

5. MECANISMOS DE COMUNICACIÓN USANDO PASO DE MENSAJES

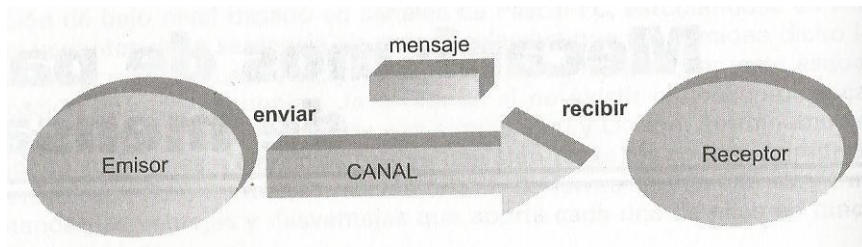
5.1. INTRODUCCIÓN.

Un sistema distribuido o Procesamiento distribuido es un Sistema débilmente acoplado que no comparte memoria, reloj, etc. y por tanto en él no es posible la comunicación entre procesos mediante variables compartidas. La comunicación y sincronización en éste tipo de sistemas es mucho más compleja, lo que introduce diversos problemas en la comunicación:

- Pérdida de mensajes
- Llegada de mensajes desordenados
- Posible heterogeneidad de los nodos (plataformas distintas de HW y SW)
- Etc.

Por tanto, la forma natural de comunicar y sincronizar procesos en este tipo de sistemas es mediante el uso de *paso de mensajes*: los procesos intercambian mensajes entre ellos mediante operaciones explícitas de envío (send) y recepción (receive) que constituyen las primitivas básicas de todo sistema de comunicaciones de éste tipo.

El modelo general de comunicación se representa en la siguiente figura, donde se identifican los elementos fundamentales que intervienen en la comunicación en sistemas con paso de mensajes (un proceso emisor, un proceso receptor, un canal de comunicación, el mensaje a enviar/recibir y las operaciones de envío y recepción).



Ventajas de éste tipo de comunicación:

- Portabilidad de las soluciones programadas para diferentes arquitecturas de computadoras incluyendo las de memoria compartida.
- No existe el problema del acceso en exclusión mutua a datos compartidos

6. IDENTIFICACIÓN EN EL PROCESO DE COMUNICACIÓN.

Es la manera en que el emisor indica a quién va dirigido el mensaje y viceversa, es decir, la forma en que el receptor indica de quién espera un mensaje. Hablamos entonces de *Comunicación Directa* o *Comunicación Indirecta*.

6.1. Sistema de Comunicación Directa.

Se caracteriza porque el emisor identifica explícitamente al receptor del mensaje en la operación de *envío*. El receptor, a su vez, identifica al emisor del mensaje y se establece un enlace de comunicación entre ambos. Las primitivas de envío/recepción tendrían la forma siguiente:

Proceso A: SEND(ProcesoB, mensaje) → Un proceso A envía un mensaje al proceso B.
Proceso B: RECEIVE(ProcesoA, mensaje) → Un proceso B recibe un mensaje del proceso A.

CARACTERÍSTICAS:

- Se identifican los procesos que intervienen en la transmisión del mensaje de forma segura
- Sólo puede existir un enlace de comunicación entre un proceso emisor y otro receptor
- Existen aplicaciones cuya implementación sería difícil e ineficiente, por ejemplo un servidor de impresoras o un servidor Web. En este tipo de aplicaciones los emisores (clientes) si podrían conocer fácilmente la identificación del receptor del mensaje, pero el receptor (servidor) no conoce a priori la identificación de los posibles procesos que pueden solicitar sus servicios.

