



VNiVERSiDAD D SALAMANCA

REDES DE COMPUTADORES I

SOCKETS

1.Introducción

Esta práctica de programación en C se ha enfocado en la implementación del modelo cliente-servidor. Se ha llevado a cabo en un entorno de desarrollo con Debian GNU/Linux y Sockets de Berkeley.

La práctica ha consistido en desarrollar el servicio "Patata Caliente", un juego interactivo basado en red que requiere de respuestas numéricas a preguntas planteadas en un número limitado de intentos.

Este servicio ha involucrado el manejo eficaz de mensajes entre el cliente y el servidor, siguiendo un protocolo específico.

2.Descripción del problema

En este juego, los clientes responden a preguntas numéricas planteadas por el servidor en un número limitado de intentos.

El objetivo consiste en el manejo eficiente y preciso de los mensajes intercambiados entre el cliente y el servidor, garantizando una comunicación fluida y sin errores.

Esta interacción sigue un protocolo de comunicación específico importante para el correcto funcionamiento del juego.

3.Desarrollo

Antes de comenzar con la programación del proyecto, hemos estudiado sobre los conceptos más importantes sobre los sockets, y realizando esta práctica hemos ido aprendiendo y adquiriendo nuevos fundamentos y conocimientos sobre ellos.

Estructura del Código.

Servidor: El código del servidor establece la conexión de red y espera a que los clientes se unan. Gestiona la lógica del juego enviando preguntas a los clientes y procesando sus respuestas.

Incluye funciones para la inicialización del socket, la espera de conexiones de clientes, y la lógica específica del juego.

Cliente: El cliente se conecta al servidor y participa en el juego. Incluye funciones para establecer la conexión con el servidor, recibir preguntas, y enviar respuestas.

Protocolo de Comunicación.

El protocolo de comunicación entre el cliente y el servidor se basa en el intercambio de mensajes de texto. Se definen formatos específicos para los mensajes de solicitud y respuesta, asegurando una comunicación clara y estructurada, asegurando siempre que las líneas de los mensajes son siempre líneas de caracteres terminadas con los caracteres CR-LF (retorno de carro '\r' (ASCII 13 (0x0D)), - avance de línea '\n' (ASCII 10 (0x0A))).

El manejo de errores y las condiciones de terminación del juego también se han incorporado en el protocolo.

4.Pruebas de funcionamiento.

Escenarios de Prueba: Para verificar el correcto funcionamiento del juego se han realizado pruebas. Estas incluyen:

- Conexiones tanto separadas como simultáneas de múltiples clientes al servidor.
- Pruebas de manejo de errores, como respuestas incorrectas y desconexiones inesperadas.
- Asegura que las preguntas se plantean y responden correctamente y que el juego sigue el flujo esperado.

Resultados y Análisis: Los resultados de las pruebas han demostrado que:

- El servidor puede manejar múltiples clientes simultáneamente sin pérdida de rendimiento.
- La lógica del juego funciona según lo esperado, con preguntas y respuestas procesadas.
- No obstante, en algunos casos, el programa no funciona como se esperaba, debido al bloqueo de algunos clientes.

Los ficheros de órdenes utilizados constan de 3 ficheros (ordenes.txt, ordenes1.txt, ordenes2.txt) teniendo cada uno las ordenes que el servidor responderá .

Todas las respuestas del servidor han sido redirigidas a un fichero cuyo nombre es el número del puerto efímero del cliente.

