Introdução às Redes e Comunicação

Ano Letivo de 2016/2017

Trabalho 2

**Implementação de um servidor de email**



João Clara nº2015245403

Renato Santos nº2015237457

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Coimbra, 2016

**Manual do utilizador:**

O objetivo do trabalho é implementar um servidor de email para um grupo específico de clientes. Para a realização deste programa, nós escolhemos a linguagem C.

O primeiro passo a realizar para a utilização desde programa é compilar e executar os dois ficheiros fornecidos em anexo.

Para a compilação, iremos escrever o seguinte no terminal:

cd <diretoria da pasta do projeto>

gcc server.c –o server –Wall –lm

gcc client.c –o cliente –Wall –lm

Para a executação do programa, iremos escrever o seguinte na janela do terminal:

./server

De seguida, abrimos uma outra janela de terminal, que digitamos o seguinte:

./cliente localhost 9000

Em que “localhost” é o HOST que nós definimos e 9000 o porto conhecido que nós definimos.

E temos assim, o nosso programa em pleno funcionamento, com a conexão dos sockets entre servidor e cliente a estarem estabelecidas.

De imediato, é pedido ao utilizador um username e uma password, com o objetivo de verificar se o login introduzido é válido ou não. Para ser válido, o username e a password têm ambos de ser autorizados no ficheiro “users.txt”. Caso o login seja válido, irá aparecer logo de seguida o seguinte menu:



Dentro do menu, temos 8 opções válidas, que garantem a comunicação entre o cliente e o servidor, que são:

1) LIST\_MESS – para listar todas as mensagens por ler.

2) LIST\_USERS – para listar todos os clientes autorizados.

3) SEND\_MESS – para enviar uma mensagem para um cliente (autorizado).

4) LIST\_READ – para listar todas as mensagens já lidas.

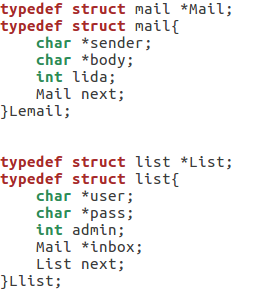
5) REMOVE\_MES – para apagar mensagens.

6) CHANGE\_PASSW – alterar a password

7) OPER – para o cliente obter os privilégios do operador.

8) QUIT – para o cliente abandonar o sistema.

**Relatório técnico**



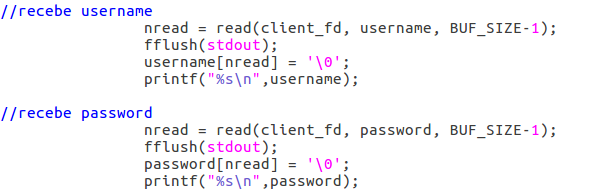
O projeto tem duas estruturas, uma de utilizadores e outra de mails, vao ser utilizadas para fazer listas ligadas, sendo que para cada elemento da lista de utilizadores há uma lista de mails.

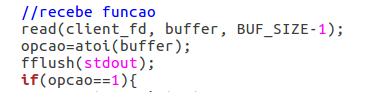


No main definimos a estrutura do socket, depois importamos a informação do ficheiro “users.txt”, e de seguida criamos o socket, fazemos bind e listen e começamos a aceitar processos dos clientes que se querem conectar.

