

Comunicación entre procesos

Universidad Arturo Jauretche
Ingeniería Informática

Docentes:

Ing. Eduardo Kunysz

Ing. Jose Vera

Ing. Daniel Alonso

Introducción

- Con frecuencia los procesos necesitan comunicarse con otros.
- Existe una necesidad de comunicación entre procesos, de preferencia en una forma **bien estructurada** sin utilización de interrupciones
- A esto se le llama "**Comunicación entre procesos**" o "**IPC**"

Cuestiones

- ¿Cómo un proceso puede pasar información a otro?
- ¿Cómo hacer que dos o más procesos no se interpongan entre sí?
- Obtener una secuencia apropiada cuando hay dependencias presentes:
 - Ejemplo: si el proceso A produce datos y el proceso B los imprime, B tiene que esperar hasta que A haya producido algún dato antes de empezar a imprimir

¡Estas cuestiones también se aplican a hilos!

Mecanismo de comunicación

- **Memoria Compartida.**

- Necesitan un mecanismo de **sincronización** externo.
- La **responsabilidad** de la comunicación recae en los **procesos** (el sistema operativo sólo proporciona llamadas para manipular dicha memoria compartida).

- **Paso de Mensajes.**

- La **responsabilidad** de la comunicación y sincronización recae en el **Sistema Operativo** que proporciona un enlace lógico entre procesos.
- Los procesos sólo tienen que invocar correctamente a dos llamadas básicas: **send y receive** (bloqueantes o no).

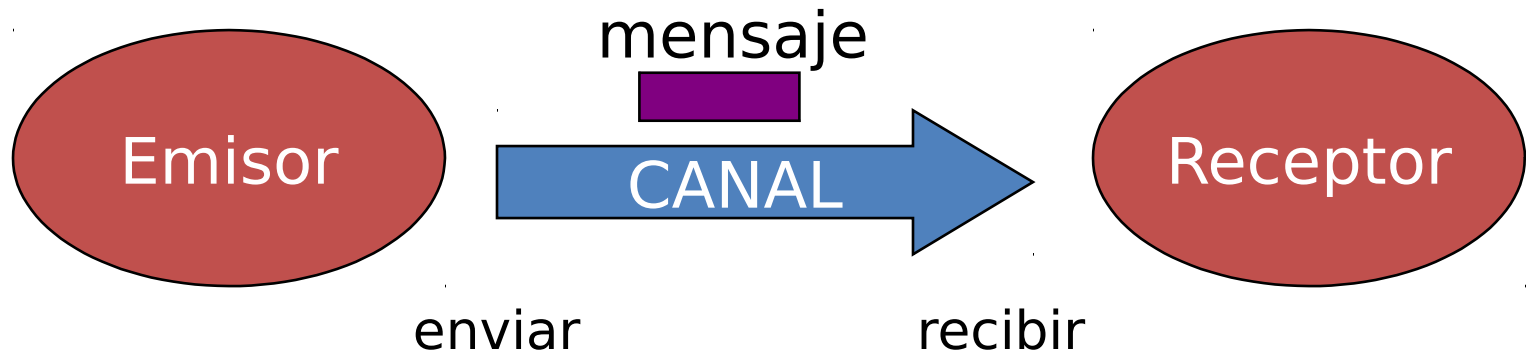
Problemas de memoria compartida

- El modelo de memoria compartida es difícil o imposible de implementar en ciertos casos:
 - en sistemas con protección de memoria, como sistemas **multiusuario**, en los que los procesos corren en espacios de direcciones diferentes
 - en sistemas donde los procesos corren en ordenadores diferentes (**sistemas distribuidos**)
 - en sistemas **multiprocesador** en los que es deseable mantener flexibilidad a la hora de decidir dónde ubicar los procesos
 - en sistemas multiprocesador en los que se quiere implementar **balance de carga** (**reasignación de procesos en tiempo de**

