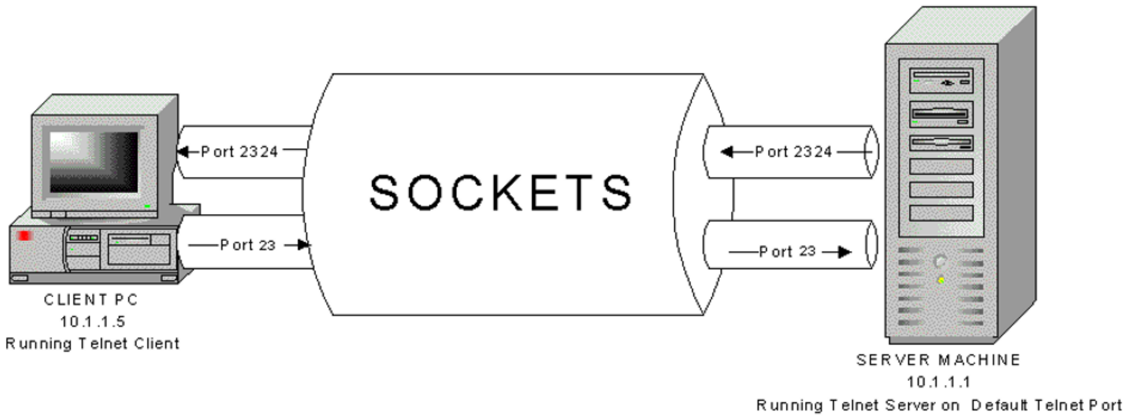
Redes y Sistemas Distribuidos

Entregable 3

Programación de sockets



Universidad de La Laguna

Escuela Superior de Ingeniería y Tecnología

Grado en Ingeniería Informática

Curso 2021-2022

26 may 2022

José Lozano Armas

Rebeca Rodríguez Rodríguez

Index

Introduction

- ### Developed application

The objective of this application is to replicate how the FTP protocol works by using two terminals as client and server, where the server provides all the resources, and the client uses commands to get all these resources.

- ### Developed protocol

To implement the application, we've used FTP (File Transfer Protocol) **[1]**. This protocol is used to transfer files between a client and a server to store or retrieve them from the server.

This protocol is based on the TCP protocol (Transmission Control Protocol) **[2]** and uses the 20 and 21 ports, which are its default ports, for the two connections involved: the control connection (that uses the port 21) and the data connection (which uses the port 20).

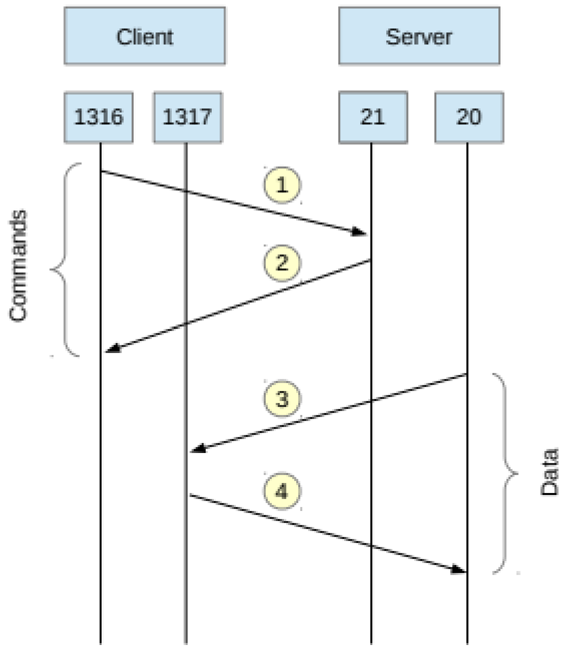
The control connection (which uses the port 21) transfers commands and replies to them. In this context, "command" refers to the textual commands defined in the FTP protocol **[3]**, not the ones that the user types into the command line of the FTP client program.

