

## Práctica 6

# Programación con sockets

En esta práctica utilizaremos la interfaz de programación de *sockets* de Berkeley con el objetivo adicional de que nuestras aplicaciones sean compatibles tanto con IPv4 como con IPv6.

Esta práctica la podemos realizar en la máquina anfitriona para que resulte más fácil la programación de las aplicaciones.

### 6.1. Protocolo UDP

Haciendo uso del ejemplo visto en clase, desarrollar una aplicación cliente-servidor con el protocolo UDP de manera que el servido quede a la escucha en el puerto pasado como primer parámetro. Esperará mensajes desde el cliente consistentes en un solo carácter. Si el carácter es 'h', el servidor enviará una respuesta con la hora actual; si el carácter es 'd', enviará la fecha; si el carácter es 'q', terminará la ejecución. Consultar las páginas del manual de las funciones `time(2)`, `localtime(3)` y `strftime(3)`. Además, el servidor mostrará la dirección IP y el puerto del cliente mediante la función `getnameinfo(3)`. Éste, por su parte, aceptará 3 parámetros: el nombre o dirección IP del servidor, el número de puerto y el mensaje ('h', 'd' ó 'q') que enviará al servidor. Mostrará la respuesta recibida así como la dirección IP y el número de puerto del servidor.

Ejemplo de ejecución:

#### Servidor

```
./udp_time_server 9100
Received 2 bytes from ::1 port 56780
Result string is "10:37:24"
Received 2 bytes from ::1 port 41599
Result string is "09/04/2018"
Received 2 bytes from ::1 port 46439
Unknown command: e
Received 2 bytes from ::1 port 59198
```

**Cliente**

```
./udp_time_client localhost 9100 h
```

```
Received 8 bytes: 10:37:24
```

```
./udp_time_client localhost 9100 d
```

```
Received 10 bytes: 09/04/2018
```

```
./udp_time_client localhost 9100 e
```

```
Received 2 bytes: e
```

```
./udp_time_client localhost 9100 q
```

```
Received 2 bytes: q
```

## 6.2. Protocolo TCP

Repetir el ejercicio anterior pero utilizando el protocolo TCP. Consultar las páginas del manual de la función `bind(2)`. Comprobar mediante `wireshark` los mensajes intercambiados entre el cliente y el servidor.