INTERBLOQUEOS

Sistemas Operativos

Prof. PhD Mirella Herrera

Definición

Dados:

- Un conjunto de procesos ejecutándose en un sistema
- Un conjunto de recursos que son utilizados por dichos procesos

  Se dice que el conjunto de procesos se encuentra en un estado de interbloqueo cuando todos los procesos se encuentran esperando un recurso que mantiene retenido otro proceso del grupo.

En esa situación:

- Ningún proceso del grupo puede evolucionar (suspendido eternamente).
- Ningún proceso podrá obtener los recursos retenidos, puesto que no pueden ser liberados.

Los interbloqueos constituyen un grave problema

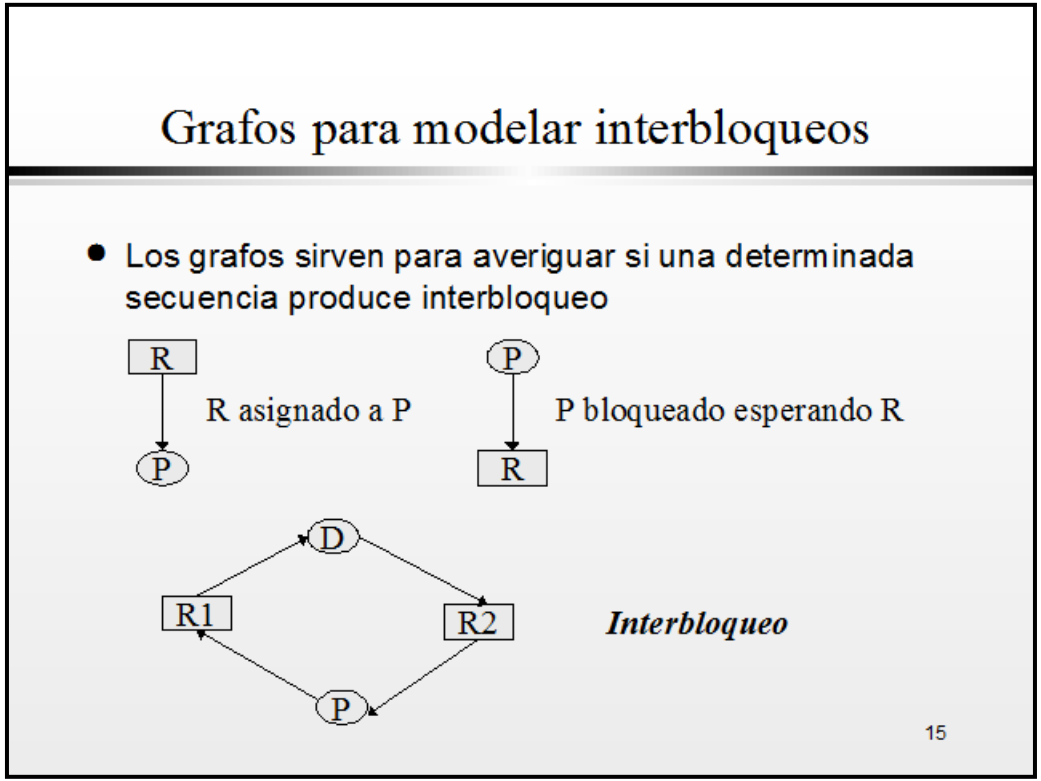
Grafos para modelar interbloqueos

Los grafos sirven para averiguar si una determinada secuencia produce interbloqueo

R -> P R asignado a P

P -> R P bloqueado esperando R

R1 -> R2 -> D -> P Interbloqueo



Condiciones de Coffman

Se ha demostrado que las siguientes cuatro condiciones son necesarias (aunque no suficientes) para que se produzca un interbloqueo:

