

Tema 5

El Interbloqueo

5.1. Introducción

5.2. Condiciones para que se produzca

5.3. Métodos para el tratamiento

5.3.1. Prevención

5.3.2. Evitación

5.3.3. Detección y recuperación

5.3.4. Estrategias combinadas

Introducción

- Una de las principales tareas del SO es la administración de recursos
- Muchos recursos no pueden ser utilizados simultáneamente por más de un proceso. Ej: Impresora, CPU, Tarjeta de sonido, etc.
- Es necesario que el SO disponga de un mecanismo que permita acceder de forma exclusiva a un recurso

Introducción

- Cuando los procesos utilizan varios recursos en exclusividad puede darse la situación de que un proceso espere recursos que tiene reservado otro proceso y viceversa
- A esta situación se la denomina *interbloqueo*, *deadlock* o *bloqueo mútuo*.

Introducción

- Se puede definir una situación de interbloqueo como:

Un conjunto de procesos se encuentra en estado de interbloqueo cuando cada uno de ellos espera un suceso que sólo puede originar otro proceso del mismo conjunto

Condiciones para que se produzca interbloqueo

- Exclusión mútua
- Retención y espera
- No expropiación
- Espera circular

Métodos para el tratamiento del interbloqueo

- Ignorar el problema (estrategia del avestruz)
- Prevención
- Evitación
- Detección y recuperación
- Estrategias combinadas

Prevención del interbloqueo

- Este método previene el interbloqueo eliminando una de sus cuatro causas
- Exclusión mútua
 - No siempre es posible
 - En algunos recursos se puede conseguir utilizando spooling y demonios.

Prevención del interbloqueo

- Retención y espera
 - Se puede eliminar haciendo que los procesos liberen los recursos que tienen cuando realizan una solicitud de un recurso no disponible
 - Dos estrategias:
 1. Los recursos se solicitan al comienzo de la ejecución
 2. Los recursos se solicitan de forma incremental pero son liberados cuando se deniega una solicitud

Prevención del interbloqueo

- No expropiación
 - Se puede eliminar permitiendo la expropiación
 - El SO debe ser capaz de restaurar el estado de los recursos
 - Se adecúa muy bien a cierto tipo de recursos como CPU y memoria.

Prevención del interbloqueo

- Espera circular
 - Se elimina ordenando los recursos
 - Los recursos sólo pueden ser solicitados en el orden establecido
 - No aprovecha adecuadamente los recursos

Evitación del interbloqueo

- Las peticiones de recursos sólo se conceden si se garantiza que no conducirán a un interbloqueo
- Dos inconvenientes:
 - Desaprovechamiento de los recursos
 - Mayor coste computacional al evaluar las peticiones de recursos

Evitación del interbloqueo

- Un algoritmo analiza cada petición de recursos antes de su concesión para comprobar que no llevará a un estado de interbloqueo
- Estos algoritmos se basan en el concepto de estado seguro

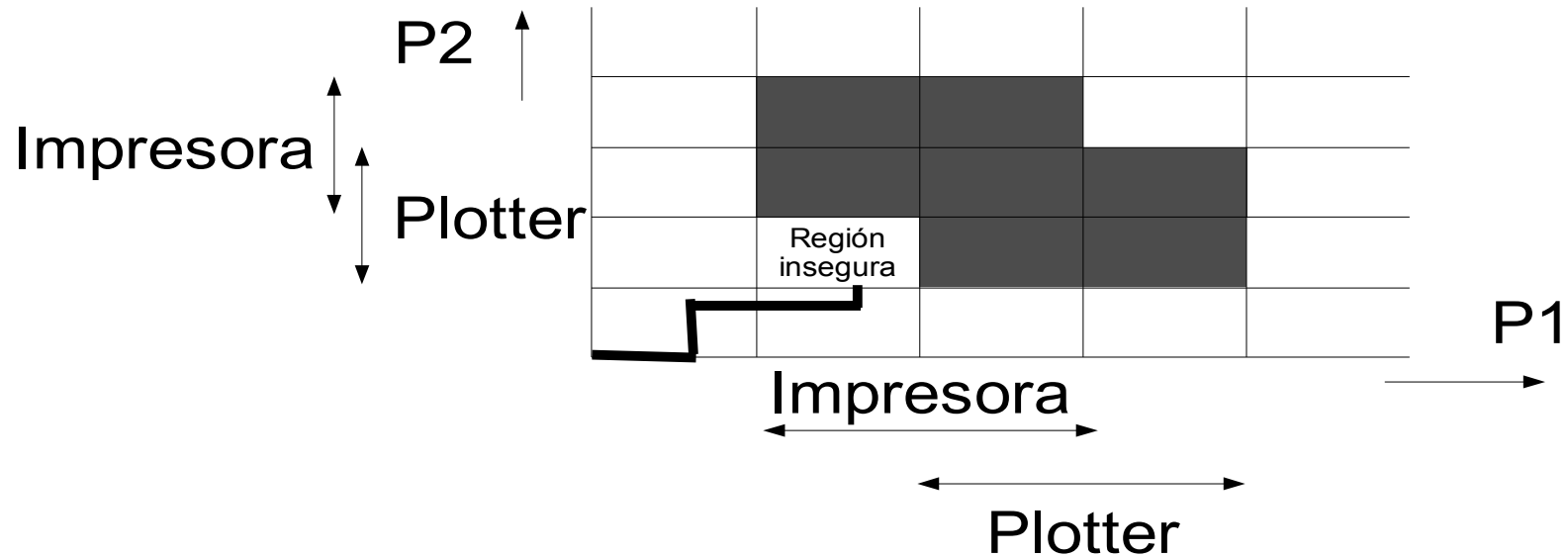
Un estado es seguro si el sistema puede asignar recursos a cada proceso hasta alcanzar el máximo de sus necesidades siguiendo algún orden arbitrario y aun así evitar el interbloqueo



