Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Grado en Ingeniería Informática

Redes y Sistemas Distribuidos

Curso 2023-2024

Report

FTP Server Application

Valeria Bosch Pérez
(alu0101485287@ull.edu.es)

Miguel Luna García
(alu0101448251@ull.edu.es)

Index:

1. Application Description.

The objective of this report is to provide an application description for a simple FTP server implementation. The FTP (File Transfer Protocol) server allows users to upload and download files from a designated folder. The server will be built based on the TCP transport protocol, utilizing the well-known ports 20 and 21.

2. Protocol Description.

The **File Transfer Protocol** is specifically designed for transferring files between a client and a server. Its purpose is to enable the storage and retrieval of files on the server. The FTP protocol operates on two TCP connections: the control connection (port 21) and the data connection (port 20). The control connection is responsible for transmitting commands and their corresponding replies. It is important to note that the term "command" in this context refers to the textual commands defined in the FTP protocol, not the commands entered by the user in the FTP client program's command line.

There are two distinct file transfer modes available:

Active mode: In this mode, the client initiates a connection from an unprivileged port N to the server's command port (port 21). Afterward, the client listens on port M and sends the FTP command "PORT M" to the FTP server. The server responds by connecting back from port 20 to the client's port M.
One limitation of the active mode is that it requires the server to establish a connection back to the client, which can pose challenges when firewalls drop incoming connections from unknown sources.

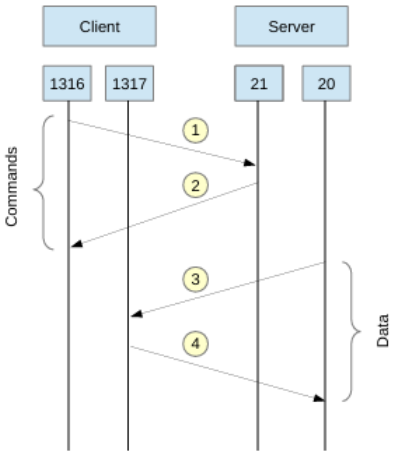

Figure 1: Active mode.

Passive mode: To overcome the limitations of the active mode, FTP clients also support passive mode. In passive mode, the client initiates both the control and data connections. When establishing an FTP connection, the client opens two random unprivileged ports locally. The first port contacts the server on port 21. Subsequently, the client issues a PASV command, and the server responds with a port number. The client then connects to that specified port for data transfer.

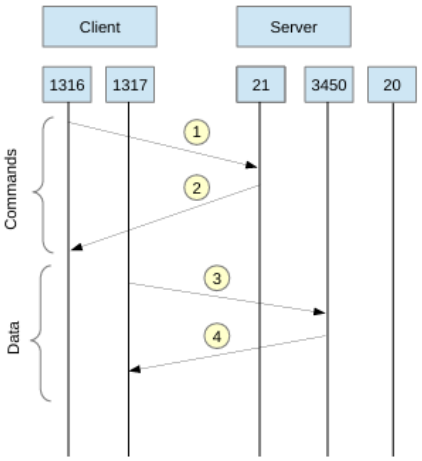

Figure 2: Passive mode.

3. Compilation and Execution Guide.

For the compilation and execution of the program, you can read the **README.md** file in the source code. This contains a mini guide of what you should do.
If not, here's a tutorial anyway:

Before compiling and running, please install Make, gcc and g++so it can work fine :)

How to Compile and Run the server
$ make clean
$ make all
$ ./ftp_server

4. Test Cases.

4.1. Open a local connection

```
FTP-Server on  main [✗]
❯ ftp -d
ftp> open localhost 2121
setupsockbufsize: rcvbuf_size determined as 131072
setupsockbufsize: sndbuf_size determined as 16384
Connected to localhost.
220 Service ready
ftp_login: user `<null>' pass `<null>' host `localhost'
Name (localhost:valeria):
---> USER valeria
331 User name okay, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
```

4.2. Get a file in active mode

```
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> get README
local: README remote: README
---> TYPE I
200 OK
---> SIZE README
213 535 bytes
ftp: setsockopt SO_DEBUG (ignored): Permiso denegado
---> EPRT |1|127.0.0.1|46041|
502 Command not implemented.
disabling epsv4 for this connection
---> PORT 127,0,0,1,179,217
200 PORT command successful.
---> RETR README
150 File status okay; about to open data connection.
100% |**********************************|   535        8.50 MiB/s    00:00 ETA
226 Closing data connection. Requested file action successful.
535 bytes received in 00:00 (2.84 MiB/s)
```

4.3. Get a file in passive mode

```
Passive mode: on; fallback to active mode: on.
ftp> get README
local: README remote: README
---> TYPE I
200 OK
---> SIZE README
213 535 bytes
ftp: setsockopt SO_DEBUG (ignored): Permiso denegado
---> EPSV
502 Command not implemented.
disabling epsv4 for this connection
---> PASV
227 Entering Passive Mode (127,0,0,1,229,161).
---> RETR README
150 File status okay; about to open data connection.
100% |**********************************|   535       11.86 MiB/s    00:00 ETA
226 Closing data connection. Requested file action successful.
535 bytes received in 00:00 (2.72 MiB/s)
```

4.4 Put a file in active mode

```
ftp> put README
local: README remote: README
ftp: setsockopt SO_DEBUG (ignored): Permiso denegado
---> PORT 127,0,0,1,218,187
200 PORT command successful.
---> STOR README
125 Data connection already open; transfer starting.
100% |****************************************|   535       10.85 MiB/s    00:00 ETA
226 Closing data connection. Requested file action successful.
535 bytes sent in 00:00 (12.38 KiB/s)
```

4.5 Put a file in passive mode

```
ftp> passive
Passive mode: on; fallback to active mode: on.
ftp> put README
local: README remote: README
ftp: setsockopt SO_DEBUG (ignored): Permiso denegado
---> PASV
227 Entering Passive Mode (127,0,0,1,186,87).
---> STOR README
125 Data connection already open; transfer starting.
100% |****************************************|   535        7.84 MiB/s    00:00 ETA
226 Closing data connection. Requested file action successful.
535 bytes sent in 00:00 (12.03 KiB/s)
```

4.6 ls in active mode

```
ftp> ls
ftp: setsockopt SO_DEBUG (ignored): Permiso denegado
---> EPRT |1|127.0.0.1|38799|
502 Command not implemented.
disabling epsv4 for this connection
---> PORT 127,0,0,1,151,143
200 PORT command successful.
---> LIST
150 File status okay; about to open data connection.
FTPServer.cpp
ClientConnection.h
Makefile
.
common.h
README
FTPServer.h
ClientConnection.cpp
ftp_server
ftp_server.cpp
..
226 Closing data connection. Requested file action successful.
```

4.7 ls in passive mode

```
ftp> passive
Passive mode: on; fallback to active mode: on.
ftp> ls
ftp: setsockopt SO_DEBUG (ignored): Permiso denegado
---> PASV
227 Entering Passive Mode (127,0,0,1,215,103).
---> LIST
150 File status okay; about to open data connection.
FTPServer.cpp
ClientConnection.h
Makefile
.
common.h
README
FTPServer.h
ClientConnection.cpp
ftp_server
ftp_server.cpp
..
226 Closing data connection. Requested file action successful.
```

5. Source Code.

Google Drive:

- 🗀 FTP-Server
- FTP-Server