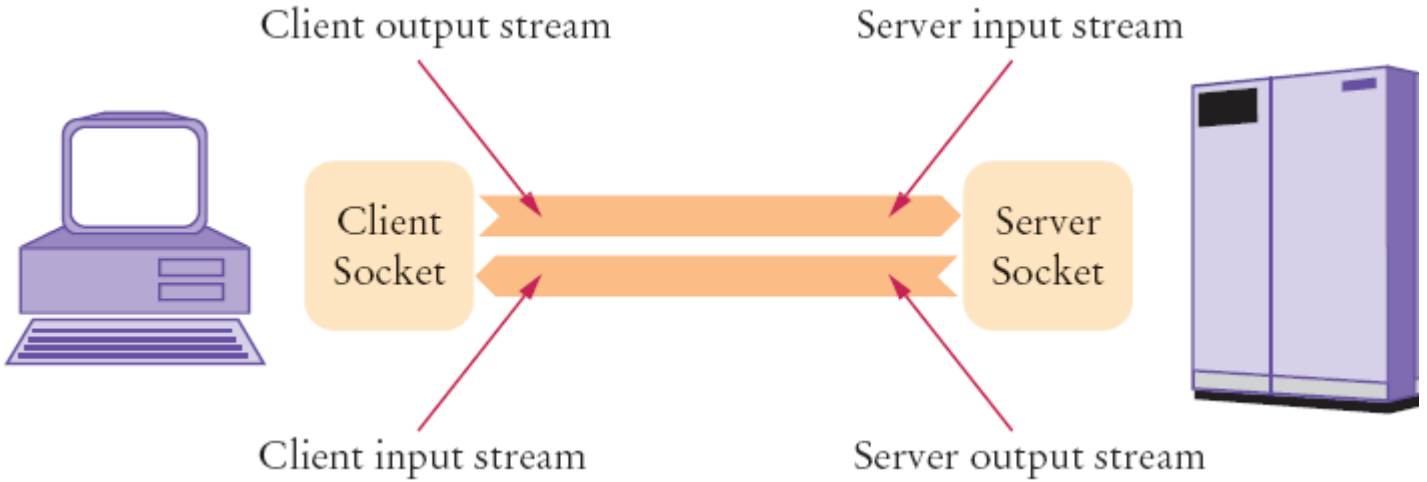


Comunicación entre procesos

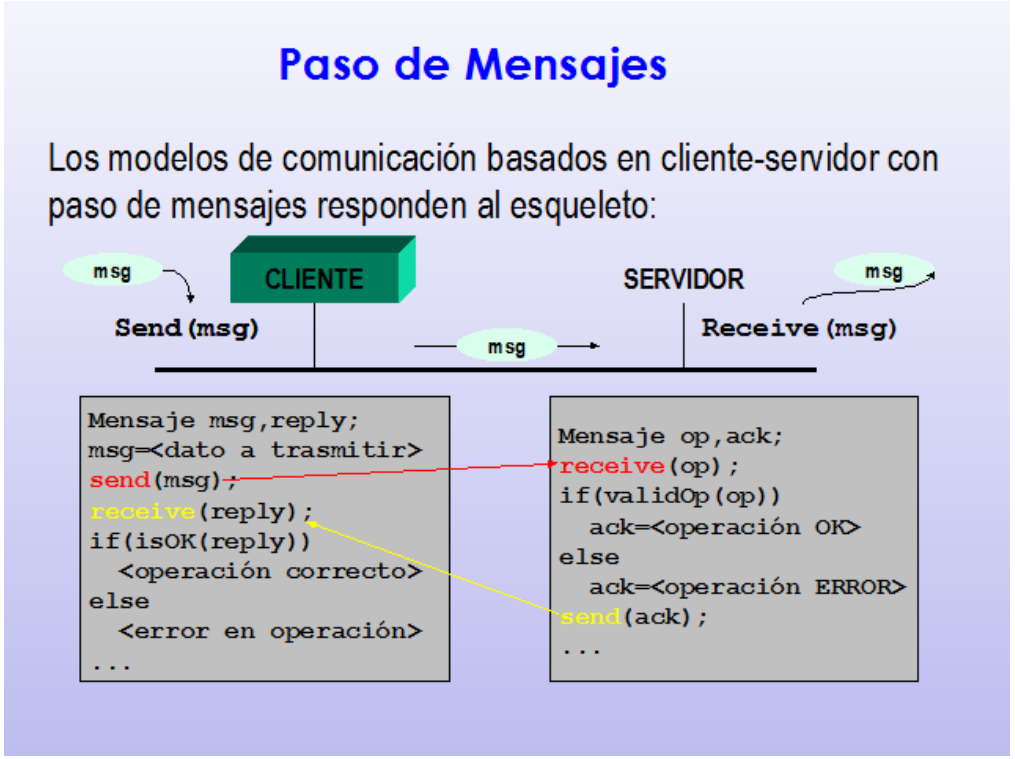
Edwin René Guamán Quinche

rene525456@gmail.com

@rene5254

Introducción

- Comunicación entre procesos es el núcleo de todos los sistemas distribuidos



- En SSDD la comunicación está basada en el paso de mensajes
- Es la cooperación entre procesos para lograr un objetivo global