Una variante de este esquema de comunicación se aquella donde la comunicación es *asimétrica*. El esquema visto hasta el momento es llamado *simétrico* porque el emisor nombra al receptor y viceversa. En la variante asimétrica, el emisor continúa identificando al receptor, pero el receptor no identifica a un emisor concreto. Las primitivas de comunicación quedarían como sigue:

Proceso A: SEND(ProcesoB, mensaje) → Un proceso A envía un mensaje al proceso B.
Proceso B: RECEIVE(id, mensaje) → Un proceso B recibe un mensaje de un emisor cuya identificación quedaría en <i>id</i>

Con este último esquema se solventa el inconveniente que presentaba el esquema anterior para aplicaciones cliente/servidor.

6.2. Sistema de Comunicación Indirecta

No se identifica explícitamente a los procesos emisor y receptor. La comunicación se realiza depositando los mensajes en un almacén intermedio que se supone conocido por los procesos interesados en la comunicación. A ese almacén intermedio se le da el nombre de *buzón*. Las primitivas de comunicación serían las siguientes:

Proceso A: SEND(buzón1, mensaje) → Un proceso A envía un mensaje al proceso B a través del buzón1.

Proceso B: RECEIVE(buzón1, mensaje) → Un proceso B recibe un mensaje del proceso A a través del buzón1.

CARACTERÍSTICAS:

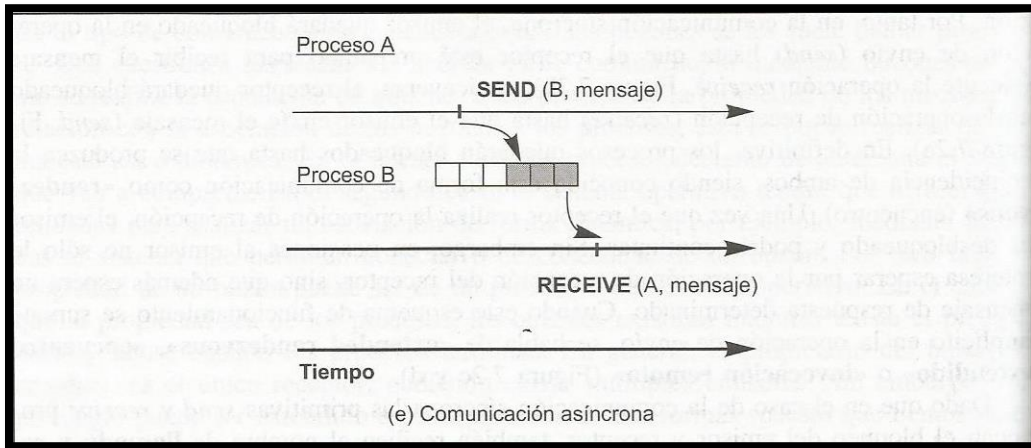
- Un buzón puede ser utilizado por más de dos procesos e incluso entre dos procesos podemos emplear diferentes buzones
- Este esquema de comunicación permite llevar a cabo comunicaciones del tipo uno a uno, uno a muchos, muchos a uno (modelo cliente/servidor) y muchos a muchos
- En el tipo de comunicación muchos a uno, a los buzones se les da el nombre de *puertos*.

Otro esquema de comunicación indirecta que estudiaremos más adelante es la **comunicación entre procesos mediante canales**. Se trata de un caso de comunicación indirecta en que las operaciones de envío y recepción se realizan a través de la especificación de **un canal** (enlace de comunicación), que generalmente tiene un tipo asociado y sobre el cual sólo se pueden enviar datos del mismo tipo. Además un canal no puede ser utilizado por múltiples emisores y receptores y es unidireccional. El concepto de canal surge en lenguajes como Occam en 1988 y Pacal-FC.