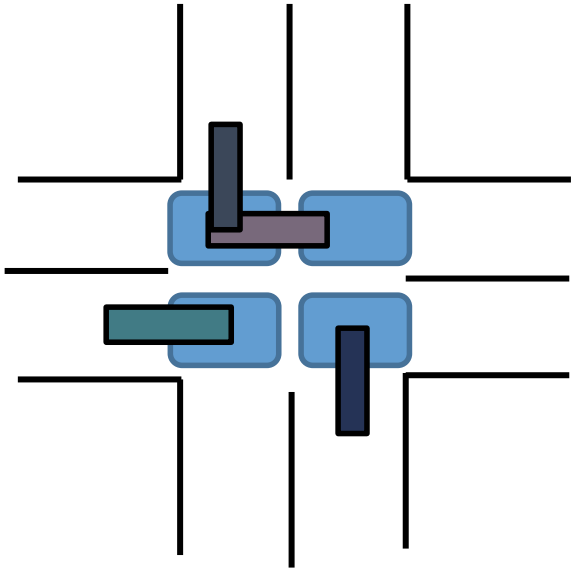
- **Exclusión mutua**: Al menos un recurso debe ser utilizado en exclusión mutua.
- **Retener y esperar**: Debe haber al menos un proceso que retenga un recurso y que haya pedido algún otro recurso que posea otro proceso, por lo que estará esperando.
- **No expropiación**: El sistema no puede arrebatar los recursos que ha asignado previamente a los procesos. Un proceso mantiene retenido un recurso hasta que deja de utilizarlo y lo libera voluntariamente.
- **Espera circular:** Debe existir un conjunto de procesos {P1, P2, ..., Pi, ..., Pn} tal que P1 se encuentra esperando un recurso que retiene P2, P2 espera un recurso que retiene P3, ..., Pn-1 espera un recurso que mantiene Pn y Pn espera un recurso que mantiene P1.

Si todas ellas se cumplen simultáneamente, el sistema se encuentra en situación de riesgo de sufrir un interbloqueo. La 4ª deriva de las 3 anteriores.

Condiciones de Coffman

Cada sección de calle se considera una sección crítica

1. **Exclusión mutua**: sólo un vehículo puede ocupar una sección de calle
2. **Retener y esperar**: cada vehículo ocupa una sección de la calle y está esperando para moverse a la siguiente sección
3. **No expropiación**: no se puede quitar una sección de la calle ocupada por un vehículo. El vehículo la liberará cuando se mueva a la siguiente sección
4. **Espera circular**: cada vehículo está esperando a que se mueva el vehículo de adelante



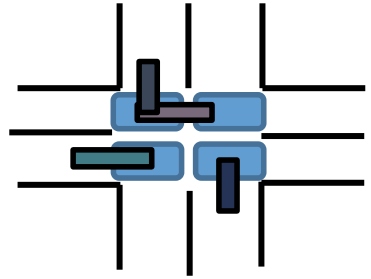
Tratamiento de Interbloqueos

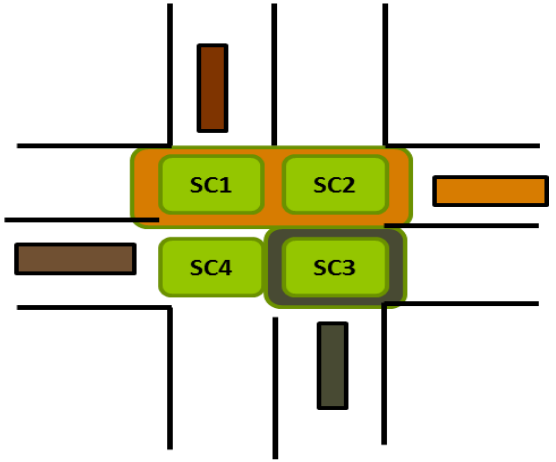
- ✓ Emplear algún algoritmo o protocolo que asegure que nunca se va a poder producir un interbloqueo. Esta solución puede adoptar dos formas alternativas:
  - ❖ **Prevención:** consiste en conseguir que no puedan darse alguna de las cuatro condiciones de Coffman. De esta forma, el interbloqueo no puede llegar a producirse.
  - ❖ **Evitación:** consiste en llevar la cuenta de los recursos disponibles en el sistema, los recursos que poseen los procesos y los que pueden llegar a solicitar. Cada vez que se hace una petición de un recurso, el sistema analiza toda esa información para conceder (o denegar) dicho recurso.
- ✓ Utilizar un algoritmo que pueda detectar una situación de interbloqueo **(detección)** y seguir alguna técnica que permita deshacer dicha situación **(recuperación).**
- ✓ Ignorar el problema **(Ostrich algorithm),** asumiendo que dicha situación nunca se dará en el sistema. Es la aproximación que mantienen muchos sistemas operativos, incluido Unix.

