Trabalho 2 IRC



Bruno José Borges Madureira nº 2011161942

Este trabalho foi dividido em 3 programas diferentes, que foram enviados em 3 ficheiros diferentes: “server.c”, “cliente.c” e “server\_ex\_2.c”. O trabalho foi feito em linguagem c já que apesar de supostamente ser mais complicado tínhamos o auxilio do código e da experiencia resultante das fichas praticas semanais.

O ficheiro “server.c “é uma implementação de um servidor que suporta um chat, recorri ao select para fazer a sincronização das mensagens, permitindo fugir de certa forma às dificuldades que advém da programação concorrente. O ficheiro “cliente.c” é uma implementação do cliente que se vai ligar ao server, ao escrever . O ficheiro “server\_ex\_2.c” é uma implementação do servidor mas adaptado a correr no servidor central do chat do dei.

É de notar que recorri ao protocolo tcp já que era necessário garantia de entrega de mensagens e não degradação do seu conteúdo.

1-Server.c

No server.c foi implementada uma solução recorrendo a função select, depois de alguma pesquisa implementei da seguinte forma:

select(fdmax+1,&ler\_fds,NULL,NULL,NULL);

Em que “fdmax” é uma copia do fd gerado pelo socket, e ler\_fds é o file descriptor temporário usado pela função.

Foi implementada uma função erro para imprimir o erro sempre que algum erro acontecer em determinadas funções, como por exemplo no bind ou no listen.

Depois de serem feito as chamadas a função socket, bind é feito o listen(), a escuta é feita até um máximo de 100 clientes, mas isto pode ser alterado facilmente alterando o define da constante Max.

O ciclo infinito implementa o servidor em si, estando lá o select e um ciclo for que itera as conexões existentes em busca de dados.

Este select auxiliado de dois ciclos for, conseguem coordenar a escrita e a recepção de mensagens de forma independente.

Ainda foram colocados “prints” que informam a conexão de novos clientes, informando o seu ip em que os sockets se ligaram.

Do ponto de vista do utilizador, o arranque do servidor é fácil, sendo apenas necessário correr o executável na linha de comandos, a saída do programa é feita a partir de “control +c”.

2-Cliente.c

Este é o programa com o qual o utilizador se vai contactar ao servidor, a chamada deste programa na linha de comandos implica 3 argumentos: ip do servidor, porto da conexão (por defeito 9999), e nome do utilizador.

Como não foi explícito no enunciado, não coloquei qualquer restrição para evitar que dois utilizadores terem o mesmo nome.

O programa é mais simples que o servidor, apenas se faz a criação de um um socket e a ligação pelo mesmo ao servidor. De seguida é feito um fork(), dividindo a execução do programa em dois processos independentes em que ambos tem toda a informação proveniente do código anterior, resumindo mais objectivamente: ambos estão conectadas ao cliente pelo socket. O processo filho fica encarre de receber a informação e escreve-la no ecrã, enquanto que o pai é responsável por fazer o “scan” ao utilizador e enviar o que ele escreve. Para isto usei as funções read e write respectivamente.

3.Server\_ex\_2.c

Neste exercício alterei o programa de forma a ligar através da função connect ao servidor do dei, o nome e o porto entram como argumentos do programa. O ip do servidor é gerado a partir da função “gethostbyname”.