SOCKET PROGRAMMING

REPORT

REDES Y SISTEMAS DISTRIBUIDOS

Pedro Hernández Alonso alu0101423944@ull.edu.es
Ramiro Difonti Domé alu0101425030@ull.edu.es

Index

Brief description of the developed application

The objective of this task is to develop our application protocol for a simple client-server application (ftp server).

We will implement the basic server functions, so that, it allows uploading and downloading files from a single folder.

Description of the developed protocol

The File Transfer protocol is thought for transferring files between a client and a server in order to store or retrieve them from the server. This protocol bases on the TCP transport protocol and uses the well-known ports 20 and 21. Two TCP connections are involved: The control connection (port 21) and the data connection (port 20). The control connection is used to transfer commands and the replies to these commands.

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as HTML editors.

Guide for the compilation and steps to execute

1 . Compiling the server aplication is as easy as executing the order make in a linux terminal command line. This will give an executable of the server named as ftp_server. We must be located in the src subdirectory of the project repossitory to do this

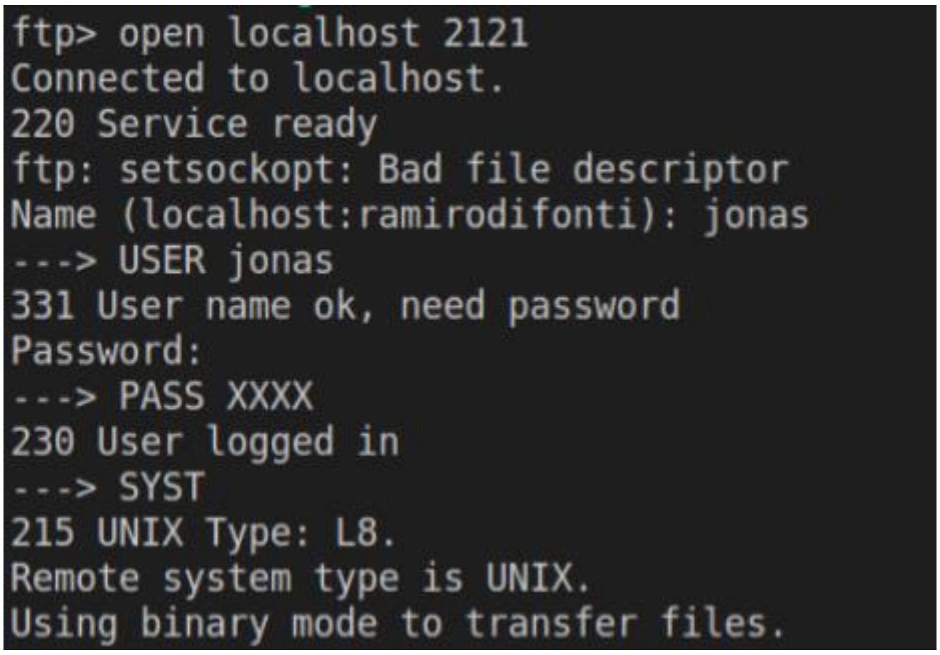
2. Once the program was compiled we can leave the server running. To do this we must run the command ./ftp_server leaving our server working on the background and waiting for a client to connec

3. To open a client we execute the ftp command in a diferent linux terminal, once done we will enter to the ftp command line interface.

4. We must connect our client to the server entering the open command in order to open the conection, as arguments we must provide the IP address which is localhost and the port 2121 in our case.

5. The next step is to introduce our username (jonas) and then our password (1234).

6. From here on we will be able to interact with the server through the ftp commands in the server, such as getting files, putting files, showing the current directory, etc.



