

Relatório - 1º Trabalho de Redes de Computadores

Daniel Barreiros 48452, Tomas Antunes 48511

Universidade de Évora - Engenharia Informática

2021/2022

Ficheiro server.c

Variáveis Globais:

PORT 1234 – Define a porta na qual o cliente se vai ligar.

BUFSIZE – Define o tamanho máximo do input realizado pelo utilizador.

MAXCLIENTS – Define o número máximo de clientes

Função `int valueinarray(char* val, char* arr[]):`

- Esta função verifica se o valor “val” se encontra no array “arr”.

```
// verifica se val está no array arr[]
int valueinarray(char* val, char* arr[]) {
    int i;
    for(i = 0; i < MAXCLIENTS; i++)
    {
        if(!strcmp(val, arr[i])) {
            return 1;
        }
    }

    return 0;
}
```

Função `void sendToAll(char* message, int index):`

- Esta função envia a mensagem escrita por um cliente para todos os clientes existentes.

- Exclui o cliente que enviou a mensagem.

- Através de um ciclo While, envia cliente a cliente a mensagem.

- Dentro deste ciclo while, é realizada uma verificação através de um if para verificar a qual cliente não se vai enviar a mensagem. Caso este cliente seja o cliente que a escreveu, realiza a iteração mas não envia a mensagem. Desta forma, o cliente que envia não recebe a sua própria mensagem.

```
// envia uma mensagem a todos os utilizadores excluindo aquele que enviou
void sendToAll(char* message, int index) {
    int i = 0;
    while(clients[i] != "0") {
        if(i != index)
            write(i + 4, message, strlen(message)+1);
        i++;
    }
}
```

Função `void sendToUser(char* user, char* message):`

- É realizada uma comparação entre o argumento “user” recebido pela função e a lista de clientes de forma a encontrar o cliente ao qual se pretende enviar a mensagem.
- Esta verificação é feita através de um ciclo while que corre o array dos clientes e realiza esta comparação.

```
// envia uma mensagem para o utilizador <user>
void sendToUser(char* user, char* message) {
    int i = 0;
    while(clients[i] != "0") {
        if(!strcmp(clients[i], user))
            break;
        i++;
    }
    write(i + 4, message, strlen(message));
}
```

Função `int process_client(int sock, char *clients[]):`

- De uma forma geral, esta função processa o input do utilizador.
- Através de um “if” é verificado se o cliente em questão se encontra ligado ao server.
- Através de um “if” é verificado se o cliente já tem nome, e caso não tenha, define o nome do utilizador.
- Através de um “if” é filtrado o input do utilizador de forma a perceber se a mensagem é para ser enviada para todos os clientes, se só para um cliente.
- No caso de a mensagem ter como destino um único cliente, através de um while é realizada a contagem do número de caracteres do nome do utilizador para o qual vai ser enviada a mensagem de forma que este nome possa ser apagado do input introduzido.
- Depois é acrescentado à mensagem que vai ser enviada, o nome do cliente que escreveu a mensagem. Desta forma, o cliente que recebe a mensagem vai visualizar o nome de quem a escreveu e a respetiva mensagem.
- No caso de a mensagem ter como destino todos os clientes existentes no server, é enviada a mensagem escrita pelo utilizador para a função sendToAll, onde posteriormente vai ser enviada.

```

int process_client(int sock, char *clients[]) {
    int n;
    int index = sock - 4;
    char buf[BUFSIZE];

    n = read(sock, buf, BUFSIZE);

    if (n <= 0) {
        return 0; /* client closed socket */
    }

    buf[n] = '\0';

    if (clients[index] == "0") {
        clients[index] = strdup(buf);
        return 1;
    }

    char* user = strtok(strdup(buf), " "); // retira do buf o nome do utilizador a ser enviada a mensagem

    if (user[0] == '-') {
        // mensagem para outro cliente

        user++; // remover primeiro caracter da string user
        if (!valueinarray(user, clients)) // se o cliente nao existir
            return 0;

        printf("%s\n", buf);

        char message[BUFSIZE] = "-";

        // adicionar o nome do utilizador que enviou
        strcat(message, clients[index]);
        strcat(message, " ");

        int sz = 0;
        int j = 0;

        char* aux = strdup(buf);

        while (user[sz] != '\0') { // tamanho do nome
            printf("%c\n", user[sz]);
            sz++;
        }
        aux += sz + 2; // remover -<user> da mensagem

        // junção da mensagem a ser enviada com o nome de utilizador de quem enviou
        strcat(message, strdup(aux));
        sendToUser(user, message);
    } else if (user[0] == '+') {
        // mensagem para todos
        printf("%s \n", buf);

        char* message = strdup(buf);
        sendToAll(message, index);
    }

    return 1;
}

```

Função `int main(int argc, char const *argv[]):`

- Utilizamos a função main existente num dos ficheiros das aulas práticas com umas pequenas alterações.
- Inicializamos o array dos clientes (`clientes[]`) com todas as posições a “0”.
- De forma a enviar para todos os utilizadores a informação de que um cliente se desconectou do servidor, dentro do while principal, utilizámos um if já existente para construir a frase que vai ser enviada a todos os clientes e é enviada posteriormente pela função `sendToAll`.

Ficheiro cliente.c:

- De uma forma geral este ficheiro permite criar o cliente, conectar o cliente na porta correta do servidor, processar o input do utilizador e receber as mensagens que vêm do servidor.