

# Práctica de Sockets - Redes I

Pablo Jesús González Rubio - 70894492M  
Sergio García González - 70921911P

# Índice

Introducción	2
1.1 Archivos incluidos	2
Solución propuesta	3
2.1 Consideraciones iniciales	4
2.2 Implementación del cliente	4
2.3 Implementación del servidor	4
2.4 Tratamiento de los mensajes	5
2.5 Ejemplo de funcionamiento correcto	5

# 1. Introducción

En este documento realizaremos una breve explicación del contenido del mismo y de cómo se ha ido desarrollando. En la práctica se pide llevar a cabo una simulación de envío de mensajes entre un servidor NNTP y su respectivo cliente. Ambos implementados con los protocolos de transmisión TCP y UDP.

## 1.1 Archivos incluidos

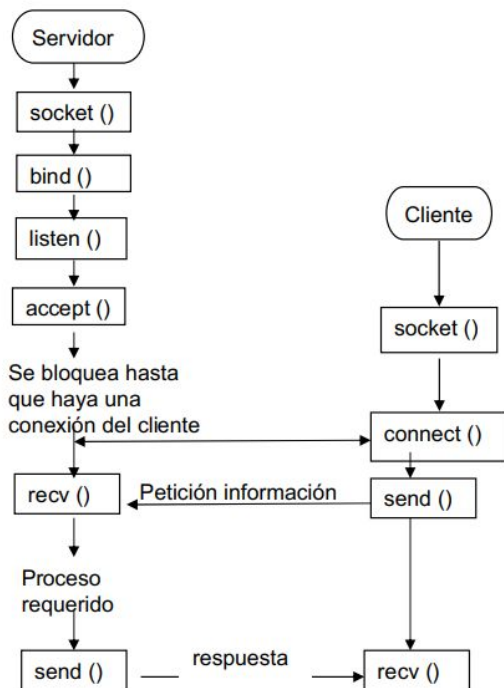
- `Servidor.c`
- `Cliente.c`
- `lanzaServidor.sh`
- `makefile`
- `Utils.c` : Fichero en el que se almacenan las funciones de las órdenes del protocolo NNTP.
- `run.sh` : Script en Bash que permite cerrar los procesos pendientes de "Servidor" y "Cliente", limpiar los ficheros compilados, volver a compilar y ejecutar el "lanzaServidor.sh".
- `killPS.sh` : Script para eliminar mediante un SIGKILL el proceso "Servidor" y "Cliente".

## 2. Solución propuesta

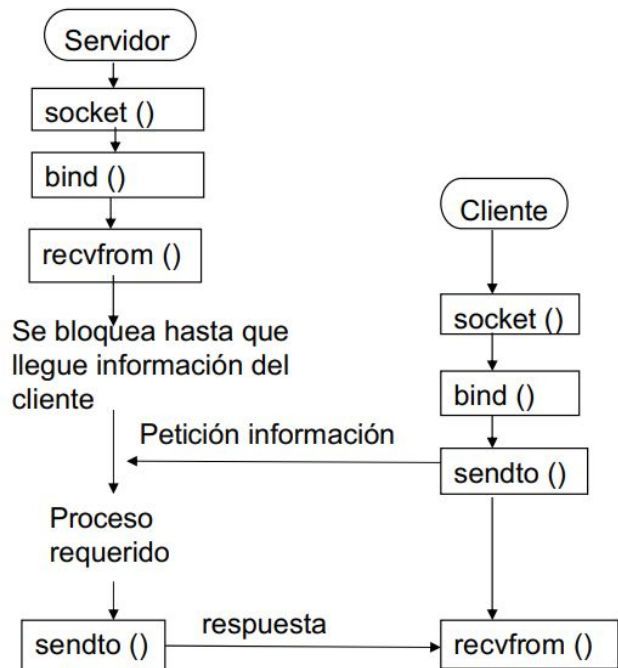
En este apartado solo comentaremos los aspectos más importantes relacionados con la propia funcionalidad del proyecto.

Se ha seguido la arquitectura de sockets de Berkeley:

### TCP



### UDP



En lo que respecta a la funcionalidad de los sockets hemos tratado el envío, la recepción de mensajes y el tratamiento de los argumentos que tiene el protocolo NNTP.