Prevención de interbloqueos

Negar una de las cuatro condiciones de Coffman

- **Exclusión Mutua**: Depende de la naturaleza del recurso, así que esta condición no se puede eliminar.
- **Retener y Esperar:** Para deshacer esta condición, se debe obligar a los procesos a :
  1. Utilizar los recursos de uno en uno, liberando cada recurso antes de solicitar el siguiente.
  2. Solicitar todos sus recursos de una vez, al principio de su ejecución

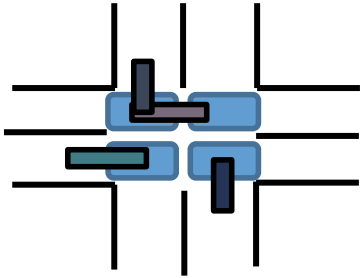


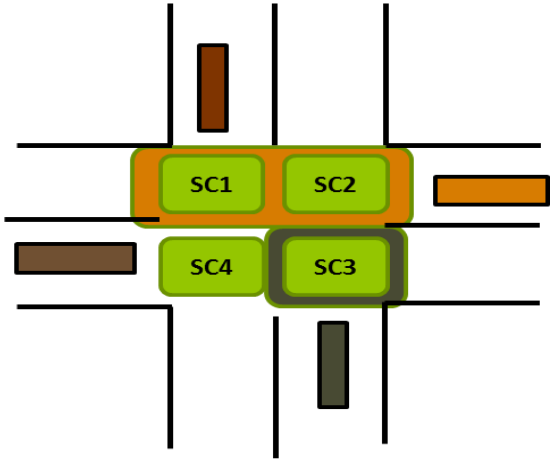


Prevención de interbloqueos

Estas aproximaciones tienen dos problemas:

- **Baja utilización de los recursos**, puesto que estarán retenidos desde el principio de la ejecución de los procesos, pero no se estarán utilizando en todo momento.
- **Inanición de los procesos** que necesiten muchos recursos solicitados muy frecuentemente por los demás procesos.



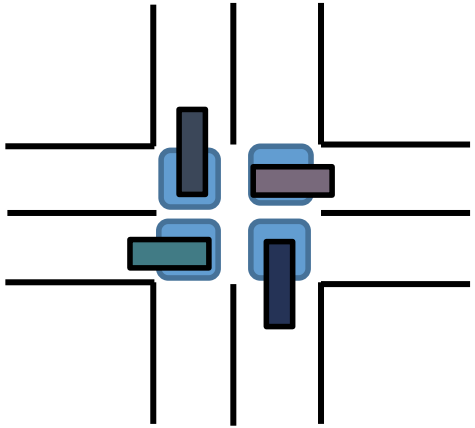


Prevención de interbloqueos

Negar una de las cuatro condiciones de Coffman

No expropiación: Permitir que el SO 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

