

Tema 2: Programación Concurrente

1. Programación concurrente.

Podemos decir que dos procesos son concurrentes, cuando la primera instrucción de un proceso se ejecuta después de la primera y antes de la última de otro proceso.

La planificación alternando los instantes de ejecución en la gestión de los procesos, hace que los procesos se ejecuten de forma concurrente. O lo que es lo mismo: multiproceso = concurrencia.

2. ¿Para qué concurrencia?

Las principales razones por las que se utiliza una estructura concurrente son:

- **Optimizar la utilización de los recursos:** Podremos simultanear las operaciones de E/S en los procesos. La CPU estará menos tiempo ociosa.
- **Proporcionar interactividad a los usuarios** (y animación gráfica).
- **Mejorar la disponibilidad:** Servidor que no realice tareas de forma concurrente, no podrá atender peticiones de clientes simultáneamente.
- **Conseguir un diseño conceptualmente más comprensible y mantenible:** El diseño concurrente de un programa nos llevará a una mayor modularidad y claridad.
- **Aumentar la protección:** Tener cada tarea aislada en un proceso permitirá depurar la seguridad de cada proceso y poder finalizarlo en caso de mal funcionamiento sin que suponga la caída del sistema.

Los actuales avances tecnológicos hacen necesario tener en cuenta la concurrencia en el diseño de las aplicaciones para aprovechar su potencial. Los nuevos entornos hardware son:

- Microprocesadores con múltiples núcleos que comparten la memoria principal del sistema.
- Entornos multiprocesador con memoria compartida.
- Entornos distribuidos.

Los beneficios que obtendremos al adoptar un modelo de programa concurrente son:

- Estructurar un programa como conjunto de procesos concurrentes que interactúan, aporta gran claridad sobre lo que cada proceso debe hacer y cuando debe hacerlo.
- Puede conducir a una reducción del tiempo de ejecución.
- Permite una mayor flexibilidad de planificación.
- La concepción concurrente del software permite un mejor modelado previo del comportamiento del programa, y en consecuencia un análisis más fiable de las diferentes opciones que requiera su diseño.

1. Condiciones de competencia.

Distinguimos los siguientes tipos básicos de interacción entre procesos concurrentes:

- **Independientes:** Solo interfieren en el uso de la CPU.
- **Cooperantes:** Un proceso genera la información o proporciona un servicio que otro necesita.
- **Competidores:** Procesos que necesitan usar los mismos recursos de forma exclusiva.

Un proceso entra en condición de competencia con otro, cuando ambos necesitan el mismo recurso, ya sea forma exclusiva o no; por lo que será necesario utilizar mecanismos de sincronización y comunicación entre ellos.

Se llama región de exclusión mutua al conjunto de instrucciones en las que el proceso utiliza un recurso y que se deben ejecutar de forma exclusiva con respecto a otros procesos competidores por ese mismo recurso. En resumen, mientras que lo esté utilizando él, ningún otro puede utilizarlo.

Se dice que un proceso hace un lock (bloqueo) sobre un recurso cuando ha obtenido su uso en exclusión mutua.

Deadlock o interbloqueo, se produce cuando los procesos no pueden obtener, nunca, los recursos necesarios para continuar su tarea. El interbloqueo es una situación muy peligrosa, ya que puede llevar al sistema a su caída o cuelgue.

3. Comunicación entre procesos.

Se da cuando un proceso da o recibe información.

Clasificaremos las interacciones entre los procesos y el resto del sistema (recursos y otros procesos), como estas tres:

- **Sincronización:** Un proceso puede conocer el punto de ejecución en el que se encuentra otro en ese determinado instante.
- **Exclusión mutua:** Mientras que un proceso accede a un recurso, ningún otro proceso accede al mismo recurso o variable compartida.
- **Sincronización condicional:** Solo se accede a un recurso cuando se encuentra en un determinado estado interno.

1. Mecanismos básicos de comunicación.

Si pensamos en la forma en la que un proceso puede comunicarse con otro, se nos ocurrirán estas dos:

- **Intercambio de mensajes:** Tendremos las primitivas enviar y recibir información.
- **Recursos (o memoria) compartidos:** Las primitivas serán escribir y leer datos en o de un recurso.

En el caso de comunicar procesos dentro de una misma máquina, el intercambio de mensajes, se puede realizar de dos formas:

- **Utilizar un buffer de memoria.**
- **Utilizar un socket.**

La diferencia entre ambos, está en que un socket se utiliza para intercambiar información entre procesos en distintas máquinas a través de la red; y un buffer de memoria, crea un canal de comunicación entre dos procesos utilizando la memoria principal del sistema.

2. Tipos de comunicación.

En cualquier comunicación, vamos a tener los siguientes elementos:

- **Mensaje:** Información que es el objeto de la comunicación.
- **Emisor:** Entidad que emite, genera o es origen del mensaje.
- **Receptor:** Entidad que recibe, recoge o es destinataria del mensaje.
- **Canal:** Medio por el que viaja o es enviado y recibido el mensaje.

Podemos clasificar el canal de comunicación según su capacidad, y los sentidos en los que puede viajar la información, como:

- **Símplex:** La comunicación se produce en un solo sentido.
- **Dúplex (Full Duplex):** Pueden viajar mensajes en ambos sentidos simultáneamente entre emisor y receptor.
- **Semidúplex (Half Duplex):** El mensaje puede viajar en ambos sentidos, pero no al mismo tiempo.

Otra clasificación dependiendo de la sincronía que mantengan el emisor y el receptor durante la comunicación, será:

- **Síncrona:** El emisor queda bloqueado hasta que el receptor recibe el mensaje. Ambos se sincronizan en el momento de la recepción del mensaje.
- **Asíncrona:** El emisor continúa con su ejecución inmediatamente después de emitir el mensaje, sin quedar bloqueado.
- **Invocación remota:** El proceso emisor queda suspendido hasta que recibe la confirmación de que el receptor ha recibido correctamente el mensaje, después emisor y receptor ejecutarán síncronamente un segmento de código común.

Dependiendo del comportamiento que tengan los interlocutores que intervienen en la comunicación, tendremos comunicación:

- **Simétrica:** Todos los procesos pueden enviar y recibir información.
- **Asimétrica:** Solo un proceso actúa de emisor, el resto solo escucharán el o los mensajes.