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Comunicación en SSDD

Red de comunicación es el elemento fundamental para la comunicación entre procesos

Dos modelos de comunicación entre procesos

- Memoria Compartida
- Zona común en el que los mensajes podrán ser compartidos entre cliente y servidor

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Pase de mensajes

- Conjunto de mecanismos para comunicar procesos mediante un enlace de comunicación

- Debe identificar el proceso origen y destino en la primitivas **recibir** o **enviar**

- Sincroniza un proceso que recibe el mensaje y otro que lo envía

Comunicación en SSDD

Pase de mensajes: es una técnica de sincronización entre procesos y permite la exclusión mutua, su principal característica es que no requiere memoria compartida

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

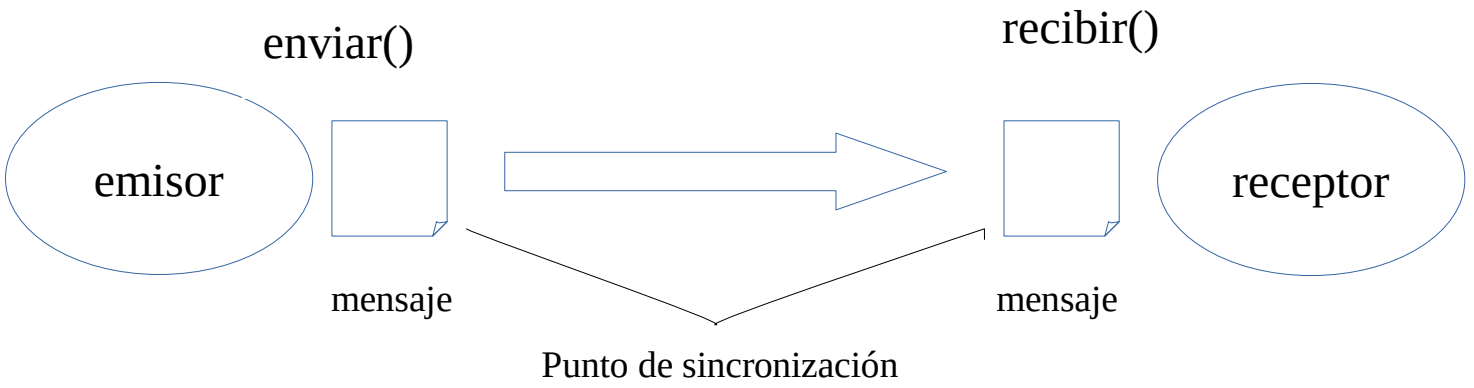
Pase de mensajes

- Conjunto de mecanismos para comunicar procesos mediante un enlace de comunicación

- Debe identificar el proceso origen y destino en la **primitivas recibir** o **enviar**

- Los sockets UNIX soportan dos formas de comunicación: identificando un socket dentro del sistema de ficheros o asociando un puerto de comunicación

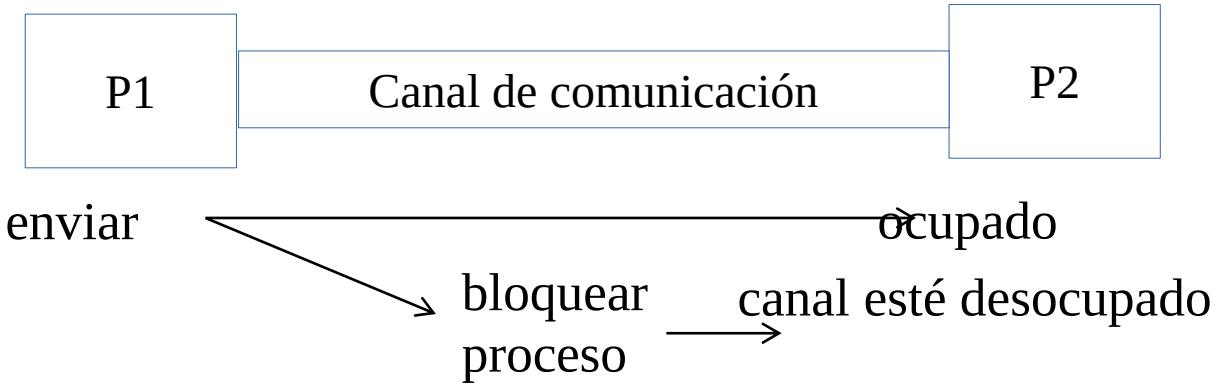
Niveles de sincronización



Síncrona: cada operación se completan cuando se completa el par de **enviar**() y **recibir**()