EJECUCIÓN DEL PROGRAMA Y PRUEBAS DE ERROR

CASOS SIN CONCURRENCIA DE PROCESOS

- `./cliente localhost TCP ordenes.txt &`

| GNU nano 4.8 | GNU nano 4.8 |
|---------------|--|
| HELO | 220 Servicio preparado |
| HOLA | 500 Error de sintaxis |
| no lo se | 250 ¿A cómo está hoy la prima de riesgo?#4 |
| RESPUESTA 67 | 500 Error de sintaxis |
| RESPUESTA 110 | 354 MAYOR#3 |
| + | 350 ACIERTO |
| RESPUESTA 51 | 250 ¿Cuántos estados tiene EEUU?#4 |
| ADIOS | 350 ACIERTO |
| | 221 Cerrando el servicio |

TCP_47408.txt

- `./cliente localhost TCP ordenes1.txt &`

| GNU nano 4.8 | GNU nano 4.8 |
|---------------|------------------------|
| RESPUESTA 67 | 220 Servicio preparado |
| RESPUESTA 110 | 500 Error de sintaxis |
| + | 500 Error de sintaxis |
| RESPUESTA 156 | 500 Error de sintaxis |
| RESPUESTA 51 | 500 Error de sintaxis |
| + | 500 Error de sintaxis |
| RESPUESTA 4 | 500 Error de sintaxis |
| ADIOS | 500 Error de sintaxis |
| | 500 Error de sintaxis |

TCP_47410.txt

- `./cliente localhost TCP ordenes2.txt &`

```
GNU nano 4.8
HOLA
HOLA
RESPUESTA no lo se
RESPUESTA 67
RESPUESTA 110
RESPUESTA 220
+
RESPUESTA 34
RESPUESTA 167
RESPUESTA 554
+
RESPU 34
RESPUESTA 5
RESPUESTA 3
RESPUESTA 4
HOLA
KDMCICE
ADIOS
```

```
GNU nano 4.8
220 Servicio preparado
250 ¿A cómo está hoy la prima de riesgo?#4
500 Error de sintaxis
500 Error de sintaxis
354 MAYOR#3
350 ACIERTO
500 Error de sintaxis
250 ¿Cuántos estados tiene EEUU?#4
354 MAYOR#3
354 MENOR#2
354 MENOR#1
250 ¿Cuántos lados tiene un cuadrado?#4
500 Error de sintaxis
354 MENOR#3
354 MAYOR#2
350 ACIERTO
500 Error de sintaxis
500 Error de sintaxis
221 Cerrando el servicio
```

TCP_47406.txt

- `./cliente localhost UDP ordenes.txt &`

```
marcos@DESKT  Seleccionar marcos@DESKTOP-U5H2COB: /mnt/c/Users/U
GNU nano 4.8
HELO
HOLA
no lo se
RESPUESTA 67
RESPUESTA 110
+
RESPUESTA 51
ADIOS
```

```
GNU nano 4.8
220 Servicio preparado
250 ¿A cómo está hoy la prima de riesgo?#4
500 Error de sintaxis
354 MAYOR#3
350 ACIERTO
250 ¿Cuántos estados tiene EEUU?#4
350 ACIERTO
221 Cerrando el servicio
```

UDP_45502.txt

- `./cliente localhost UDP ordenes1.txt &`

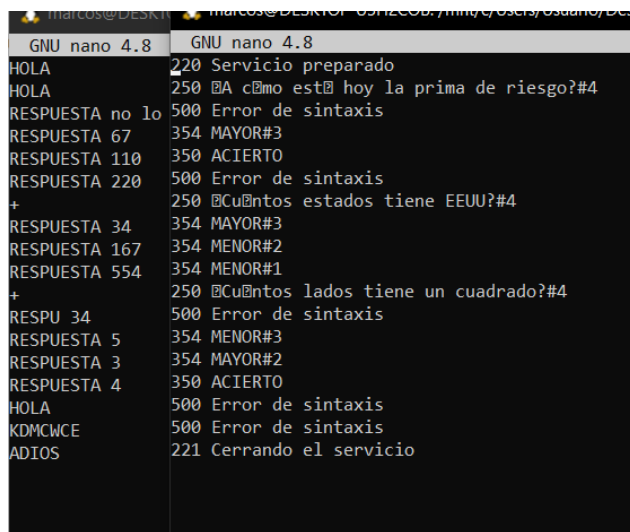
```
GNU nano 4.8
RESPUESTA 67
RESPUESTA 110
+
RESPUESTA 156
RESPUESTA 51
+
RESPUESTA 4
ADIOS
```

```
GNU nano 4.8
220 Servicio preparado
500 Error de sintxis
500 Error de sintxis
500 Error de sintxis
500 Error de sintxis
500 Error de sintxis
500 Error de sintxis
500 Error de sintxis
```