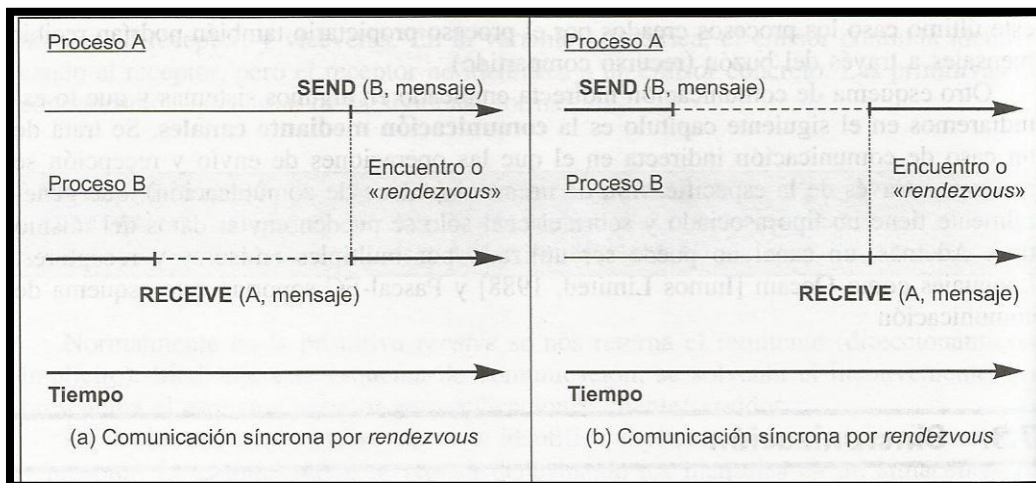
7. SINCRONIZACIÓN

En todo proceso de comunicación, emisores y receptores pueden no coincidir en el tiempo a la hora de realizar la operación de envío y recepción del mensaje. Según el funcionamiento de las primitivas SEND y RECEIVE, podemos hablar de dos tipos de comunicación: *Comunicación Síncrona* y *Comunicación Asíncrona*.

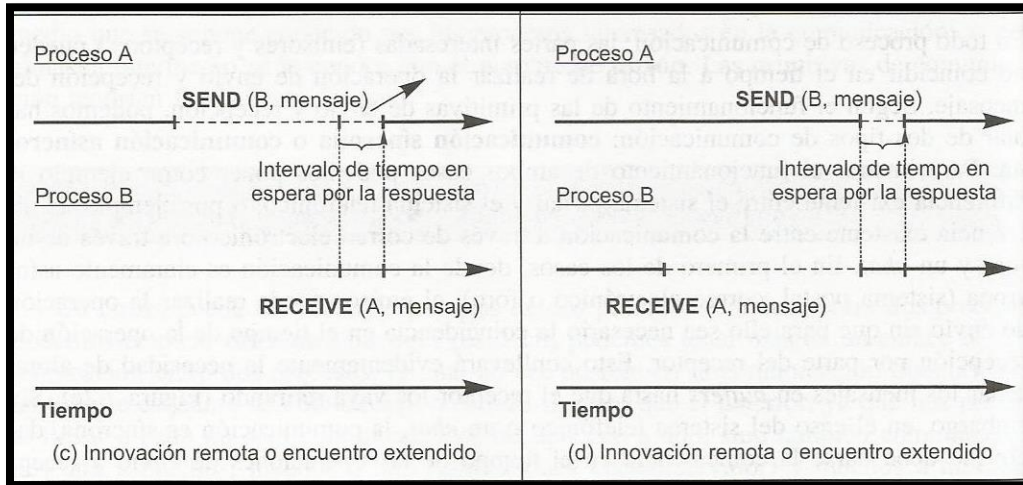
Comunicación Asíncrona. El emisor puede realizar la operación de envío sin que para ello sea necesario que coincida en el tiempo con la operación de recepción por parte del receptor. Esto conlleva la necesidad de almacenar los mensajes en buffers hasta que el receptor los vaya retirando. Ejemplos de sistemas asíncronos son el sistema postal, el sistema del correo electrónico, los sistemas de foros de discusión, etc.