Paso de mensajes

- Los procesos no comparten memoria (variables, objetos, etc.)
- La comunicación se hace mediante operaciones explícitas de envío y recepción

Modelo general



Ventajas del paso de mensajes

- Válido para cualquier arquitectura de computadores
 - sistemas distribuidos
 - arquitecturas paralelas sin memoria compartida
 - también en sistemas de memoria compartida
- **No existe** el problema universal del acceso en **exclusión mutua** a datos compartidos

Memoria Compartida Vs Paso de Mensajes

- Ambos esquemas de comunicación **NO son mutuamente exclusivos**
- Podrían utilizarse simultáneamente dentro de un mismo SO, o incluso dentro de un mismo proceso

Cuestiones básicas de la comunicación

- Sincronización entre emisor y receptor
 - Comunicación síncrona/asíncrona
- Identificación en el proceso de comunicación
 - Comunicación directa/indirecta
 - Comunicación simétrica/asimétrica
- Características del canal

Comunicación síncrona o asíncrona

- **Com. síncrona.** El intercambio de un mensaje es una operación atómica que exige la participación simultánea del emisor y el receptor (rendezvous).
- **Com. asíncrona.** El emisor puede enviar un mensaje sin bloquearse; el receptor lo puede recoger más tarde.

Comunicación síncrona o asíncrona

- Símil:
 - Comunicación síncrona □ llamada telefónica
 - Comunicación asíncrona □ correo postal
- La comunicación **síncrona** es en principio más **sencilla** de implementar
- Podemos emular comunicación síncrona usando primitivas de comunicación asíncrona. P.ej. usando SEND y REPLY.
- Podemos emular comunicación asíncrona usando primitivas de comunicación síncrona. Símil: contestador automático

Repercusiones de la comunicación asíncrona

- El emisor puede enviar varios mensajes:
 - NECESIDAD de disponer de **búferes**
- ¿Cuándo sabe el emisor que su mensaje ha llegado/se ha atendido?
 - conveniencia de operaciones de “acuse de recibo” o de respuesta
(send □ receive □ send_reply □ receive_reply)

Llamadas bloqueantes / no bloqueantes

- Las operaciones de envío y recepción pueden estar definidas como bloqueantes o no bloqueantes
- Un envío/recepción con bloqueo es un ejemplo de comunicación síncrona
- Un envío/recepción sin bloqueo es un ejemplo de comunicación asíncrona

Identificación: Directa

- Comunicación directa
 - Cada proceso que desea comunicarse debe nombrar explícitamente el destinatario o el remitente de la comunicación
 - **enviar**(P , *mensaje*)
 - Enviar un mensaje al proceso P
 - **recibir**(Q , *mensaje*)
 - Recibir un mensaje del proceso Q

Identificación: Indirecta

- Comunicación indirecta
 - Con la comunicación indirecta, los mensajes se envían a, y se reciben de, **buzones de correo** (también llamados **puertos**)
 - **enviar**(*A, mensaje*)
 - Enviar un mensaje al buzón A
 - **recibir**(*A, mensaje*)
 - Recibir un mensaje del buzón A

Comunicación indirecta: PROPIEDADES

- Puede establecerse un enlace entre un par de procesos sólo si **ambos** tienen un **buzón de correo compartido**
- Un enlace puede asociarse con mas de dos procesos
- Entre **cada par** de procesos en comunicación, puede haber una serie de enlaces diferentes, correspondiendo cada enlace a **un buzón** de correo.