UDP_47112.txt

Javier Sánchez Cacho 70959150X
Marcos Rivas Kyoguro 70962760D

- `./cliente localhost UDP ordenes2.txt &`



```
GNU nano 4.8
HOLA 220 Servicio preparado
HOLA 250 ¿A cómo está hoy la prima de riesgo?#4
RESPUESTA no lo 500 Error de sintaxis
RESPUESTA 67 354 MAYOR#3
RESPUESTA 110 350 ACIERTO
RESPUESTA 220 500 Error de sintaxis
+ 250 ¿Cuántos estados tiene EEUU?#4
RESPUESTA 34 354 MAYOR#3
RESPUESTA 167 354 MENOR#2
RESPUESTA 554 354 MENOR#1
+ 250 ¿Cuántos lados tiene un cuadrado?#4
RESPU 34 500 Error de sintaxis
RESPUESTA 5 354 MENOR#3
RESPUESTA 3 354 MAYOR#2
RESPUESTA 4 350 ACIERTO
HOLA 500 Error de sintaxis
KDMCWE 500 Error de sintaxis
ADIOS 221 Cerrando el servicio
```

UDP_56625.txt

Los ficheros escritos son los ficheros correspondientes a cada cliente y donde se encuentran las respuestas del servidor.

CASOS CON CONCURRENCIA DE PROCESOS

TCP

```
# lanzaServidor.sh
# Lanza el servidor que es un daemon y varios clientes
# las ordenes están en un fichero que se pasa como tercer parámetro
./servidor
./cliente localhost TCP ordenes.txt &
./cliente localhost TCP ordenes1.txt &
./cliente localhost TCP ordenes2.txt &
./cliente localhost TCP ordenes.txt &
```

Esta ejecución debería crear 4 ficheros con las respuestas del servidor al cliente en cada una de ellas. (Hemos añadido el prefijo del protocolo para poder distinguir clientes TCP de UDP).

| | | | |
|-----------|------------------|---------------------|------|
| TCP_47424 | 12/12/2023 17:56 | Documento de tex... | 1 KB |
| TCP_47426 | 12/12/2023 17:56 | Documento de tex... | 1 KB |
| TCP_47428 | 12/12/2023 17:56 | Documento de tex... | 1 KB |
| TCP_47430 | 12/12/2023 17:56 | Documento de tex... | 1 KB |

Dentro de cada fichero se encuentran las respuestas del servidor a cada cliente correspondiente.

Javier Sánchez Cacho 70959150X
Marcos Rivas Kyoguro 70962760D

Para comprobar el correcto funcionamiento y la comunicación hemos creado un fichero log, llamado “peticiones.log”, donde pueden verse todos los eventos ocurridos incluyendo la hora, nombre del host, ip, protocolo, y el mensaje enviado o recibido.

Nombre: Cacho, J. Sánchez

```
RESPUESTA ENVIADA [2023-12-12 17:51:15]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47422, Mensaje: 221 Cerrando el servicio
COMUNICACION FINALIZADA [2023-12-12 17:51:15]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47422, Mensaje:
#####
#####
COMUNICACION REALIZADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje:
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: RESPUESTA 67

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: 500 Error de sintaxis
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: RESPUESTA 110

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: 500 Error de sintaxis
#####
COMUNICACION REALIZADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47426, Mensaje:
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: +

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: 500 Error de sintaxis
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47426, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47426, Mensaje: 250 ¿A cómo está hoy la prima de riesgo?#4
#####
COMUNICACION REALIZADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47428, Mensaje:
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: RESPUESTA 156

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47424, Mensaje: 500 Error de sintaxis
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47426, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47426, Mensaje: 500 Error de sintaxis
#####
COMUNICACION REALIZADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47430, Mensaje:
ORDEN RECIBIDA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47428, Mensaje: HELLO

RESPUESTA ENVIADA [2023-12-12 17:56:09]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47428, Mensaje: 500 Error de sintaxis
```

