

HTTP – Sockets

Fernando Olivares Naranjo – 54126671N

Diego Sánchez Martín – 44435121H

En esta práctica hemos implementado una versión rudimentaria del protocolo HTTP versión 1.1. En el documento vamos a explicar la estructura de ficheros de la práctica entregada, algunos detalles sobre nuestra implementación, una serie de errores conocidos y unos ejemplos de funcionamiento.

Estructura de ficheros:

- servidor.c: contiene el código referente a la implementación del servidor. Realiza la creación de los sockets, responde a las peticiones de los clientes y maneja el fichero de log.
- cliente.c: es referente al código del cliente. Aunque HTTP versión 1.1 solo funciona sobre TCP, tanto este cliente como el servidor anterior incorporan una versión funcional de UDP.
- utils.c (utils.h): recogen las funciones referentes a la escritura en ficheros (fichero de log y ficheros de puerto efímero).
- Makefile: genera el ejecutable.
- lanzaServidor.sh: lanza el servidor y seis clientes (tres de TCP y tres de UDP).
- ordenes*.txt: ficheros con órdenes que serán leídos por los clientes y contienen las órdenes del programa.
- peticiones.log: fichero que recoge el log del servidor. Incluye el inicio de conexión, los detalles sobre el envío y el cierre.
- www: directorio que contiene los ficheros .html.
- salidaPuerto: directorio que contiene los múltiples ficheros generados de tipo puertoeffimero.txt.

Detalles sobre nuestra implementación:

Hemos hecho una implementación acorde con el enunciado entregado y los ficheros de ejemplo disponibles en Diaweb.

Nuestra práctica funciona correctamente y sigue todas las estructuras de órdenes y respuestas mandadas por cliente y servidor que se especificaron en el enunciado. Es por ello que nos vamos a centrar en explicar algunas decisiones que hemos tomado en ciertos puntos del ejercicio.

- En UDP la comunicación siempre será close (se establece, se atiende y se cierra). Independientemente del valor de k o c expresado en la orden leída

del fichero, el servidor siempre devolverá "Connection: close". Esto es indiferente y no afecta a la correcta ejecución de la práctica.

- Tanto TCP como UDP generarán múltiples ficheros de puerto efímero (en la mayoría de los casos, depende del fichero ordenes*.txt). En el caso de UDP, se generará un fichero de nombre puertoefímero.txt para cada una de las órdenes individuales. En cambio, en TCP se tendrán en cuenta los valores de k (keep alive) y c (close).

Por ejemplo, para un fichero ordenes.txt cuyas órdenes terminen en una secuencia "k-c-c-k-c" se generarán tres ficheros de puertoefímero.txt: el primero contiene "k-c", el segundo contiene "c" y el tercero contiene "k-c". Este es el comportamiento deseado respecto a nuestra implementación de la cabecera "Connection". Para un correcto funcionamiento de TCP el fichero de ordenes*.txt debe terminar en "c".

- El funcionamiento entre TCP y UDP es ligeramente distinto (como cabría esperar) y se ve reflejado en los ficheros de puerto efímero que se generan. Además, el funcionamiento de TCP también depende del valor del campo "Connection". Se creará un proceso para cada conexión y, en el caso de UDP, un socket nuevo. Se registraron los signal para una finalización correcta de los distintos procesos.
- En los ficheros puertoefímero.txt aparecerá el mensaje o mensajes completos devueltos por el servidor al cliente correspondiente. Esto es la estructura completa de HTTP (línea inicial, múltiples cabeceras y cuerpo).

Errores conocidos:

A lo largo de la práctica nos hemos enfrentado a numerosos problemas que no hemos sido capaces de resolver en su plenitud.

- El tamaño de los paquetes es reducido ya que no hemos sido capaces de implementar una forma correcta (y sobre todo fiable y funcional) de enviar paquetes de gran tamaño. Se mostrará el tamaño de los distintos mensajes en el campo "Content Length" de las respuestas del servidor.
- El fichero peticiones.log y los ficheros puertoefímero.log podrían mejorar con una recogida de errores más exhaustiva.
- En el caso de Encina hemos encontrado un error donde, dependiendo del estado del servidor, es posible que los mensajes de TCP se redirijan a un fichero erróneo llamado "0.txt" en lugar de al fichero de puerto efímero que correspondía. Creemos que este comportamiento es un problema de concurrencia que pensamos haber detectado, pero no hemos sido capaces de resolver. Su funcionamiento en el Linux del hogar es óptimo.