Evitación del interbloqueo

- Existen diferentes modelos para determinar si un estado es inseguro. Dos de ellos son:
 - **Trayectoria de recursos.** Utiliza una representación gráfica de la evolución del sistema
 - **Algoritmo del banquero.** Algoritmo desarrollado por Dijkstra

Trayectoria de recursos



Última modificación 07/03/20



Algoritmo del banquero

- Este algoritmo se ejecutará cada vez que un proceso solicite un recurso y decidirá si la petición se concede (no conduce a un estado inseguro)
- Datos de entrada:
 - `Solicitud: ARRAY [1..m] DE NATURAL;`
Representa el número de unidades de cada recurso solicitadas (m es el número de tipos de recursos diferentes en el sistema).

Algoritmo del banquero

- **Datos internos:**

- Disponible: ARRAY [1..m] DE NATURAL; **Número de recursos disponibles de cada tipo**
- Maximo: ARRAY [1..n][1..m] DE NATURAL;
Matriz que indica la cantidad máxima de cada recurso que puede necesitar un proceso (n es el número de procesos)
- Asignacion: ARRAY [1..n][1..m] DE NATURAL;
Matriz que indica la cantidad de cada recurso asignados a cada proceso
- Necesidad: ARRAY [1..n][1..m] DE NATURAL;
Necesidad = Maximo - Asignacion

Algoritmo del banquero

1. SI $\exists i$, $Solicitud[i] \geq Necesidad[p][i]$ ENTONCES denegar solicitud
// El proceso ha excedido su demanda máxima
2. SI $\exists i$, $Solicitud[i] > Disponible[i]$ ENTONCES denegar solicitud
// No hay recursos suficientes
3. Calcular nuevo estado:
 1. $Disponible'[i] := Disponible[i] - Solicitud[i]$, para $1 \leq i \leq m$
 2. $Asignacion'[p][i] := Asignacion[p][i] + Solicitud[i]$, para $1 \leq i \leq m$
// p es el identificador del proceso que realiza la reserva
 3. $Necesidad'[p][i] := Necesidad[p][i] - Solicitud[i]$, para $1 \leq i \leq m$
4. SI $estadoSeguro(Disponible', Asignacion', Necesidad')$
ENTONCES aceptar solicitud de recursos, EN OTRO CASO denegar la solicitud

Algoritmo estadoSeguro

- Variables
Trabajo: ARRAY [1..m] DE NATURAL;
Fin: ARRAY [1..n] DE LOGICO;
- Algoritmo
 1. Fin[i] := FALSO, para $1 \leq i \leq n$
 2. Trabajo[i] := Disponible[i], para $1 \leq i \leq m$
 3. MIENTRAS $\exists p, \neg \text{Fin}[p] \wedge (\forall i, \text{Necesidad}[p][i] \leq \text{Trabajo}[i])$
 4. Trabajo[i] := Trabajo[i] + Asignacion[p][i], para $1 \leq i \leq m$
 5. Fin[p] := CIERTO
 6. SI $\forall i, \text{Fin}[i]$ ENTONCES el estado es seguro, EN OTRO CASO el estado es inseguro

Detección y Recuperación

- Esta estrategia de tratamiento del interbloqueo se basa en dejar que este ocurra, detectarlo y realizar un proceso de recuperación del interbloqueo
- Para el proceso de detección se utiliza el algoritmo `estadoSeguro`

Detección y Recuperación

- Dos aspectos fundamentales para la aplicación de esta estrategia:
 1. Con que frecuencia se utiliza el algoritmo de detección de interbloqueo
 2. Como se recupera el interbloqueo

Detección del interbloqueo

- El algoritmo de detección puede utilizarse cada vez que se solicita un recurso o de forma periódica
- Si se ejecuta con mucha frecuencia implica con consumo excesivo de computo
- Si se ejecuta con poca frecuencia se desaprovechan recursos bloqueados por los procesos en interbloqueo

Recuperación del interbloqueo

- Aviso al operador
- Terminación de procesos. Alternativas:
 - Abortar todos los procesos interbloqueados
 - Abortar los procesos de uno en uno hasta que desaparezca el interbloqueo
- Expropiación de recursos. Problemas:
 - Selección de la víctima
 - Retroceso
 - Postergación indefinida



Terminación de procesos

- Factores que influyen en la elección del proceso a eliminar:
 - Prioridad
 - Tiempo de ejecución consumido y previsto
 - Recursos usados y tipo
 - Necesidades futuras de recursos
 - Número de procesos
 - Tipo de procesos

Estrategias combinadas

- Basa su funcionamiento en utilizar la mejor estrategia de tratamiento de interbloqueo para cada recurso en función de las características del recurso
- Los recursos se agrupan en 4 categorías dependiendo de sus características

Categorías de recursos

- **Recursos internos** (recursos del sistema, ej. tablas de procesos). Prevención mediante ordenación de recursos
- **Memoria principal y CPU.** Prevención mediante expropiación
- **Recursos usados por los procesos.** Evitación, prevención mediante ordenación de recursos. prevención de la exclusión mutua mediante spooling.
- **Espacio de intercambio.** Prevención mediante adquisición anticipada