

# ***Sistemas Operativos*** ***2020/2021***

## Práctica 2

Comunicaciones en red  
Y  
Concurrencia

## “COMUNICACIONES EN RED”

Implementar una aplicación cliente-servidor mediante la utilización de sockets cuya funcionalidad será la ejecución remota de comandos sin parámetros. El ejercicio constará de dos partes (procesos): el cliente que enviará el comando a ejecutar al servidor y el servidor que recibirá la petición desde el cliente, ejecutará el comando y devolverá los resultados al cliente.

El cliente se ejecutará en la máquina local y su función es establecer la comunicación con el servidor, enviar al servidor el comando, recibir los resultados y mostrarlos por pantalla. Para ello, el usuario escribirá en el prompt el comando:

### **ClienteRemoto IP\_Servidor Comando**

El servidor se lanzará con la orden **ServidorRemoto** y debe estar en todo momento escuchando por el puerto 9999, preparado para aceptar conexiones. Tras una petición de conexión por parte de un cliente, debe crear un hijo que será el encargado de realizar todas las operaciones necesarias relacionadas con la ejecución de comando solicitado por el cliente.

La puntuación del ejercicio será la siguiente:

- Implementación correcta del cliente y su estructura. **(1,5 puntos)**
- Implementación correcta del servidor y su estructura. **(1,5 puntos)**
- Ejecución remota de comandos. **(3 puntos)**

## **LLAMADAS AL SISTEMA RELACIONADAS CON LAS COMUNICACIONES EN RED**

### **Estructuras necesarias**

<sys/socket.h>,            **struct sockaddr.**  
<netinet/in.h>,           **struct in\_addr; struct sockaddr\_in** (incluye un campo in\_addr).  
<sys/types.h>

### **Llamadas al sistema principales**

#### **socket**

Permite establecer el canal de comunicación, está definida en <sys/socket.h> y necesita <sys/types.h>.

Ejemplo:

```
conector = socket(AF_INET, SOCK_STREAM, 0) //(Familia protocolos, tipo  
transporte, protocolo).
```

#### **bind**

Se emplea para comunicar a la red la dirección del canal. Utiliza como parámetro la estructura sockaddr\_in definida en el archivo de cabecera <netinet/in.h>

Bind(conector, addr, addrlen) //(descriptor del conector, dirección servidor, longitud dir)

#### **listen**

Disponibilidad para recibir peticiones de servicio.

listen(conector, tamañoCola) // (descriptor del conector, cola peticiones pendientes)

## **accept**

Bloquea al servidor en espera de peticiones de conexión por parte de los clientes.

`accept(conector, addr, addrlen) //(descriptor del conector, dir. servidor, longitud dir)`

## **connect**

Es invocada por el proceso cliente para establecer una conexión, el servidor debe estar esperando.

`connect(conector_cliente, addr, addrlen) // (descriptor conector, dirección servidor, longitud)`

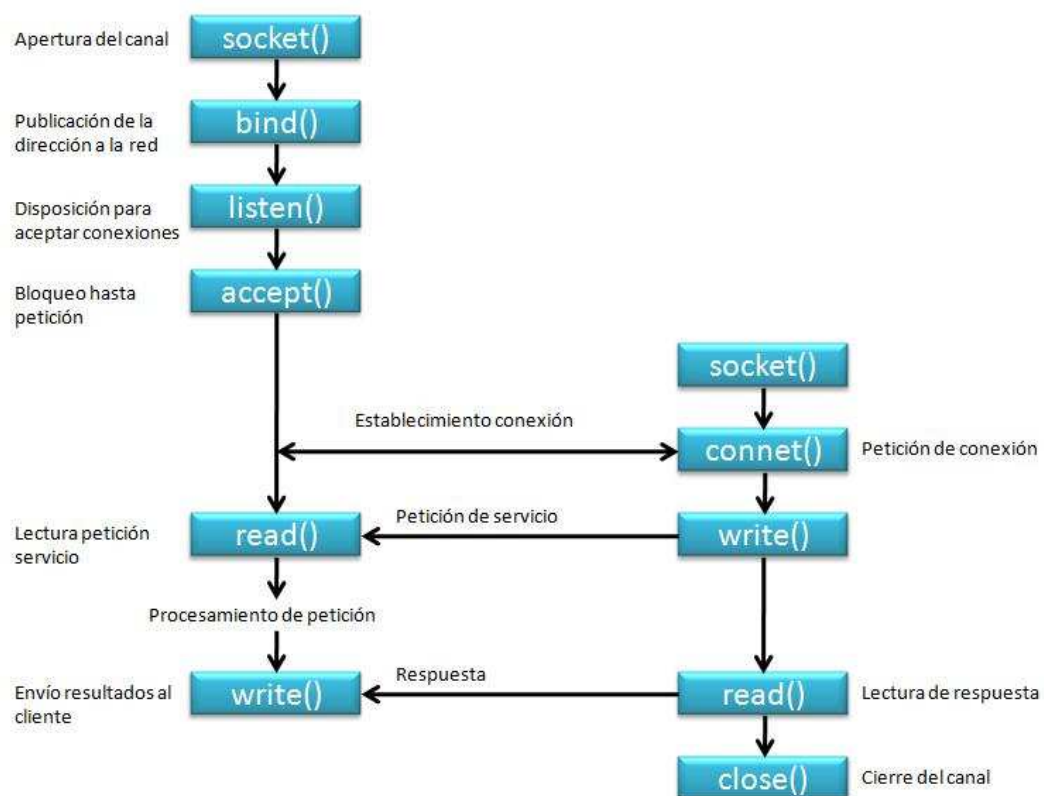
## **Funciones de ayuda**

`inet_addr( cadena) //traduce una cadena de caracteres en dirección IP.`

`inet_ntoa( in_addr) //conversión inversa.`

`htonl, htons, ntohl, ntohs.` Convierten datos unsigned long y unsigned short del formato de host al formato de la red y viceversa.

## **Estructura de toda aplicación cliente-servidor**



# “CONCURRENCIA”

**Sincronización de procesos. Semáforos. Diseñar un programa de concurrencia mediante el entorno JBACI.**

<https://code.google.com/archive/p/jbaci/downloads/>

## **Enunciado**

En una nave industrial hay una máquina de inyectar que produce piezas de una en una y las deja en una cinta transportadora de tamaño 10. Además, hay un robot que coge las piezas de una en una y las deja en la caja de embalaje. Implementar mediante semáforos la solución al problema suponiendo que la máquina produce 100 piezas y que la cinta es la sección crítica.

## **Notas**

Para la corrección y calificación del ejercicio además de la corrección del programa se valorarán los siguientes aspectos:

- Uso correcto de los semáforos atendiendo a los dos principales aspectos de exclusión mutua y sincronización
- Uso del interfaz gráfico para la claridad del problema

La puntuación del ejercicio será de **4 puntos**.

## **NORMAS DE ENTREGA**

- La entrega de la práctica será durante la semana del **16 al 20 de noviembre**.
- Se entregará una **memoria** indicando cómo se han resuelto los distintos ejercicios.