



UD 4. Interbloqueos



Concurrencia y Sistemas Distribuidos



Objetivos de la Unidad Didáctica

- ▶ Comprender los problemas que se pueden producir con el uso incorrecto de los mecanismos de sincronización.
 - ▶ Analizar las situaciones de interbloqueo.

- ▶ Conocer las técnicas de gestión de interbloqueos:
 - ▶ Prevención
 - ▶ Evitación
 - ▶ Detección y recuperación
 - ▶ No actuación (ignorar el problema)



Bibliografía

- ▶ Capítulo 4 libro base de la asignatura
- ▶ Operating system concepts with Java. Abraham Silberschatz Peter B Galvin; Greg Gagne Hoboken, NJ : John Wiley & Sons cop. 2007 7th ed. ISBN 9780471769071. Chapter 7 – Deadlocks.
- ▶ Documentos:
 - ▶ **System Deadlocks.** Edward G. Coffman Jr., M. J. Elphick y Arie Shoshani. ACM Comput. Surv., 3(2):67–78, 1971.
<http://dl.acm.org/citation.cfm?id=356588>
 - ▶ Richard C. Holt. **Some deadlock properties of computer systems.** ACM Comput. Surv., 4(3):179–196, 1972.
<http://dl.acm.org/citation.cfm?id=356607>
- ▶ Java concurrency: *Liveness* \rightarrow *Deadlock*; *Starvation* and *Livelock*
<http://docs.oracle.com/javase/tutorial/essential/concurrency/>



Contenido

- ▶ Concepto
- ▶ Condiciones de Coffman
- ▶ Ejemplos de interbloqueo
- ▶ Representación gráfica: GAR
- ▶ Soluciones
- ▶ Estrategias de prevención
- ▶ Ejemplos de solución



- ▶ **Recurso:** cualquier elemento físico o lógico que solicita un hilo (impresora, semáforo, ...)
- ▶ Un recurso puede tener varias **instancias**
 - ▶ son equivalentes para el hilo
 - ▶ ej.- pedimos una impresora y nos sirve cualquiera de ellas
- ▶ Un hilo A espera a otro hilo B cuando
 - ▶ solicita un recurso usado por B
 - ▶ y dicho recurso no es compartible (no pueden usarlo ambos a la vez)



- ▶ Los hilos usan recursos con el siguiente protocolo:
 - ▶ **Petición:** Si no está disponible, el hilo se suspende hasta que lo esté.
 - ▶ **Uso** del recurso.
 - ▶ **Liberación**
- ▶ **Interbloqueo:** existe un conjunto de hilos que no pueden evolucionar porque se esperan mutuamente



Contenido

- ▶ Concepto
- ▶ Condiciones de Coffman
- ▶ Ejemplos de interbloqueo
- ▶ Representación gráfica: GAR
- ▶ Soluciones
- ▶ Estrategias de prevención
- ▶ Ejemplos de solución



Condiciones de Coffman

▶ Condiciones de Coffman:

▶ Exclusión mutua

- ▶ Mientras un recurso está asignado a un hilo, otros no pueden usarlo

▶ Retención y espera

- ▶ Los recursos se solicitan a medida que se necesitan, de forma que podemos tener un recurso asignado y solicitar otro no disponible (espera)

▶ No expulsión

- ▶ Un recurso asignado sólo lo puede liberar su dueño (no se puede expropiar)

▶ Espera circular

- ▶ En el grupo de hilos interbloqueados, cada uno está esperando un recurso que mantiene otro del grupo, y así sucesivamente hasta cerrar el círculo

- ▶ Si todas se cumplen simultáneamente, **riesgo de interbloqueo**.
- ▶ En caso de interbloqueo, se cumplen todas.
- ▶ Son condiciones necesarias pero no suficientes.



Ejemplos

▶ **Hormigas**

- ▶ Las hormigas A y B ocupan celdas vecinas
- ▶ A quiere desplazarse donde está B (debe esperar), y B donde está A (debe esperar)

▶ Condiciones de Coffman

- ▶ **Exclusión mutua.**- una celda no es compartible
- ▶ **Retención y espera.**- cada una utiliza una celda y espera que quede libre la celda destino
- ▶ **No expulsión.**- una hormiga no puede empujar a otra fuera de su celda
- ▶ **Espera circular.**- A espera a B, y B espera a A



Ejemplos

▶ 5 Filósofos

- ▶ Llegan todos a la mesa a la vez
 - ▶ Todos tienen tiempo de coger su tenedor derecho
 - ▶ Cuando van a coger el izquierdo, todos tienen que esperar

