Introdução às Redes de Comunicação

Trabalho 2

Realizado por:

Rui Mendes 2013136967

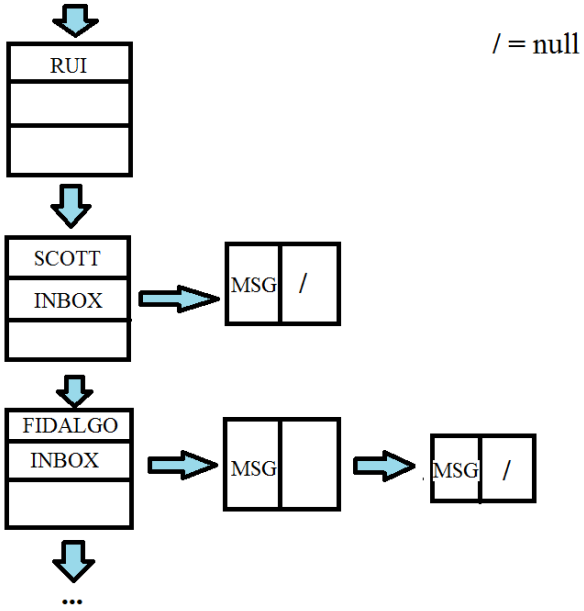
Ana Fidalgo 2013134819

Introdução

O objetivo do trabalho é implementar um serviço simplificado de correio eletrónico. Para isso criamos um servidor de correio eletrónico, um programa cliente e um segundo servidor caso o cliente não encontrasse o destinatário desejado. É necessário referir que a utilização de sockets foi necessária, estes são os elementos principais para fazer a ligação entre o cliente e o servidor e permite que as mensagens sejam enviadas. Além disso foi utilizado o protocolo TCP na totalidade do projecto.

O cliente permite a ligação ao servidor com autenticação do utilizador com login e password. Permite ainda enviar mensagens para um ou mais destinatários, lista-las, lê-las e apaga-las. O Servidor recebe as mensagens destinadas aos utilizadores e guarda-as em memória, respondendo aos pedidos de um ou mais programas clientes simultaneamente. Neste projeto temos ainda um segundo servidor que apenas recebe as caso contenha o destinatário especificado pelo cliente, ou caso o destinatário não seja encontrado no primeiro servidor, este irá ser procurado nele e caso mais uma vez não seja encontrado devolve uma mensagem de erro para o utilizador original.

Visão Geral:



Ao correrm o programa cliente, recebemos como argumento o porto do servidor, ao qual nos conectamos. Antes de ser aceite, será necessário que se introduza um username com a password correspondente, caso o login seja bem sucedido, o

utilizador tem como opção enviar mensagens a outros utilizadores, todas estas interações cliente-utilizador são realizadas através da criação um socket.

Manual de Utilizador

Para conseguir iniciar corretamente o programa é necessário seguir alguns passos:

1- Num primeiro terminar correr o servidor para o por à “escuta” de ligações.
   >gcc servidor1.c –o sevidor.out
   >.\servidor
2- Seguidamente noutro terminal, corro o cliente com os argumentos corretos referentes ao servidor1.
   >gcc cliente.c –o cliente.out
   >./cliente localhost 9002
3- Insere-se o username e a password.
4- Surge um menu que diz: "Que pretende fazer? Escrever mensagens-1 ; Ler mensagens-2 ; Apagar Uma Mensagem-3 ; Sair-5"
5- Seguindo as instruções, é possível gerir o email.

Manual de Programador

1 – cliente.c - começa-se por se inserir os seguintes argumentos para se estabelecer a ligação: *cliente <host> <port>*. Caso o número de argumentos esteja errado, pede para voltar a inseri-los e caso o porto adicionado não seja encontrado devolve uma mensagem de erro.

Seguidamente estabelece-se a ligação através de um socket, que depois de ser aceite pelo servidor, irá ser estabelecida pelo *connect(),* caso isto não aconteça, é novamente devolvida uma mensagem de erro.

Depois de estabelecida a ligação é efetuado o login do utilizador, pede primeiramente o username, caso seja inserido com mais de 20 caracteres é devolvida a mensagem de que é demasiado longo. Depois de se inserir o username é pedido a password, que também limitamos para um máximo de 20 caracteres, caso a pass e o user estejam corretos é efetuado o login com sucesso.

Se o utilizador inserido não estiver guardado nos ficheiros, ele pode-se registar e assim criar um novo utilizador.

Seguidamente são apresentadas ao utilizador várias opções, escrever mensagens, ler, apagar uma mensagem ou a inbox inteira ou apenas sair.