Ejemplos de funcionamiento:

A continuación, vamos a mostrar una serie de imágenes del funcionamiento en Encina de la práctica. En primer lugar, la estructura de ficheros explicada:

```
<ENCINA>/home/i4126671/entregaSockets$ ls -l
total 96
-rw-rw-r-- 1 i4126671 domainusers 14346 dic 14 06:31 cliente.c
-rwxrwxr-x 1 i4126671 domainusers 238 dic 14 06:38 lanzaServidor.sh
-rw-rw-r-- 1 i4126671 domainusers 275 dic 14 06:32 Makefile
-rw-rw-r-- 1 i4126671 domainusers 37 dic 14 05:33 ordenes1.txt
-rw-rw-r-- 1 i4126671 domainusers 81 dic 14 05:33 ordenes2.txt
-rw-rw-r-- 1 i4126671 domainusers 100 dic 14 05:33 ordenes.txt
-rw-rw-r-- 1 i4126671 domainusers 0 dic 14 06:33 peticiones.log
drwxrwxr-x 2 i4126671 domainusers 4096 dic 14 06:35 salidaPuerto
-rw-rw-r-- 1 i4126671 domainusers 25386 dic 14 06:31 servidor.c
-rw-r--r-- 1 i4126671 domainusers 12444 dic 14 06:43 sockets.zip
-rw-rw-r-- 1 i4126671 domainusers 2025 dic 14 06:32 utils.c
-rw-rw-r-- 1 i4126671 domainusers 350 dic 14 06:32 utils.h
drwxrwxr-x 2 i4126671 domainusers 4096 dic 14 06:35 www
```

Compilaríamos utilizando el Makefile añadido a la práctica:

```
<ENCINA>/home/i4126671/entregaSockets$ make
gcc -g -c utils.c
gcc -g servidor.c utils.o -o servidor
gcc -g cliente.c utils.o -o cliente
<ENCINA>/home/i4126671/entregaSockets$
```

Lanzamos el servidor y 6 clientes (3 de TCP y 3 de UDP) con ./lanzaServidor.sh:

```
<ENCINA>/home/i4126671/entregaSockets$ ./lanzaServidor.sh
<ENCINA>/home/i4126671/entregaSockets$
```

Una vez realizado el ejercicio, podemos comprobar el correcto transcurso de este en el fichero peticiones.log:

```

<ENCINA>/home/i4126671/entregaSockets$ cat peticiones.log
Tue Dec 14 06:59:06 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56700 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:06 2021
HOST: nogal.fis.usal.es | UDP en puerto: 55745 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:06 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56698 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:06 2021
HOST: nogal.fis.usal.es | UDP en puerto: 50551 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:06 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56696 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:06 2021
HOST: nogal.fis.usal.es | UDP en puerto: 56355 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 55745 | Mensaje: Envío correcto de 200 OK
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 55745 | Mensaje: Close socket.
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56700 | Mensaje: Envío correcto de 200 OK
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 50551 | Mensaje: Envío correcto de 501 Not Implemented
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56696 | Mensaje: Envío correcto de 501 Not Implemented
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56698 | Mensaje: Envío correcto de 200 OK
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 50551 | Mensaje: Close socket.
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 50916 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 56355 | Mensaje: Envío correcto de 200 OK
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 44274 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 56355 | Mensaje: Close socket.
Tue Dec 14 06:59:07 2021
HOST: nogal.fis.usal.es | UDP en puerto: 58291 | Mensaje: Inicio correcto.
Tue Dec 14 06:59:08 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56700 | Mensaje: Envío correcto de 200 OK
Tue Dec 14 06:59:08 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56700 | Mensaje: Close socket.
Tue Dec 14 06:59:08 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56696 | Mensaje: Envío correcto de 200 OK
Tue Dec 14 06:59:08 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56698 | Mensaje: Envío correcto de 501 Not Implemented
Tue Dec 14 06:59:08 2021
HOST: nogal.fis.usal.es | TCP en puerto: 56696 | Mensaje: Close socket.
Tue Dec 14 06:59:08 2021