▶ Condiciones de Coffman

- ▶ **Exclusión mutua.**- Un tenedor no se puede usar simultáneamente por 2 filósofos
- ▶ **Retención y espera.**- Usan (retienen) el tenedor de la derecha, y esperan el de la izquierda
- ▶ **No expulsión.**- No se puede robar el tenedor del vecino cuando lo está usando
- ▶ **Espera circular.**- Cada uno del grupo espera al vecino, hasta cerrar el círculo



Ejemplos

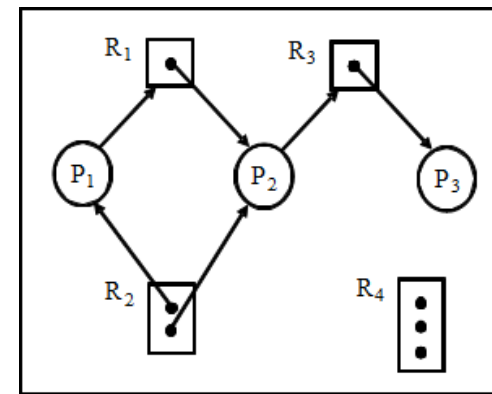
- ▶ **Transferencias entre cuentas bancarias**
 - ▶ Durante la ejecución concurrente de transferencia(a,b) y transferencia(b,a), ambas pueden reservar la cuenta origen antes de que ninguna reserve la cuenta destino
- ▶ **Condiciones de Coffman**
 - ▶ **Exclusión mutua.-** Cada cuenta debe reservarse antes de su uso
 - ▶ **Retención y espera.-** Cada hilo tiene una cuenta reservada, y pide reservar la otra
 - ▶ **No expulsión.-** No podemos apropiarnos de una cuenta reservada por otro hilo
 - ▶ **Espera circular.-** Cada hilo espera al otro



Contenido

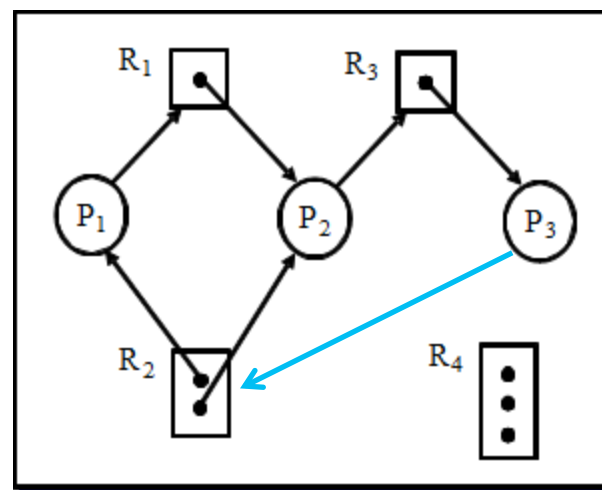
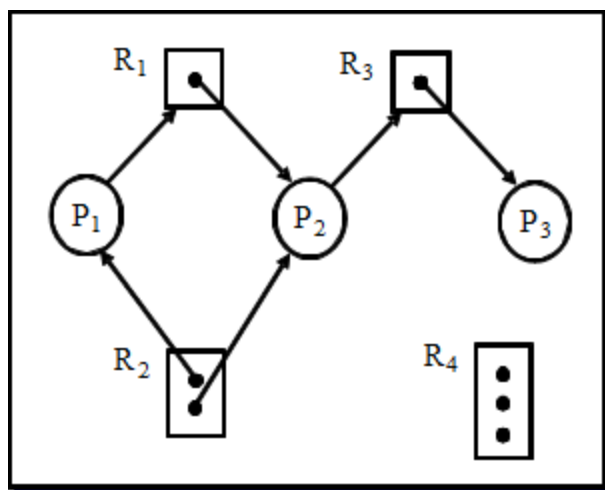
- ▶ Concepto
- ▶ Condiciones de Coffman
- ▶ Ejemplos de interbloqueo
- ▶ Representación gráfica: GAR
- ▶ Soluciones
- ▶ Estrategias de prevención
- ▶ Ejemplos de solución

- ▶ **GAR = grafo de asignación de recursos**
(representación gráfica del estado del sistema)