Utilizando dicho fichero .log podemos confirmar que los ficheros creados corresponden a clientes diferentes.

UDP

En el caso de UDP, las respuestas del servidor a los clientes las hemos redirigido a un unico fichero cuyo nombre es el puerto efimero del cliente que inició la conexión UDP.

```
# lanzaServidor.sh
# Lanza el servidor que es un daemon y varios clientes
# las ordenes están en un fichero que se pasa como tercer parámetro
./servidor
./cliente localhost UDP ordenes.txt &
./cliente localhost UDP ordenes.txt &
./cliente localhost UDP ordenes1.txt &
./cliente localhost UDP ordenes2.txt &
```

| | | | |
|-----------|------------------|---------------------|-------|
| servidor | 12/12/2023 16:38 | Archivo O | 39 KB |
| UDP_36550 | 12/12/2023 18:23 | Documento de tex... | 0 KB |

Javier Sánchez Cacho 70959150X
Marcos Rivas Kyoguro 70962760D

```
=====
OMUNICACION REALIZADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 36550, Mensaje:
ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 57840, Mensaje: HELLO

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 57840, Mensaje: 500 Error de sintxis

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 56975, Mensaje: RESPUESTA 67

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 56975, Mensaje: 500 Error de sintxis

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 36550, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 36550, Mensaje: 250 ¿A cómo está hoy la prima de riesgo

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 57840, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 57840, Mensaje: 500 Error de sintaxis

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 52436, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 52436, Mensaje: 500 Error de sintaxis

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 56975, Mensaje: RESPUESTA 110

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 56975, Mensaje: 350 ACIERTO

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 36550, Mensaje: no lo se

RESPUESTA ENVIADA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 36550, Mensaje: 500 Error de sintaxis

ORDEN RECIBIDA [2023-12-12 18:23:49]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 57840, Mensaje: no lo se
=====
```

En “peticiones.log” vemos que el número del puerto efímero del cliente que inició la conexión es 36550, y posteriormente aparecieron los nuevos clientes (57840, 56975, 52436). Todos los mensajes de estos clientes aparecen en el fichero “UDP_36550.txt”.

Hemos intentado distinguir los números de los puertos efímeros de cada cliente, pero no lo conseguimos a tiempo. No obstante, tras enviar la práctica seguiremos intentándolo.

```
UDP_36550: Bloc de notas
Archivo Edición Formato Ver Ayuda
220 Servicio preparado
500 Error de sintxis
500 Error de sintxis
250 ¿A cómo está hoy la prima de riesgo?#4
500 Error de sintaxis
500 Error de sintaxis
350 ACIERTO
500 Error de sintaxis
500 Error de sintaxis
500 Error de sintaxis
250 ¿Cuántos estados tiene EEUU?#4
354 MENOR#3
354 MENOR#2
500 Error de sintaxis
354 MENOR#1
375 FALLO
500 Error de sintaxis
500 Error de sintaxis
500 Error de sintaxis
250 ¿Cuántos lados tiene un cuadrado?#4
500 Error de sintaxis
354 MENOR#3
500 Error de sintaxis
354 MENOR#2
354 MENOR#1
375 FALLO
500 Error de sintaxis
221 Cerrando el servicio
221 Cerrando el servicio
250 ¿Cuál es la raíz cuadrada de 25?#4
500 Error de sintaxis
354 MENOR#3
354 MENOR#2
354 MENOR#1
250 ¿En qué año nació Albert Einstein?#4
500 Error de sintaxis
354 MAYOR#3
354 MAYOR#2
354 MAYOR#1
500 Error de sintaxis
500 Error de sintaxis
221 Cerrando el servicio
<
```

