

Sincronismo

Algoritmo / protocolo es sincrónico si sus **acciones pueden ser delimitadas en el tiempo**.

- **Sincrónico.** Entrega de msj posee timeout conocido.
- **Parcialmente sincrónico.** No posee timeout conocido o es variable.
- **Asincrónico.** No posee timeout asociado.

Propiedades

- **Tiempo de Delivery:** tiempo que tarda mensaje en ser recibido luego de haber sido enviado.
- **Timeout de Delivery:** todo mensaje enviado va a ser recibido antes de un tiempo conocido.
- **Steadiness:** máxima diferencia entre el mínimo y máximo tiempo de delivery de cualquier mensaje recibido por un proceso.
 - Define varianza con la cual un proceso observa que recibe los msjs.
 - Qué tan constante es la recepción de mensajes.
- **Tightness:** máxima diferencia entre los tiempos de delivery para cualquier mensaje.
 - Define simultaneidad con la cual un mensaje es definido por múltiples procesos.

Protocolos

- Time-driven.
- Clock-driven.

Orden

- **Delivery de Mensajes** != Envío.
- Delivery: **procesar** mensaje, provocando **cambios en el estado**.
- Cola de mensajes, permite:
 - **Hold-back queue** y **delivery queue**.
 - Demorar el **delivery**.
 - Re-ordenar mensajes en la cola.

Órdenes

- **FIFO.** Orden en el que fueron enviados entre un mismo emisor y un mismo receptor.
- **Causal.** Todo mensaje que implique la generación de un nuevo mensaje es entregado **manteniendo esta secuencia de causalidad**, sin importar receptor.
- **Total.** Todo par de mensajes entregados a los mismos receptores es recibido en el mismo orden por esos receptores.

Estado y consistencia

- **Estado Local.** Valores de variables en un momento dado.
- **Estado Global.** Unión de todos los estados locales del sistema.

Máquina de estados

- Modelar el sistema como **serie de estados**.
- Evolución de estados debido a **eventos**.
- Con múltiples procesos hay **estados globales inválidos**.

Historia y Corte

- **Historia (corrida).** Secuencia de todos los eventos procesados por un proceso P_i .
- **Corte.** Unión del subconjunto de historias de todos los procesos del sistema hasta cierto evento k de cada proceso.
 - **Consistente** si por cada evento que contiene también contiene a aquellos que **ocurren antes**.

Algoritmo de Chandy & Lamport

- **Snapshots de estados globales** en sistemas distribuidos.
- Objetivo: estado global almacenado es **consistente**.

Comunicación fiable

Si se garantiza **integridad, validez y atomicidad** en el **delivery** de los mensajes.

- **Uno a uno.** Trivial sobre TCP/IP y red segura.
- **Uno a muchos.**
 - El grupo debe proveer las 3 propiedades.
 - Bajo TCP/IP, ojo con atomicidad.
 - Definir qué orden de mensajes se garantiza.