comunicación

Programas concurrentes o/y distribuidos necesitan algún tipo de comunicación entre los procesos.
Hay dos razones principales:

1. Los procesos compiten para obtener acceso a recursos compartidos.
2. Los procesos quieren intercambiar datos.

metódos de comunicación

Para cualquier tipo de comunicación hace falta un método de sincronización entre los procesos que quieren comunicarse entre ellos.
Al nivel del programador existen tres variantes como realizar las interacciones entre procesos:

1. Usar memoria compartida (*shared memory*).
2. Mandar mensajes (*message passing*).
3. Lanzar procedimientos remotos (*remote procedure call* RPC).

síncrono y asíncrono

- La comunicación no tiene que ser síncrona en todos los casos.
- Existe también la forma asíncrona donde un proceso deja su mensaje en una estructura de datos compartida por los procesos.
- El proceso que ha mandado los datos puede seguir con otras tareas.
- El proceso que debe leer los datos, lo hace en su momento.

canal de comunicación

Una comunicación entre procesos sobre algún canal físico puede ser no fiable en los sistemas.
Se puede usar el canal

- para mandar paquetes individuales del mensaje (por ejemplo protocolo UDP del IP)
- para realizar flujos de datos (por ejemplo protocolo TCP de IP)
- Muchas veces se realiza los flujos con una comunicación con paquetes añadiendo capas de control (pila de control).

posibles fallos en canales de paquetes

Para los canales de paquetes, existen varias posibilidades de fallos:

1. se pierden mensajes
2. se cambia el orden de los mensajes
3. se modifican mensajes
4. se añaden mensajes que nunca fueron mandados

técnicas para superar los problemas

1. protocolo de recepción
   (¿Cuándo se sabe que ha llegado el último mensaje?)
2. enumeración de los mensajes
3. uso de código de correción de errores (CRC)
4. protocolo de autenticación

comunicación segura sin mensajes de asentimiento

- Existen protocolos de transmisión de paquetes que no necesitan un canal de retorno pero que garantizan la distribución de los mensajes bajo leves condiciones al canal (*digital fountain codes*).
- Ejemplos: Reed/Solomon códigos (clásicos), Tornado códigos (finales de los 90), Raptor códigos (*rap*id *tom*ado, en los últimos años))
- Ideas básicas presentadas en pizarra.
- Raptor código: se manda 1.002 MByte, y de cualquier 1 MByte recibido se puede recuperar el mensaje original (con tiempo de cálculo lineal en el longitud del mensaje)
- base para varios nuevos estandares de transmisión de datos en la red