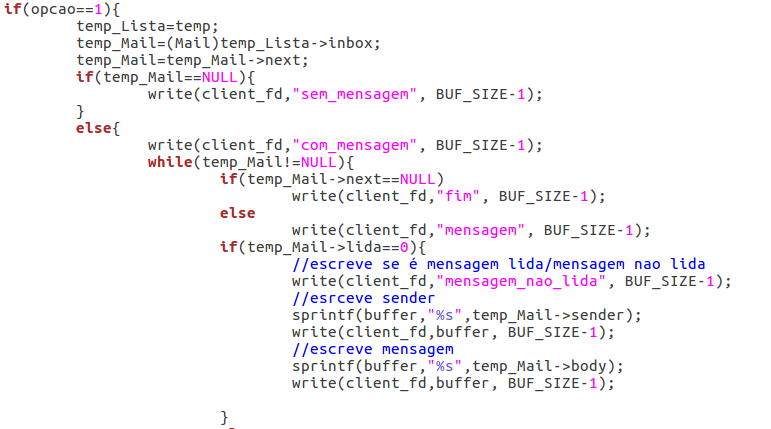


No process\_client

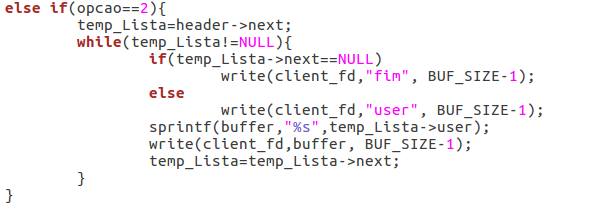
 Recebemos um user e uma pass com a função read, depois com um ciclo while, percorremos a lista de users e com um strcmp() comparamos os dados recebidos com cada um dos dados da lista, se encontrar um user e pass iguais envia com um write() uma mensagem a a dizer que encontrou, se não, manda uma mensagem a dizer que não existe aquela combinação de user/pass.



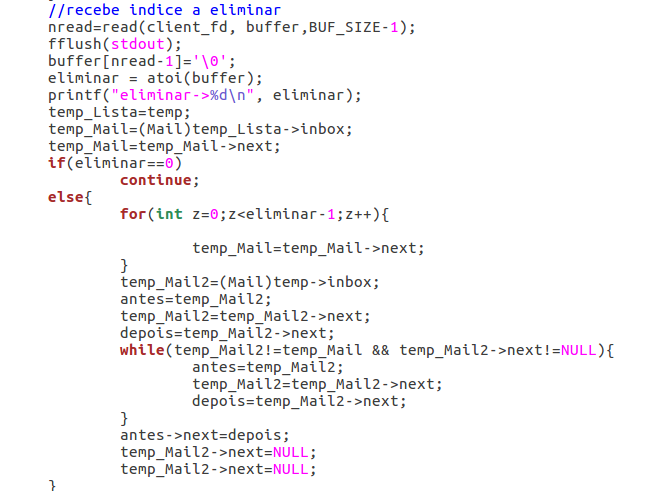
Se o o user/pass existir vai aparecer um menu de opçoes na pagina do cliente, que por sua vez vai introduzir a opção que deseja, que é enviada, e recebida no server com um read da string, que depois é convertida num inteiro que vai ser utilizado numa cadeia de if’s e else if’s dependendo do numero recebido.

****

Se é recebido um 1 do cliente o server vai buscar a estrutura do cliente que está a usar o server, e a sua inbox, com um if vai verificar se o cliente tem algum mail. Se não tiver manda um write() ao cliente a dizer que não tem mails, se tiver vai percorrer os mails disponiveis com um while(temp\_Mail->next!=NULL), este while serve para ver se existe mais um mail a seguir do atual, ou se o atual é o ultimo. Se o atual for o ultimo, manda uma mensagem a dizer “fim” que representa o fim dos mails, para o cliente saber quando tem que parar de ler, se não for o ultimo envia “mensagem” para continuar a ler.

No if seguinte, aplicamos o mesmo principio, a opção 1 é para listar os mails não lidos, o if verifica se a mensagem já foi lida ou não, se for não a mostra , caso contrário mostra.

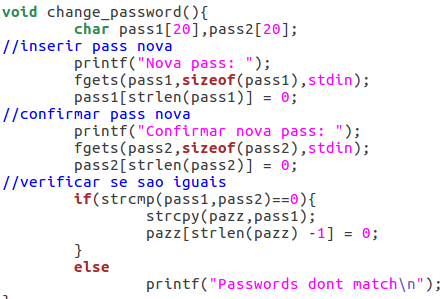
Na opção 2 é usado um while para listar todos os clientes com o mesmo principio de if/else para verificar se é o ultimo user.

