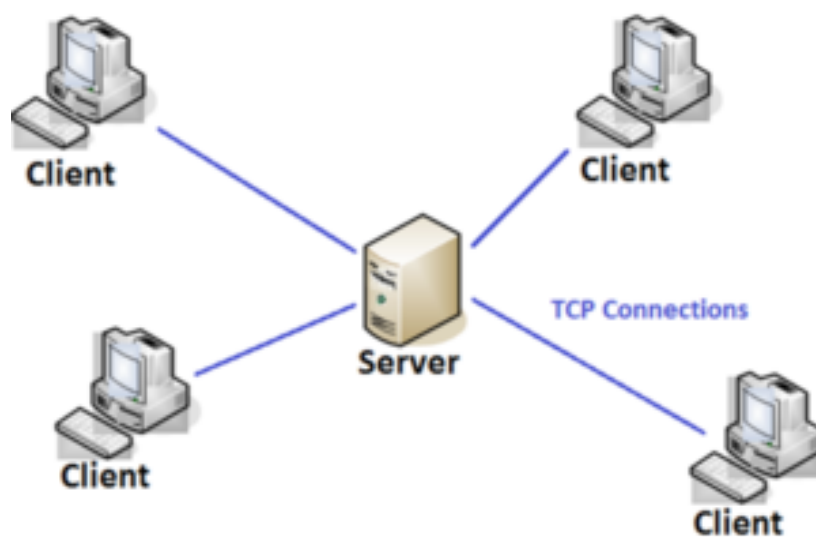


Relatório 2º Trabalho Prático

Redes de Computadores



Trabalho realizado por:
Diogo Carreiro nº48729
Rodrigo Alves nº48681

Docentes:
Professor Pedro Salgueiro
Professor Pedro Patinho



1. Introdução

Neste trabalho pretende-se desenvolver um sistema de comunicação para equipas de trabalho, em que é possível o envio de mensagens de um para um ou para vários clientes. Neste “*chat*” também existe a possibilidade da criação de equipas/grupos a partir das *tags* que são geradas pelos clientes que estão ligados ao servidor. Além disso entre clientes existe a possibilidade de haver transmissão de ficheiros entre os clientes, como também existe uma operação adicional que permite a visualização dos grupos em que cada cliente está inserido. O sistema assenta num protocolo de comunicação bem definido, de forma a possibilitar a interação entre clientes e servidores.

2. Comandos utilizados no trabalho

Quando existe a ligação de um cliente ao servidor este é recebido e aceite pelo mesmo e posteriormente é mostrado no “*chat*” todos os comandos disponíveis, bem como a forma como deve ser escrita a instrução pretendida pelo utilizador. O grupo decidiu adicionar esta informação ao trabalho, pois achou que iria facilitar o processo de troca de mensagens, como o funcionamento do “*chat*”. Os comandos apresentados no chat são os seguintes:

MENU

NICK <nickname> --> Atribui/altera o *nick* do utilizador

MSG <tag> <message> --> Envia a mensagem <message> para a tag <tag> (GLOBAL ou USER <NICKNAME>)

TAG <nickname> <tag>--> Adiciona a tag <tag> às tags do utilizador <nickname>

INFO --> Lista e mostra a informação dos utilizadores no chat(NICKNAME : TAG)

POST <tag> <message> -->Envia um *post* com a mensagem <message> para a tag <tag> (GLOBAL apenas)

READ <tag> --> Pede ao servidor a lista de posts já enviada para a tag <tag>

FILE <tag> <filename> <bytes> --> Inicia o envio do ficheiro <filename>, cujo tamanho é de <bytes> , para a tag <tag> (USER apenas)

EXIT --> Para o utilizador sair do *chat*;

3. Decisões tomadas ao longo do desenvolvimento do trabalho

Para este trabalho o grupo decidiu que cada cliente iria ser representado por um *array* de *structs* denominado list_c em que o seu tamanho máximo é de 10 utilizadores. Cada cliente irá ser definido por um *socket*, por um nome (*nick*), pela porta onde entrou, pelo numero de IP do servidor, por uma variável que contém o número de *tags* associado ao mesmo, um *array* de *strings* que guarda todas as *tags* que este está inserido, e por uma variável de controlo que verifica se um utilizador inseriu primeiramente o comando “NICK”.

