SOCKETS

Sistemas operativos

Antoni Lorenzana

laSalle ENG
Universitat Ramon Llull

Introducción

- ¿Qué *file descriptors* conocemos?
  - ✓Pipes
  - ✓I/O locales (pantalla, teclado, etc.)
  - ✓Ficheros (.txt, .dat, etc.)

PROBLEMA!

Introducción

¿Cómo conectamos dos procesos que funcionan en diferentes sistemas? ¿Y en diferentes ciudades?

PROBLEMA!

SOLUCIÓN: SOCKETS

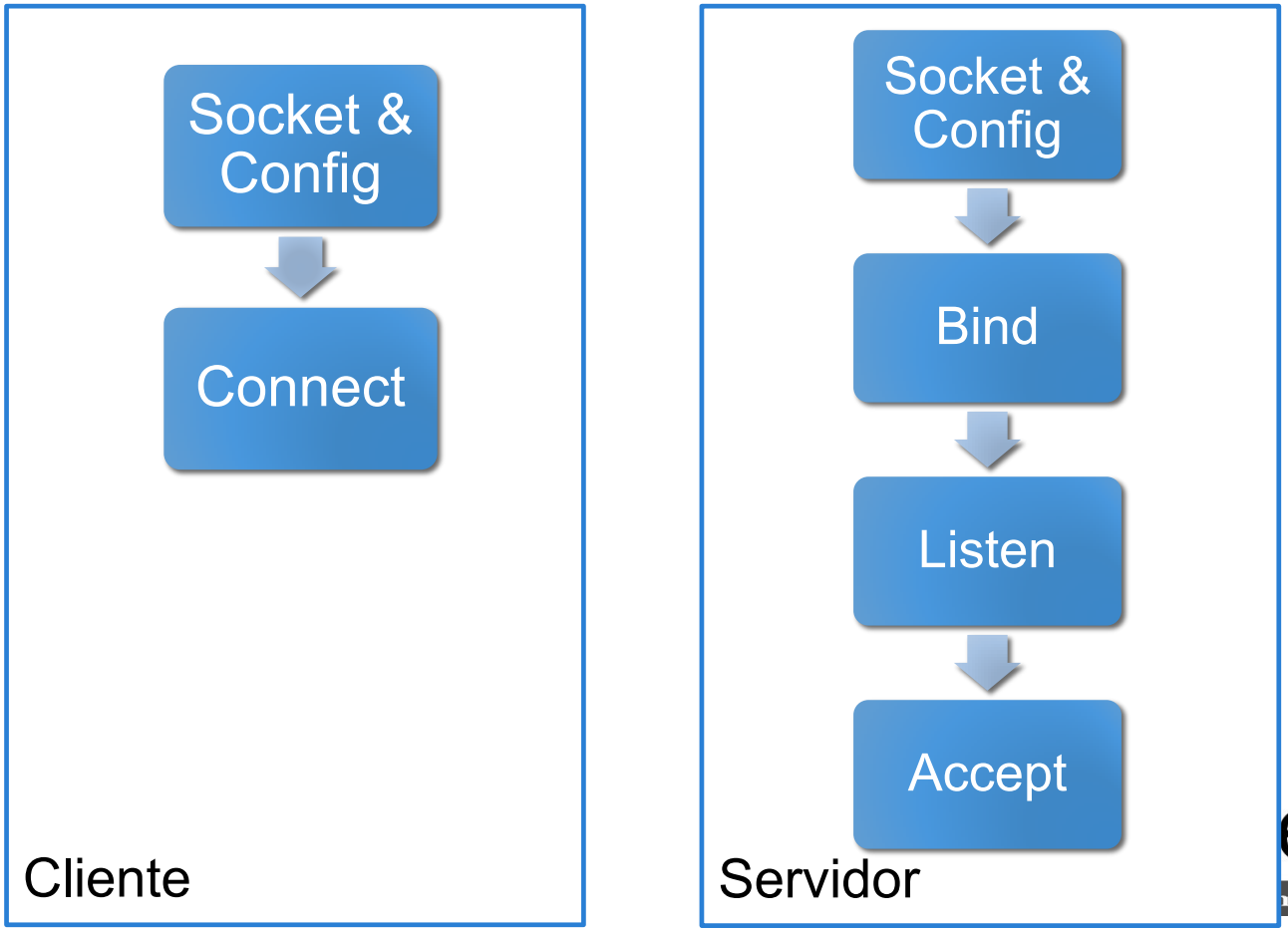
Introducción

- ¿Qué es un socket?
  - ✓Es una herramienta de comunicación
- ¿Qué tienen de nuevo?
  - ✓Comunicaciones locales y remotas
  - ✓Configurados mediante IP y puerto
  - ✓Son un file descriptor

