



FEUP **FACULDADE DE ENGENHARIA**
UNIVERSIDADE DO PORTO

Redes de computadores

2º trabalho prático

Licenciatura em Engenharia Informática e Computação

Diogo Fernandes (202108752)
José Sereno (202108729)

Dezembro 2023

Contents

1	Introdução	1
2	Desenvolvimento da aplicação	1
2.1	Estrutura do código	1
2.2	Fluxo do programa	1
3	Configurar e analisar o funcionamento de uma rede	2
3.1	Exp 1 - Configurar o IP de uma rede	2
3.2	Exp 4 - Configuração de um Router Comercial e Implementação de NAT	2

1 Introdução

Objetivos Este projeto teve dois objetivos:

1. Desenvolver uma aplicação em **C** que implemente o protocolo **FTP** - descrito no **RFC959** - para fazer o download de um ficheiro através de um **URL** - a sintaxe deste URL deveria seguir o **RFC1738**.
2. Configurar e analisar o funcionamento de uma rede de computadores.

2 Desenvolvimento da aplicação

O programa **download** foi desenvolvido em **C** e tem como objetivo fazer o download de um ficheiro através de um **URL** que segue a sintaxe do **RFC1738**. Através da realização deste programa aprofundamos os nossos conhecimentos sobre os seguintes temas:

Client-Server & TCP/IP Aprendemos como se dá a comunicação entre um cliente e um servidor através do protocolo **TCP/IP**.

RFCs RFCs são documentos que descrevem os padrões da internet. Consultámos, como referido anteriormente, o **RFC959** e o **RFC1738**, para além de outras documentações, para entender o funcionamento do protocolo **FTP** e da sintaxe do **URL**.

Sockets Aprendemos a usar **sockets** em **C** como forma de comunicação entre o cliente e o servidor.

DNS (Domain Name System) Procurámos entender o funcionamento do **DNS** e como é que este é usado para traduzir um **URL** num endereço **IP**.

UNIX Aprendemos alguns comandos de **UNIX** que usamos no nosso programa (ex.: **getaddrinfo**, **socket**, **connect**, **recv**, **send**)

2.1 Estrutura do código

O código do programa **download** está dividido em 3 ficheiros: **download.c**, **server.c** e **url.h**. É no ficheiro **download.c** onde se encontra a função **main** - onde se percebe o fluxo do programa. O ficheiro **server.c** contém as funções que implementam o protocolo **FTP** e o ficheiro **url.h** contém as funções que permitem fazer o **parse** do **URL**. O programa **download** é compilado através do comando **make** e é executado da seguinte forma:

2.2 Fluxo do programa

O programa é executado do seguinte comando:

```
./download ftp://[<user>:<password>@]<host>
>/<url-path>
```

O fluxo do programa será o seguinte:

1. **Parse** do **URL** para obter os seguintes campos: **user** (opcional), **password** (opcional), **host**, **port** (opcional, é usado o valor 21 por default no protocolo FTP) e **url-path**.
2. Criação de uma **socket** que, através de uma ligação TCP/IP, inicia uma conexão com o **host** na porta **port**.
3. Envio dos comandos **USER** e **PASS** para o servidor juntos dos valores **user** e **password**, respetivamente. Caso estes valores não tenham sido especificados, usa-se o valor **anonymous**.
4. Envio do comando **PASV** para o servidor para que este abra uma porta à qual nos conectaremos para futuramente recebermos o ficheiro.
5. Criação de uma nova **socket** que, através de uma ligação TCP/IP, inicia uma conexão com a nova porta aberta pelo servidor.
6. Envio do comando **RETR** através da porta inicial para o servidor junto do valor **url-path** para que o servidor nos envie o recurso especificado.
7. Leitura do ficheiro através da **socket** criada no ponto 5 e escrita do mesmo para um ficheiro local.
8. Fecho das **sockets** criadas.

O `parse` do `URL` é feito a partir da função `parse_url` que recebe uma `string` e devolve uma `struct URL` com os campos especificados no ponto 1. Esta função usa uma máquina de estados e expressões regulares.

Todo o processo de comunicação entre o cliente e o servidor é feito através de `sockets` e o protocolo usado é o `TCP/IP`. A criação das `sockets` é feita através da função `getaddrinfo` que recebe o `hostname` e a `porta` e devolve uma `struct addrinfo` com os campos necessários para a criação da `socket`.

As `sockets` são criadas através da função `socket` e são feitas as respetivas ligações através da função `connect`.

O envio de comandos pelas `sockets` é feito através da função `send` e a leitura das respostas do servidor é feita através da função `recv`.

Antes de ser enviado qualquer comando para o servidor, primeiro é feita a leitura do código de status do servidor através da função `recv`, de modo a termos uma noção do estado do servidor.

A escrita do ficheiro no disco é feita através da função `write`.

3 Configurar e analisar o funcionamento de uma rede

O objetivo deste conjunto de experiências é configurar uma rede de computadores de modo a que estes tenham acesso à internet para instalar ficheiros a partir de um servidor remoto usando o protocolo FTP desenvolvido ou seja, a aplicação.

3.1 Exp 1 - Configurar o IP de uma rede

O objetivo desta experiência é conectar o Tux-3 ao Tux-4 a partir do MicroTik Switch.

3.2 Exp 4 - Configuração de um Router Comercial e Implementação de NAT