Comunicación indirecta

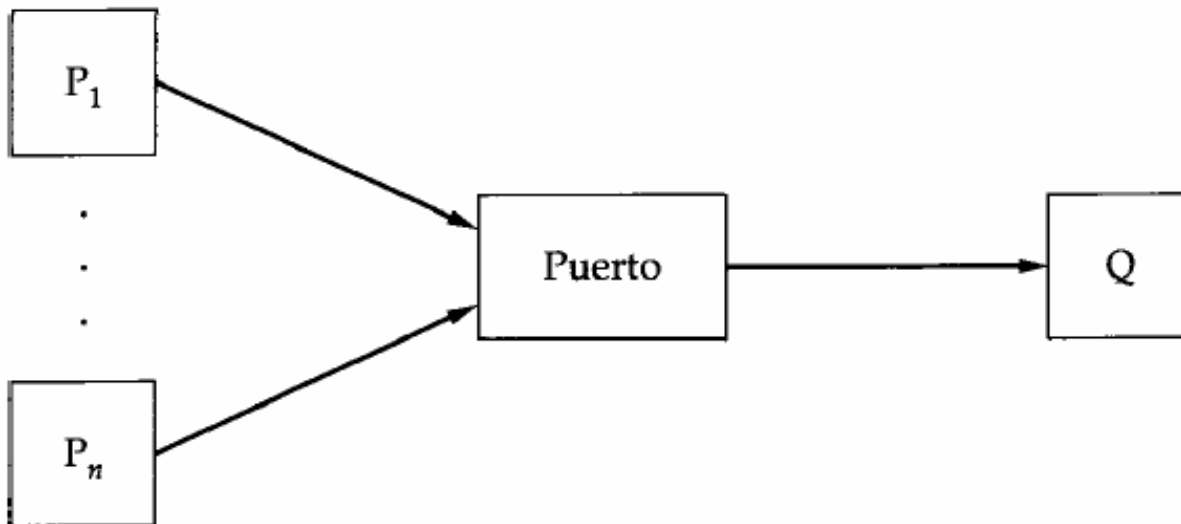
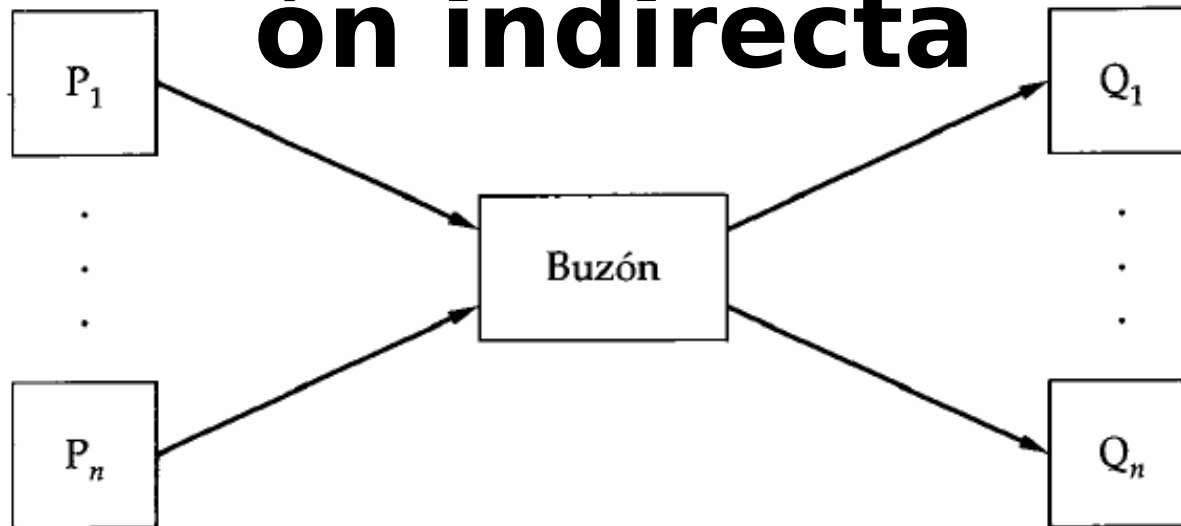
Relaciones emisor - receptor

- **Uno a uno:** enlace privado entre dos procesos
- **Muchos a uno:** un proceso ofrece un servicio a un conjunto de procesos (buzón llamado **Puerto**)
- **Uno a muchos:** útil para aplicaciones en las que un mensaje se difunda a un conjunto de procesos

Procesos
Emisores

Comunicación indirecta

Procesos
Receptores



Comunicación indirecta

- Un buzón puede ser propiedad de un proceso o del sistema

Comunicación indirecta

ejemplo:

- P1, P2, P3 comparten un buzón A. P1 envía un mensaje a A, y P2 y P3 ejecutan cada uno un recibir de A.