Tipos de socket

- Orientados a conexión (TCP)
  - ✓Se establece una conexión entre dos puntos y se garantiza que los datagramas enviados se reciban correctamente
- No orientados a conexión (UDP)
  - ✓No se establece una conexión
  - ✓No se garantiza que la información llegue correctamente
  - ✓Es más rápido

Creación de sockets



Funciones: socket

- Función: devuelve un file descriptor sin conexión.
- Prototipo:

```
int socket (int family, int type, int protocol)
```

  - family: AF_INET, AF_INET6, etc.
  - type: especifica el tipo de conexión
    - ✓ SOCK_STREAM (orientado a conexión)
  - protocolo: define el protocolo a usar
    - ✓ 0 (por defecto lo determina el S.O.)
  - Return: fd del socket creado

Funciones: connect (TCP)

- Función: ejecuta el Three-way-handshake si el destino lo permite
- Prototipo:

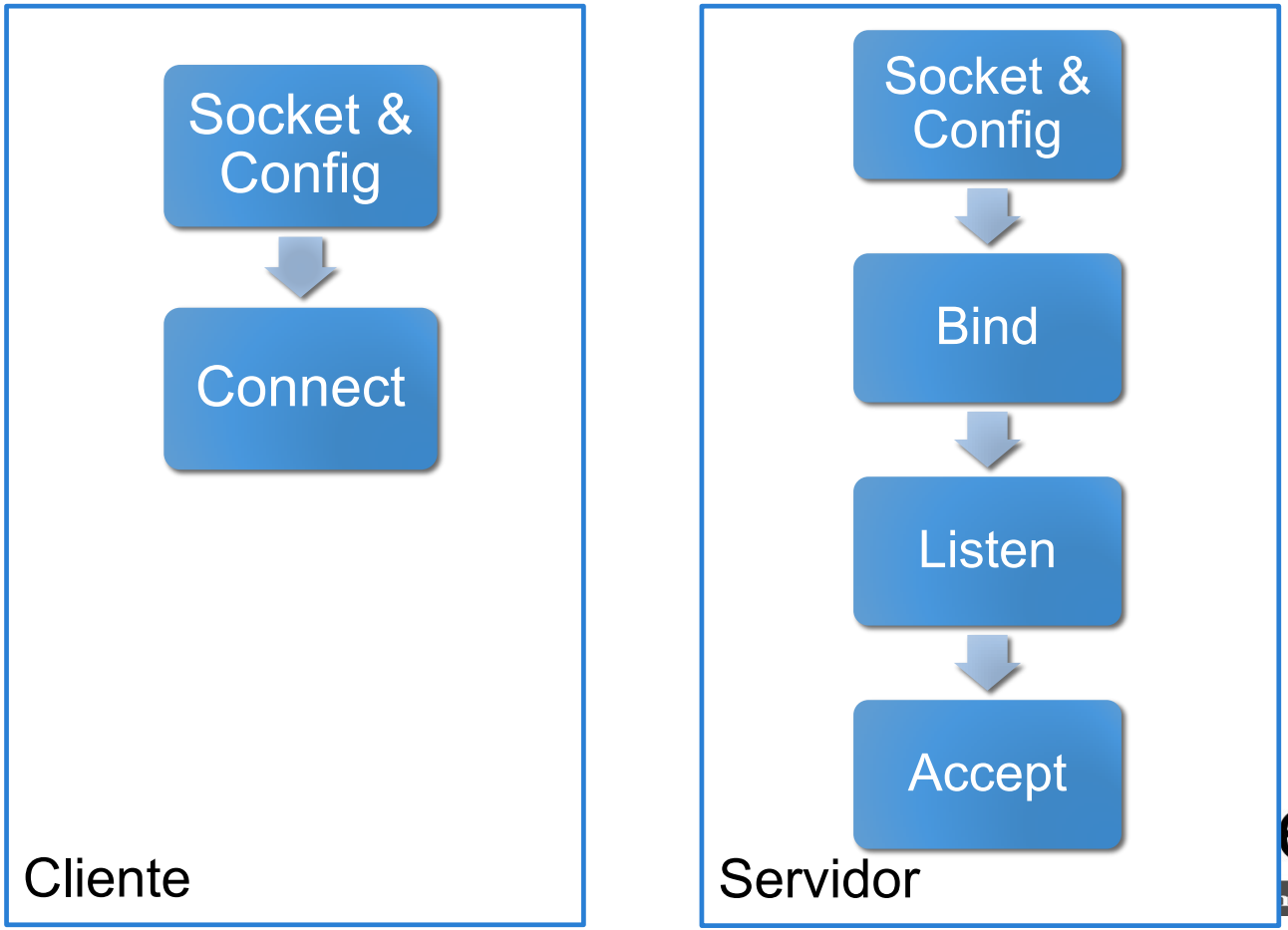
```
int connect (int sockfd, const struct sockaddr * servaddr, socklen_t addrlen)
```