Asíncrona: pueden completarse por separado

Modo de sincronización

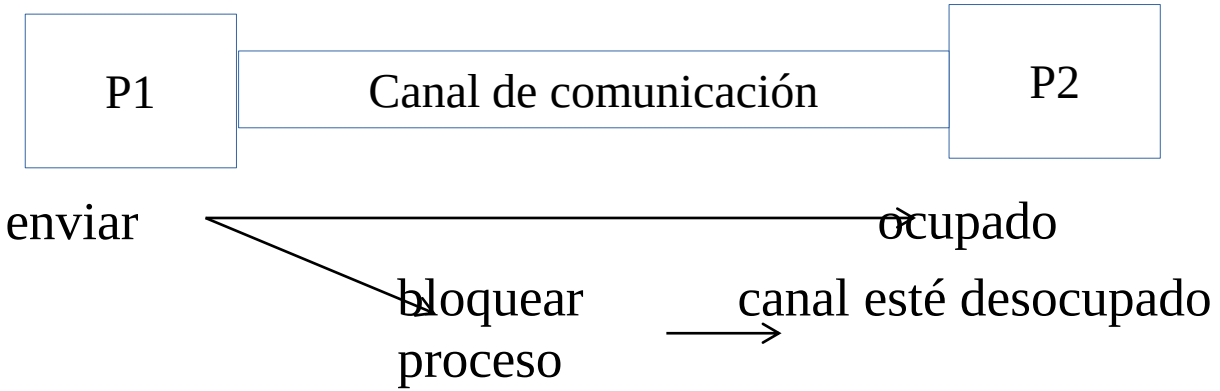


Modo bloqueante o comunicación síncrona

Canal de comunicación está ocupado, enviar puede bloquear al proceso hasta que se libere el canal

- Depende también de si permite buffering o no)
- Se deposita el mensaje (**modo bloqueante** o síncrono).

Modo de sincronización



Modo bloqueante o comunicación síncrona

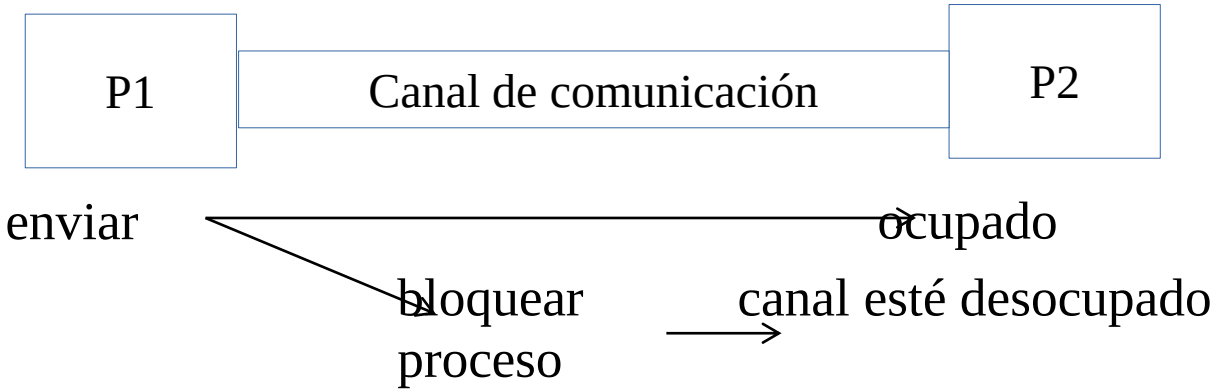
Los procesos de envío y recepción se sincronizan en cada mensaje

- El send y recibe son operaciones bloqueantes

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Modo de sincronización



Modo bloqueante o comunicación síncrona

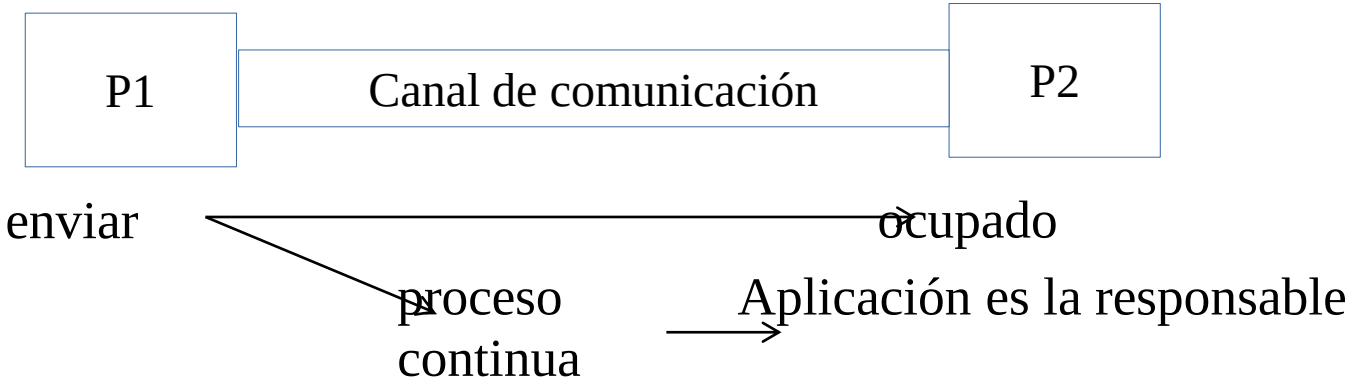
El send es ejecutado, el proceso de envío está bloqueado hasta que la recepción correspondiente se emita