While using this protocol, there are two different types of transfer modes:

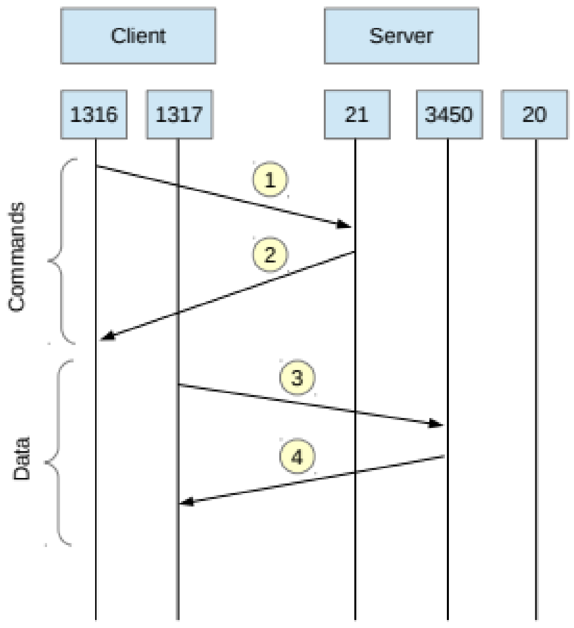
- **Active mode:** the client is connected to a non privileged port N to the server command port (port 21). Then, the client starts listening to port M and sends the FTP command PORT M to the FTP server, which will then connect back from port 20 to port M of the client.

  There's a problem with this procedure, which consists on the server connecting back to the client. This often causes the firewalls to drop unknown incoming connections. To solve this issue, there's the passive mode.

- **Passive mode:** the client initiates both connections (control and data). When opening an FTP connection, the client opens two random non privileged ports locally. The first port contacts the server on port 21, and then the client issues a PASV command and the server responds with a port number. After that, the client connects to that port to transfer the data.



Active mode



Passive mode

Guide of compilation

- ### Getting started

In order to make good use of the program, first of all it is essential to compile the code in one directory, and use the command "ftp -d" for working as a client in another. With the aim of avoiding problems when we are going to act as client and server. Especially if we are sending and receiving files.

- ### Using the server mode

Now that we've made all the compilation, we can start to use our program. To begin with the terminal that will act as server, the only thing that you need to do is activate the executable, called "ftp_server". Using the command ./ftp_server, then this terminal will stay in a loop waiting for the client to send its commands.