  - sockfd: es el file descriptor del socket
  - servaddr: registro de configuración con la informacion del destino
  - addrlen: tamaño de la estructura servaddr
  - Return: retorna -1 en caso de error y 0 en caso contrario

Implementación:

- server.sin_addr.s_addr = inet_addr ("127.0.0.1");
- struct sockaddr_in s_addr;
- s_addr.sin_family = AF_INET;
- s_addr.sin_port = htons (PUERTO);

Creación de sockets



Estructura de configuración

- Función: esta estructura sirve para configurar diversos parámetros de los socket como la IP, puerto, etc.
- Definición:

```
struct sockaddr_in {
        uint8_t sin_len; //Tamaño del objeto
        sa_family_t sin_family;//Familia del socket
        in_port_t sin_port; //Numero del puerto en Big Endian
        struct in_addr sin_addr; //Struct con entero indicando @IP
        char sin_zero[8]; //'\0'
};
```

Funciones útiles

- uint16_t htons (uint16_t port);
  - Cambia el formato del entero a big endian
- int inet_aton (const char * cp, struct in_addr * inp);
  - Convierte una cadena en formato binario
- struct hostent * gethostbyname (const char *name);
  - Retorna la información de un host a partir de una dirección
  - Campo h_addr es igual a sin_addr.s_addr

Funciones: bind

- Función: indica al S.O. que la aplicación espera información de un puerto determinado
- Prototipo:

```
int bind (int sockfd, const struct sockaddr * myaddr, socklen_t addrlen)
```

  - sockf: el fd del socket
  - myaddr: es necesario cast a (void *), uso de INADDR_ANY como IP
  - addrlen: tamaño de la estructura myaddr
  - Return: retorna -1 en caso de error y 0 en caso contrario

Implementación:

- struct sockaddr_in s_addr;
- s_addr.sin_family = AF_INET;
- s_addr.sin_port = htons (PORT);
- s_addr.sin_addr.s_addr = INADDR_ANY;

Funciones: listen

- Función: abre el puerto asignado al socket y determina el numero máximo de conexiones a aceptar
- Prototipo:

int listen (int sockfd, int backlog)

  - sockfd: el fd del socket
  - backlog: conexiones en cola pendientes de aceptar

Funciones: accept

- Función: crea un nuevo file descriptor con conexión activa, función bloqueante
- Prototipo:

```
int accept (int sockfd, struct sockaddr * cliaddr, socklen_t * addrlen)
```

