

Arquitectura de Redes y Servicios

Práctica tema 6: Cliente-servidor TCP concurrente

Dr. Diego R. Llanos
Departamento de Informática
Universidad de Valladolid
Versión 1.0

1. Introducción

En esta práctica desarrollaremos un servidor y un cliente TCP para suministrar el servicio DAYTIME, siguiendo el formato de la práctica anterior. Esta práctica está dimensionada para su realización en seis horas lectivas.

2. Desarrollo de un cliente DAYTIME TCP

En el caso de la versión TCP del servicio DAYTIME, el cliente no necesita enviar un mensaje. En cuanto el cliente establezca la conexión de circuito virtual con el servidor, éste le enviará el mensaje con la cadena correspondiente y cerrará la conexión. El cliente se llamará `daytime-tcp-client-apellido1-apellido2`, y recibirá los mismos parámetros que el cliente UDP desarrollado en la práctica anterior.

Como en el caso del cliente UDP, el profesor pondrá a disposición de los alumnos un servidor DAYTIME que utiliza el protocolo TCP. El servidor se encontrará disponible en la dirección 10.0.25.250.

3. Desarrollo de un servidor DAYTIME TCP concurrente

Se desarrollará un servidor TCP concurrente (`daytime-tcp-server-apellido1-apellido2`) para el servicio DAYTIME, con la misma estructura que el servidor UDP desarrollado con anterioridad.

Funcionalidad añadida: Los servidores TCP, tanto iterativos como concurrentes, presentan el siguiente problema. Si detenemos el servidor con `<ctrl>C`, el socket de escucha por donde llegan las conexiones entrantes no se libera hasta pasado un tiempo. En consecuencia, si detenemos el servidor, no podemos volver a iniciarlo inmediatamente, porque el puerto asociado al servicio sigue en uso.

Cuando pulsamos `<ctrl>C`, el sistema operativo envía una señal al servidor. Lo que haremos es crear una función que acepte esa señal y cierre los sockets con `shutdown()`. Hacerlo es sencillo y consta de dos pasos:

- Escribir una función que se activará cuando el usuario pulse `<ctrl>C`. Esa función, a la que llamaremos `signal_handler()`, recibe como parámetro de entrada un número que identifica la señal recibida. Si esa señal es `SIGINT`, deberá invocar a `shutdown()` para cerrar los dos sockets abiertos. Para que esta función pueda acceder a las variables que identifican a los dos sockets, dichas variables deberán declararse como variables globales, es decir, fuera del `main()`.
- Al principio del programa, hay que *registrar* nuestra función de captura de señales, indicando ante qué señal va a reaccionar. Para ello basta con invocar a la función `signal(2)`, pasándole dos parámetros: la señal y el nombre de la función, en nuestro caso, `signal_handler()`. Ver la página *man* de `signal` para más detalles.

4. Condiciones de entrega y evaluación

1. Esta práctica deberá realizarse en la máquina virtual instalada a principios de curso.
2. Los dos ficheros de código fuente (correspondientes al cliente y al servidor) deberán 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 deberán estar adecuadamente comentados. De lo contrario, se restarán dos puntos de la nota final.
4. Tanto el cliente como el servidor deberán compilar sin advertencias (opción `-Wall` del compilador `gcc`). De lo contrario, se restarán tres puntos de la nota final.
5. Cuando esté finalizada, se creará un fichero comprimido en formato ZIP que contenga exclusivamente los dos ficheros fuente (sin directorios) y se subirá al entorno virtual. Un fallo en las condiciones de entrega supondrá otro punto menos en la nota final.
6. El plazo de entrega finalizará el **jueves 14 de noviembre de 2019 a las 23:55**.
7. En la sesión del **viernes 15 de noviembre de 2019** se evaluará el trabajo de seis alumnos de manera *pseudoaleatoria*, por lo que la asistencia a esa sesión es obligatoria para todos los alumnos.
8. Esta práctica supondrá un 20% de la calificación de prácticas de la asignatura.
9. Se utilizará un sistema automático de detección de copias. En caso de copia, todos los alumnos involucrados figurarán como suspensos en la convocatoria ordinaria, debiendo presentar todas las prácticas por e-mail al profesor para poder presentarse a la convocatoria extraordinaria. Todas las prácticas se corregirán sobre 7.
10. No se admitirán entregas fuera de plazo.