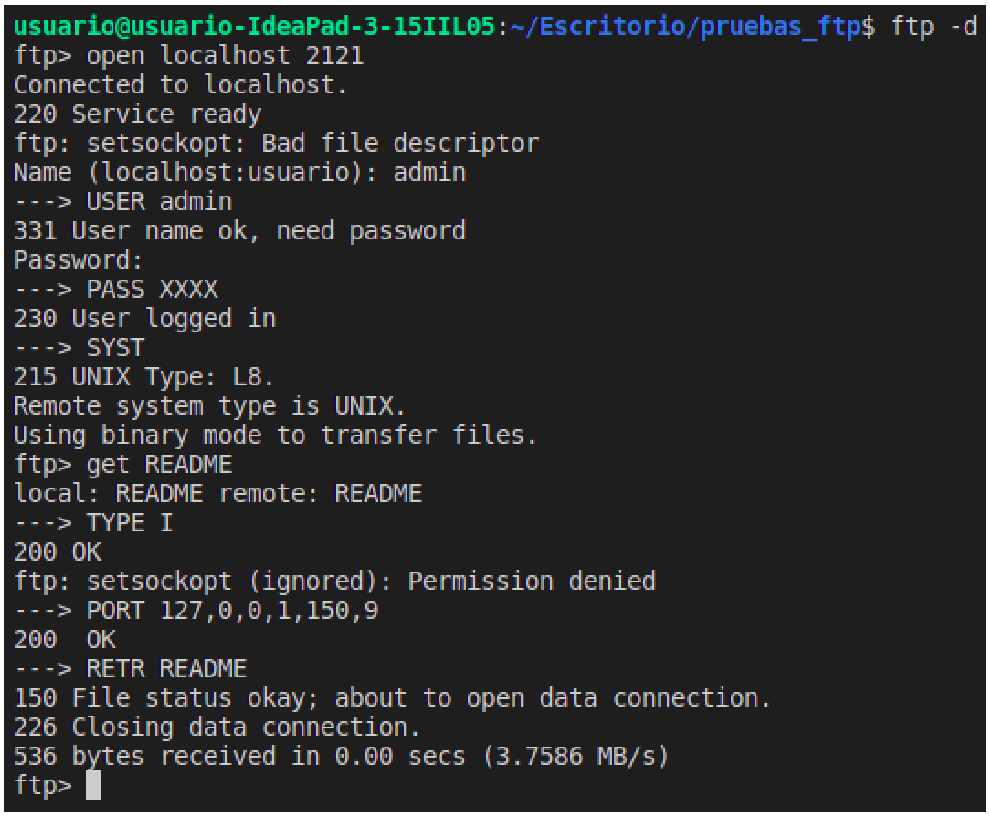
- ### Using the client mode

On the other hand, if you want to act like a client, firstly you need to use the command "ftp -d" to activate the client mode. Then, the application will ask you for a connection, so you need to introduce the next information: "open localhost 2121". If the process has been successful, then you will go to the next step. But if the program prints any type of advice you could try to wait a little and repeat or use "make clean" and "make" to compile again.

If at first you can make a session using "open localhost 2121" the next step is to introduce a user and their password. In this case, you must introduce "admin" as the user, and 1234 as the password. If everything has gone right, then you can start working as a client, using the commands that you prefer (`get`, `put`, `ls`, `exit`, ...).

