

# Arquitectura de Redes y Servicios

## Práctica Tema 5: Cliente-Servidor UDP Iterativo

Grado en Ingeniería Informática  
Dpto. de Informática (ATC, CCIA, LSI)  
Universidad de Valladolid

Octubre, 2024

### Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Cliente UDP</b>	<b>1</b>
<b>3. Servidor Iterativo</b>	<b>2</b>
<b>4. Entrega y Evaluación</b>	<b>3</b>

## 1. Introducción

En esta práctica se va a aplicar el modelo Cliente-Servidor estudiado en el Tema 3 de la asignatura para llevar a cabo la implementación de un cliente y un servidor iterativo de forma que, mediante el empleo de sockets UDP, se pueda hacer uso del servicio `DAYTIME`. Este servicio tiene asociado un WKP a través del cual se pueden realizar peticiones para obtener la fecha y hora actuales. Su funcionamiento es muy simple: cuando el servidor recibe una petición de un cliente, envía como respuesta una cadena que contiene la fecha y la hora del día actual con el siguiente formato:

`Mon Oct 14 11:24:14 CEST 2024`

Tanto el cliente como el servidor se implementarán utilizando el lenguaje C. Es imprescindible que el alumno comprenda perfectamente el significado de cada línea del código desarrollado. Los únicos recursos necesarios para realizar la práctica son (a) los apuntes proporcionados por el profesor y (b) las páginas `man` ofrecidas por el Sistema Operativo.

## 2. Cliente UDP

La primera tarea a realizar consiste en escribir el código C necesario para implementar la funcionalidad del cliente UDP. El código debe estar en el fichero: `daytime-udp-client-grupoLab.c`, donde `grupoLab` será proporcionado por el profesor. Una vez compilado el código con `gcc`, el cliente se iniciará desde la terminal de comandos de la siguiente forma:

```
./daytime-udp-client-grupoLab ip_servidor [-p puerto_servidor]
```

Si no se especifica el número de puerto del servidor, el cliente obtendrá el puerto asociado al servicio DAYTIME mediante la función `getservbyname(3)`. En caso contrario, se utilizará como número de puerto el proporcionado tras la opción `-p`. El número indicado debe corresponder a un puerto libre del servidor<sup>1</sup>. En ambos casos, el número de puerto debe almacenarse como un entero sin signo de 16 bits en formato de red (*network byte order*). Esto se consigue con la función `htons(3)`, que recibe como único parámetro una variable de tipo entero y devuelve como resultado el número de puerto en formato de red. Además, se debe usar la función `inet_aton(3)` para convertir la IP del servidor proporcionada como parámetro (en notación punto) en un entero de 32 bits.

Tras almacenar los parámetros de entrada, el cliente creará un socket UDP y enviará al servidor un datagrama con la cadena: ¿Qué día es hoy?. A continuación, quedará a la espera del datagrama de respuesta enviado por el servidor. Cuando lo reciba, mostrará por pantalla el mensaje recibido y finalizará.

Para probar el funcionamiento del cliente, se puede utilizar el servidor desarrollado por el compañero de prácticas o el servidor disponible en la máquina virtual del profesor, cuya dirección IP es 10.0.25.250.

### 3. Servidor Iterativo

El siguiente paso consiste en escribir el código C necesario para implementar la funcionalidad del servidor UDP iterativo. El código debe estar en el fichero: `daytime-udp-server-grupoLab.c`, donde `grupoLab` será proporcionado por el profesor. Una vez compilado el código, el servidor se iniciará desde la terminal de comandos de la siguiente forma:

```
./daytime-udp-server-grupoLab [-p puerto]
```

Si no se proporciona el número de puerto, el servidor utilizará la función `getservbyname(3)` para obtener el puerto asociado al servicio DAYTIME. En otro caso, el número de puerto suministrado debe estar disponible y coincidir con el especificado en el cliente. A continuación, el servidor creará un socket UDP y entrará en bucle infinito donde, para cada datagrama recibido, generará una cadena formada por:

- Nombre del Servidor: obtener mediante `gethostname(2)`
- Fecha y Hora: utilizar la función `system(3)`, que permite ejecutar un comando del *shell* y redirigir la salida a un fichero. En este caso, el servicio DAYTIME se ofrecerá mediante la ejecución del comando `date`. El siguiente código muestra cómo se puede obtener la información generada por dicho comando:

```
#define BUFF_SIZE 30

FILE *fd;
char buff[BUFF_SIZE];
...
system("date > /tmp/date.txt");
fd = fopen("/tmp/date.txt", "r");
if (fgets(buff, BUFF_SIZE, fd) == NULL) {
    printf("Error en system(), fopen() o fgets()\n");
    exit(EXIT_FAILURE);
}
```

La cadena enviada como respuesta debe tener **exactamente** el siguiente aspecto:

```
vm25XY: Mon Oct 14 11:24:14 CEST 2024
```

---

<sup>1</sup>El comando `netstat -tulpn` muestra información sobre los puertos que están siendo utilizados.

Para finalizar el servidor existen dos opciones, en función de cómo se haya iniciado:

- 1º Plano: pulsar combinación de teclas `Ctrl+C`.
- 2º Plano: obtener ID del proceso con el comando: `ps aux | grep daytime-udp-server-grupoLab | head -1 | tr -s " " | cut -f2 -d " "` y, a continuación, finalizar su ejecución: `kill -9 ID`

**Nota:** tanto el cliente como el servidor deben compilarse incluyendo la opción `-Wall`, que permite mostrar por pantalla advertencias relacionadas con el código desarrollado.

## 4. Entrega y Evaluación

1. Esta práctica supone un **20 %** en la calificación final de las prácticas de la asignatura.
2. Los ficheros de código deben incluir una cabecera que indique el título de la práctica y el nombre de los componentes del grupo como se muestra a continuación:

```
/**
 * Practica Tema 5: DAYTIME UDP
 *
 * Apellido, Nombre
 * Apellido, Nombre
 *
 */
```

3. El código debe estar documentado mediante el uso de comentarios que justifiquen brevemente cómo se ha implementado toda la funcionalidad requerida. De lo contrario, se penalizará con un máximo de dos puntos la práctica.
4. Es fundamental que la práctica **funcione correctamente** en el entorno de máquinas virtuales. Los ficheros de código deben compilar sin mostrar advertencias. De lo contrario, se penalizará con dos puntos la práctica. La presencia de errores durante la compilación o ejecución del código supondrá una calificación igual a tres puntos con independencia del código desarrollado y del cumplimiento de los requisitos de entrega.
5. Una vez finalizada la práctica, se subirá a la tarea habilitada en el Campus Virtual un fichero comprimido en formato ZIP con el nombre `grupoLab-Prac5.zip`. Este fichero contendrá exclusivamente los dos ficheros de código fuente. Un fallo en las condiciones de entrega se penalizará con un punto.
6. Se hará uso de un sistema automático de detección de copias. En caso de copia, los estudiantes de los grupos involucrados figurarán como suspensos en la convocatoria ordinaria, debiendo enviar todas las prácticas por correo al profesor para poder presentarse en la convocatoria extraordinaria.
7. El profesor se reserva el derecho de citar a aquellos grupos de prácticas para los que considere necesario realizar una revisión de la práctica.
8. Fecha de Entrega: **28 de Octubre de 2024 a las 23:55** (No se admitirán entregas fuera de plazo)