

Sistemas Operativos

Tema 11. Interbloqueo



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

© 1998-2008 José Miguel Santos - C. Rubén García - Alexis Quesada

1

Contenidos

- Caracterización del interbloqueo
- Estrategias de tratamiento del interbloqueo
- Métodos de prevención
- Métodos de evitación
- Detección del interbloqueo
- Recuperación tras un interbloqueo



2

Modelo del sistema

- El **interbloqueo** es un problema que afecta a *procesos* concurrentes que utilizan *recursos* en un sistema.
- Los procesos solicitan recursos al sistema y los liberan cuando ya no los necesitan. Un recurso puede estar disponible o bien asignado a algún proceso.



3

Modelo del sistema

- **Ejemplares.** Puede haber varios ejemplares de un mismo tipo de recurso (ej. varias impresoras). En este caso, cuando un proceso solicita un recurso, se le concede *cualquiera* de los ejemplares que esté disponible.
- Si un proceso solicita un recurso que no tiene ejemplares disponibles, el proceso queda bloqueado, esperando hasta que se le asigna un ejemplar.



4

Modelo del Sistema

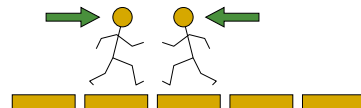
- Esquema de funcionamiento normal
 - 1.- Solicitud
 - 2.- Uso
 - 3.- Liberación
- **Llamadas al Sistema**
- ¿ Como saber en qué estados están los recursos ?
 - Tabla de sistema podría registrar si cada recurso está libre o asignado, y si un recurso está asignado, a qué proceso se le asignó



5

El problema

- Un conjunto de procesos bloqueados, cada uno de ellos esperando por un recurso que retiene otro proceso de ese conjunto.
 - ningún proceso del conjunto puede avanzar
 - interbloqueo, bloqueo mutuo, abrazo mortal (deadlock).



6

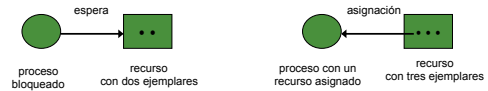
¿De quién es la culpa del interbloqueo?

- Muchas veces, el interbloqueo no es responsabilidad de las aplicaciones, sino del sistema de gestión de recursos
- Ejemplo: Los procesos A y B se pueden interbloquear, aunque están escritos correctamente

Proceso A	Proceso B
Pide (escáner)	Pide (impresora)
Pide (impresora)	Pide (escáner)
usa impr. y escáner	usa impr. y escáner
Libera (impresora)	Libera (escáner)
Libera (escáner)	Libera (impresora)

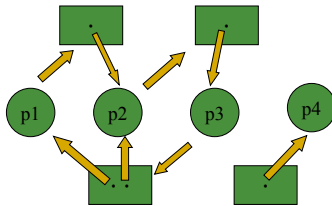
Grafo de asignación de recursos

- Sirve para representar el estado de un sistema de asignación de recursos. Muestra esta información:
 - cuántos ejemplares hay de cada tipo de recurso
 - los procesos activos en el sistema
 - qué recursos están asignados y a qué proceso
 - qué procesos están bloqueados y por cuáles recursos



Ejemplo de grafo de asignación

P1, P2 y P3 están en interbloqueo



Condiciones para el interbloqueo

- Si en un sistema se produce una situación de interbloqueo, entonces se cumplen simultáneamente estas cuatro condiciones:
 - **Exclusión mutua.** Los recursos no se pueden compartir.
 - **Retención y espera.** Un proceso que retiene uno o varios recursos se encuentra esperando por recursos asignados a otros procesos.
 - **No expropiación.** Un recurso sólo puede ser liberado por el proceso que lo retiene, voluntariamente.
 - **Espera circular.** Existe una serie de procesos en espera $\{P_0, P_1, \dots, P_n\}$ en la que todo P_i espera por un recurso retenido por P_{i+1} ; y P_n espera por un recurso retenido por P_0 .

Tratamiento del interbloqueo

- Garantizar que en el sistema nunca ocurren interbloqueos
 - **prevención:** diseñar el sistema de manera que nunca se cumpla alguna de las cuatro condiciones del interbloqueo.
 - **evitación:** tratar de no caer nunca en un estado de interbloqueo.
- Permitir la aparición de interbloqueos y recuperarse cuando ocurran
 - necesitamos un sistema de detección y un mecanismo de recuperación
- No tratar el problema
 - si hay interbloqueos, el usuario tiene que intervenir

Prevención del interbloqueo

- Se trata de eliminar la aparición de alguna de las cuatro condiciones necesarias para el interbloqueo.
 - **Exclusión mutua.** Depende de la naturaleza del recurso, así que esta condición no se puede eliminar.