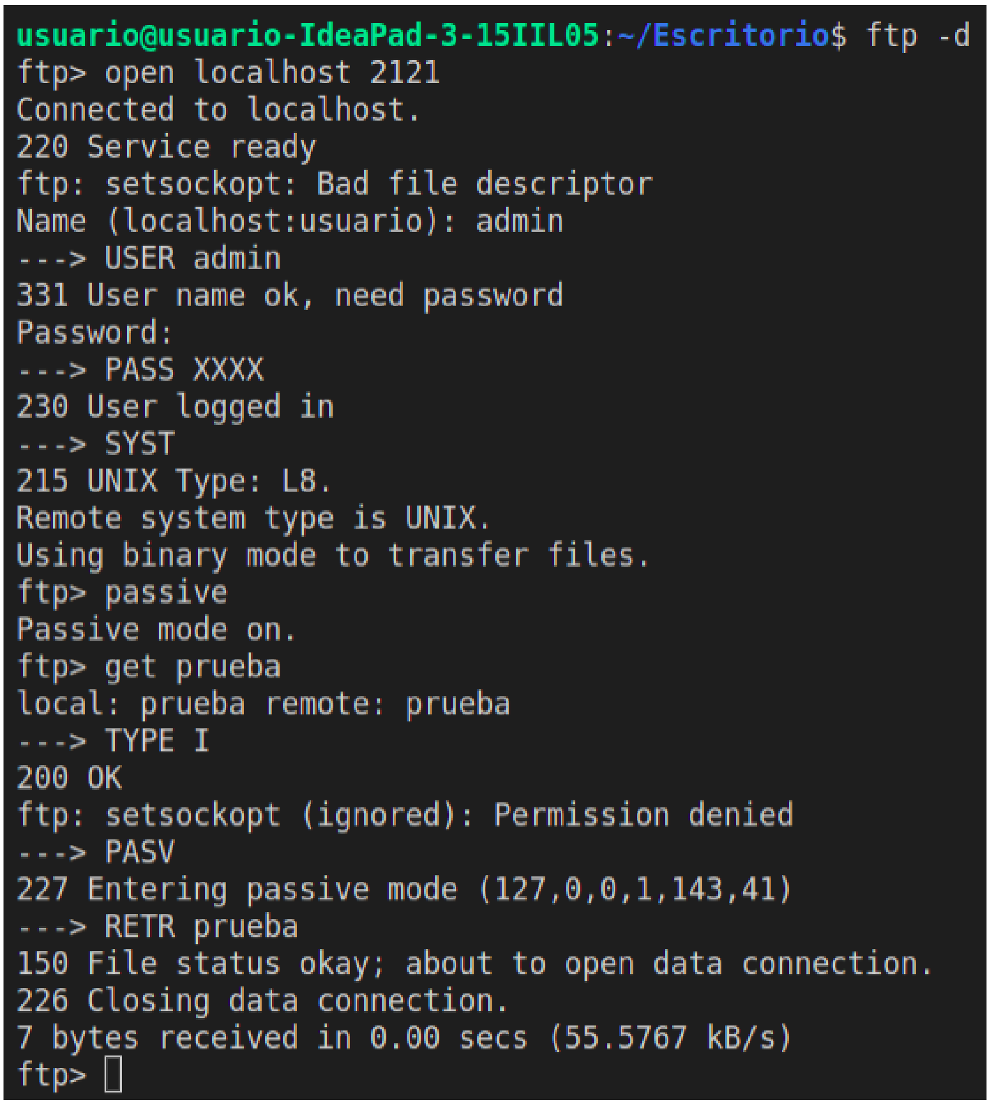
Test cases

- ### Getting a file in active mode



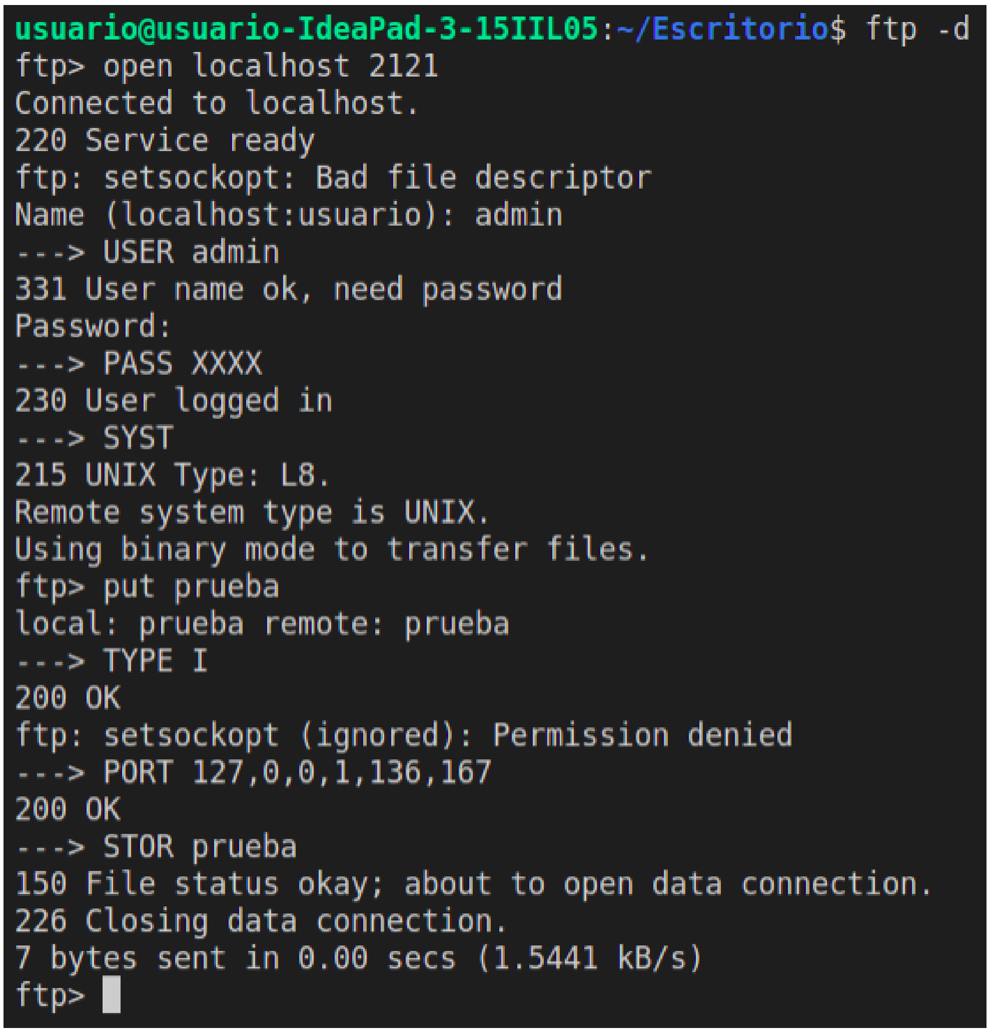
```
usuario@usuario-IdeaPad-3-15IIL05:~/Escritorio/pruebas_ftp$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:usuario): admin
---> USER admin
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get README
local: README remote: README
---> TYPE I
200 OK
ftp: setsockopt (ignored): Permission denied
---> PORT 127,0,0,1,150,9
200  OK
---> RETR README
150 File status okay; about to open data connection.
226 Closing data connection.
536 bytes received in 0.00 secs (3.7586 MB/s)
ftp> 
```