## 2.1 Consideraciones iniciales

- Se ha creado un fichero "utils.c" dónde se encuentran las funciones para cubrir cada orden del protocolo NNTP como pueda ser "LIST" o "NEWSGROUP".
- La gestión de los "timeouts" y "retries" para el protocolo UDP, tanto en cliente como en servidor, se realiza principalmente en la manejadora de "SIGALRM". Al provocarse un SIGALRM, se reduce su valor (siendo el valor inicial "RETRIES" en "cliente.c", teniendo este valor 5) hasta llegar a 0. En este caso, se procede a la siguiente orden, y en el caso de que no hubiera se dejan de enviar más órdenes.

## 2.2 Implementación del cliente

Tanto en TCP como en UDP se ha implementado siguiendo la plantilla y tratando la lectura de las órdenes de los respectivos ficheros.

Sin embargo, como ya sabemos TCP y UDP son protocolos diferentes por lo que no realizan los envíos y las recepciones de la misma forma.

En TCP se establece una comunicación y por cada orden leída del fichero de órdenes, se envía al servidor para que este la procese y en función del mensaje que le haya llegado envía una respuesta al cliente.

En cambio en UDP el cliente está constantemente leyendo las órdenes del fichero y enviándolas hasta que las reciba el servidor, en caso contrario se producen "timeouts" hasta cierto límite (en nuestro caso 5 intentos) y pasa a enviar la siguiente orden.

## 2.3 Implementación del servidor

En cuanto a la implementación del servidor NNTP hemos realizado dos funciones independientes, una para el funcionamiento del servidor en protocolo TCP y otra para el funcionamiento del servidor en protocolo UDP.

La función "serverTCP" se encarga del tratamiento de los mensajes recibidos por parte del "clienteTCP" (función TCP), dependiendo del mensaje se realizará una de las funciones del fichero "utils.c". Si la orden es incorrecta o hay un error de sintaxis o de contenido enviará un mensaje de error al cliente, el cual estará esperando un mensaje de confirmación del servidor.

La función "serverUDP" realizará el mismo tratamiento de los de los mensajes recibidos por parte del "clienteUDP" (función UDP). Al contrario que en el protocolo TCP este recibirá todos los mensajes que provengan del cliente y los irá contestando a medida que los reciba, pues puede haber errores en el envío. Es por ello que puede realizarse la misma petición hasta un máximo de 5 veces, y que por la misma razón el servidor pueda responder 5 veces.

## 2.4 Tratamiento de los mensajes

Tanto en TCP como en UDP se tratan los mensajes con unos caracteres de retorno predefinidos que son el retorno de carro (“\r”) y el salto de línea (“\n”) y con los que se trata el mensaje enviado por el cliente. En base a ello se trocean y verifican algunos parámetros necesarios para cada orden. Esto se realiza principalmente con la ayuda de la función “fgets()” y “strtok()”.

Más importancia aún ha cobrado el vaciado del contenido de las variables tipo “char \*” mediante funciones como “memset()” y “strcpy()”, pues sin ellas se estaría sobrescribiendo continuamente la información recibida y enviada.

## 2.5 Ejemplo de funcionamiento correcto

Para realizar una ejecución correcta de nuestro programa utilizaríamos el script “run.sh” que es el que utilizamos para cerrar los procesos tanto del cliente como del servidor, además de realizar el “make clean”, el posterior “make” y lanzar el script “lanzaServidor.sh”.

Una vez hayamos lanzado el run empezaremos a ver por pantalla las peticiones al servidor y las respuestas que obtienen los clientes. También a su vez se generarán diferentes archivos uno de ellos el “nntpd.log” que contendrá todas las peticiones realizadas al servidor y 6 archivos, puesto que se lanzan 6 clientes concurrentes, con el nombre de su puerto efímero “XXXXXX.txt” que contendrá los mensajes de progreso de una ejecución completa.

