

Arquitectura de Redes y Servicios

Práctica Tema 6: Cliente-Servidor TCP Concurrente

Jesús Cámara y Diego R. Llanos
Dpto. de Informática, Universidad de Valladolid

23 de octubre de 2023

Índice

1. Introducción	1
2. Implementación del Cliente TCP	1
3. Implementación del Servidor TCP Concurrente	2
4. Criterios de Evaluación y Entrega	2

1. Introducción

En esta práctica se llevará a cabo el desarrollo de un cliente y un servidor (concurrente) que permitan, mediante el protocolo TCP, hacer uso del servicio **EchoCon** presentado en la práctica anterior.

La práctica está dimensionada para su realización en seis horas lectivas. Se desarrollará exclusivamente en lenguaje C utilizando la máquina virtual asignada en Matrix. Es fundamental que el alumno comprenda perfectamente el significado del código implementado. Los únicos recursos necesarios para realizar la práctica son (a) los apuntes proporcionados por el profesor y (b) las páginas **man** ofrecidas por propio S.O

2. Implementación del Cliente TCP

El cliente TCP funcionará de forma similar a su homólogo UDP. En este caso, en cuanto el cliente establezca el circuito virtual con el servidor y le envíe la cadena a convertir, el servidor le responderá con la nueva cadena convertida y procederá al cierre de la conexión.

Se implementará en el fichero `echocon-tcp-client-apellidos.c` y recibirá los mismos parámetros que el cliente UDP desarrollado en la práctica anterior. Para probar su correcto funcionamiento, se puede establecer la conexión TCP con el servidor **EchoCon** desarrollado por el profesor (ubicado en la máquina con IP 10.0.25.250), tal como se hizo en la práctica anterior con el cliente UDP.

3. Implementación del Servidor TCP Concurrente

El servidor TCP se implementará en el fichero `echocon-tcp-server-apellidos.c` y recibirá el mismo parámetro de entrada que el servidor UDP desarrollado en la práctica anterior para el servicio `EchoCon`. En este caso, las solicitudes enviadas por el cliente se atenderán de forma concurrente, creando un proceso hijo para cada solicitud mediante la llamada al sistema `fork()`

Los servidores TCP, tanto iterativos como concurrentes, presentan el siguiente problema. Al finalizar el servidor pulsando `Ctrl+C`, el socket principal por donde se reciben las conexiones entrantes no se libera hasta que transcurre un tiempo. En consecuencia, no se puede volver a iniciar el servidor inmediatamente porque el puerto asociado al servicio sigue estando en uso. La solución consiste en crear una función que permita capturar la señal enviada por el sistema operativo al pulsar `Ctrl+C` y, a continuación, notificar el cierre del socket usando la función `shutdown(2)`. Para ello, es necesario llevar a cabo los siguientes pasos:

- Registrar al comienzo del programa la función encargada de capturar la señal. Basta con invocar a la función `signal(2)` pasándole dos parámetros: la señal a capturar y el nombre de la función que se debe ejecutar cuando el usuario pulse `Ctrl+C`.
- Escribir el código de la función, que llamaremos `signal_handler` y recibirá como parámetro de entrada un número que identificará la señal recibida. Si la señal es `SIGINT`, invocará a `shutdown(2)` para notificar el cierre del socket principal y, a continuación, finalizará el programa. Para que esta función pueda acceder a la variable que identifica al socket principal, debe estar declarada de forma global (fuera del `main`)

4. Criterios de Evaluación y Entrega

1. La práctica debe realizarse en la máquina virtual asignada en Matrix.
2. Los dos ficheros de código fuente (correspondientes al cliente y al servidor) deben comenzar con un comentario indicando el nombre de su autor, con el siguiente formato:

```
// Practica Tema 6: Apellido1 Apellido2, Nombre
```

3. Tanto el cliente como el servidor deben incluir comentarios que indiquen cómo se ha implementado la funcionalidad en cada uno. De lo contrario, se penalizará con dos puntos la calificación obtenida.
4. Tanto el cliente como el servidor deben compilar sin advertencias (opción `-Wall` de compilador `gcc`) ni errores. De lo contrario, se penalizará con tres puntos la calificación obtenida.
5. Esta práctica supone un 20 % en la calificación de las prácticas de la asignatura.
6. Cuando esté finalizada, se subirá a la tarea habilitada en el Campus Virtual un fichero comprimido en formato ZIP que contenga **única y exclusivamente** los ficheros de código fuente (sin directorios) El nombre del fichero ZIP será `Apellido1-Apellido2-Prac5.zip`. Un fallo en las condiciones de entrega supondrá un punto menos en la calificación.
7. Se hará uso de un sistema automático de detección de copias. En caso de copia, los alumnos involucrados figurarán como suspensos en la convocatoria ordinaria, debiendo enviar todas las prácticas por e-mail al profesor para poder presentarse a la convocatoria extraordinaria. Todas las prácticas se corregirán sobre 7.
8. Fecha de Entrega: **5 de noviembre de 2023 a las 23:55.**
9. No se admitirán entregas fuera de plazo.