- **Getting a file in passive mode**



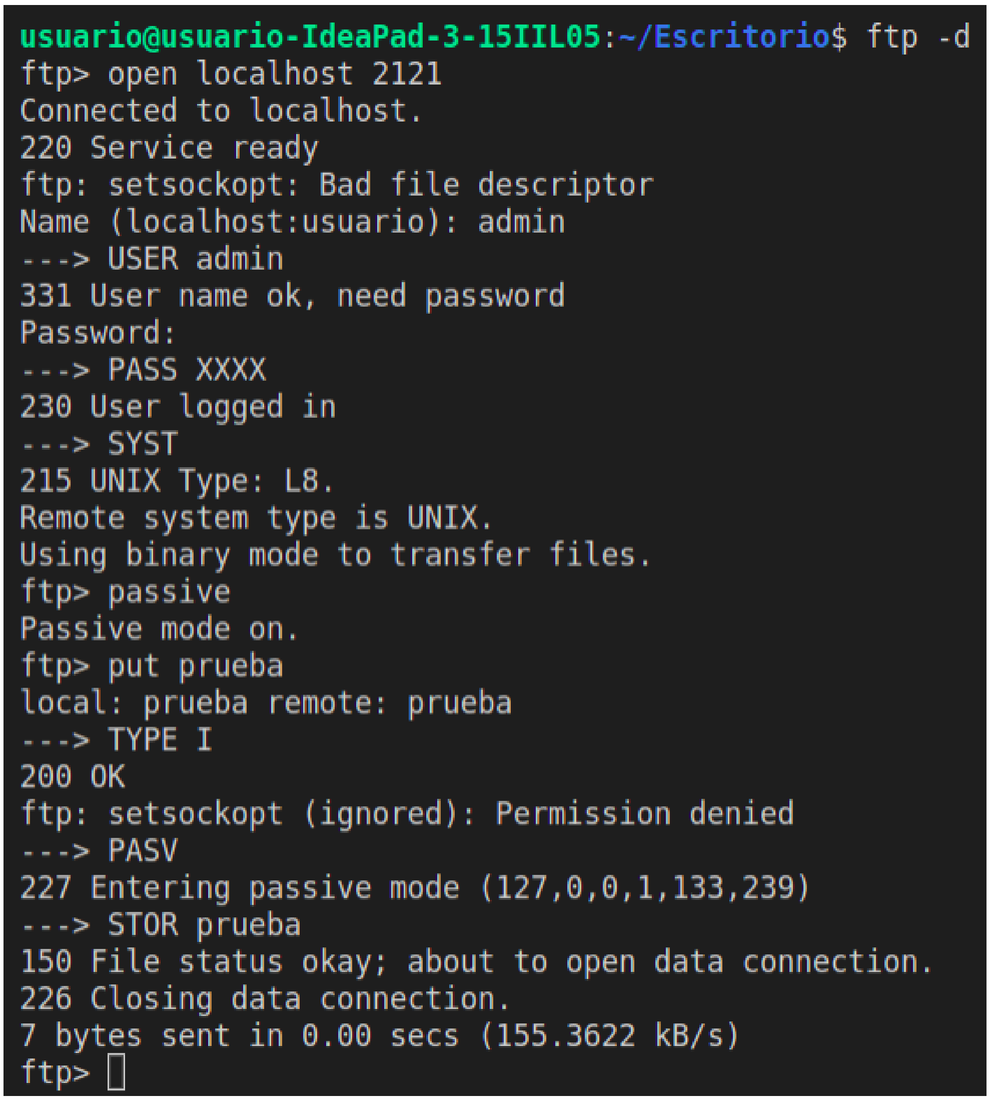
```
usuario@usuario-IdeaPad-3-15IIL05:~/Escritorio$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:usuario): admin
---> USER admin
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> get prueba
local: prueba remote: prueba
---> TYPE I
200 OK
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering passive mode (127,0,0,1,143,41)
---> RETR prueba
150 File status okay; about to open data connection.
226 Closing data connection.
7 bytes received in 0.00 secs (55.5767 kB/s)
ftp> 
```

