Concurrencia

Exclusión Mutua y Sincronización

Concurrencia

En los sistemas de tiempo compartido se presentan muchos problemas debido a que los procesos compiten por los recursos del sistema. Imagine que un proceso está escribiendo en la unidad de cinta y se le termina su turno de ejecución e inmediatamente después el proceso elegido para ejecutarse comienza a escribir sobre la misma cinta.

Concurrencia

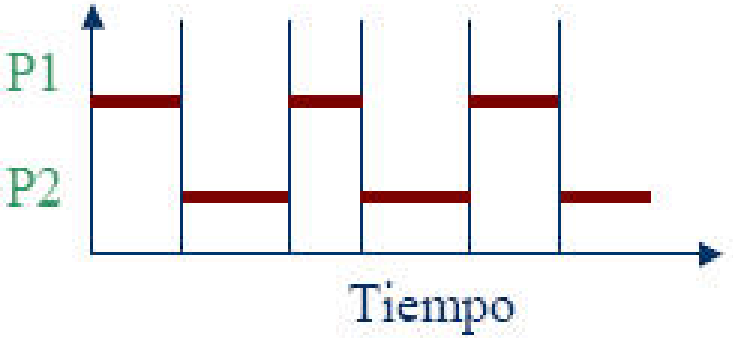
El resultado es una cinta cuyo contenido es un desastre de datos mezclados. Así como la cinta, existen una multitud de recursos cuyo acceso debe ser controlado para evitar los problemas de la concurrencia.

Concurrencia

- La concurrencia es el solapamiento en el tiempo de la ejecución de varias actividades.
- Dos procesos, P1 y P2, se ejecutan concurrentemente si la primera instrucción de P1 se ejecuta entre la primera y la última instrucción de P2.

Real o paralelismo

Nº procesadores >= Nº procesos



Aparente, simulada o seudoparalelismo

Nº procesadores < Nº procesos

Concurrencia

- **Comunicación:** Necesidad de **transmisión de información** entre procesos concurrentes.

- **Sincronización:** Necesidad de que las ejecuciones de los procesos concurrentes se produzcan según una **secuenciación temporal**, conocida y establecida entre los propios procesos.

Concurrencia

Cooperación entre procesos:

- Dos técnicas de operación:
  - **Sistemas multiprogramados con un solo procesador:** los procesos se intercalan en el tiempo para dar la apariencia de ejecución simultánea.
  - **Sistemas multiprocesador:** además de intercalarse los procesos se superponen.

Concurrencia

- Estas técnicas son ejemplos de procesamiento concurrente y plantean varios problemas:

1. Compartir recursos globales conlleva riesgos. Se debe ser muy **cuidadoso en el orden** en que dos procesos modifican independientemente el valor de una variable global que ambos usan.
2. Administrar la asignación óptima de recursos. Se debe **impedir que se produzcan situaciones de interbloqueo.**
3. Localizar **errores de programación.**

Problemas generales de la Concurrencia

- **Condiciones de Carrera o Competencia:** La condición de carrera (race condition) ocurre cuando dos o más procesos accesan un recurso compartido sin control, de manera que el resultado combinado de este acceso depende del orden de llegada. Suponga, por ejemplo, que dos clientes de un banco realizan cada uno una operación en cajeros diferentes al mismo tiempo.

Problemas generales de la Concurrencia

- **Postergación o Aplazamiento Indefinido(a)**: Consiste en el hecho de que uno o varios procesos nunca reciban el suficiente tiempo de ejecución para terminar su tarea. Por ejemplo, que un proceso ocupe un recurso y lo marque como 'ocupado' y que termine sin marcarlo como 'desocupado'. Si algún otro proceso pide ese recurso, lo verá 'ocupado' y esperará indefinidamente a que se 'desocupe'.