Cada vez que un recibe es ejecutado, se bloquea el proceso de recepción hasta que llegue el mensaje

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Modo de sincronización



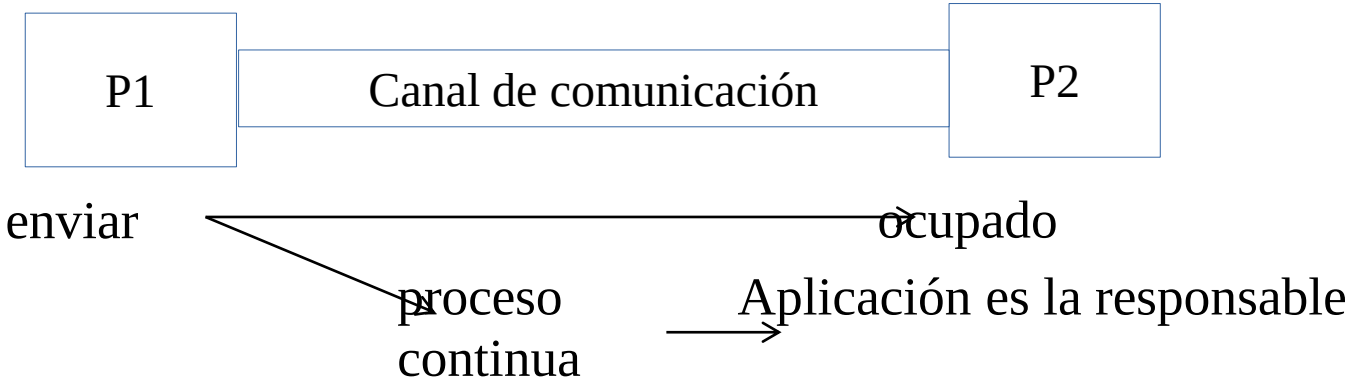
modo no bloqueante o comunicación asíncrona

- Permite que el proceso continúe aunque el mensaje **NO** se haya podido **enviar**
- Transfiriendo a la aplicación la responsabilidad de gestionar la sincronización en el uso del buffer de usuario donde se ubica el mensaje enviad

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Modo de sincronización



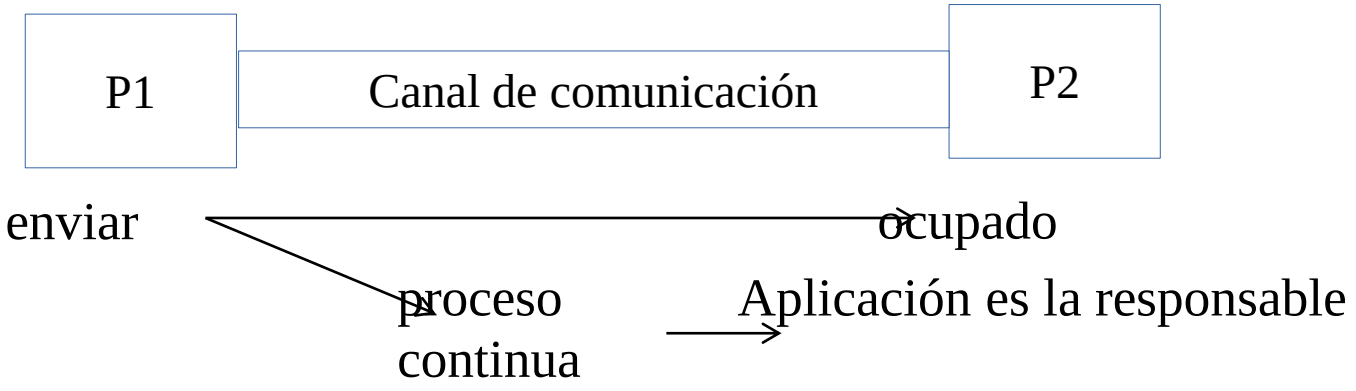
modo no bloqueante o comunicación asíncrona

- La operación send() es no bloqueante
- El envío termina tan pronto como el mensaje ha sido copiado por el buffer local
- La transmisión del mensaje se realiza en paralelo con el procesos de envio.