Javier Sánchez Cacho 70959150X
Marcos Rivas Kyoguro 70962760D

TCP con UDP

Al mezclar clientes con protocolos diferentes, el funcionamiento no debería alterar. No obstante si algún cliente UDP cierra su servicio y queda otro cliente UDP puede que no se envíen todos los datagramas del cliente pendiente, provocando pérdidas de información.

```
# lanzaServidor.sh
# Lanza el servidor que es un daemon y varios clientes
# las ordenes están en un fichero que se pasa como tercer parámetro
./servidor
./cliente localhost TCP ordenes.txt &
./cliente localhost TCP ordenes.txt &
./cliente localhost UDP ordenes1.txt &
./cliente localhost UDP ordenes1.txt &
```

| | | | |
|-----------|------------------|---------------------|------|
| TCP_47456 | 12/12/2023 18:36 | Documento de tex... | 1 KB |
| TCP_47458 | 12/12/2023 18:36 | Documento de tex... | 1 KB |
| UDP_33020 | 12/12/2023 18:36 | Documento de tex... | 0 KB |

Eventos de apertura de clientes:

Archivo | Opcion | Formato | Ver | Ayuda

COMUNICACION REALIZADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje:
ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: HELO

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: 500 Error de sintaxis

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: 250 ¿A cómo está hoy la prima de riesgo?#4

COMUNICACION REALIZADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje:
ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: no lo se

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: 500 Error de sintaxis

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: HELO

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: 500 Error de sintaxis

COMUNICACION REALIZADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 33020, Mensaje:
ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: RESPUESTA 67

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: 354 MAYOR#3

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: HOLA

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: 250 ¿A cómo está hoy la prima de riesgo?#4

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: RESPUESTA 67

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: 500 Error de sintaxis

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: RESPUESTA 110

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje: 350 ACIERTO

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: no lo se

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: 500 Error de sintaxis

Javier Sánchez Cacho 70959150X
Marcos Rivas Kyoguro 70962760D

Eventos de Cierre de clientes:

peticiones: Bloc de notas

Archivo Edición Formato Ver Ayuda

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: 500 Error de sintxis

COMUNICACION FINALIZADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47458, Mensaje:

#####

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: RESPUESTA 51

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 33020, Mensaje: RESPUESTA 156

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: 350 ACIERTO

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 33020, Mensaje: 500 Error de sintxis

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: ADIOS

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: RESPUESTA 156

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje: 221 Cerrando el servicio

COMUNICACION FINALIZADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: TCP, Puerto: 47456, Mensaje:

#####

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: 500 Error de sintxis

|

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 33020, Mensaje: RESPUESTA 51

...

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 33020, Mensaje: ADIOS

RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 33020, Mensaje: 500 Error de sintxis

ORDEN RECIBIDA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: ADIOS




RESPUESTA ENVIADA [2023-12-12 18:36:54]: Host: localhost, IP: 127.0.0.1, Protocolo: UDP, Puerto: 59898, Mensaje: 500 Error de sintxis

<

El contenido de los ficheros TCP es correcto, pero el de los UDP (33020, 59898), al haber un cierre temprano del servicio para el cliente 59898, todas las órdenes que aún no había mandado no se acabarían de procesar. Esto causaría una pérdida de datagramas inesperado.