Caso o utilizador pretenda escrever uma mensagem, ser-lhe-á pedido que introduza um destinatário e um assunto, sendo que ambos também estão limitados por 20 caracteres. Seguidamente o utilizador escreve a mensagem que pretende enviar a outro utilizador, limitada a 100 caracteres.

Pode ainda ler uma determinada mensagem da sua inbox através do seu id ou lista-la por completo, no caso de ler apenas uma primeiramente irá verificar se a mensagem selecionada existe, caso não exista irá devolver uma mensagem de erro, caso exista lê a mensagem e pode ainda apaga-la se pretender através da função *apagaMsg*.
Imaginemos que ele antes de ler uma mensagem em especifico pretende ver a inbox, ou seja, as mensagens todas que esse utilizador recebeu, verá então a sua inbox a partir da função *imprimeMsg*. Caso a função *aLer* seja null, quer dizer que a dita inbox do utilizador está vazia e devolve esta informação.

2 – Servidor1.c – Começou-se por se definir um porto de 9002 e criar um *socket* para se poder estabelecer uma ligação com o cliente. A função *bind()* é que associa este socket a um determinado endereço, a *listen()* que irá aguardar pela receção de ligações e quando é detetada uma ligação esta irá ser aceite pela accept(). A variável *resposta* (na função *void process_client(int client_fd))* seguidamente vai informar o cliente se foi aceite ou não a ligação com o servidor e este recebe do cliente através do *master* o que é que o cliente pretende realizar, escrever, ler, apagar mensagens, apagar a inbox (todo o conjunto de mensagens associadas a esse utilizador) ou simplesmente sair.

Ao se efetuar o login, se o nome de utilizador não for encontrado pode-se criar um novo que será adicionado à listaU e escrito no ficheiro dos users.

Durante o login é pedido o nome de utilizador, este é procurado na função *procuraUser* e faz-se o mesmo procedimento para a password, neste caso na função *procuraPass.*

Caso o cliente escolha que pretenda escrever uma mensagem, este dirá o seu destinatário que será procurado na lista dos utilizadores, seguidamente irá ser chamada a função void insereMsg(listaU user,char* msg,char* assunto) e insere a mensagem na inbox desse destinatário.

Se o utilizador quiser ler uma mensagem, através do seu respetivo id, esta mensagem será procurada no ficheiro e depois devolvida se existir esse mesmo id.

É também através da procura do id que uma mensagem é apagada do ficheiro que contem todas as mensagens relativas a um utilizador (inbox) .Temos uma função que conta as mensagem que estão na inbox, a função *contaMsg.*

Funções Utilizadas na manipulação de mensagens:

void printListaU(listaU p) – começa por pedir o username e a password correspondente do utilizador;

listaU procuraUser(listaU p,char* username) – seguidamente nesta função vai verificar se o utilizador inserido existe, se não existir irá devolver null.

listaU insereListaU(listaU p,char* nome,char* pass)- insere na listaU um novo utilizador.

void escreveFicheiroU(char* nomeFich,listaU l)- passa para ficheiro todos os utilizadores de l.

listaU leFicheiroU(char* nomeFich)- lê do ficheiro todos os utilizadores que ele possui.

void escreveFicheiroMsg(listaU l)- passa para cada ficheiro de mensagem de cada utilizador (por exemplo, "rui.msg") a sua inbox através de um ciclo que percorre os utilizadores e que para chama a função escreveFicheiroMsgUser.

void escreveFicheiroMsgUsers(listaU u)- é chamada pela escreveFicheiroMsg. A sua função está explicada em cima.

listaM leFicheiroMsgUsers(listaU l)- lê do ficheiro as inbox e guarda-as nos utilizadores respetivos através de um ciclo que percorre os utilizadores e que para chama a função leFicheiroMsgUser.

void leFicheiroMsg (char* nomeFich)- é chamada pela leFicheiroMsg. A sua função está explicada em cima.

listaM apagaMsg(listaU user,char* assunto)- apaga uma mensagem da inbox do utilizador principal.

void insereMsg(listaU user,char* msg,char* assunto)- insere mensagens na inbox do utilizador especificado no destinatário.

void imprimeMsgsUser(listaU u)- imprime as mensagens do utilizador principal.

int contaMsg(listaU u)- devolve um inteiro que corresponde ao número de mensagens na inbox do utilizador u.

Conclusão

No momento de entrega do trabalho, verifica-se que conseguimos atingir alguns dos parâmetros pedidos, como a criação de uma ligação cliente-servidor, das estruturas de utilizador e de inbox (através das listas ligadas), da sua manutenção em ficheiros e das funcionalidades de envio de mensagem e de listagem de mensagem. Contudo, não foram concretizadas a tempo algumas funcionalidades pedidas, nomeadamente a capacidade de, caso o utilizador não exista no servidor, ir procurá-lo num segundo servidor, enviar mensagens para mais que um destinatário.