Problemas generales de la Concurrencia

- **Condición de Espera Circular**: Esto ocurre cuando dos o más procesos forman una cadena de espera que los involucra a todos. Por ejemplo, suponga que el proceso A tiene asignado el recurso 'cinta' y el proceso B tiene asignado el recurso 'disco'. En ese momento al proceso A se le ocurre pedir el recurso 'disco' y al proceso B el recurso 'cinta'. Ahí se forma una espera circular entre esos dos procesos que se puede evitar quitándole a la fuerza un recurso a cualquiera de los dos procesos.

Problemas generales de la Concurrencia

- **Condición de No Apropiación:** Esta condición no resulta precisamente de la concurrencia, pero juega un papel importante en este ambiente. Esta condición especifica que si un proceso tiene asignado un recurso, dicho recurso no puede arrebatársele por ningún motivo, y estará disponible hasta que el proceso lo 'suelte' por su voluntad.

Problemas generales de la Concurrencia

Condición de Espera Ocupada: Esta condición consiste en que un proceso pide un recurso que ya está asignado a otro proceso y la condición de no apropiación se debe cumplir. Entonces el proceso estará gastando el resto de su tiempo verificando si el recurso fue liberado. Es decir, desperdicia su tiempo de ejecución en esperar. La solución más común a este problema consiste en que el sistema operativo se dé cuenta de esta situación y mande a una cola de espera al proceso, otorgándole inmediatamente el turno de ejecución a otro proceso.

Problemas generales de la Concurrencia

Condición de Ocupar y Esperar un Recurso:

Consiste en que un proceso pide un recurso y se le asigna. Antes de soltarlo, pide otro recurso que otro proceso ya tiene asignado.

Problemas generales de la Concurrencia

Condición de Exclusión Mutua, lo revisaremos más adelante.

Concurrencia

Áreas de comunicación entre procesos:

- Los procesos de un sistema **no** actúan de forma aislada. A veces tienen que:
  - **Cooperar** para alcanzar el objetivo ⇒ **sincronización**
  - **Competir** por el uso de recursos ⇒ **exclusión mutua**

Concurrencia

- Tipos de recursos:
  - **Compartibles**: pueden ser usados por varios procesos de forma concurrente;
  - **No compartibles**: en cada momento solo pueden ser usados por un único proceso.

Razones:

- por la naturaleza física del recurso;
- porque la acción sobre el recurso por parte de un proceso puede interferir en la acción de otro proceso.

Concurrencia

- La interacción entre procesos se clasifica de acuerdo al nivel de conocimiento que cada proceso tiene de los demás:

- **Los procesos no tienen conocimiento de la existencia de los demás:** Independientes, no van a operar juntos, entre ellos se establece una **competencia por los recursos** (disco, archivos, impresora, …).

Concurrencia

- **•Los procesos tienen un conocimiento indirecto de los otros:** No se conocen específicamente unos a otros pero comparten el acceso a algunos objetos como el buffer de E/S. Relación de **cooperación** para compartir.

- **•Los procesos tienen un conocimiento directo de los otros:** Diseñados para **cooperar** trabajando en conjunto en alguna actividad y comunicándose entre ellos.

Concurrencia

Tipos de Procesos

Independientes	Cooperantes
• Son deterministas y reproducibles: no intercambian información con el resto de procesos y por tanto no dependen de ellos • Ejemplo: Un programa que calcula la transformada discreta de Fourier a partir de una secuencia de entrada	• No son deterministas y su funcionamiento puede no ser reproducibles: intercambian información con el resto de procesos y por tanto dependen de ellos • Ejemplo: Un programa que escribe en la pantalla los datos que otros programas el dejan en un buffer

Ejemplo

- Supongamos que en un entorno UNIX se ejecuta la orden

- `$ cat archivo1 archivo2 archivo3 | wc -l`

Ejemplo

- Esta orden va a provocar que el intérprete de comandos cree dos procesos concurrentes, el primero ejecutará el programa *cat*, que concatenará el contenido de los archivos de texto *archivo1*, archivo2 y archivo3. El segundo ejecutará el programa *wc*, que contará el número de líneas de la salida producida por *cat*.
- Estos dos procesos cooperan, y será preciso algún tipo de sincronización entre ellos, concretamente hasta que *cat* no produzca alguna salida *wc* debería bloquearse.