*En esta prueba hemos tenido muchos resultados, siempre aleatorios. No obstante, en algunos casos como en este uno de los clientes TCP no se crea y por tanto se queda pendiente al terminar el programa.

```
# lanzaServidor.sh
# Lanza el servidor que es un daemon y varios clientes
# las ordenes están en un fichero que se pasa como tercer parámetro
./servidor
./cliente localhost TCP ordenes.txt &
./cliente localhost TCP ordenes1.txt &
./cliente localhost TCP ordenes2.txt &
./cliente localhost UDP ordenes.txt &
./cliente localhost UDP ordenes1.txt &
./cliente localhost UDP ordenes1.txt &
```

| | | | |
|---|------------------|---------------------|------|
|  TCP_47476 | 12/12/2023 18:50 | Documento de tex... | 1 KB |
|  TCP_47478 | 12/12/2023 18:50 | Documento de tex... | 1 KB |
|  UDP_40890 | 12/12/2023 18:50 | Documento de tex... | 0 KB |

5.Código a destacar

```
marcos@DESKTOP-U5H2COB:/mnt/c/Users/Usuario/Desktop/redes2$ ./lanzaservidor.sh
marcos@DESKTOP-U5H2COB:/mnt/c/Users/Usuario/Desktop/redes2$ Servicio cerrado
Servicio cerrado
Servicio cerrado
Servicio cerrado
ps
  PID TTY          TIME CMD
   10 pts/0    00:00:01 bash
  1465 pts/0    00:00:00 servidor
  1474 pts/0    00:00:00 ps
marcos@DESKTOP-U5H2COB:/mnt/c/Users/Usuario/Desktop/redes2$ kill -term 1465
marcos@DESKTOP-U5H2COB:/mnt/c/Users/Usuario/Desktop/redes2$
Ha finalizado el programa servidor.
```

Cuando los clientes terminan de enviar sus órdenes no necesitan seguir conectados con el servidor, luego se desconecta de él, indicando “Servicio cerrado”.

Para finalizar el servidor y los clientes pendientes se debe usar la señal SIGTERM, dado a que hemos implementado una manejadora que finaliza la ejecución infinita del servidor al recibir dicha señal (kill -sigterm pid).

6.Desafíos y Soluciones

Durante la implementación de esta práctica, aparecieron varios desafíos:

Manejo de Múltiples Conexiones: Uno de los desafíos ha sido gestionar múltiples conexiones de clientes de manera eficiente.

Se ha resuelto utilizando un enfoque multihilo en el servidor, permitiendo que cada conexión de cliente se manejara en un hilo separado, lo que mejoró la escalabilidad y el rendimiento del servidor.

Además, debido a la aleatoriedad con la que el SO maneja los clientes, hemos encontrado muchos casos diferentes, en los que cada vez se gestionan los clientes de manera diferente.

Sincronización de Datos: Hemos utilizado semáforos para evitar la sobreescritura en los archivos compartidos, “peticiones.log”.

También, hemos implementado mecanismos de control de flujo y confirmación de mensajes para asegurar que los datos se transmitieran de manera confiable.

Por ejemplo: Cuando un cliente quiere mandar una orden usando el protocolo TCP debe esperar a que el servidor le confirme que está preparado.

Manejo de Errores y Excepciones: Se desarrollaron procedimientos de manejo de errores para garantizar que el servidor permaneciera estable y funcional bajo estas condiciones.

7.Conclusiones y Aprendizajes

La realización de esta práctica ha sido fundamental para comprender en profundidad la programación en C, especialmente en el contexto de la programación de sockets.

Javier Sánchez Cacho 70959150X

Marcos Rivas Kyoguro 70962760D

El desarrollo del juego "Patata Caliente" ha destacado la importancia del manejo eficiente de conexiones y la comunicación entre cliente y servidor.

Además, se ha reforzado la habilidad para resolver problemas y manejar errores, elementos esenciales en cualquier desarrollo de software.

Esta experiencia ha aportado una valiosa perspectiva sobre los desafíos y soluciones en la programación de aplicaciones de red, estableciendo una base sólida para futuros proyectos en esta área.

Referencias

Código fuente de la asignatura "Sistemas Operativos I" y "Sistemas Operativos II"

Apuntes de clase "Redes de Computadores I"