

# Projeto Introdução às Redes de Comunicação

## Relatório



UNIVERSIDADE D  
COIMBRA



Trabalho realizado por:

- Alexandre Cortez dos Santos – 2018277730
- Guilherme Eusébio Reis Gaspar – 2018278327

# Manual do utilizador

1. No Ubuntu/Lubuntu, inicie num terminal o executável server, que foi compilado através do ficheiro server.c com “gcc -Wall server.c -o server”, da seguinte forma: “./server 9050 10”, sendo 9050 o porto do servidor e 10 o nº de clientes que o server pode receber.
2. Abra outro terminal, inicie o executável ircproxy que foi compilado através do ficheiro ircproxy.c com “gcc -Wall ircproxy.c -o ircproxy”, da seguinte forma “./ircproxy 9050”, sendo 9050 o porto do servidor e também da proxy. Este porto terá de ser sempre igual ao que foi inserido na chamada do executável do servidor.
3. Abra outro terminal, inicie o executável ircproxy que foi compilado através do ficheiro client.c com “gcc -Wall client.c -o client”, da seguinte forma “./client 127.0.0.2 127.0.0.1 9020 tcp”, sendo 127.0.0.2 o endereço da proxy, 127.0.0.1 o endereço do server, 9020 o porto do cliente e tcp o protocolo inicial de comunicação.
4. O servidor ficará à espera de um comando vindo do cliente através da proxy. A proxy está preparada para executar o chat de comunicação entre o servidor e o cliente. O cliente terá de inserir um de três comandos a enviar ao servidor (“LIST”, “DOWNLOAD <TCP7UDP> <ENC/NOR> <nome>”, “QUIT”), sendo que a proxy só enviará ao servidor os comandos LIST e DOWNLOAD.
5. Se o cliente enviar ao server o comando “LIST”, o server vai devolver ao cliente a lista de todos os ficheiros disponíveis na diretoria “server\_files”, isto tudo sempre com a proxy como intermediária.
6. O comando DOWNLOAD é um pouco mais complexo. O cliente tem a opção de escolher o protocolo que irá ser utilizado para a transferência do seu download, se quer que o ficheiro seja transferido com recurso à encriptação e por fim, o nome do ficheiro que pretende transferir. Após a transferência ter sido realizada, o cliente irá imprimir o nome do ficheiro recebido, o total de bytes transferidos, o protocolo utilizado para a transferência e o tempo total de download.
7. Com o QUIT, o cliente irá ser desconectado do servidor.

## Explicação do código

Para este projeto foi-nos pedido que implementássemos um sistema cliente-servidor, com uma proxy como mediadora, para a transferência de ficheiros entre dois sistemas, recorrendo aos protocolos TCP ou UDP. A encriptação dos ficheiros transferidos deverá ser possível e o servidor e o cliente devem autenticar-se um ao outro.

**Server:** No servidor abrimos um socket TCP que vai fazer comunicação com a proxy e receber os comandos enviados pelo cliente através da proxy, com recurso a reads e writes, tal como nos foi introduzido na Ficha 03.

Se o servidor receber o comando “LIST”, vai abrir a diretoria dos server\_files e listá-los todos no buffer, enviando esse buffer de volta para a proxy que, por sua vez, vai enviar para o cliente.

Ao receber o comando “DOWNLOAD <TCP/UDP> <ENC/NOR> <nome>”, o servidor com auxílio do strtok vai dividir os diferentes argumentos e direcioná-los para o que fazer através de if's. Encripta o ficheiro dependendo do que o cliente pediu e envia através de um buffer para a proxy a informação do ficheiro pedido.

**Client:** No cliente abrimos um socket TCP que vai comunicar com a proxy. O cliente prepara-se para receber um comando e envia-o para a proxy.

Se o cliente enviar o comando “LIST”, prepara-se para receber um buffer da proxy com a lista dos ficheiros presentes na diretoria. Quando o recebe, imprime-os na consola e volta a fazer o pedido ao cliente para inserir um comando.

Se o comando enviado for “DOWNLOAD <TCP/UDP> <ENC/NOR> <nome>”, através de if's, o cliente vai-se preparar para receber o ficheiro por tcp/udp (definido pelos if's), encriptado ou não (definido por if's) e se do tipo txt, jpg ou wav (definido por if's também), criando assim um novo ficheiro com a informação que foi enviada pelo servidor através da proxy. Ao receber o ficheiro, vai verificar se a diretoria “downloads” existe, para armazenar lá o ficheiro.

Se enviar o comando “QUIT”, a consola irá imprimir “Bye bye” e o socket será fechado, saindo com exit(0);

**Proxy:** Na proxy será aberto um socket TCP que vai ficar à espera de mensagens por parte do cliente. Quando o socket fica no modo “accept” para receber os comandos, é utilizada como argumento para a função process\_chat. Nesta função, tal como o nome indica, vai ser um chat entre o servidor e o cliente, recebendo os pedidos do cliente para o servidor e a respetiva resposta do servidor.