Exclusión mutua

El **acceso a ciertos recursos es exclusivo de un proceso cada vez**.

Mientras los otros deben permanecer a la espera hasta que finalice la utilización de dicho recurso por el proceso al que se le asigno. Cuando este proceso termine, el recurso será asignado a uno de los procesos en espera. *Se asegura el correcto uso del recurso.*

A la parte del programa que los utiliza se le llama sección crítica.

Requisitos para la Exclusión mutua

Cualquier solución que de soporte a la EM debe de cumplir los requisitos:

1. **Sólo un proceso** debe tener permiso para **entrar en la sección crítica** por un recurso en un instante dado.
2. Un **proceso que se interrumpe** en una sección no crítica debe hacerlo **sin interferir con los otros procesos.**
3. No pueden permitirse **el interbloqueo o la inanición.**
4. Cuando ningún proceso está en su sección crítica, **cualquier proceso que solicite entrar en la suya debe poder hacerlo sin tardanza.**
5. **No** hacer **suposiciones sobre la velocidad relativa** de los procesos o el número de procesadores.
6. Un proceso permanece en su sección crítica sólo por un **tiempo finito.**

Sección Crítica

- Segmento de código.
- **No está permitido que más de un proceso, estén simultáneamente en su sección crítica.**
- Un protocolo rige la forma de entrar y salir de la sección crítica.

Hacer que se cumpla la exclusión mutua puede dar lugar a dos problemas:

- Interbloqueo
- Inanición

Problema de la Sección Crítica

Cualquier solución al problema de la sección crítica debe satisfacer los tres requisitos:

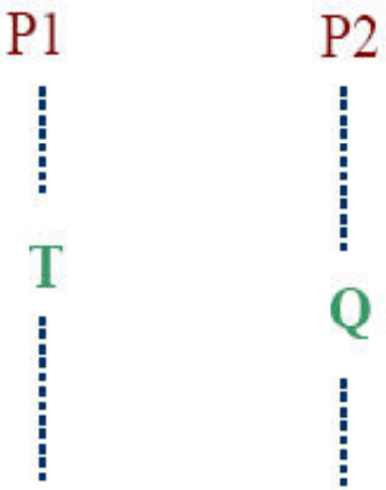
- **Exclusión Mutua:** Sólo un proceso ejecuta simultáneamente su sección crítica.
- **Progreso:** Cuando ningún proceso ejecuta su sección crítica, algún proceso que lo solicite podrá entrar utilizando un protocolo, que impida la entrada simultánea de varios. La decisión de quién entra no se puede posponer indefinidamente.
- **Espera limitada:** Ningún proceso debe esperar ilimitadamente la entrada en la sección crítica.

Semáforos

- Mecanismo **frente a problemas de concurrencia**.
- Funcionamiento basado en la **cooperación** entre procesos.
- **Uso de señales**, podemos obligar a un proceso a detenerse hasta que reciba una señal.
- Para esta señalización se usan unas variables especiales llamadas semáforos:
  - Para **transmitir una señal** por el semáforo s, los procesos ejecutan la primitiva **signal (s).**
  - Para **recibir una señal** del semáforo s, los procesos ejecutan la primitiva **wait (s).**Si la señal aún no se ha emitido, el proceso es suspendido hasta que se emite la señal

Sincronización

Es la comunicación requerida entre dos o más procesos con el fin de sincronizar sus actividades.
Gráficamente:



Un proceso, P1, llegado a un punto T, no puede proseguir su ejecución hasta que otro proceso, P2, haya llegado a otro punto, Q, de su ejecución.