```

Hemos incorporado 3 ficheros de ordenes*.txt generados para probar los distintos casos de las implementaciones realizadas en el ejercicio (k y c, respuestas 200, 404, 501).

```

<ENCINA>/home/i4126671/entregaSockets$ cat ordenes.txt
GET /docencia.html k
GET /facultad.html c
DAME /index.html c
GET /docencia.html k
GET /index.html c

```

```

<ENCINA>/home/i4126671/entregaSockets$ cat ordenes1.txt
DAME /index.html k
GET /index.html c

```

```

<ENCINA>/home/i4126671/entregaSockets$ cat ordenes2.txt
GET /index.html k
HOLA /docencia.html k
GET /docencia.htm k
GET /facultad.html c

```

En esta ejecución, hemos sufrido el error de concurrencia donde algunas de las salidas se redirigen al archivo "0.txt". Adjuntamos una captura de cómo se

generan una serie de ficheros extra (y desaparece el “0.txt”) en nuestros sistemas de casa:

```
<ENCINA>/home/i4126671/entregaSockets$ cd salidaPuerto/
<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$ ls
0.txt 38231.txt 42042.txt 44274.txt 50551.txt 50916.txt 51199.txt 55745.txt 56355.txt 56702.txt 57935.txt 58291.txt 60839.txt
<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$ ls -l
total 52
-rw-r--r-- 1 i4126671 domainusers 1790 dic 14 06:59 0.txt
-rw-r--r-- 1 i4126671 domainusers 154 dic 14 06:59 38231.txt
-rw-r--r-- 1 i4126671 domainusers 147 dic 14 06:59 42042.txt
-rw-r--r-- 1 i4126671 domainusers 147 dic 14 06:59 44274.txt
-rw-r--r-- 1 i4126671 domainusers 154 dic 14 06:59 50551.txt
-rw-r--r-- 1 i4126671 domainusers 156 dic 14 06:59 50916.txt
-rw-r--r-- 1 i4126671 domainusers 156 dic 14 06:59 51199.txt
-rw-r--r-- 1 i4126671 domainusers 287 dic 14 06:59 55745.txt
-rw-r--r-- 1 i4126671 domainusers 147 dic 14 06:59 56355.txt
-rw-r--r-- 1 i4126671 domainusers 153 dic 14 06:59 56702.txt
-rw-r--r-- 1 i4126671 domainusers 287 dic 14 06:59 57935.txt
-rw-r--r-- 1 i4126671 domainusers 154 dic 14 06:59 58291.txt
-rw-r--r-- 1 i4126671 domainusers 142 dic 14 06:59 60839.txt
<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$
```

```
fer@ferMint:~/Clase/RedesI/SocketsHTML/EntregaSockets$ ./lanzaServidor.sh
fer@ferMint:~/Clase/RedesI/SocketsHTML/EntregaSockets$ cd salidaPuerto/
fer@ferMint:~/Clase/RedesI/SocketsHTML/EntregaSockets/salidaPuerto$ ls -l
total 64
-rw-rw-r-- 1 fer fer 139 Dec 14 07:19 33751.txt
-rw-rw-r-- 1 fer fer 139 Dec 14 07:19 36095.txt
-rw-rw-r-- 1 fer fer 279 Dec 14 07:19 37932.txt
-rw-rw-r-- 1 fer fer 146 Dec 14 07:19 44350.txt
-rw-rw-r-- 1 fer fer 148 Dec 14 07:19 47352.txt
-rw-rw-r-- 1 fer fer 279 Dec 14 07:19 47938.txt
-rw-rw-r-- 1 fer fer 139 Dec 14 07:19 50670.txt
-rw-rw-r-- 1 fer fer 146 Dec 14 07:19 51797.txt
-rw-rw-r-- 1 fer fer 146 Dec 14 07:19 52188.txt
-rw-rw-r-- 1 fer fer 134 Dec 14 07:19 56441.txt
-rw-rw-r-- 1 fer fer 148 Dec 14 07:19 59689.txt
-rw-rw-r-- 1 fer fer 428 Dec 14 07:19 60112.txt
-rw-rw-r-- 1 fer fer 287 Dec 14 07:19 60114.txt
-rw-rw-r-- 1 fer fer 576 Dec 14 07:19 60116.txt
-rw-rw-r-- 1 fer fer 145 Dec 14 07:19 60118.txt
-rw-rw-r-- 1 fer fer 419 Dec 14 07:19 60120.txt
fer@ferMint:~/Clase/RedesI/SocketsHTML/EntregaSockets/salidaPuerto$
```