Un ejemplo del fichero “nntpd.log” sería:

```
Sun Jan  3 20:10:29 2021
[C] localhost 127.0.0.1 UDP 43649
[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - LISTADO

[Sun Jan  3 20:10:29 2021][localhost:58465][UDP] - LIST

[Sun Jan  3 20:10:29 2021][localhost:51633][UDP] - LIST

[Sun Jan  3 20:10:29 2021][localhost:58465][UDP] - NEWGROUPS 201126
120000

[Sun Jan  3 20:10:29 2021][localhost:51633][UDP] - NEWGROUPS 201130
000000

[Sun Jan  3 20:10:29 2021][localhost:58465][UDP] - POST
```

```

Newsgroups: local.deporte
El pádel es un deporte divertido.
Se puede practicar a cualquier edad.
En poco tiempo se puede mejorar mucho.
.

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - LIST

[Sun Jan  3 20:10:29 2021][localhost:58465][UDP] - POST
Newsgroups: local.deporte
Subject: PAdel mi deporte preferido
El pAdel es un deporte divertido.
Se puede practicar a cualquier edad.
En poco tiempo se puede mejorar mucho.
.

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - GROUP redes

[Sun Jan  3 20:10:29 2021][localhost:51633][UDP] - NEWGROUPS 201126
120000

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - GROUP local.redes

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - ARTICLE 4

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - ARTICLE 1

[Sun Jan  3 20:10:29 2021][localhost:51633][UDP] - NEWNEWS local.test
201127 120000

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - HEAD 1

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - BODY 1

[Sun Jan  3 20:10:29 2021][localhost:51633][UDP] - POST
Newsgroups: local.redes
Subject: Sobre los sockets
Los sockets son un API de programación en red.
Permite utilizar los servicios de los protocolos de transporte TCP y
UDP.
.

[Sun Jan  3 20:10:29 2021][localhost:43649][UDP] - QUIT

Sun Jan  3 20:10:29 2021
[C] localhost 127.0.0.1 TCP 52640

```

```

200 Servidor de noticias localhost 127.0.0.1 Sun Jan  3 20:10:29 2021
Sun Jan  3 20:10:29 2021
[C] localhost 127.0.0.1 TCP 52642
200 Servidor de noticias localhost 127.0.0.1 Sun Jan  3 20:10:29 2021
Sun Jan  3 20:10:29 2021
[C] localhost 127.0.0.1 TCP 52644
200 Servidor de noticias localhost 127.0.0.1 Sun Jan  3 20:10:29 2021
[Sun Jan  3 20:10:35 2021][localhost:52640][TCP] - LIST

[Sun Jan  3 20:10:35 2021][localhost:52642][TCP] - LIST

[Sun Jan  3 20:10:35 2021][localhost:52644][TCP] - LISTADO

[Sun Jan  3 20:10:41 2021][localhost:52640][TCP] - NEWGROUPS 201130
000000

[Sun Jan  3 20:10:41 2021][localhost:52642][TCP] - NEWGROUPS 201126
120000

[Sun Jan  3 20:10:41 2021][localhost:52644][TCP] - LIST

[Sun Jan  3 20:10:47 2021][localhost:52642][TCP] - POST
Newsgroups: local.deporte
El pádel es un deporte divertido.
Se puede practicar a cualquier edad.
En poco tiempo se puede mejorar mucho.
.

[Sun Jan  3 20:10:47 2021][localhost:52640][TCP] - NEWGROUPS 201126
120000

[Sun Jan  3 20:10:47 2021][localhost:52644][TCP] - GROUP redes

[Sun Jan  3 20:10:53 2021][localhost:52640][TCP] - NEWNEWS local.test
201127 120000

[Sun Jan  3 20:10:53 2021][localhost:52642][TCP] - POST
Newsgroups: local.deporte
Subject: PAdel mi deporte preferido
El pAdel es un deporte divertido.
Se puede practicar a cualquier edad.
En poco tiempo se puede mejorar mucho.
.

[Sun Jan  3 20:10:53 2021][localhost:52644][TCP] - GROUP local.redes

```



```
[Sun Jan  3 20:10:58 2021][localhost:52642][TCP] -  
[Sun Jan  3 20:10:59 2021][localhost:52640][TCP] - POST  
Newsgroups: local.redes  
Subject: Sobre los sockets  
Los sockets son un API de programación en red.  
Permite utilizar los servicios de los protocolos de transporte TCP y  
UDP.  
.  
  
[Sun Jan  3 20:10:59 2021][localhost:52644][TCP] - ARTICLE 4  
  
[Sun Jan  3 20:11:05 2021][localhost:52640][TCP] - POST  
Newsgroups: local.redes  
Subject: Sobre los sockets otra vez  
El número de puerto a utilizar en los servidores en nogal es 9999.  
Siendo 9999 los últimos 4 dígitos del DNI del alumno.  
Si el número es menor de 1024 utilizar los últimos 5 dígitos.  
.  
  
[Sun Jan  3 20:11:05 2021][localhost:52644][TCP] - ARTICLE 1  
  
[Sun Jan  3 20:11:11 2021][localhost:52640][TCP] - QUIT  
[Sun Jan  3 20:11:11 2021][localhost:52644][TCP] - HEAD 1  
  
[Sun Jan  3 20:11:12 2021][localhost:52644][TCP] - BODY 1  
  
[Sun Jan  3 20:11:18 2021][localhost:52644][TCP] - QUIT
```