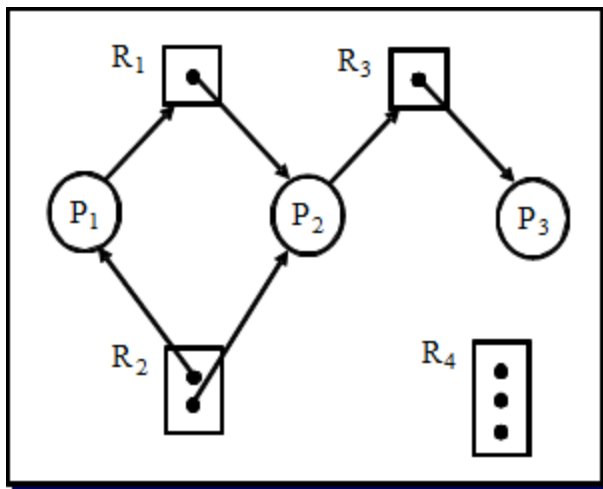


- ▶ Un **círculo** representa al hilo (o proceso)
- ▶ Un **rectángulo** representa al recurso (y contiene tantos puntos internos como instancias tiene el recurso)
- ▶ Arista de asignación: un **recurso asignado** es una flecha desde una instancia concreta al hilo que la utiliza
- ▶ Arista de petición: una **solicitud no resuelta** (no hay instancias libres del recurso, por lo que el hilo debe esperar) es una flecha que va desde el hilo al recurso (y no a una instancia concreta)

- ▶ En la figura izquierda:
 - ▶ Hilos P1, P2 y P3
 - ▶ Recursos R1, R2, R3, R4 (con 1,2,1 y 3 instancias respectivamente)
 - ▶ La instancia de R1 está asignada a P2, la de R3 a P3, y las de R2 a P1 y P2 (una instancia a cada uno)
 - ▶ P1 está esperando una instancia de R1, y P2 una instancia de R3
- ▶ Si P3 solicita R2 (flecha azul), llegamos a la figura derecha

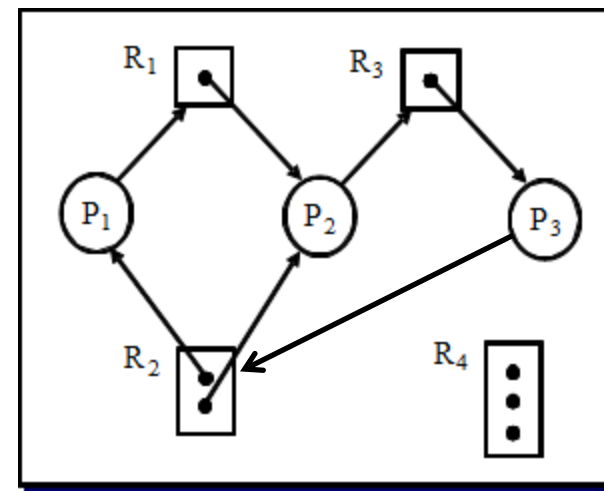
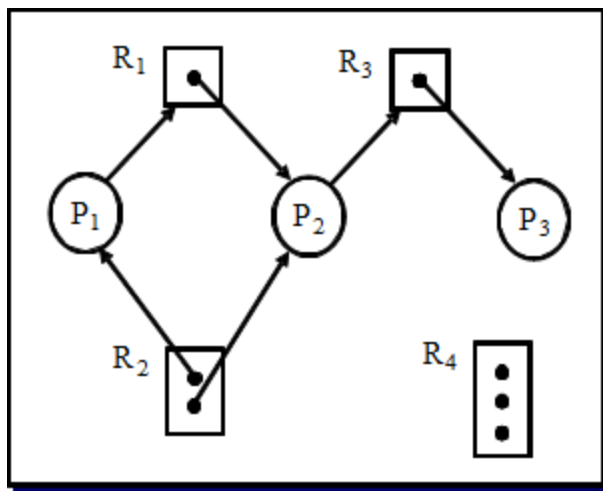


- ▶ Un **ciclo dirigido** en el GAR implica *riesgo de interbloqueo*
 - ▶ Si todos los recursos que participan en el ciclo tienen una única instancia y el ciclo dirigido, entonces tenemos un **interbloqueo**.
- ▶ Si existe un interbloqueo, seguro que hay un ciclo dirigido en el GAR
- ▶ La figura presenta el camino P1,R1,P2,R3,P3. No hay ciclos, por lo que no hay interbloqueo