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Modo de sincronización



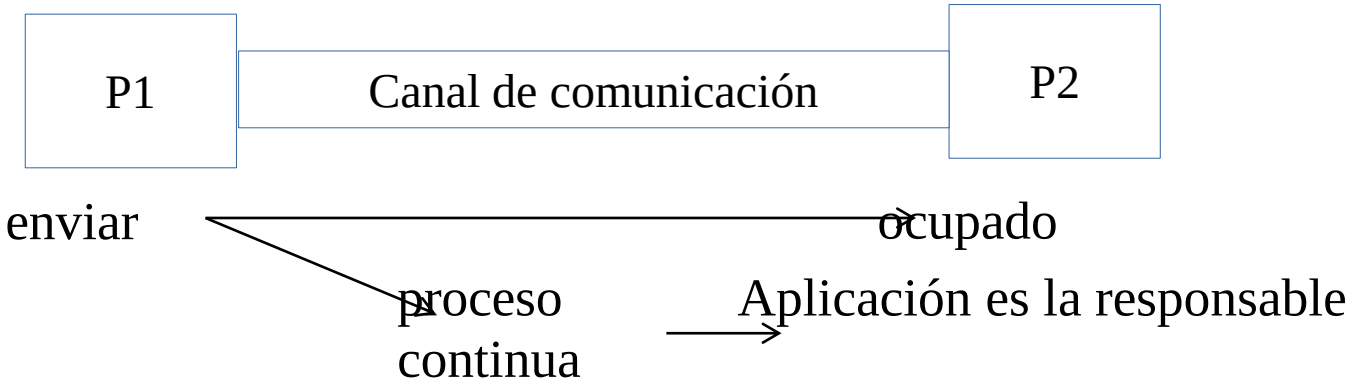
modo no bloqueante o comunicación asíncrona

- La operación receive() no puede ser bloqueante o no bloqueante
- No bloqueante: la recepción se realiza normalmente aun se hay ejecutado la operación receive()
- Bloqueante: la recepción se bloquea hasta que llegue el mensaje

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Modo de sincronización



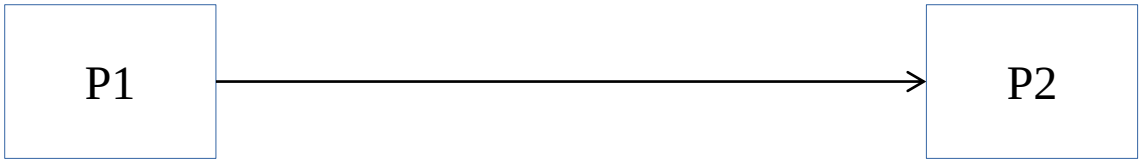
- Los sockets de UNIX son en principio bloqueantes, pero pueden configurarse como no bloqueantes.

Fiabilidad

- En la comunicación, el pase de mensajes depende del soporte que proporcione la red

- UNIX ofrece dos formas de comunicación con sockets:
- Comunicación **orientada a conexión**, basada en TCP/IP, fiable
- Comunicación por **datagramas**, basada en UDP/IP, no fiable.

Modo de comunicación



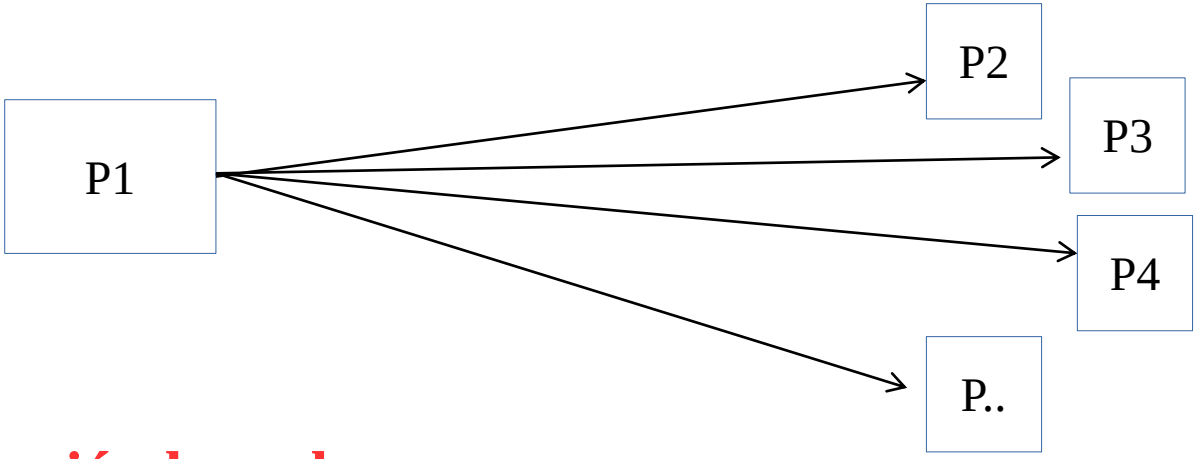
Comunicación unicast

- El proceso por el cual se envía un paquete de un host a un host individual
- Es una transmisión punto a punto con cada destinatario