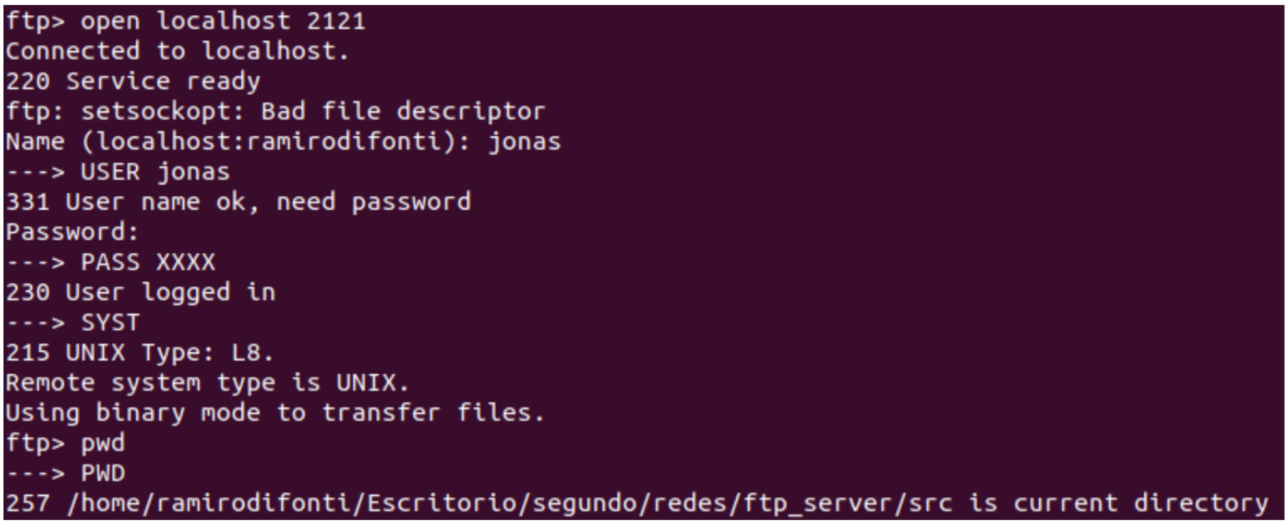
```
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:ramirodifonti): jonas
---> USER jonas
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Test cases

Once the program is compiled and excuted (instructions above), we are able to use the ftp server, in which we can manage multiple commands.

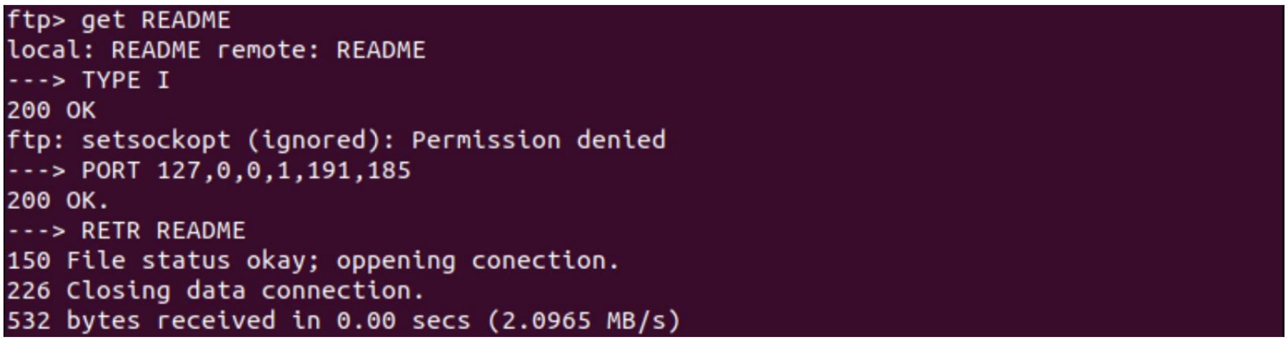
- Active mode

Login and pwd:



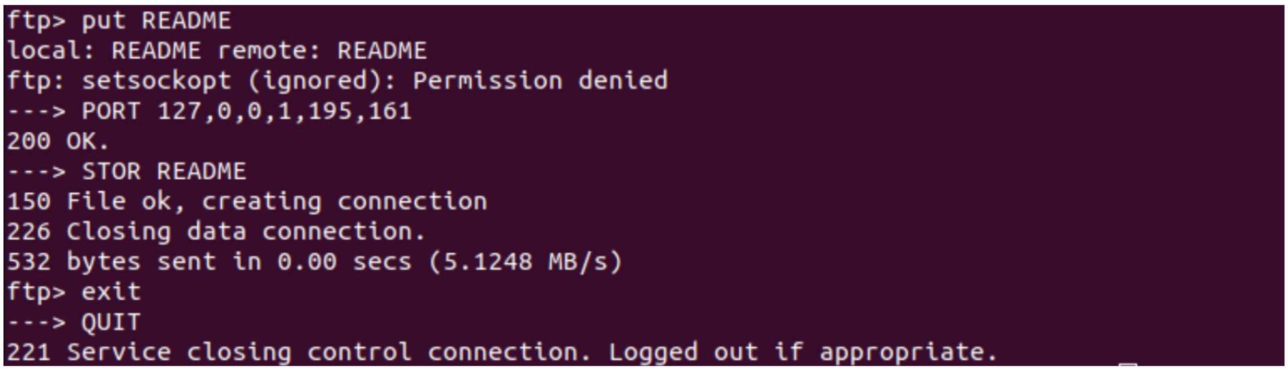
```
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:ramirodifonti): jonas