¿Qué proceso recibirá el mensaje que envió P1?

Depende de los siguientes métodos:

Comunicación indirecta

ejemplo:

- Permitir que c/ enlace esté asociado como máximo a dos procesos
- Permitir que sólo un proceso, como máximo, ejecute una operación de recepción en c/ momento.
- Permitir que el sistema seleccione arbitrariamente que proceso recibirá el mensaje.

Características del buzón de correo

Buzón de correo

```
graph TD; A[Buzón de correo] --> B[Propiedad Proceso]; A --> C[Propiedad Sistema Operativo]; B --> D[• Propietario: aquél que sólo recibe mensajes a través de este buzón  
• Usuario: aquél que sólo puede enviar mensajes a dicho buzón de correo];
```

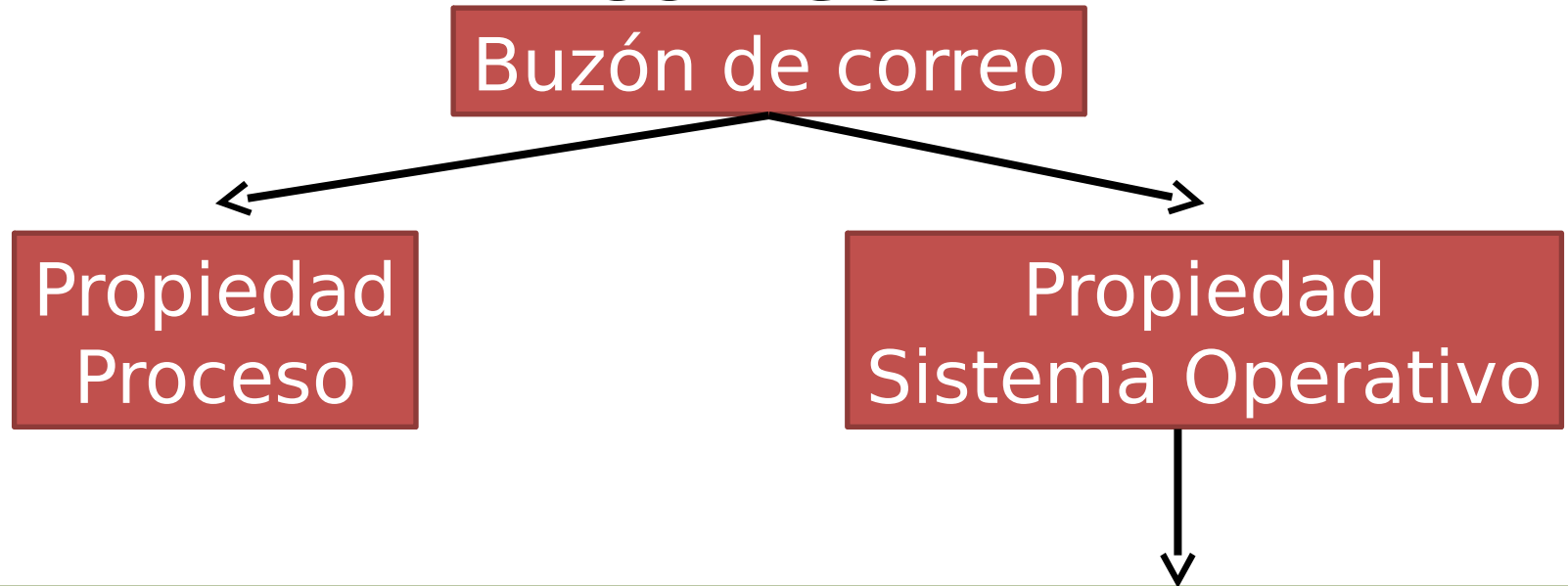
Propiedad
Proceso

Propiedad
Sistema Operativo

- **Propietario:** aquél que sólo recibe mensajes a través de este buzón
- **Usuario:** aquél que sólo puede enviar mensajes a dicho buzón de correo