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

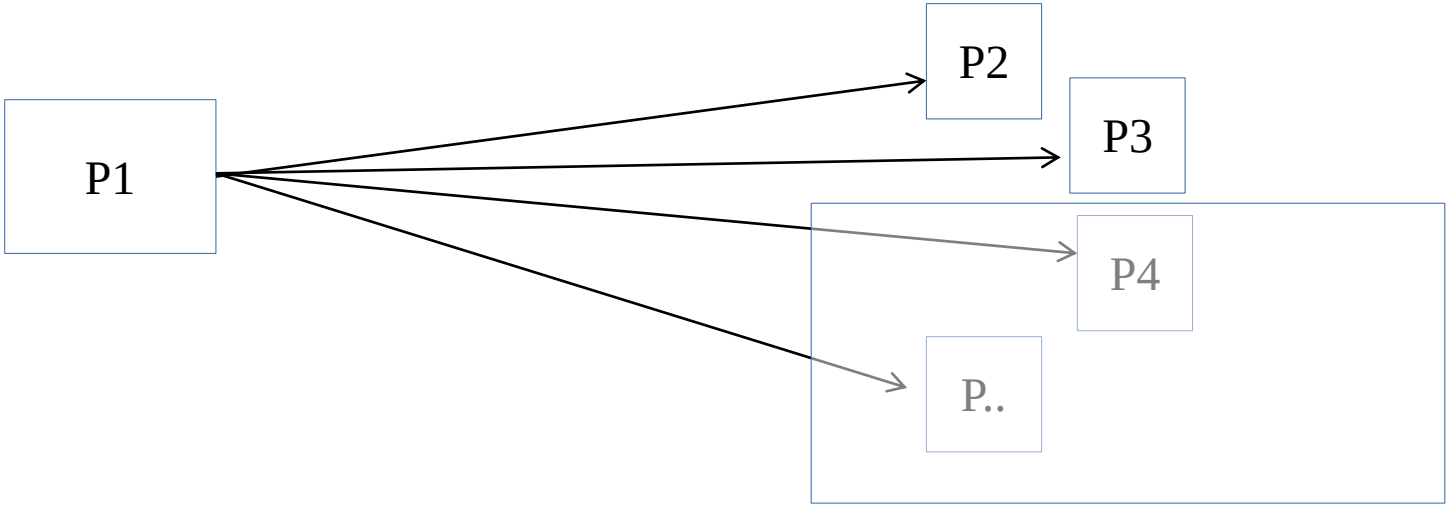
Modo de comunicación



Comunicación broadcast

- El proceso por el cual se envía un paquete de un host a todos los host de la red
- Se basa en un único proceso de envío, independientemente del número de potenciales máquinas receptoras, de una misma información
- Desde la máquina origen sólo se envía una vez la pertinente información y no se transmiten “n”copias de la misma aunque haya “n” destinatarios

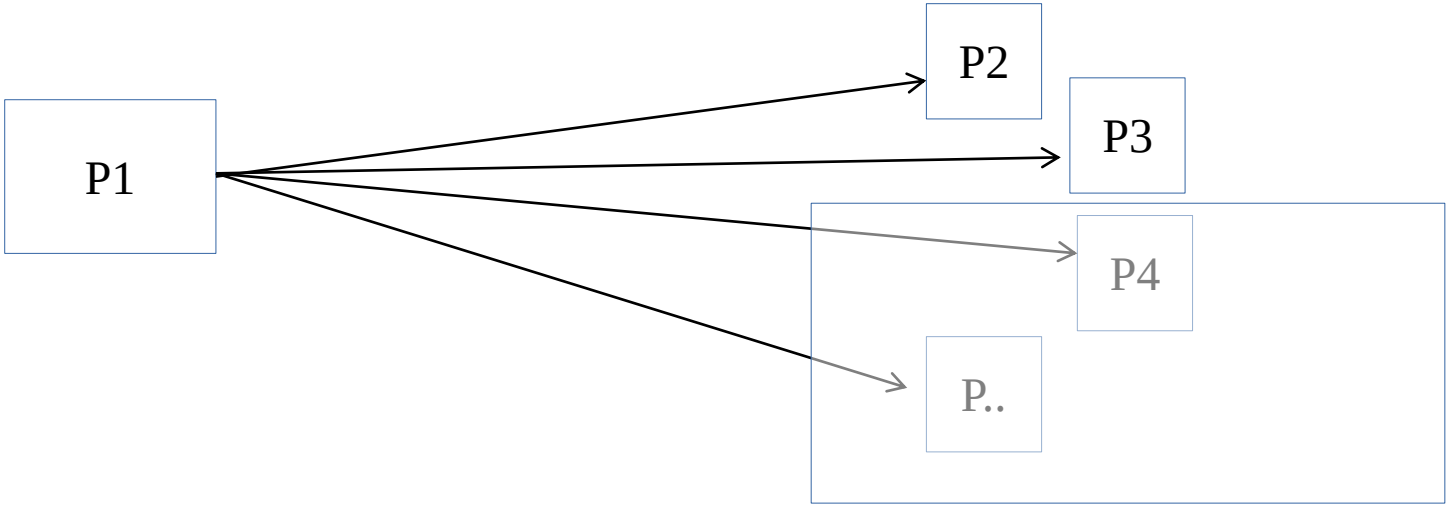
Modo de comunicación



Comunicación multicast

- El proceso por el cual se envía un paquete de un host a un grupo seleccionado de hosts

Modo de comunicación



El soporte para **multicast** es muy útil en sistemas Replicados. Como ejemplo, el protocolo IP reserva un conjunto de direcciones para multicast.