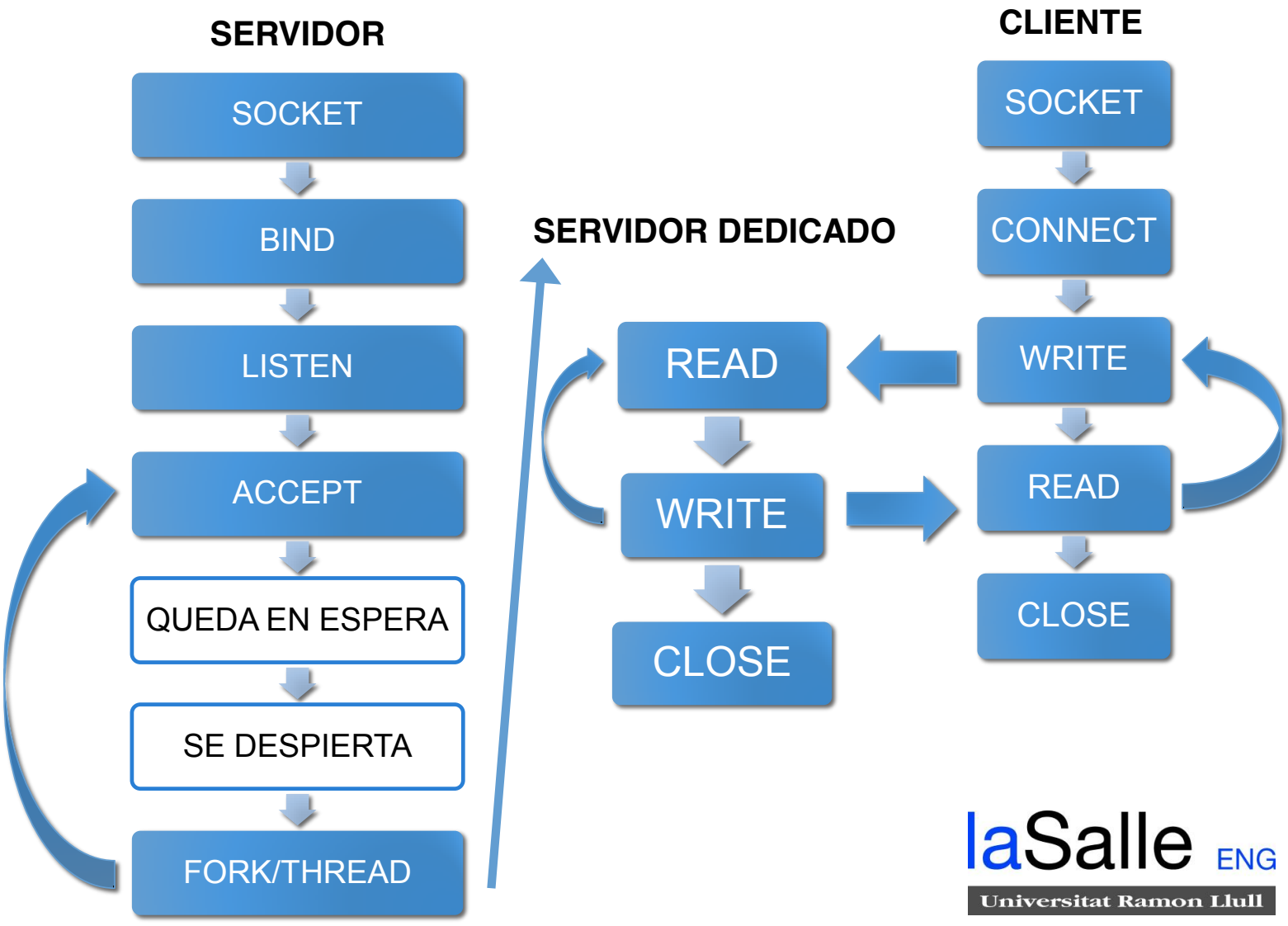
  - sockfd: fd del socket
  - cliaddr: indica la dirección del que a solicitado la conexión (void *)
  - addrlen: tamaño de la estructura cliaddr
  - Return: nuevo fd activo para comunicarse

  Implementación:
  - struct sockaddr_in s_addr;
  - socklen_t len = sizeof (s_addr);
  - int newsock = accept (sockfd, (void *) &s_addr, &len);

Funciones: close

- Función:
  - Socket pasivo: no se aceptan más conexiones
  - Socket activo: se cierra la conexión activa
- Prototipo:

int close (int sockfd)



¿Preguntas?

laSalle ENG
Universitat Ramon Llull