Cuando termina, dicho buzón desaparece

Características del buzón de correo



- Tiene existencia propia, debe proporcionar mecanismo para:
 - Crear un buzón nuevo
 - Enviar y recibir mensajes a través del buzón
 - Eliminar un buzón de correo

Identificación: Simétrica /asimétrica

- Comunicación simétrica
 - Los procesos tanto receptor como emisor necesitan nombrar al otro para comunicarse
- Comunicación asimétrica
 - Sólo el emisor nombra al destinatario. El destinatario no tiene que nombrar al emisor
- Otra forma más flexible: bulletin boards (tablones) => nadie nombra a nadie

Identificación: Simétrica /asimétrica

- En este esquema las primitivas **send()** y **receive()** se definen:
 - `send(p, mensaje)` -> envía un mensaje al proceso P
 - `receive(id, mensaje)` -> recibe un ***mensaje*** de cualquier proceso; a la variable ***id*** se le asigna el nombre del proceso con el que se ha llevado a cabo la comunicación.

Desventajas simétrico / asimétrico

- Limitada modularidad de las definiciones de procesos resultantes:
 - Cambiar el id de un proceso requiere que se modifiquen las definiciones del resto de los procesos.

Cualquier técnica de **precodificación**, en la que los identificadores deban establecerse explícitamente es menos deseable que las técnicas basadas en **indirecciones**

Características del canal (1)

- Punto a punto, multipunto
- Unidireccional, bidireccional
- Capacidad del canal
 - cero
 - limitada
 - infinita (teórico)

Características del canal (2)

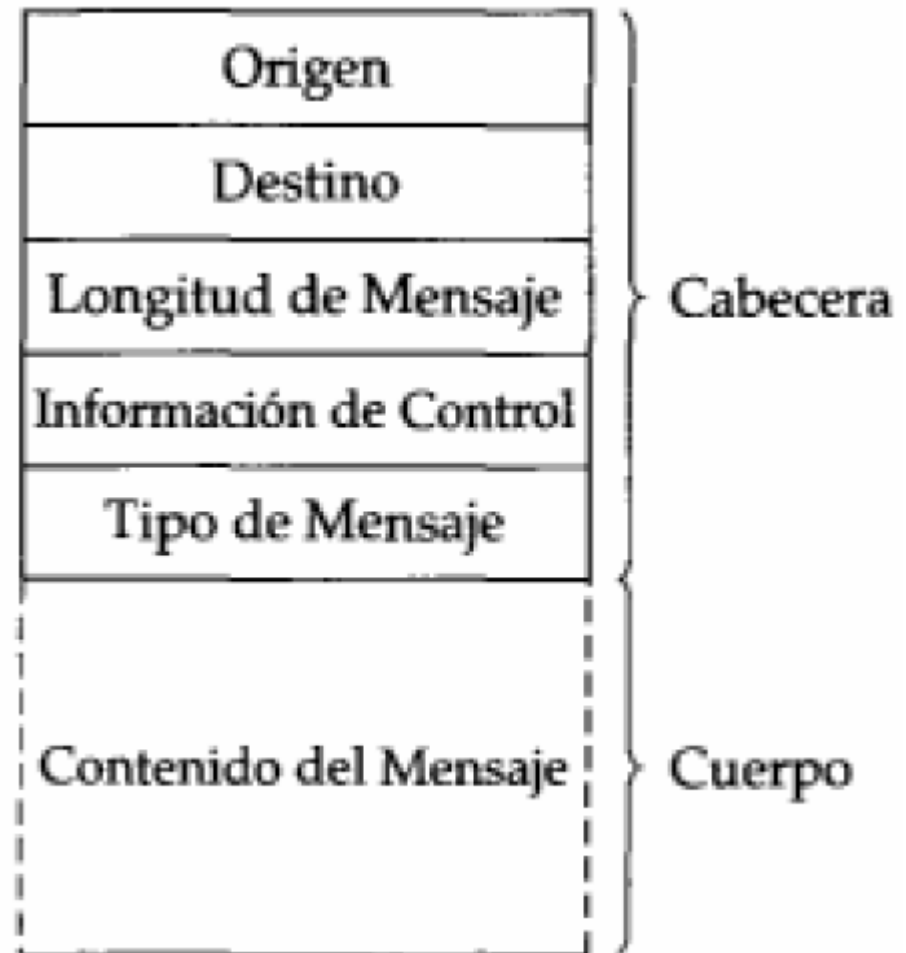
- Mensajes:
 - Tamaño fijo o variable
 - Canales con tipo o sin tipo
 - Paso por copia o por referencia (icuidado!)