ESPAMMFL

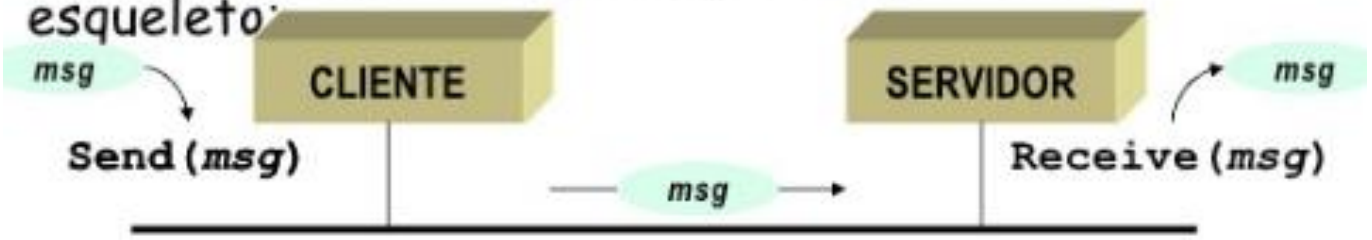
MTI
Maestría en Tecnologías de la Información

Primitivas para comunicación

CanalComunicación
enviar() recibir() conectar() desconectar()

Algoritmos de pase de mensajes

Los modelos de comunicación basados en cliente-servidor con paso de mensajes responden al esqueleto:




```
Mensaje msg,reply;
msg=<dato a trasmitir>
send(msg);
receive(reply);
if(isOK(reply))
  <operación correcto>
else
  <error en operación>
...
```

```
Mensaje op,ack;
receive(op);
if(validOp(op))
  ack=<operación OK>
else
  ack=<operación ERROR>
send(ack);
...
```

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

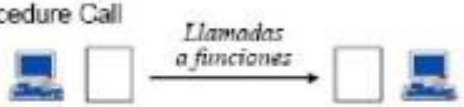
Tipos de pases de Mensajes

Paso de mensajes puro

- Sockets

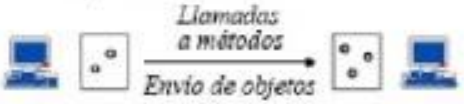
Llamada a procedimientos remotos

- Remote Procedure Call



Invocación a métodos remotos

- Objetos distribuidos (RMI, CORBA)



Puerto

Códigos de identificación que generalmente está asignado de antemano por el fabricante de la aplicación que genera el proceso

- Ftp (80)
- Smtp(25)
- Www(80)
- Telnet(23)
- Pop3(110)
- Mysql(3306)

Los puertos del 1 al 1024 están reservados para el Sistema Operativo, los demás puertos se pueden utilizar en cualquier aplicación

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Hilos

Se define como una secuencia única de control de flujo dentro de un programa, en un programa puede haber más de una secuencia de control de hilos

Hilos: comparten los mismos recursos del programa que las contiene

Procesos: tienen separado su código, así como sus datos

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Hilos

Hilos: comparten los mismos recursos del programa que las contiene

Procesos: tienen separado su código, así como sus datos

Se identifican dos tipos de hilos:

- **Flujo único**: aplicación utiliza únicamente un hilo para controlar el su ejecución

- **Flujo múltiple**: aplicación utilizan varios contextos de ejecución para realizar su trabajo

Hilos

En un sistemas multihilos:

- Cada tarea se inicia y termina tan pronto como sea posible
- Tiene un hilo principal del programa en ejecución
- Hay hilos o tareas paralelas en ejecución
- Facilita la entrada de datos en sistemas en tiempo real, especialmente si estos datos provienen de diferentes fuentes

Hilos

Un hilo:

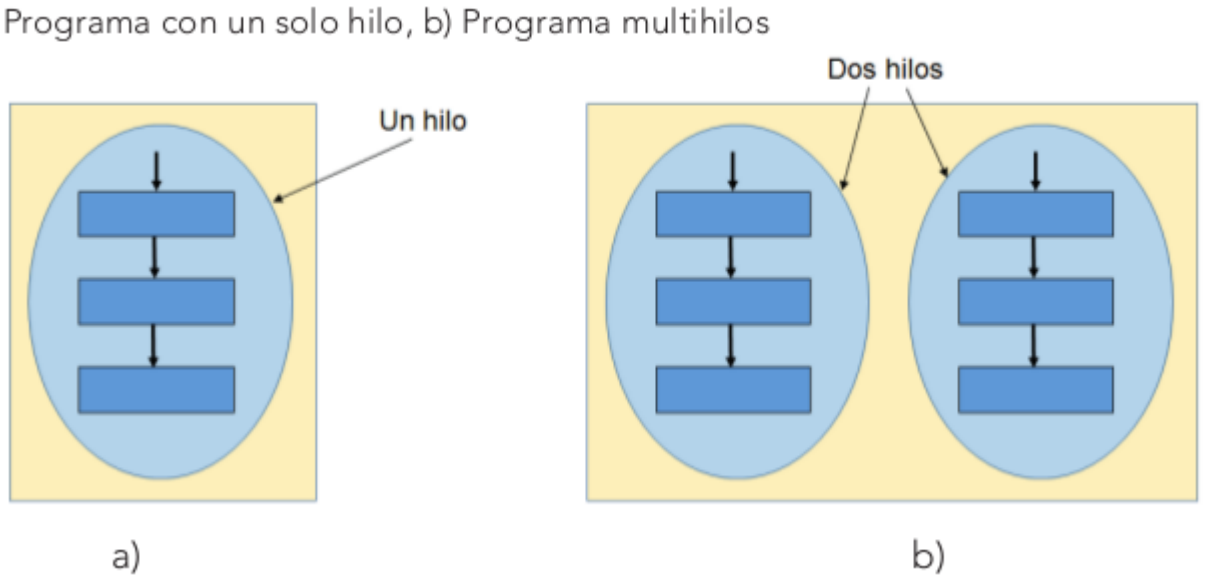
- Es una unidad de código más pequeña que se pueda ejecutar en un entorno multitareas

