

# Trabalho prático de Redes de Computadores

Universidade de Évora – Departamento de Informática 2020

O trabalho pressupõe o desenvolvimento de um cliente e servidor de acordo com o protocolo definido abaixo, e de interfaces nos clientes e no servidor. No entanto, o foco do trabalho não é o interface de utilizador, pelo que deve optar por soluções simples, e ser pragmático na sua implementação. Considere portanto que a prioridade é a implementação das funcionalidades do ponto de vista lógico devendo o interface de utilizador ser tão simples quanto possível desde que se possam testar as funcionalidades requeridas.

O servidor deverá gerir vários clientes em simultâneo usando **select** (sem utilizar fork ou threads) e será responsável pela gestão de: receção e re-envio das **mensagens**; dos **canais**; e dos dados do **utilizadores**, enquanto os clientes poderão inserir, e ler mensagens, e alterar os dados na sua conta/perfil.

O servidor tem vários canais. Este terá um canal "default", e vários canais temáticos (considere por exemplo pelo menos mais dois canais o canal "cn" (computer networks) e o canal "oss" (open source software)

Os clientes podem funcionar em dois modos: o **modo registado** e **não registado**, em qualquer um deles é obrigatório ter um nickname/nome definido (que funcionará como um username visível e que deve ser único no servidor). No modo não registado o cliente funciona apenas no canal default, enquanto no modo registado inicia no canal preferido desse utilizador, que o próprio utilizador pode alterar. Os utilizadores registados podem ter a categoria especial de **operador** que permite gerir canais e gerir outros utilizadores.

## 1. Protocolo de comunicação

A comunicação entre clientes e servidor deve obedecer a um protocolo de texto simples (plain text single line protocol), ou seja um protocolo baseado em mensagens de texto legíveis por um humano, por oposição a protocolos binários (que são mais eficientes do ponto de vista da comunicação apesar de ilegíveis).

Cada mensagem tem um comprimento máximo de 512 bytes; esta é iniciada por um comando com 4 caracteres seguida de um ou mais parâmetros separados por espaços em branco. Uma mensagem pode dar origem (ou não) a uma resposta do servidor.

### Comandos de utilizador (todos os utilizadores):

**NICK** nome - atribui ou muda o nome/nickname com um máximo de 9 caracteres.

**MSSG** mensagem - envia uma mensagem para o canal ativo (a mensagem re-enviada pelo servidor deve ser composta pela mensagem original precedida da identificação de quem a enviou).

### Comandos de utilizador registado (pode ser operador ou não):

**PASS** password - autenticação de um utilizador registado.

**JOIN** canal - muda o canal ativo.

**LIST** - lista os canais existentes.

**WHOS** - lista e mostra informação dos utilizadores conectados no canal.

### **Comandos de Operador:**

**KICK** nome - expulsa o utilizador (remove da lista dos utilizadores registados).

**REGS** nome password - regista o utilizador (inclui na lista dos utilizadores registados).

**OPER** nome - promove o utilizador (registado) à categoria de operador.

**QUIT** - desiste de ser operador.

### **Mensagens do servidor**

**RPLY** código (código de resposta) / [listas] - respostas a comandos.

**MSSG** "origem nome/categoria:> mensagem" - outras mensagens.

### **Descrição detalhada**

O protocolo consiste nas seguintes mensagens.

Pedido do cliente:

#### **NICK <nome>**

Atribui ou muda o nome/nickname com um máximo de 9 caracteres (apenas letras ASCII e algarismos). Numa nova ligação a definição do nome é obrigatoriamente o primeiro pedido que o cliente tem de fazer. Só após a definição do nome é que o cliente recebe e envia mensagens.

Respostas do servidor:

RPLY 001 - Nome atribuído com sucesso

RPLY 002 - Erro: Falta introdução do nome.

RPLY 003 - Erro: Nome pedido não válido (caso não tenha apenas letras e algarismos e 9 caracteres no máximo)

RPLY 004 - Erro: nome já em uso (num outro utilizador registado ou em uso por um utilizador não registado, e o comando não tem qualquer efeito

Mensagens do servidor enviadas a todos os clientes ativos no mesmo canal do cliente que faz o pedido,

MSSG "server :> novo utilizador <nome>"

se for um utilizador novo, ou

MSSG "server :> <nome\_antigo> mudou o seu nome para <nome>"

se já tinha nome atribuído.

Pedido do cliente:

#### **MSSG <mensagem>**

Envia uma mensagem para o canal ativo (a mensagem será re-enviada pelo servidor, composta pela mensagem original precedida da identificação de quem a enviou) O total do comando e da mensagem original não pode exceder os 512 bytes.

Respostas do servidor:

RPLY 101 - mensagem enviada com sucesso.

RPLY 102 - Erro. Não há texto na mensagem.

RPLY 103 - Erro. Mensagem demasiado longa.

Mensagem do servidor enviada a todos os clientes:

MSSG "<nome\_da\_origem/categoria> :> mensagem\_original"

Pedido do cliente:

### **PASS <password>**

Autenticação de um utilizador registado (só após o nome estar definido) password com um máximo de 9 caracteres

Respostas do servidor:

RPLY 201 - Autenticação com sucesso.

RPLY 202 - Erro. Nome não está definido.

RPLY 203 - Erro. Password incorreta.