Algunos de los ficheros generados:

```
<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$ cat 58291.txt
HTTP/1.1 501 Not Implemented
Server: nogal.fis.usal.es
Connection: close
Content-Length: 54

<html><body><h1>501 Not Implemented</h1></body></html>

<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$
```

```
<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$ cat 51199.txt
HTTP/1.1 200 OK
Server: nogal.fis.usal.es
Connection: close
Content-Length: 69

<html><body><h1>Facultad de Ciencias de Salamanca</h1></body></html>
```

```

<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$ cat 57935.txt
HTTP/1.1 200 OK
Server: nogal.fis.usal.es
Connection: close
Content-Length: 199

<HTML lang="es">
<head>
<title>Departamento de Informática y Automática - Universidad de Salamanca</title>
</head>
<body>
<table>
<tr>
<td>Redes de Computadores I</td>
</tr>
</table>
</body>
</html>

<ENCINA>/home/i4126671/entregaSockets/salidaPuerto$ █

```

Por último, añadimos algunas de las salidas de una ejecución correcta de las salidas de TCP (los ficheros ordenes*.txt eran ligeramente distintos). Las respuestas a las distintas órdenes del cliente se escriben en un mismo fichero:

```

[<ENCINA>/home/i4435121/EntregaSockets/salidaPuerto$ cat 54974
cat: 54974: No existe el fichero o el directorio
[<ENCINA>/home/i4435121/EntregaSockets/salidaPuerto$ cat 54974.txt
HTTP/1.1 501 Not Implemented
Server: nogal.fis.usal.es
Connection: keep-alive
Content-Length: 55

<html><body><h1>501 Not Implemented</h1></body></html>
HTTP/1.1 200 OK
Server: nogal.fis.usal.es
Connection: close
Content-Length: 60

<html><body><h1>Universidad de Salamanca</h1></body></html>
<ENCINA>/home/i4435121/EntregaSockets/salidaPuerto$ █

```

```

<ENCINA>/home/i4435121/EntregaSockets/salidaPuerto$ cat 54972.txt
HTTP/1.1 200 OK
Server: nogal.fis.usal.es
Connection: keep-alive
Content-Length: 199

<HTML lang="es">
<head>
<title>Departamento de Informática y Automática - Universidad de Salamanca</title>
</head>
<body>
<table>
<tr>
<td>Redes de Computadores I</td>
</tr>
</table>
</body>
</html>
HTTP/1.1 200 OK
Server: nogal.fis.usal.es
Connection: keep-alive
Content-Length: 69

<html><body><h1>Facultad de Ciencias de Salamanca</h1></body></html>
HTTP/1.1 200 OK
Server: nogal.fis.usal.es
Connection: close
Content-Length: 60

<html><body><h1>Universidad de Salamanca</h1></body></html>
<ENCINA>/home/i4435121/EntregaSockets/salidaPuerto$ █

```

Bibliografía:

Consultas generales, close y shutdown. Especialmente shutdown:

<https://beej.us/guide/bgnet/html/#close-and-shutdownget-outta-my-face>

Getnameinfo:

<https://linux.die.net/man/3/getaddrinfo>

Sendto:

<https://www.ibm.com/docs/en/zos/2.3.0?topic=functions-sendto-send-data-socket>

TCP y UDP:

<http://www.faqs.org/faqs/internet/tcp-ip/tcp-ip-faq/part1/>