- Permite escribir programas más eficientes

- Permite optimizar recursos tan importantes como el mejor desempeño del CPU al minimizar sus tiempos de inactividad

- Permite sincronizar la diferencia entre la velocidad de transmisión de la red con las de procesamientos del CPU

Hilos



Una de las razones de importancia para el estudio de la programación multihilos es que permite acceder a los recursos de tiempo libre de la CPU mientras se realizan otras tareas

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Interfaz de programación (API)

- Las aplicaciones clientes – servidor se vale de protocolos de transporte para comunicarse (UDP y TCP)

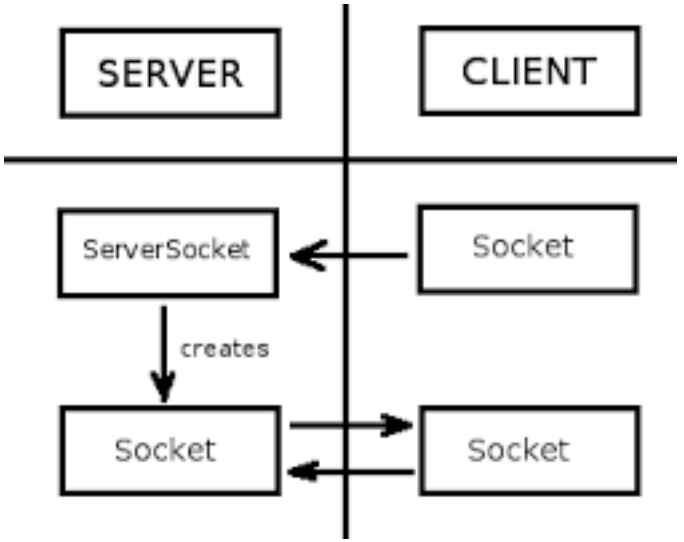
- Las aplicaciones deben especificar los detalles sobre el tipo de servicio, datos a transmitir y el receptor donde situar los datos

- La interfaz entre la aplicación y los protocolos de comunicación de un sistemas operativo es **Interfaz de programación de aplicaciones (API)**

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Interfaz de socket



- **Socket**: punto de referencia hacia donde los mensajes pueden ser enviados o donde pueden ser recibidos

- Al llamar a un procedimiento socket, el control pasa a una rutina de la biblioteca de sockets, que realiza las llamadas al SO para implementar el socket

Funciones de los sockets

socket()

- Esta subrutina se usa para crear un socket y regresa un descriptor correspondiente a este socket
- El descriptor es usado en el lado del cliente y en el lado del servidor
- El descriptor de archivo es el final de un canal de comunicación
- La rutina retorna -1 si ocurres un error

Funciones de los sockets

close()

- Indica al sistema que el uso del socket debe de ser finalizado
- Si se usa un TCP, close termina la conexión antes de cerrarlo
- Cuando se cierra el socket, el descriptor se libera
- La aplicación ya no transmite datos
- El protocolo de transporte yo no acepta mensajes de entrada para el socket

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

Funciones de los sockets

listen()

- Esta subrutina prepara al socket para aceptar conexciones
- Sólo pueden ser utilizadas en socket que utilicen el canal de comunicación
- Se usa del lado del servidor antes que pueda aceptar alguna solicitud de conexión del lado del cliente
- El servidor encola las solicitudes de los clientes conforme estas lleguen

MTI
Maestría en Tecnologías de la Información

Funciones de los sockets

accept()

- Es usada del lado del servidor, para aceptar las conexiones de los clientes

- Servidor llama accept, cesa su actividad y espera una solicitud de conexión un programa cliente

- Cuando llega una solicitud al socket accept(), llena la estructura de dirección con dirección del cliente

- Crea un socket para la conexión.

ESPAMMFL

MTI
Maestría en Tecnologías de la Información

MTI

Maestría en Tecnologías de la Información