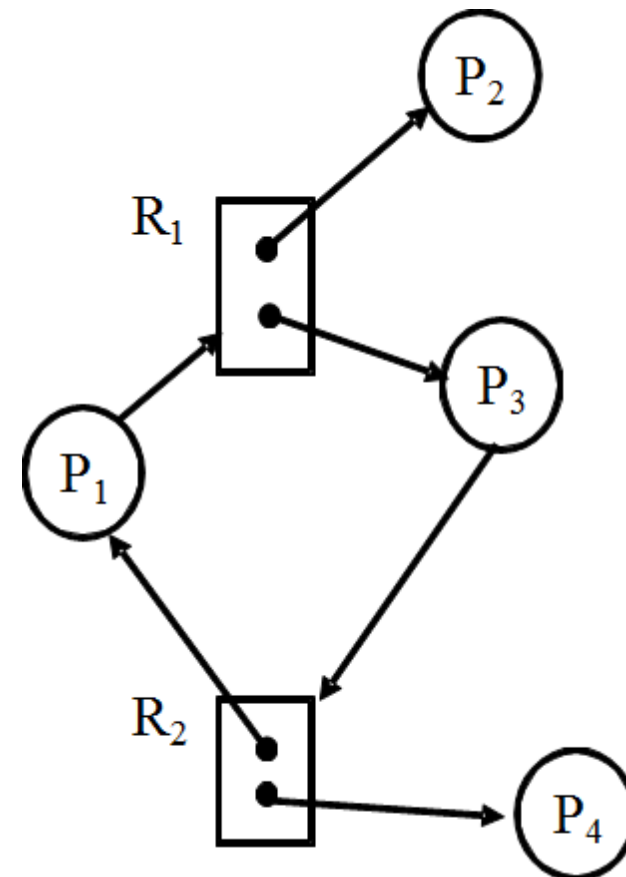


Representación Gráfica: GAR

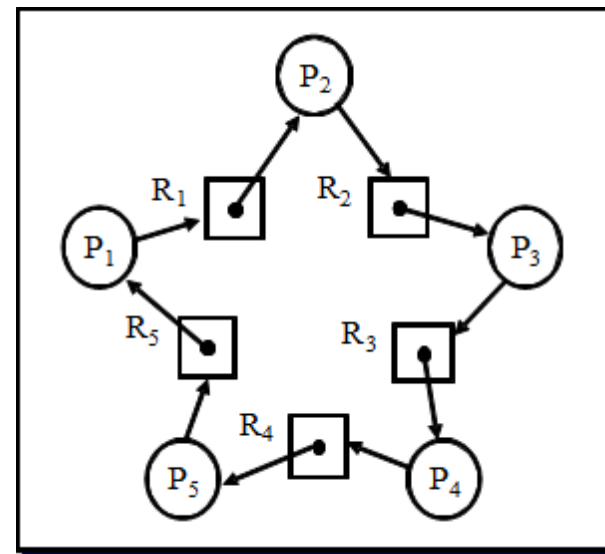
- ▶ La figura derecha presenta el ciclo $P_1, R_1, P_2, R_3, P_3, R_2, P_1$.
- ▶ R_2 posee más de una instancia: hay **riesgo** (no certeza) de interbloqueo.
- ▶ Si encontramos un orden en que puedan terminar todos los hilos (**secuencia segura**), significa que no hay interbloqueo.
- ▶ En este ejemplo no existe una secuencia segura, porque los tres hilos están esperando. Por tanto, **sí hay interbloqueo**.



- ▶ Aparece el ciclo P_1, R_1, P_3, R_2, P_1
- ▶ R_1 y R_2 tienen más de una instancia: *riesgo de interbloqueo*
- ▶ Encontramos la secuencia segura P_2, P_1, P_3, P_4 (hay otras)
 - ▶ NO hay interbloqueo



- ▶ GAR para el ejemplo de los 5 Filósofos
- ▶ Hay un ciclo y todos los recursos poseen una única instancia:
 - ▶ Hay **interbloqueo**
- ▶ Pregunta:
 - ▶ ¿Por qué representamos cada tenedor como un recurso con una única instancia en lugar de utilizar un único recurso tenedor con 5 instancias?





Contenido

- ▶ Concepto
- ▶ Condiciones de Coffman
- ▶ Ejemplos de interbloqueo
- ▶ Representación gráfica: GAR
- ▶ Soluciones
- ▶ Estrategias de prevención
- ▶ Ejemplos de solución



▶ De mejor a peor:

1. Prevención

- ▶ Diseñar un sistema que rompa alguna Condición de Coffman.
- ▶ El interbloqueo NO es posible.