O nome de utilizador pode conter no máximo 10 caracteres, cada *tag* não pode conter mais do que 25 caracteres, cada utilizador só pode conter 5 *tags*, e o número de *posts* que pode ser guardado no servidor é de 20.

4. Descrição do código

Ao ser inicializado o servidor pelo comando ./server 1234 este estará à escuta para possivelmente ser adicionado um novo utilizador ao chat. Feito isto inicializamos o nosso cliente a partir do comando ./client 1234.

Em primeiro lugar é mostrado o menu referido anteriormente com as informações acerca dos comandos disponíveis para usar no *chat*. De seguida é pedido ao utilizador que insira um nome que

o identifique (*nick_func*), esta operação pode ser realizada a partir do comando NICK seguido do *nickname*. Só após a mesma ser efetuada de forma correta, ou seja é criado um cliente com sucesso e adicionado ao servidor (*pushClient*), este utilizador terá permissão de utilizar o resto dos comandos. Na hipótese de o utilizador querer mudar a sua identidade, é possível fazê-lo inserindo novamente o comando NICK seguido do novo nome de utilizador que pretende. Se o utilizador pretender enviar uma mensagem, seja para todos, seja para uma *tag* em específico, é escrito o comando MSG seguido da *tag* e da mensagem. Posteriormente o programa irá seleccionar a *tag* e verificar quais são os utilizadores que a possuem, enviando assim a mensagem para eles (*message_func*). Como foi dito anteriormente que cada utilizador pode estar inserido numa ou mais *tags* é possível assim adicioná-lo a uma *tag* inserindo o comando TAG seguido do *nickname* e do nome da *tag* em que o utilizador pretenda ser adicionado (*addtag*).

No caso de o utilizador pretender a informação sobre todos os utilizadores ligados ao servidor, bem como as suas *tags*, basta introduzir o comando INFO no terminal. Assim o programa vai a cada *struct* correspondente a um dado utilizador e mostra todas as *tags* contidas na matriz de *tags* da *struct* (*user_tag*). Caso seja introduzido o comando POST seguido da *tag* GLOBAL e de uma mensagem o programa irá armazenar a mensagem introduzida pelo utilizador num *array* de *strings* (*add_post*). Se for chamado o comando READ seguido da *tag* GLOBAL irão ser mostradas no terminal todas as *strings* contidas no *array* referido no comando anterior (*read_all_posts*). No caso de ser chamado o comando FILE seguido da *tag* USER <*nickname*>, do nome do ficheiro, e dos *bytes* que representam o tamanho do ficheiro (este tamanho não pode ultrapassar os 256 *bytes*) é enviado o ficheiro ao destinatário (*send_file*). O grupo teve algumas dificuldades em relação a este comando, e por isso a chamada do mesmo efetua a leitura do ficheiro e, de seguida, o envio do conteúdo do mesmo para o destinatário definido no terminal.

Finalmente se um utilizador pretender sair do servidor basta escrever o comando EXIT. Este comando vai chamar a função *exit_func* e a função *popClient*.

5. Conclusão

Concluindo, neste trabalho, para além dos comandos que foram pedidos foram adicionadas duas extensões ao trabalho, uma delas sendo a possibilidade dos utilizadores serem inseridos em *tags* representantes de uma dada equipa, o que possibilita troca de mensagens de grupo para além da troca de mensagens privadas entre dois clientes. A outra extensão implementada foi o facto de todos os utilizadores terem acesso aos utilizadores que estão ligados ao servidor, assim como a todos os grupos a que eles pertencem.