



Faculdade de Ciências e Tecnologias da
Universidade de Coimbra

Relatório do Trabalho 2

Implementação de um Servidor de Email

Leonardo Machado Alves Vieira Nº2015236155

Tiago José Rodrigues Menezes Nº2015236199

Índice

Introdução	3
Requisitos	3
Implementação	4
header.h	4
server.c	4
cliente.c	5
Execução	6
Conclusão	6

Introdução

O objetivo deste trabalho é a implementar um servidor de emails para um grupo específico de clientes, o qual deverá ter suporte para o uso simultâneo de múltiplos utilizadores.

Todo o projeto foi desenvolvido em C.

Requisitos

Cada cliente terá uma identificação com um login único com um comprimento máximo de 9 caracteres. Também deverá existir uma password que será guardada pelo server. Ambas as credenciais serão encriptadas.

O servidor irá estar à espera de mensagens dos clientes num porto conhecido e que é configurável no arranque, na linha de comando, utilizando o switch `-p` seguido do número do porto pretendido.

No início, o servidor deverá solicitar ao utilizador o seu login e password. De seguida, o servidor deverá verificar se esse login e password existem no ficheiro de clientes autorizados (`"cliente.aut.txt"` – o qual contém o login e a password encriptadas).

O servidor deverá suportar varios operadores que terão permissão para enviar mensagens para outros utilizadores.

A comunicação entre o cliente e o servidor deverá suportar as seguintes primitivas com um conjunto de parâmetros adequados:

- 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.

O servidor deverá suportar também o envio de mensagens para múltiplos utilizadores. Quando um cliente recebe uma mensagem e se encontra ligado ao servidor deverá ser informado da nova mensagem.

Todas as ligações são TCP.

Implementação

[header.h](#)

No *headerfile* *header.h* dos ficheiros *server.c* e *cliente.c*, são feitos todos os *includes* e *defines* necessários para o funcionamento do servidor. É também definida a estrutura cliente, que contém os campos *name* e *pass*.

São também definidas no *headerfile* todas as funções criadas para que não existam warnings.

[server.c](#)

Ao iniciar, o servidor irá configurar as sockets de modo a que estas estejam ligadas ao porto previamente fornecido na linha de comando pelos argumentos.

De seguida o server ficará à espera que um cliente se conecte e que forneça a sua informação de login e a password. Caso esta informação seja válida o cliente liga-se ao servidor e é-lhe apresentado um menu com várias opções de funcionalidades existentes no server. Caso contrário a ligação é interrompida.

No caso de ser seleccionada a opção *LIST_MESS*, o servidor irá aceder ao ficheiro específico do cliente. Vai percorre-lo e enviar ao cliente todos os emails não lidos, que estão identificados com uma flag '+'. Depois de cada envio, vai alterar a flag do email agora lido para '-', para que estes sejam depois identificados como já lidos.

A opção *LIST_USERS*, o servidor irá aceder ao ficheiro "clientes.aut.txt" e irá verificar se cada utilizador é um operador. No caso de o ser, o nome do cliente será enviado para o cliente que chamou a função.

A opção *SEND_MESS* irá verificar se o cliente é um operador e no caso de o ser, irá pedir o nome dos clientes para o qual o email será enviado sendo de seguida pedida a mensagem a ser enviada. Usando a função *strtok*, os vários destinatários serão separados, e para cada um deles, o servidor irá enviar o email caso o utilizador exista. É também adicionada no início a flag '+', para referencia de que este email ainda não foi lido.

No caso de ser seleccionada a opção *LIST_READ*, irá acontecer o mesmo que na opção *LIST_MESS* mas em vez de verificar se a flag é '+' verifica se a flag é '-', que é a flag utilizada para os emails já lidos.

Quando é escolhida a opção *REMOVE_MESS*, o servidor irá apresentar 3 opções: Apagar um email, apagar todos os emails lidos e apagar todos os emails.