2. Evitación

- ▶ El sistema, usando un GAR, considera cada solicitud y decide si es seguro atenderla en ese momento.
- ▶ Si una solicitud crea un ciclo (riesgo de interbloqueo), deniega esa solicitud (*algoritmo del banquero*).
- ▶ No puede llegar a interbloqueo, pero requiere comprobar cada solicitud.



▶ De mejor a peor (cont.):

3. Detección y recuperación

- ▶ Monitoriza periódicamente el sistema (detección).
- ▶ Si hemos llegado a una situación de interbloqueo, aborta alguna(s) de las actividades involucradas (proceso de recuperación).

4. Ignorar el problema

- ▶ No resuelve nada, pero es una postura cómoda y frecuente.
- ▶ Ej. muchos Sistemas Operativos utilizan esta solución, como Unix y Windows.



Contenido

- ▶ Concepto
- ▶ Condiciones de Coffman
- ▶ Ejemplos de interbloqueo
- ▶ Representación gráfica: GAR
- ▶ Soluciones
- ▶ Estrategias de prevención
- ▶ Ejemplos de solución



Prevención

- ▶ La prevención implica que se rompa alguna de las *condiciones de Coffman*.
- ▶ **Exclusión mutua**
 - ▶ Depende del sistema, pero en muchos casos la mayor parte de recursos deben usarse en exclusión mutua por definición (recursos no compartibles)
- ▶ **Retención y espera**
 - ▶ La forma normal de trabajar es solicitar recursos a medida que los necesitamos, lo que conduce a retención y espera.
 - ▶ **Solución 1.-** Pedir desde el principio todo lo que podemos llegar a necesitar.
 - ▶ **Solución 2.-** Solicitar recursos de forma no bloqueante
 - ▶ Si el recurso está en uso, el hilo no se bloquea, sino que recibe un valor que indica dicha situación
 - ▶ Si un hilo no puede obtener todos los recursos que necesita, libera aquellos que ha retenido.



- ▶ **Retención y espera (cont.):**
 - ▶ **Ambas soluciones...**
 - ▶ ...reducen drásticamente la concurrencia.
 - muchos hilos tienen que esperar, o bien hay que reintentar repetidamente las reservas.
 - ▶ ...suponen una baja utilización de los recursos.
 - ▶ Pueden provocar inanición de los hilos que necesiten muchos recursos solicitados con frecuencia por otros hilos.



Prevención

▶ **No expulsión**

- ▶ Permitimos que un hilo expropie recursos de otro
- ▶ El hilo expropiado tendría que volver a solicitar recursos (mayor coste)
- ▶ Podríamos tener a varios hilos que no avanzan porque se están expropiando mutuamente recursos y nadie llega a tener todo el conjunto que necesita (*livelock*)

▶ **Espera circular**

- ▶ Establecemos un orden total entre los recursos, y obligamos a solicitar los recursos en orden
- ▶ Suele ser la condición más fácil de romper



Solución a los ejemplos

- ▶ Soluciones al problema de los **5 filósofos** (*prevención de interbloqueos*)
 - ▶ **Asimetría: rompe “espera circular”**
 - ▶ Todos igual excepto el último (todos cogen primero el tenedor derecho y luego el izquierdo, excepto el último, que utiliza el orden contrario)
 - ▶ Pares e impares (los pares cogen primero el tenedor derecho y luego el izquierdo, y los impares justo en orden contrario)
 - ▶ **Cada filósofo coge ambos tenedores a la vez, o ninguno**
 - ▶ Rompe “retener y esperar”
 - ▶ **Comedor** (máximo 4 filósofos sentados simultáneamente a la mesa):
 - ▶ Rompe “retener y esperar” (al menos un filósofo dispone de ambos tenedores) y “espera circular”



Solución a los ejemplos

- ▶ Solución a la **transferencia entre cuentas**
 - ▶ Existe un orden total en las cuentas (numeradas 0..N)
 - ▶ Modificamos el código de las transferencias
 - ▶ se reserva primero la cuenta de menor índice, y posteriormente la de mayor índice, etc.
 - ▶ Se rompe la “espera circular” (prevención)



Resultados de aprendizaje de la Unidad Didáctica

- ▶ Al finalizar esta unidad, el alumno deberá ser capaz de:
 - ▶ Identificar los problemas que puede originar un uso incorrecto de los mecanismos de sincronización.
 - ▶ En concreto, se identificará el problema de interbloqueo.
 - ▶ Caracterizar las situaciones de interbloqueo.
 - ▶ Identificar las técnicas de gestión de interbloqueos.
 - ▶ Describir ejemplos de solución de estrategias de prevención de interbloqueos.