Prevención del interbloqueo (2)

- **Retención y espera.** Hay que garantizar que un proceso no pueda quedar bloqueado si retiene algún recurso. ¿Cómo conseguirlo?
 - el proceso tiene que pedir todos sus recursos de una vez, p.ej. antes de empezar a ejecutarse
 - efecto negativo: muchos recursos retenidos pero no usados,
 - un proceso sólo puede solicitar recursos cuando no tiene ninguno asignado
 - Efecto negativo: puede ocurrir que tengamos que liberar un recurso y volver a pedirlo para poder solicitar otros recursos
 - En ambos caso puede que un proceso nunca se ejecute (**inanición**)

Prevención del interbloqueo (3)

- **No expropiación.** Permitir que el S.O. desasigne recursos a un proceso bloqueado.
 - Si un proceso se bloquea por un recurso, los recursos retenidos quedan a disposición de los procesos activos
 - El proceso bloqueado tiene ahora que esperar por *todos* los recursos
 - penaliza a los procesos que necesitan muchos recursos
 - Es posible seguir este protocolo en recursos cuyo estado se puede guardar fácilmente y después restaurarse (registros de CPU, espacio de memoria, ...). Generalmente no puede aplicarse a recursos tales como impresoras y unidades de cinta

Prevención del interbloqueo (4)

- **Espera circular.** Se puede evitar forzando un orden en la petición de los recursos.
 - Cada recurso tiene asignado un número de orden
 - Los recursos se deben pedir en orden ascendente
 - Aconsejable: el orden de petición de los recursos se establezca según el orden de uso normal de los recursos de un sistema
 - Efectos negativo
 - se limita la libertad de escritura de código
 - se puede inducir a una mala utilización de los recursos

Evitación del interbloqueo: algoritmo del banquero

- Se trata de conceder los recursos sólo cuando no representen un riesgo futuro de interbloqueo.
- Lo procesos han de declarar por anticipado la cantidad máxima de recursos que van a utilizar a lo largo de su vida
- **Estado seguro:** un estado en el cual no hay riesgo inminente de interbloqueo. Un estado es seguro si en él podemos encontrar una **secuencia segura** con todos los procesos del sistema
- $\{P_1, P_2, \dots, P_N\}$ es una **secuencia segura** si los recursos que P_i puede pedir en el peor caso se pueden atender con lo que hay disponible más los recursos poseídos por todos los procesos $P_{j < i}$
- Sólo concedemos recursos si el estado resultante tras la petición es seguro

Algoritmo del banquero (2)

- ¿ Qué significa una secuencia segura ?
 - Nos ponemos en el peor caso del sistema: que todos los procesos soliciten al mismo tiempo el máximo de recursos a los que tiene derecho
 - El primer proceso de la secuencia es uno que podría finalizar en ese peor caso, con los recursos disponibles en el sistema
 - El segundo proceso es uno que puede finalizar con lo que hay disponible más los recursos que liberaría el primer proceso
 - De la misma forma, los siguientes procesos pueden finalizar con los recursos que han liberado los anteriores en la secuencia
 - Y si todos los procesos pueden terminar, es que no hay interbloqueo

Algoritmo del banquero (3)

- Cuando un proceso realiza una petición, el SO calcula si tras conceder los recursos el sistema pasa a un estado seguro
 - si el nuevo estado es seguro, se concede la petición
 - si el nuevo estado no es seguro, el proceso queda bloqueado (aunque existan recursos suficientes para atender la petición). La petición se concede cuando se observa que no hay riesgo de interbloqueo

Algoritmo del banquero: ejemplo

- Dos recursos R1 y R2, con 5 y 6 ejemplares
- En el instante actual quedan libres 1 y 1 ejemplares

	Asignado	Máximo	Necesidades
Pa	1 0	2 2	1 2
Pb	1 3	3 4	2 1
Pc	2 2	3 2	1 0

- El estado es seguro porque existe la secuencia segura {Pc, Pa, Pb}
- ¿ Qué pasa si Pa pide un ejemplar de R1 y se lo damos ?
 - El sistema quedará en un estado inseguro

Detección del interbloqueo

- El interbloqueo se puede detectar comprobando si existe una secuencia de terminación de procesos (similar a la sec. segura):
 - Sea L la lista de procesos del sistema y R el conjunto de recursos disponibles
 - 1. Buscar en L un proceso que puede continuar con los recursos disponibles en R
 - 2. si no se encuentra ningún proceso, ir al paso 5
 - 3. suponer que P termina (lo retiramos de L) y que libera los recursos que retiene (los añadimos a R)
 - 4. volver al paso 1
 - 5. Si L no está vacía, hay interbloqueo

Recuperación del interbloqueo

- Un sistema que pretenda recuperarse del interbloqueo, debe invocar a un algoritmo de detección cuando lo considere oportuno (ej. periódicamente)
- Formas de intentar la recuperación:
 - Terminación de procesos
 - Expropiación de recursos

Recuperación del interbloqueo (2)

- Terminación de procesos
 - matando a todos los procesos implicados (drástico)
 - matando a uno de los procesos ¿cuál?
 - el que más recursos libere
 - el que menos tiempo lleve en ejecución...
 - retrocediendo la ejecución de algún proceso (*rollback*)
 - muy complicado de implementar y necesita que el programa esté diseñado para que pueda retroceder

Recuperación del interbloqueo (3)

- Expropiación de recursos
 - Selección de la víctima
 - ¿ Qué recursos y de que procesos se expropián ?
 - Retroceso
 - Si expropiamos un recurso de un proceso, ¿ qué hacemos con ese proceso ?
- En ambos casos (terminación de procesos o expropiación de recursos) hay que tener cuidado de no provocar la inanición de procesos