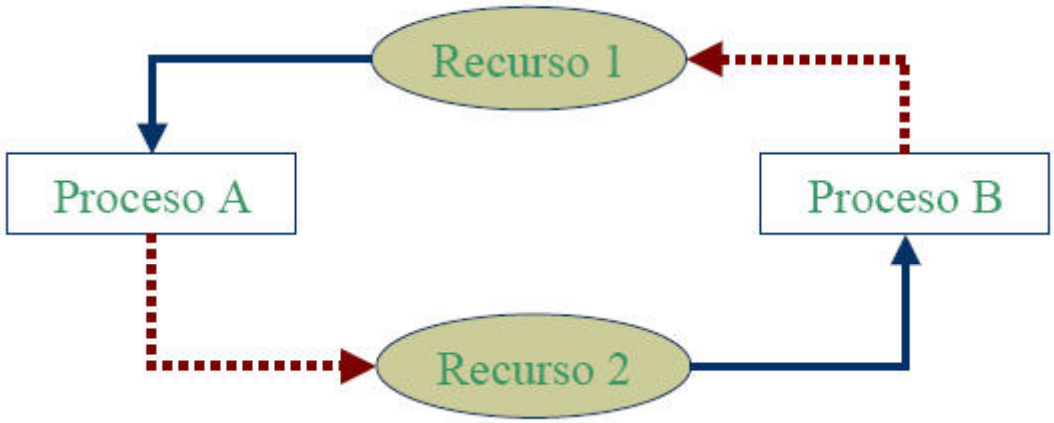
Bloqueos Mutuos

Postergación indefinida (inanición o Lockout)

- Cuando un proceso espera por un evento que puede ocurrir pero no se sabe cuando.

Interbloqueo (DeadLocks)

Un proceso esta interbloqueado si está esperando por un evento determinado que nunca va a ocurrir.



Interbloqueo entre dos procesos por el uso de recursos no compartibles (abrazo mortal)

Condiciones necesarias

Coffman, Elphick y Shoshani establecieron las **cuatro condiciones necesarias** para que se produzca interbloqueo:

- Condición de ***exclusión mutua***: los procesos reclaman control exclusivo de los recursos que piden.
- Condición de ***esperar por*** : los procesos mantienen los recursos que ya les han sido asignados mientras esperan por recursos adicionales.
- Condición de ***no apropiatividad***: los recursos no pueden ser extraídos de los procesos que los tienen hasta su completa utilización.
- Condición de ***espera circular***: existe una cadena circular de procesos en la cual cada uno de ellos mantiene a uno o más recursos que son requeridos por el siguiente proceso de la cadena.

Métodos

- Usar un protocolo que asegure que el sistema ***nunca*** llegará a un estado de bloqueo mutuo.
- Permitir que el sistema entre en bloqueo mutuo y luego se recupere.
- Desatendernos del problema y hacer como si nunca ocurrieran bloqueos mutuos en el sistema. Esta solución es la que adopta la mayor parte de los sistemas operativos, incluido UNIX.

Puntos a Considerar

A. **Prevención del interbloqueo:** condiciona al sistema para que elimine toda posibilidad de que se produzca interbloqueo. Estrategias:

- *Negar la condición de esperar por*: cada proceso debe pedir todos los recursos que va a necesitar de golpe. Si el conjunto de todos ellos está disponible, se le asigna todos. Si no esta disponible todo el conjunto completo, no se le asigna ninguno al proceso y tendrá que esperar hasta que estén todos disponibles.

Puntos a Considerar

- *Negar la condición de no apropiatividad*: cuando un proceso que tiene recursos le es negada una petición de recursos adicionales, deberá liberar sus recursos y, si es necesario, pedirlos de nuevo junto con los recursos adicionales.

- *Negar la condición de espera circular*: cuando se instala un recurso se le asigna un número exclusivo, de forma que los procesos deben de solicitar los recursos en orden ascendente de acuerdo a los números asignados a dichos recursos.

Puntos a Considerar

B. Evitar el interbloqueo: condiciona al sistema para que esquive determinadas situaciones con posibilidad de interbloqueo.

C. Detección del interbloqueo: determinan si existe o no un interbloqueo, e identifica cuáles son los procesos y recursos implicados en él.

D. Recuperación del interbloqueo: pretenden romper el interbloqueo retirando una o más de las condiciones necesarias.