- **Putting a file in active mode**



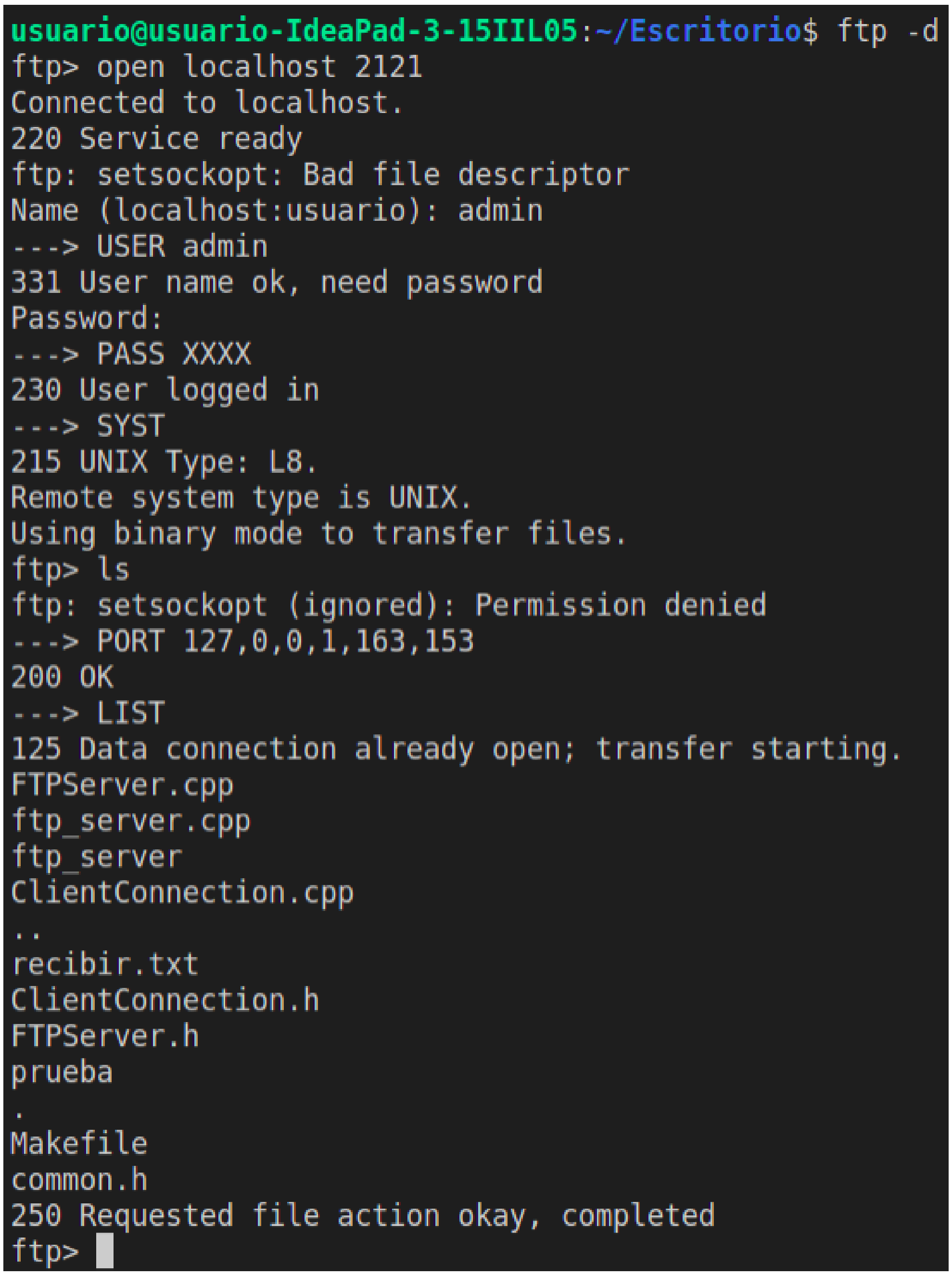
```
usuario@usuario-IdeaPad-3-15IIL05:~/Escritorio$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:usuario): admin
---> USER admin
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put prueba
local: prueba remote: prueba
---> TYPE I
200 OK
ftp: setsockopt (ignored): Permission denied
---> PORT 127,0,0,1,136,167
200 OK
---> STOR prueba
150 File status okay; about to open data connection.
226 Closing data connection.
7 bytes sent in 0.00 secs (1.5441 kB/s)
ftp> 
```

