia a posição para o seu adversário.
3. Comportamento do Server e do Cliente

Após a conexão do cliente ao servidor, é feita uma pergunta ao cliente para verificar se já possui uma identificação. Se a resposta for ‘sim’, é-lhe pedido depois o username e a sua password que serão procuradas num ficheiro que contém os nomes e as palavras passes dos clientes com identificação já feita. Caso o cliente não tenha identificação, responderá à pergunta inicial ‘nao’ e terá que fazer a sua identificação introduzindo um username e uma password à escolha em que ambas serão introduzidas no ficheiro. Depois será preciso responder ‘sim’ à pergunta que foi feita no princípio e meter os novos dados.

Após a identificação do cliente, será preciso escolher o tipo de chat que o cliente deseja: caso o input seja ‘1’ para a comunicação 1-1 em tempo real; ‘2’ para a comunicação 1-todos em tempo real; ‘3’ para a comunicação 1-1 em diferido; ‘4’ para jogar o jogo do galo.

* 1. Na comunicação em tempo real 1-1

Caso a resposta seja ‘1’, será pedido ao cliente o nome de quem quererá falar e verificará se este está online ou não. Quando ambos esta online e prontos para conversar um com o outro, ambos ficam a espera para escrever ou para receber uma mensagem. Quando o cliente escreve, é enviado primeiramente a frase “1-1 em tempo real” para o servidor para que este saiba com que tipo de comunicação irá trabalhar e o servidor responde de volta. Posteriormente o cliente envia o nome do utilizador com quem quer falar para o servidor e ele vai verificar dentro de um ficheiro o socket correspondente, sendo assim possível saber qual o socket a quem vai enviar a mensagem. Por fim , o servidor recebe a mensagem por parte do cliente e envia-a para o socket correspondente ao nome fornecido pelo mesmo. O cliente pode sair para o menu principal , escrevendo a mensagem "VOLTAR ATRAS" e terminar o programa com a mensagem "SAIR". Quando o cliente recebe uma mensagem de outro, é lida e faz-se um print.

* 1. Na comunicação em tempo real 1-todos

Caso a resposta seja ‘2’, o cliente será avisado em que comunicação entrou e assim que entra, uma mensagem é enviada ao servidor contento “1-todos” para que o servidor saiba em que tipo de comunicação se irá utilizar. Quando há mais clientes online e no mesmo modo de comunicação, cada vez que um envia uma mensagem, todos recebem a mensagem. Qualquer cliente poderá enviar mensagens. Caso um deles escreva “VOLTAR ATRAS”, volta ao menu, caso escreva “SAIR”, o cliente sai do programa.

* 1. Na comunicação em diferido 1-1

Caso a resposta do cliente seja ‘3’, o cliente entra na comunicação 1-1 em diferido e é-lhe pedido o username da pessoa com quem quer conversar. Depois de o introduzir, é enviado ao server a mensagem “1-1 em diferido” para que este saiba em que modo se vai trabalhar. Posteriormente o cliente envia o nome do utilizador com quem quer falar para o servidor e ele vai verificar dentro de um ficheiro o socket correspondente, sendo assim possível saber qual o socket a quem vai enviar a mensagem.

Assim que o outro cliente ficar online, este receberá as mensagens que estão destinadas a ele e que estiveram à espera de serem enviadas quando ele estivesse online.

* 1. No jogo do galo

Caso a resposta do cliente seja ‘4’, o cliente entra no jogo do galo. É-lhe logo apresentado um tabuleiro vazio para jogar e é dado a hipótese de ser o primeiro a jogar ou o segundo, caso espere pela jogada do oponente. Tanto o cliente como o oponente têm um tabuleiro cada um.

Quando o cliente joga, haverá uma variável fiscal (inicializada a 0) que será incrementada porque, caso o cliente ou o oponente tente jogar duas vezes seguidas, será impossibilitado porque o fiscal não pode ser maior que 1. Depois desta incrementação, é pedido a jogada. Caso a posição não esteja ocupada, é introduzido um ‘O’, caso contrário aparece uma mensagem a avisar que a posição escolhida já está ocupada e terá que escolher outra. Depois de escolhida e realizada a jogada, é enviado ao oponente o buffer que a contém. Caso haja vitória, o tabuleiro é imprimido, avisa que o cliente ganhou e é enviado ao oponente um buffer com a letra ‘w’. Caso haja empate, é contato o número de posições ocupadas e se for igual a 9, então avisa o cliente que houve empate.

Quando o cliente recebe a jogada do oponente, haverá a variável fiscal, já referida anteriormente, que será posta a zeros para que quem lê a jogada do oponente seja o seguinte jogador a escolher uma posição. Depois do cliente receber a jogada, irá a jogar no seu próprio tabuleiro e verificar se o oponente, com a jogada que enviou, ganhou o jogo. Caso o que recebeu seja um ‘w’ quer dizer que o cliente ganhou o jogo, se não irá verificar se o vencedor da partida foi o oponente. Se nenhum ganhar o jogo, ou se verifica se houve empate da mesma maneira de quando o cliente joga ou então continua-se a jogar ate haver um vitoria e derrota ou empate para ambos.

Quando o jogo é terminado, o programa continua a correr, pedindo que tipo de comunicação se quer ou se quer jogar outra vez.