

# REDES Y SISTEMAS DISTRIBUIDOS

Curso 2013-2014

## Práctica de sockets

### Implementación de un servidor FTP sencillo.

**Objetivo:** El objetivo de esta práctica es el estudio e implementación de un servidor FTP sencillo. Se implementarán las funciones básicas para el funcionamiento del servidor, de modo que éste permita subir y bajar ficheros de un único directorio.

#### 1. Documentación previa.

Antes de comenzar con la implementación de la práctica deberá estudiar las llamadas al sistema de Linux que permiten el manejo de sockets. En la mayoría de los sistemas podrá encontrar documentación referente a las mismas en las propias páginas del manual de Linux. También podrá encontrar ejemplos de uso de estas funciones en el Aula Virtual de la asignatura.

Llamadas al sistema en Linux para el manejo de sockets: `socket`, `bind`, `connect`, `listen`, `accept`, `send`, `recv`, `shutdown`, `close`.

Además, conviene que revise la función `fdopen`.

#### 2. Plantilla de código.

Para facilitar la implementación de la práctica ésta va acompañada de una plantilla de código que podrá utilizar para su implementación. Dicha plantilla consta de los siguientes ficheros:

- `common.h`: Funciones comunes.
- `ftp_server.cpp`: Programa principal.
- `FTPServer.h`: Clase correspondiente al servidor FTP.
- `FTPServer.cpp`: Implementación del servidor FTP.
- `ClientConnection.h`: Clase encargada de atender las peticiones asociadas a cada cliente.
- `ClientConnection.cpp`: Implementación de la clase anterior.
- `Makefile`.

#### 3. Comandos FTP a implementar.

Se deberán implementar los siguientes comandos: `USER`, `PWD`, `PASS`, `PORT`, `PASV`, `CWD`, `STOR`, `SYST`, `TYPE`, `RETR`, `QUIT`, `LIST`. La descripción de los mismos, así como los códigos de control a devolver en cada caso, se puede encontrar en el RFC959 que se adjunta. Tenga en cuenta que cada comando enviado por la conexión de control deberá ser respondido mediante un código de control y un texto descriptivo.

Por otra parte, la implementación deberá contemplar ambos modos de transferencia, activo y pasivo. La conexión de control se hará por el puerto 2121, en vez de por el 21, para que se pueda realizar el desarrollo como usuario normal.

#### 4. Comprobación del funcionamiento y depuración.

Para comprobar funcionamiento puede utilizar el comando `ftp` con la opción `-d` (debug). A continuación, se muestra una traza que demuestra el funcionamiento esperado en los seis casos de uso con los que deberá cumplir la implementación (las líneas en rojo se deben al modo de depuración):

##### get en modo activo

```
user@localhost:~/pruebas_ftp$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost: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> get README
local: README remote: README
----> TYPE I
200 OK.
ftp: setsockopt (ignored): Permission denied
----> PORT 127,0,0,1,229,250
200 OK.
----> RETR README
150 File status okay; about to open data connection
226 Closing data connection.
2480 bytes received in 0.00 secs (33176.4 kB/s)
ftp>
```

##### get en modo pasivo

```
user@localhost:~/pruebas_ftp$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:jonas): 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> passive
Passive mode on.
ftp> get README
local: README remote: README
----> TYPE I
200 OK.
ftp: setsockopt (ignored): Permission denied
----> PASV
227 Entering Passive Mode (127,0,0,1,129,232).
----> RETR README
150 File status okay
226 Closing data connection.
2480 bytes received in 0.00 secs (41756.5 kB/s)
```

## put en modo activo

```
user@localhost:~/pruebas_ftp$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost:jonas): 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> put README
local: README remote: README
----> TYPE I
200 OK.
ftp: setsockopt (ignored): Permission denied
----> PORT 127,0,0,1,223,168
200 OK.
----> STOR README
150 File creation ok, about to open data connection
226 Closing data connection.
2480 bytes sent in 0.00 secs (86495.5 kB/s)
ftp>
```

## put en modo pasivo

```
user@localhost:~/pruebas_ftp$ ftp -d
ftp> open localhost 2121
Connected to localhost.
220 Service ready
ftp: setsockopt: Bad file descriptor
Name (localhost: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> passive
Passive mode on.
ftp> put README
local: README remote: README
----> TYPE I
200 OK.
ftp: setsockopt (ignored): Permission denied
----> PASV
227 Entering Passive Mode (127,0,0,1,190,81).
----> STOR README
150 File creation ok
226 Closing data connection.
2480 bytes sent in 0.00 secs (89699.1 kB/s)
```

## ls en modo activo

```
ftp> ls
ftp: setsockopt (ignored): Permission denied
----> PORT 127,0,0,1,162,188
200 OK.
----> LIST
125 List started OK.
-rwxr-xr-x  1 jonas  jonas      142300    Sun Mar 23 18:34:55 2014    ftp_server
-rw-r--r--  1 jonas  jonas       2480    Sun Mar 23 19:28:58 2014    README
-rw-r--r--  1 jonas  jonas        283    Sun Mar 23 12:41:43 2014    Makefile
250 List completed successfully.
ftp>
```

## ls en modo pasivo

```
ftp> ls
----> TYPE A
200 OK.
ftp: setsockopt (ignored): Permission denied
----> PASV
227 Entering Passive Mode (127,0,0,1,231,72).
----> LIST
125 List started OK.
-rwxr-xr-x  1 jonas  jonas      142300    Sun Mar 23 18:34:55 2014    ftp_server
-rw-r--r--  1 jonas  jonas       2480    Sun Mar 23 19:28:58 2014    README
-rw-r--r--  1 jonas  jonas        283    Sun Mar 23 12:41:43 2014    Makefile
250 List completed successfully.
```

## 5. Informe.

Una vez finalizado el desarrollo deberá elaborar un informe utilizando la plantilla que se adjunta.