- **Putting a file in passive mode**



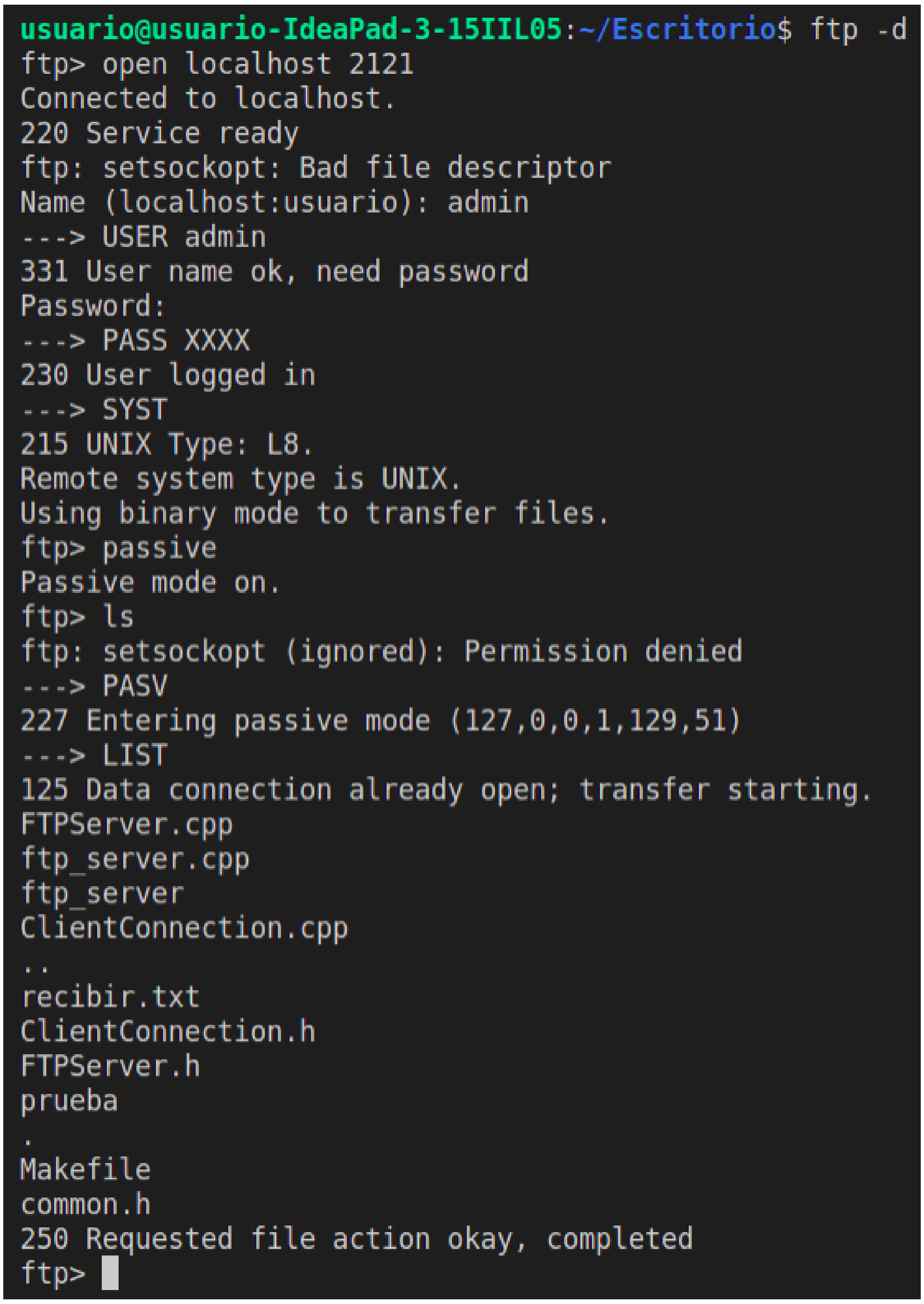
```
usuario@usuario-IdeaPad-3-15IIL05:~/Escritorio$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:usuario): admin
---> USER admin
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> put prueba
local: prueba remote: prueba
---> TYPE I
200 OK
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering passive mode (127,0,0,1,133,239)
---> STOR prueba
150 File status okay; about to open data connection.
226 Closing data connection.
7 bytes sent in 0.00 secs (155.3622 kB/s)
ftp> 
```

- **Using `ls` command in active mode**



```
usuario@usuario-IdeaPad-3-15IIL05:~/Escritorio$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:usuario): admin
---> USER admin
331 User name ok, 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.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PORT 127,0,0,1,163,153
200 OK
---> LIST
125 Data connection already open; transfer starting.
FTPServer.cpp
ftp_server.cpp
ftp_server
ClientConnection.cpp
..
recibir.txt
ClientConnection.h
FTPServer.h
prueba
.
Makefile
common.h
250 Requested file action okay, completed
ftp> 
```

- **Using `ls` command in passive mode**



```
usuario@usuario-IdeaPad-3-15IIL05:~/Escritorio$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:usuario): admin
---> USER admin
331 User name ok, 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.
ftp> passive
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering passive mode (127,0,0,1,129,51)
---> LIST
125 Data connection already open; transfer starting.
FTPServer.cpp
ftp_server.cpp
ftp_server
ClientConnection.cpp
..
recibir.txt
ClientConnection.h
FTPServer.h
prueba
.
Makefile
common.h
250 Requested file action okay, completed
ftp>
```

Appendix: Source code

Here you can find a link to access our Github Repository: Link to FTP_RSD

References

[1] https://en.wikipedia.org/wiki/File_Transfer_Protocol

[2] https://en.wikipedia.org/wiki/Transmission_Control_Protocol

[3] https://en.wikipedia.org/wiki/List_of_FTP_commands