Comunicación Síncrona. Debe darse la coincidencia en el tiempo de las operaciones de envío y recepción por parte de emisores y receptores, es decir, el emisor quedará bloqueado en la operación de envío (send) hasta que el receptor esté preparado para recibir el mensaje (ejecute la operación receive), o viceversa, el receptor quedará bloqueado en la operación de recepción (receive) hasta que el emisor envíe el mensaje (send). En definitiva, los procesos quedarán bloqueados hasta que se produzca la coincidencia de ambos, llamada **rendezvous (cita o encuentro)**. Ejemplos de sistemas síncronos son los sistemas telefónicos. Los sistemas de chats, etc.



En ocasiones al emisor no sólo le interesa esperar por la operación de recepción del receptor, sino que además espera un mensaje de respuesta determinado. Cuando este esquema de funcionamiento se supone implícito en la operación de envío, se habla de **Invocación remota**.



Observación: En el caso de una comunicación asíncrona el envío podría llegar a ser bloqueante si se produce una situación en la que el buffer intermedio donde se almacenan los mensajes tiene un tamaño finito y el envío continuo de mensajes llena el buffer. En este caso el proceso emisor debe esperar hasta que exista algún hueco en el buffer para almacenar el mensaje.

8. CANALES DE COMUNICACIÓN Y MENSAJES.

a) **Flujo de Datos.** Una vez establecido un canal de comunicación entre emisor y receptor, éste, de acuerdo al flujo de datos que pasan por él puede ser de dos tipos: *unidireccional* o *bidireccional*. Para el primero, la información fluye siempre en un sentido entre los dos interlocutores mientras que para el segundo la información fluye en ambos sentidos.

De forma general, para el caso de las comunicaciones asíncronas, el enlace será unidireccional. Cuando la comunicación es síncrona las dos opciones son posibles.

b) **Capacidad del Canal.** Es la posibilidad que tiene el enlace de comunicación de almacenar los mensajes enviados por el emisor cuando éstos no son recogidos de forma inmediata por el receptor. Se tienen en general *canales de capacidad cero* (en donde no existe un buffer donde se vayan almacenando los mensajes), *canales de capacidad finita* (en donde el buffer existente tiene un tamaño fijo) y *canales de capacidad infinita* (donde el buffer asociado al enlace de comunicación se supone infinito en su capacidad).

- c) **Tamaño de los mensajes.** Los mensajes que viajan por el canal pueden ser de longitud fija o de longitud variable.
- d) **Canales con tipo o sin tipo.** Algunos esquemas de comunicación exigen definir el tipo de datos que va a fluir por el canal, imponiendo la restricción del envío de datos únicamente del tipo para el que fue declarado el canal.
- e) **Paso por copia o por referencia.** La comunicación mediante paso de mensajes exige enviar información entre los procesos implicados en la comunicación. Esto puede realizarse de dos maneras: efectuar una copia exacta de los datos (mensaje) que el emisor quiere enviar desde su espacio de direcciones al espacio de direcciones del proceso receptor (paso por copia o valor) o simplemente enviarle al receptor la dirección en el espacio de direcciones del emisor donde se encuentra el mensaje (paso por referencia). Esto último requiere que los procesos interlocutores compartan memoria.

9. CONDICIONES DE ERROR EN SISTEMAS DE PASO DE MENSAJES.

Cuando existe un conjunto de procesos concurrentes y cooperantes que se comunican por medio de paso de mensajes, existen situaciones excepcionales que pueden producir problemas producto de errores en la comunicación tales como:

- a) Pérdida de mensajes entre los procesos que se comunican mediante canales
- b) Alteración de mensajes como consecuencia por ejemplo de ruidos en la transmisión del mensaje por el canal
- c) Bloqueo de procesos (emisor y/o receptor)

El estudio a fondo de soluciones a los errores de comunicación entre procesos concurrentes y cooperantes por medio de paso de mensajes en sistemas distribuidos o de red cae dentro del ámbito de los llamados Sistemas Tolerantes a Fallos.