---> USER jonas
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
---> PWD
257 /home/ramirodifonti/Escritorio/segundo/redes/ftp_server/src is current directory
```

get readme:



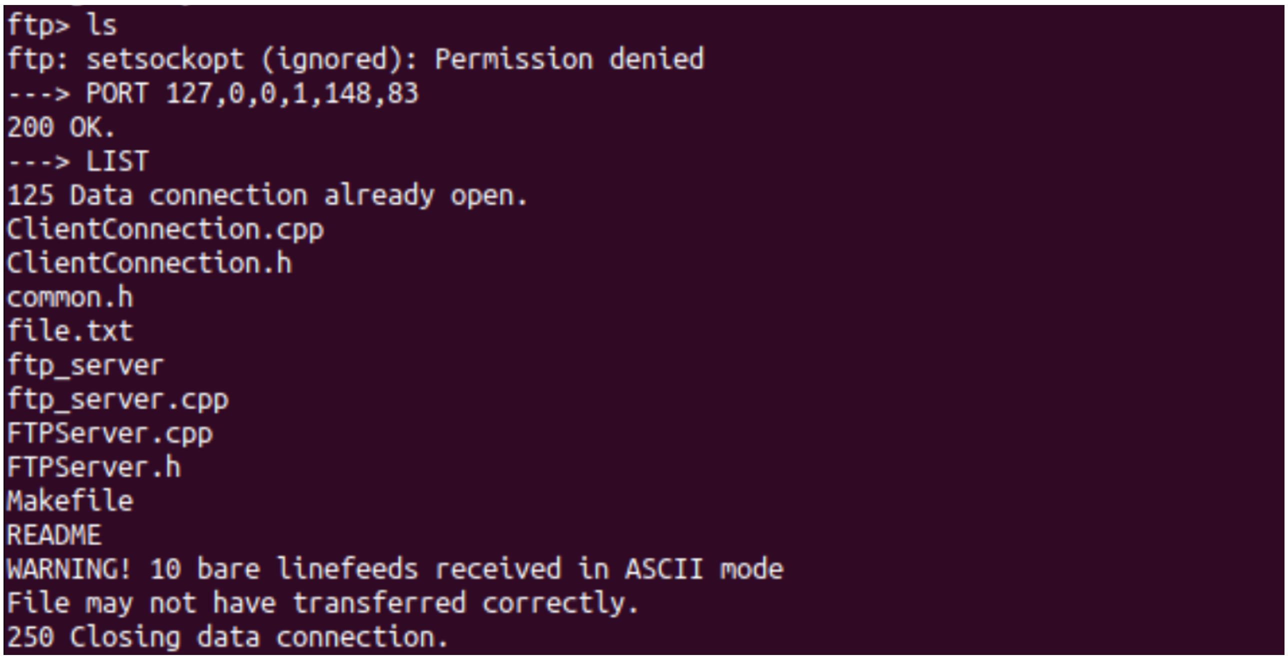
```
ftp> get README
local: README remote: README
---> TYPE I
200 OK
ftp: setsockopt (ignored): Permission denied
---> PORT 127,0,0,1,191,185
200 OK.
---> RETR README
150 File status okay; oppening conection.
226 Closing data connection.
532 bytes received in 0.00 secs (2.0965 MB/s)
```

put readme:



```
ftp> put README
local: README remote: README
ftp: setsockopt (ignored): Permission denied
---> PORT 127,0,0,1,195,161
200 OK.
---> STOR README
150 File ok, creating connection
226 Closing data connection.
532 bytes sent in 0.00 secs (5.1248 MB/s)
ftp> exit
---> QUIT
221 Service closing control connection. Logged out if appropriate.
```

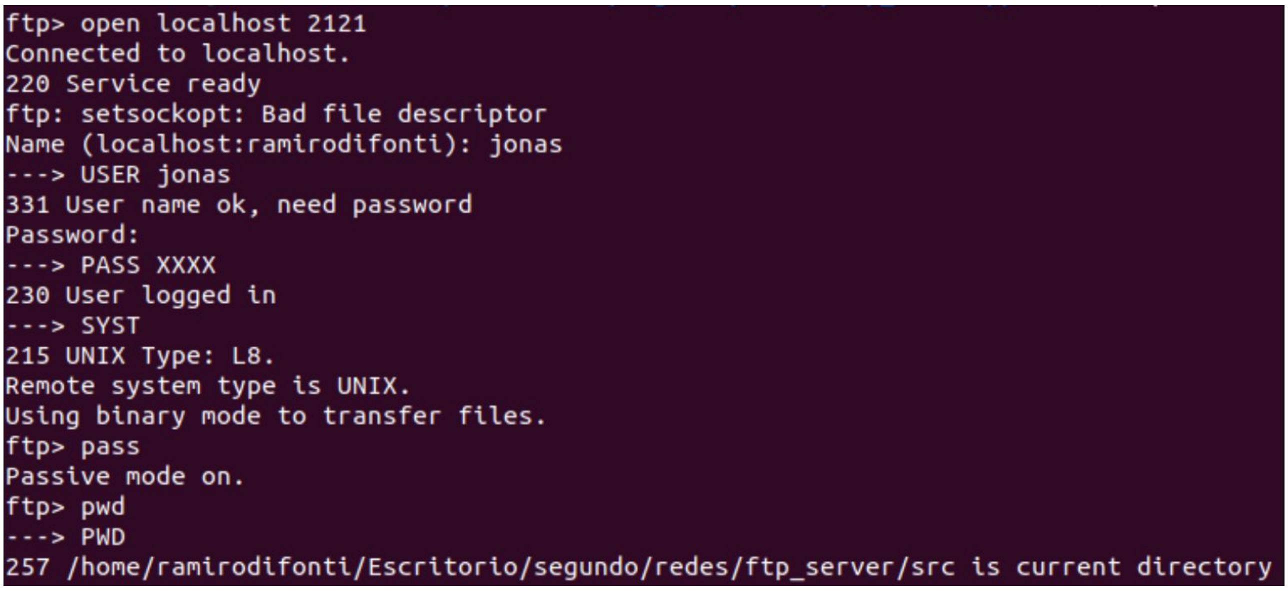
ls:



```
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PORT 127,0,0,1,148,83
200 OK.
---> LIST
125 Data connection already open.
ClientConnection.cpp
ClientConnection.h
common.h
file.txt
ftp_server
ftp_server.cpp
FTPServer.cpp
FTPServer.h
Makefile
README
WARNING! 10 bare linefeeds received in ASCII mode
File may not have transferred correctly.
250 Closing data connection.
```

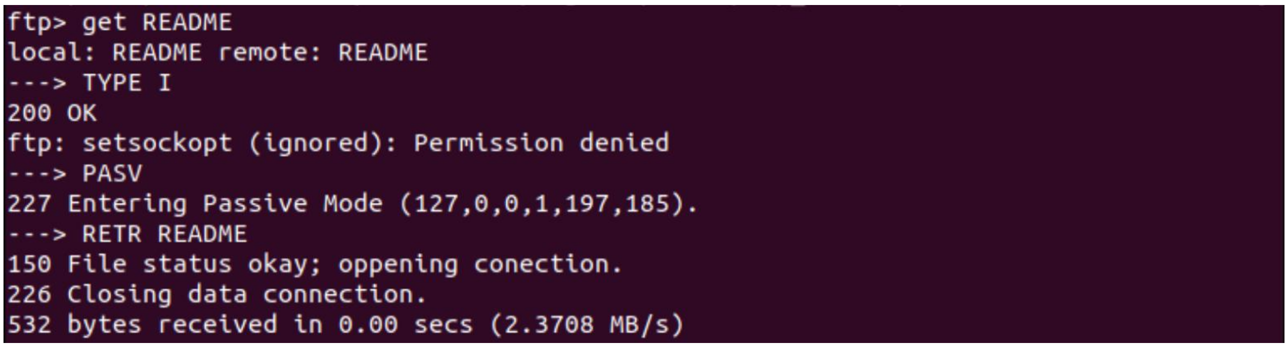
- Passive mode

Login and pwd:



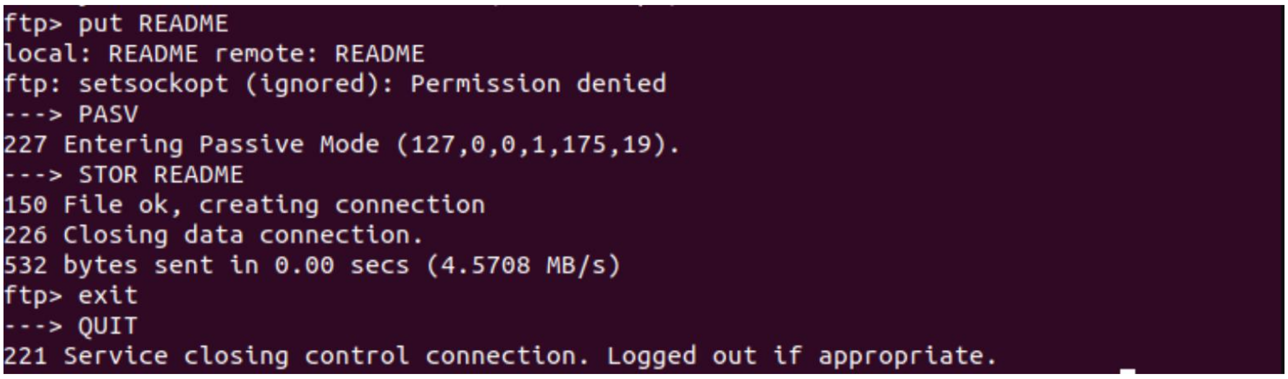
```
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:ramirodifonti): jonas
---> USER jonas
331 User name ok, need password
Password:
---> PASS XXXX
230 User logged in
---> SYST
215 UNIX Type: L8.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pass
Passive mode on.
ftp> pwd
---> PWD
257 /home/ramirodifonti/Escritorio/segundo/redes/ftp_server/src is current directory
```

get readme:



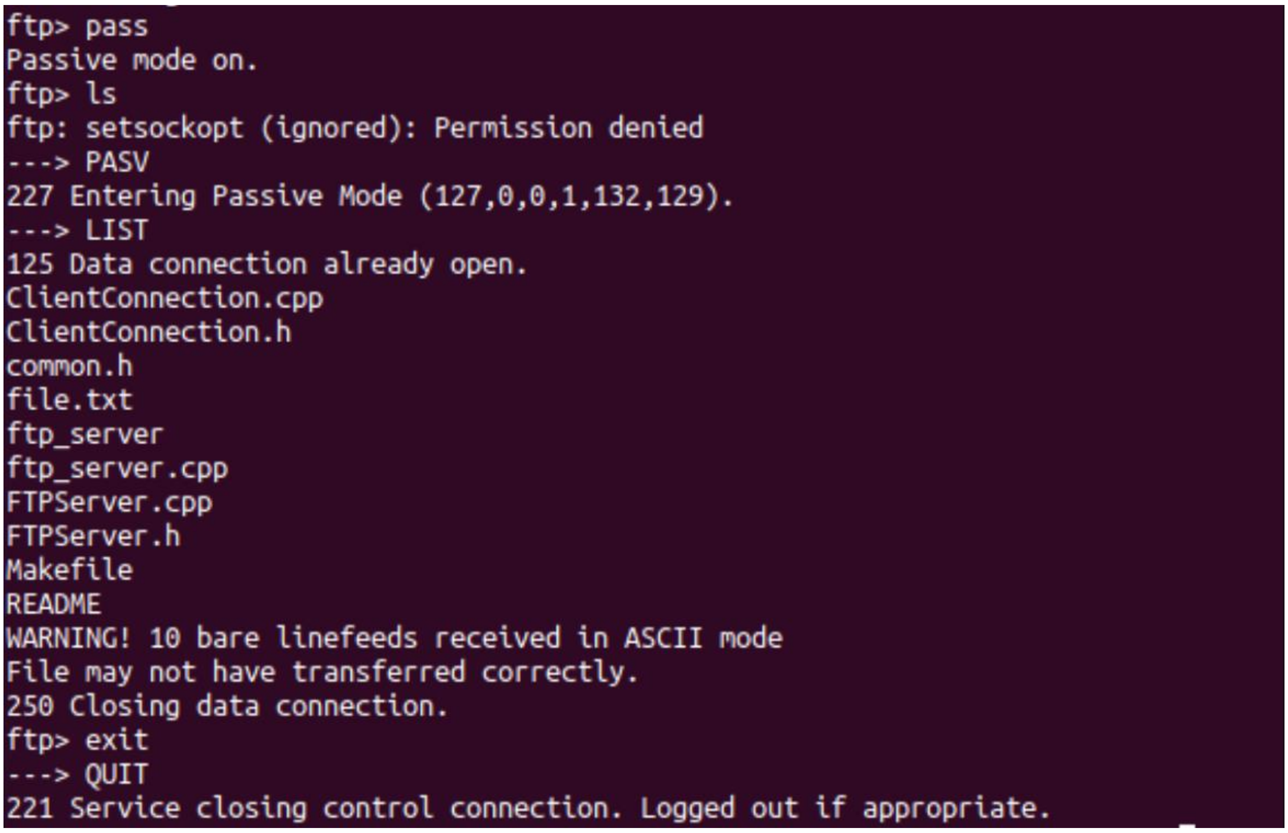
```
ftp> get README
local: README remote: README
---> TYPE I
200 OK
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (127,0,0,1,197,185).
---> RETR README
150 File status okay; oppening conection.
226 Closing data connection.
532 bytes received in 0.00 secs (2.3708 MB/s)
```

put readme:



```
ftp> put README
local: README remote: README
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (127,0,0,1,175,19).
---> STOR README
150 File ok, creating connection
226 Closing data connection.
532 bytes sent in 0.00 secs (4.5708 MB/s)
ftp> exit
---> QUIT
221 Service closing control connection. Logged out if appropriate.
```

ls:



```
ftp> pass
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (127,0,0,1,132,129).
---> LIST
125 Data connection already open.
ClientConnection.cpp
ClientConnection.h
common.h
file.txt
ftp_server
ftp_server.cpp
FTPServer.cpp
FTPServer.h
Makefile
README
WARNING! 10 bare linefeeds received in ASCII mode
File may not have transferred correctly.
250 Closing data connection.
ftp> exit
---> QUIT
221 Service closing control connection. Logged out if appropriate.
```