Pedido do cliente:

### **JOIN <canal>**

Muda o canal ativo para "canal"

Respostas do servidor:

RPLY 301 - Mudança de canal com sucesso

RPLY 302 – Erro. canal não existente

RPLY 303 - Erro. Não pode mudar para o canal (utilizador não autenticado)

Mensagem do servidor enviada a todos os clientes ativos no <canal>:

MSSG "server :> <nome> entrou neste canal"

Mensagem do servidor enviada a todos os clientes ativos no <canal\_anterior>:

MSSG "server :> <nome> deixou este canal"

Pedido do cliente:

### **LIST**

Lista os canais existentes

Resposta do servidor:

RPLY 401 <canal1>; <canal2>; ... <canalN>

Pedido do cliente:

### **WHOS**

Lista e mostra informação dos utilizadores conectados no canal ativo

Resposta do servidor:

RPLY 501 <user1/tipo1/sign\_on\_time1> <user2/tipo2/sign\_on\_time2>;  
...<userN/tipoN/sign\_on\_timeN>

Pedido do cliente:

**KICK <nome>**

Operador expulsa o utilizador <nome> (remove da lista dos utilizadores registados)

Respostas do servidor:

RPLY 601 – Utilizador expulso.

RPLY 602 – Erro. Ação não autorizada, utilizador cliente não é um operador.

Mensagem do servidor enviada a todos os clientes:

MSSG "server :> <nome> foi expulso"

Pedido do cliente:

**REGS <nome> <password>**

Regista o utilizador (inclui na lista dos utilizadores registados)

Respostas do servidor:

RPLY 701 – Utilizador foi registado com sucesso.

RPLY 702 – Erro. Ação não autorizada, utilizador cliente não é um operador.

Mensagem do servidor enviada a todos os clientes:

MSSG "server :> <nome> foi registado"

Pedido do cliente:

**OPER <nome>**

Promove o utilizador <nome> (registado) à categoria de operador.

Respostas do servidor:

RPLY 801 – Foi promovido a operador.

RPLY 802 – Erro. Ação não autorizada, utilizador cliente não é um operador.

RPLY 803 – Erro. Ação não autorizada, utilizador cliente não está autenticado.

RPLY 804 – Erro. Ação não autorizada, utilizador <nome> não é um utilizador registado.

Mensagem do servidor enviada a todos os clientes:

MSSG "server :> <nome> foi promovido a operador"

Pedido do cliente:

**QUIT**

Desiste de ser operador

Respostas do servidor:

RPLY 901 – Deixou de ser operador.

RPLY 902 – Erro. Ação não autorizada, utilizador cliente não é um operador.

Mensagem do servidor enviada a todos os clientes:

MSSG "server :> <nome> deixou de ser operador

## 2. Interface

Considere uma interface simples que permita, nos clientes:

- a) visualizar o fluxo de mensagens, e
- b) dar os comandos definidos acima (e.g. linha de comando, ou um sistema menus/opções)

## 3. Sistema base a implementar [8 valores]

O servidor deve aceitar ligações de vários clientes através do porto TCP 5555, e o sistema implementar os comandos **NICK** e **MSSG**.

## 4. Registo de clientes/autenticação [4 valores]

O servidor deve possuir um sistema para guardar em disco os dados dos clientes registados (nome e password, e categoria). O sistema deve implementar os comandos **PASS**, e **JOIN**.

## 5. Gestão de utilizadores registados [3 valores]

O servidor deve fazer a gestão de utilizadores registados (criar e/ou remover) por um operador. O sistema deve implementar os comandos **REGS**, e **KICK**, (note que o sistema deve ser inicializado com um operador por default, por exemplo “admin” com password “admin”).

## 6. Gestão de utilizadores operadores [3 valores]

O servidor deve conceder poder de gestão de operadores (promover e/ou desistir) aos operadores existentes. O sistema deve implementar os comandos **OPER**, e **QUIT**.

## 7. Correção de erros [2 valores]

Considere que a comunicação não é fiável e existe uma pequena probabilidade de ocorrerem erros no envio que não são detetados e corrigidos nos níveis inferiores da hierarquia de software de redes. Implemente uma variante do cliente e servidor (client-ham.c e server-ham.c) com um sistema de correção de erros usando código de Hamming (4 bits de dados para 3 checkbits).

Considere a função que simula a ocorrência de erros no envio, e que já foi usada anteriormente:

```
void canal( char* in, char* out ){
    int i = 1, erro=0;
    while (*in != '\0')
        {
            erro=0;
            i=(i+5)%7;
            printf(" %i ",i);
```

```
        if(i>3){erro=0;} else{erro=pow(2,i);} //função que gera os erros
        *out = (*in)^(char)(erro);
        out++;
        in++;
        i++;
    }
    *out = '\0'; }
```

Note que nesta variante o protocolo deixa de ser um protocolo “plain text”.

### **Prazo e forma de entrega**

O trabalho deve ser realizado em grupos de (no máximo) dois alunos. A entrega será feita através do moodle, na área reservada para o efeito, até dia 6/02/2021 às 23.59h.

No ficheiro (.zip) entregue devem constar o código-fonte do servidor e do cliente, (server.c e client.c) bem como um relatório (.txt .odt ou pdf) de implementação, onde serão descritas as opções tomadas no desenvolvimento do trabalho, e as opções necessárias para a compilação no gcc, e teste do software.

O nome do ficheiro deve ser no formato "redes-12345-67890.zip", onde 12345 e 67890 são os números de aluno dos elementos do grupo. Ao descompactar, o conteúdo do trabalho deve ficar numa directoria chamada "redes-12345-67890/".

*Nota: O presente enunciado poderá ser atualizado, de forma a incluir esclarecimentos adicionais.*