No caso de ser selecionada a opção que permite apagar um email específico o servidor irá aceder ao ficheiro do cliente irá permitir que o cliente escolha o email que pretende apagar. De seguida irá ser criado um ficheiro para o qual serão copiados todos os emails exceto o email que se pretende apagar e o ficheiro original será eliminado, alterando de seguida o nome do novo ficheiro para o do ficheiro original.

No caso de ser selecionada a opção de apagar todos os emails lidos o servidor irá ao ficheiro do cliente e copiar para um novo ficheiro todos os emails que tenham a flag '+', ou seja, copia os emails ainda não lidos e de seguida o ficheiro original será eliminado, alterando de seguida o nome do novo ficheiro para o do ficheiro original.

No caso de ser selecionada a opção que apaga todos os emails será apenas criado um novo ficheiro, de seguida o ficheiro original é eliminado e por fim é alterado o nome do novo ficheiro para o nome do original.

Ao escolher a opção *CHANGE_PASSW* irá ser pedida ao utilizador uma nova password que irá ser enviada para o servidor. Este por sua vez irá percorrer o ficheiro *client.aut.txt* e irá criar um novo ficheiro onde irá ser escrito todas as informações dos clientes iguais, alterando apenas a password do utilizador que fez o pedido. No final apaga-se o ficheiro original e muda-se o nome ao novo ficheiro.

Quando é escolhida a opção *OPER*, as permissões do cliente serão alteradas e ele passará a ser considerado um operador. Se este já o for, é notificado de tal e nada acontece. Caso ainda não seja, irá ser criado um ficheiro idêntico ao original, apenas com a alteração da flag de operador do cliente que pediu a alteração

Ao escolher a opção *QUIT*, o cliente é desconectado do servidor.

Durante todas estas opções, a informação escrita e lida no ficheiro "cliente.aut.txt" é encriptada e desencriptada, respetivamente.

[cliente.c](#)

No ficheiro *cliente.c* é feita uma ligação entre o cliente e o server através de uma socket criada na inicialização do cliente. Para inicializar o cliente é necessário que seja recebido o porto do servidor, o login, a password e o host (que nesta situação é o localhost). O login e a password serão enviados ao servidor para verificar se o utilizador existe e se a password é a correta. Se o utilizador for válido, será apresentado o menu com as várias opções.

Para as opções, *LIST_MESS*, *LIST_USERS* e *LIST_READ*, as funções executadas pelo cliente apenas farão um *read()* das informações enviadas pelo servidor e de seguida mostrar essa informação, caso ela exista, e voltar ao menu.

No caso da opção *REMOVE_MESS* o utilizador irá escolher uma das 3 opções já referenciadas e caso escolha a primeira (apenas uma

mensagem) irá receber uma lista de todas as mensagens numeradas podendo depois escolher umas delas. Caso escolha algumas das restantes duas, não irá receber mais nada.

No caso da opção *OPER*, é apenas recebida por parte do cliente a confirmação de que a operação foi efetuada e é impressa no ecrã essa confirmação e de seguida o menu é voltado a ser apresentado para que seja escolhida uma nova opção.

Quando é escolhida a opção *SEND_MESS* vai ser verificado se o cliente tem permissão para enviar emails e, caso tenha, o cliente vai inserir o(s) login(s) do(s) utilizador(es) para que pretende que o email seja enviado. De seguida insere a mensagem que pretende enviar e essa informação é enviada ao servidor que conclui o processo.

Na utilização da opção *CHANGE_PASSW* o cliente envia a nova password que pretende ter ao server.

A opção *QUIT* termina a conexão entre o cliente e o servidor.

Execução

Para executar o programa utiliza-se a seguinte linha no terminal:

```
./server -p<port>
```

O argumento *port* refere-se ao número do porto que se pretende utilizar, tendo este de ser entre 1025 e 65000.

Conclusão

Apesar de todo o trabalho ser desenvolvido com relativa facilidade, não foi implementado o sistema de notificação de clientes pois tivemos dificuldades relativamente a como o fazer, visto que seria necessário recorrer a algumas matérias que não foram lecionadas nesta cadeira.