Tamaño de mensaje fijo

- Minimiza el procesamiento
- Si se pasan gran cantidad de datos, los datos se ponen en un archivo y el mensaje hace referencia al archivo

Tamaño de mensaje variable

- **Cabecera:**
información sobre el mensaje
- **Cuerpo:**
contenido real del mensaje



Características del canal de comunicación: directa

- Comunicación directa
 - Se establece automáticamente
 - Un canal se asocia a **exactamente dos procesos**
 - Entre cada par de procesos **sólo existe un canal**
 - El enlace puede ser **unidireccional**, pero suele ser **bidireccional**

Características del canal de comunicación: indirecta

- Comunicación indirecta
 - Se establece un **canal entre un par de procesos** sólo si tienen un **buzón** compartido
 - Un canal puede estar **asociado a más de dos procesos**
 - Entre cada par de procesos en comunicación puede haber **varios enlaces distintos**, cada uno de los cuales corresponderá a un buzón
 - Los enlaces pueden ser unidireccionales o bidireccionales

Error de comunicación

- 1 máquina => los mensajes se implementan (generalmente) en memoria compartida
 - Fallo => falla todo el sistema
- Entornos distribuidos => procesos residen en diferentes máquinas
 - Los mensajes se transfieren **por líneas de comunicación**
 - La **probabilidad** de que ocurra un error durante la comunicación y el procesamiento es mucho **mayor que** en un entorno de **una sola máquina**

Error de comunicación

- Cuando ocurre un fallo en un sistema, sea centralizado o distribuido, el sistema debe intentar recuperarse del error
- Algunas situaciones de error:
 - El emisor o el receptor podría terminar antes de que se procese un mensaje
 - P espera un mensaje de Q (proceso terminado)
 - P envía un mensaje a Q (proceso terminado)

Error de comunicación

- Mensajes perdidos
 - Fallo *hardware* o de la línea de comunicación
- Tres métodos para enfrentar este suceso en función de quien asume la responsabilidad de detectar el fallo:
 - SO
 - Emisor
 - SO/Emisor

Error de comunicación

- No siempre es necesario detectar los mensajes perdidos => protocolos de red que garantizan la confiabilidad
- ¿Cómo detectar la pérdida de mensajes?
 - El método de detección más común consiste en emplear ***tiempos límite*** o ***plazos***

Error de comunicación

- Mensajes alterados
 - es común el uso de códigos de verificación de errores (paridad, etc...)

Bibliografía

- Cap 2, Tanenbaum "Sistemas Operativos modernos"
- Cap 4, William Stallings "Sistemas Operativos"
- Cap 3, Silberschatz "Fundamentos de sistemas operativos"