 Na opção 3 recebemos com um read o numero de pessoas a que é pretendido mandar uma mensagem, o conteudo da mensagem e fazemos um for com o numero de iteraçoes igual ao de numero de pessoas recebido, dentro do for recebemos o nome de cada uma dessas pessoas e percorremos um while em que comparamos o nome recebido a cada um dos da lista, para inserir os dados dessa pessoa como recetor.

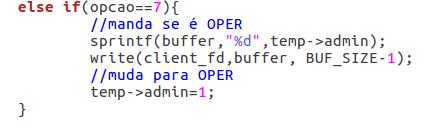
A opção 4 é igual à 1 mas invertidas as mensagens lidas e não lidas.

A opção 5 começa da mesma forma que a 1 e a 4, mas listando todas mensagens, lidas e não lidas,

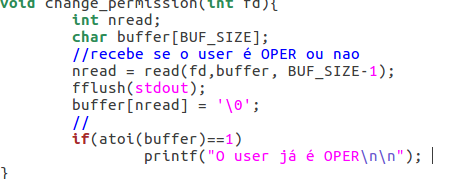
mas depois de imprimidas todas as mensagens com um indice atras, o utilizador vai inserir o indice da mensagem que pretende apagar, atraves desse indice vamos buscar o mail respetivo e de seguida apaga-lo, apontando o next do mail anterior para o mail seguinte ao que pretendemos apagar.



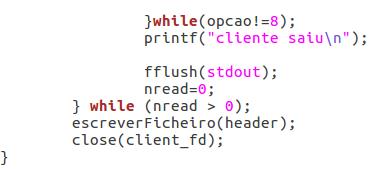
A opção 6 vai ser feita totalmente no ficheiro do cliente enviando apenas atraves do write() a nova password para o server.



A opção 7 vai mandar ao ficheiro do cliente se esse cliente é operador ou não,

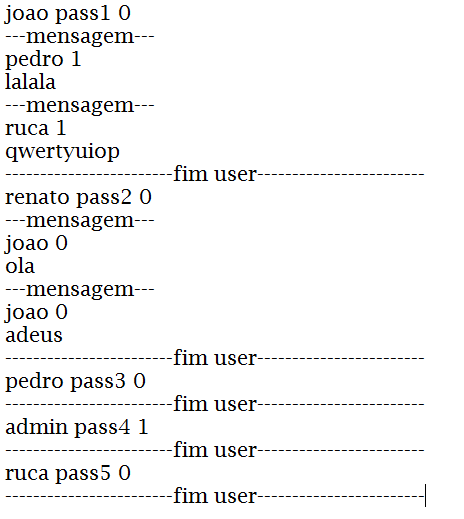


O cliente vai receber essa informação e se não for operador, vai passar a ser, se já for dá print de uma mensagem a dizer que já é.



Por fim, se a opção for 8 vai sair do ciclo do-while e escrever toda a informação alterada de volta no ficheiro “users.txt”.

Temos também as funçoes secundarias create\_list() e create\_message() que vão simplesmente criar um header para a lista ligada, e a função insert\_list() e insert\_message() que vão pegar na informação de uma estrutura dada e colocar como proximo elemento da lista ligada.



Aqui está o ficheiro que guarda a informação, cada utilizador é definido inicialmente com “nome password admin” sendo admin 1 se for operador e 0 se não for. Na proxima linha aparece “---mensagem---“ se tiver uma mensagem, e por baixo da mensagem “nome leitura” o nome do utilizador que enviou seguido de se a mensagem está lida(1) ou não (0). Podendo aparecer qualquer numero de mensagens, é deifinido o fim de um utilizador com “---------fim user---------“ voltando ao inicio.