# Arquitectura de Redes y Servicios

## Práctica Tema 5: Cliente-Servidor UDP Iterativo

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

5 de octubre de 2023

## Índice



## 1. Introducción

La práctica consiste en desarrollar un cliente y un servidor (iterativo) que permita hacer uso de un nuevo servicio utilizando sockets UDP.

El nuevo servicio, denominado `EchoCon`, añade una nueva funcionalidad al servicio estándar `echo`[1]. En este caso, el servidor recibirá una cadena de caracteres, realizará la conversión de mayúsculas a minúsculas (y viceversa) y devolverá el resultado al cliente. Dado que se trata de un servicio no estándar, no aparece listado en el fichero `/etc/services`. Por tanto, se le asociará el puerto: 5/udp[2].

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

## 2. Implementación del Cliente UDP

La primera tarea a realizar consiste en desarrollar el cliente EchoCon utilizando el protocolo UDP. Este cliente, implementado en el fichero `echocon-udp-client-apellidos.c`, se iniciará desde la terminal de comandos de la siguiente forma:

[1] Recibe una cadena de caracteres y la devuelve al cliente.
[2] El servicio `echo` recibe las peticiones a través del puerto: 7/udp.

```
./echocon-udp-client-apellidos ip_servidor [-p puerto_servidor] cadena
```

El cliente hará uso de la función `inet_aton(3)` para convertir la dirección IP recibida como parámetro (en notación punto) en un entero de 32 bits en formato 'network byte order'

Si el usuario no especifica el número de puerto del servidor, se utilizará por defecto el puerto 5. En caso contrario, el cliente utilizará como puerto del servidor el número proporcionado. Para ello, el cliente deberá transformar la cadena con el número de puerto a un entero sin signo de 16 bits en formato "network byte order". Esto se consigue mediante las funciones `sscanf(3)` y `htons(3)`

El tercer argumento es la cadena a transformar. Esta cadena tendrá, como máximo, 80 caracteres. El cliente enviará un datagrama UDP con dicha cadena al puerto y servidor correspondiente. El servidor enviará como respuesta un datagrama UDP con la cadena convertida según se indica en el apartado 1 de este documento. Los caracteres numéricos o especiales no se transformarán. Cuando el cliente reciba la nueva cadena del servidor, la mostrará por pantalla y finalizará la ejecución.

Para probar el funcionamiento del cliente, se puede intentar establecer la comunicación con el servidor desarrollado por el profesor (cuya IP es 10.0.25.250), que estará escuchando en el puerto 5/udp.

# 3. Implementación del Servidor UDP Iterativo

El siguiente paso consiste en el desarrollo del servidor EchoCon UDP. Este servidor, implementado en el fichero `echocon-udp-server-apellidos.c`, se iniciará del siguiente modo:

```
./echocon-udp-server-apellidos [-p puerto-servidor]
```

Si no se indica el número de puerto, el servidor utilizará por defecto el 5. A continuación, el servidor pondrá un socket UDP a la escucha en el puerto especificado y entrará en un bucle infinito. Cada vez que reciba un datagrama del cliente, convertirá la cadena tal como se indica en el apartado 1 de este documento y la enviará como respuesta. Tras ello, el servidor quedará a la espera del siguiente datagrama.

**Nota:** El comando "`netstat -tulpn`" muestra información sobre los puertos que están siendo utilizados en la máquina local. Para obtener toda la información es necesario ejecutar este comando como `root`.

**Nota:** Tanto el cliente como el servidor deben compilarse con la opción `-Wall`, que muestra por pantalla todas las advertencias del compilador. Se penalizarán las advertencias no resueltas.

# 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 5: 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: **22 de octubre de 2023 a las 23:55**.

9. No se admitirán entregas fuera de plazo.