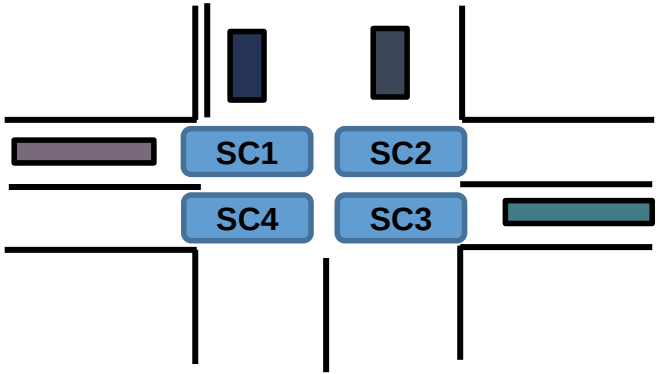


Prevención de interbloqueos

Negar una de las cuatro condiciones de Coffman

Espera Circular: Esta condición se puede romper si imponemos un orden total a los recursos, y se obliga a que los procesos soliciten siempre los recursos siguiendo dicho orden. Es decir:

- Definimos la función f: R -> N, que asocia a cada recurso un número natural
  Un proceso que posee un recurso Ri puede solicitar otro recurso Rj si y sólo si f(Ri) < f(Rj). De difícil implementación en sistemas con muchos recursos
- Bastante ineficiente



SC1 antes que SC2
SC2 antes que SC3
SC3 antes que SC4
SC1 antes que SC4

Evitación de interbloqueos

Las técnicas de prevención no son utilizables en algunas situaciones, presentan algunos inconvenientes y resultan confusas de utilizar

La evitación (también llamada predicción) permite las tres primeras condiciones de Coffman, pero controla que no se pueda producir la cuarta

Para este tipo de técnicas, el sistema debe conocer:

- La cantidad total de recursos disponibles en el sistema
- La cantidad actual de recursos disponibles en el sistema
- La necesidad máxima de recursos de los procesos
- La asignación actual de recursos a procesos

Dos técnicas:

1. Denegación de inicio de proceso
2. Denegación de asignación de recurso.
    - Concepto de estado seguro
    - Algoritmo del banquero

Evitación de interbloqueos

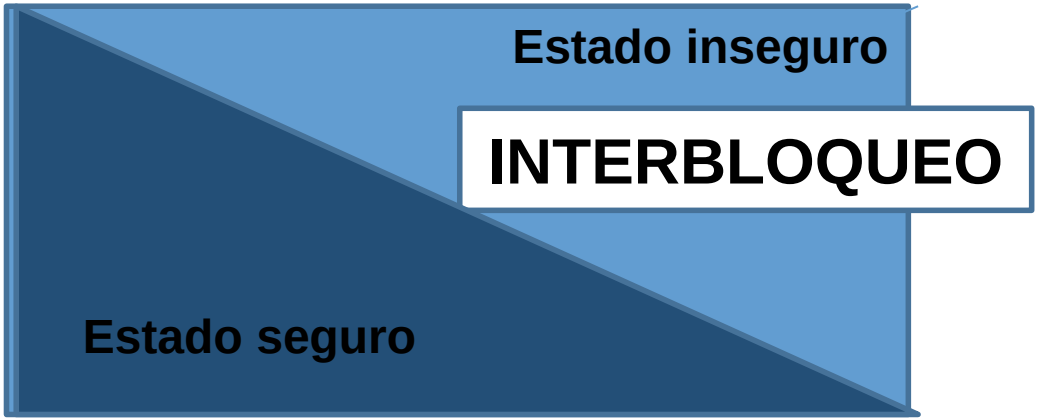
- ✓ **Denegación de inicio de proceso**
  - Evita el inicio de procesos
  - Se impide que un proceso se inicie si los recursos máximos necesarios son superiores a los disponibles
  - Incluso aunque...
    - No hubiera llegado a usarlos todos
    - Pueda ir avanzando con los disponibles
- ✓ **Denegación de asignación de recurso**
  - Evita la asignación de recursos
  - Se impide que un recurso se asigne si esto llevaría a un estado inseguro
  - Se emplea el algoritmo del banquero

Evitación de interbloqueos

Estado seguro

Un estado (de asignación de recursos) se considera cuando en él no hay posibilidad de interbloqueo

Para que un estado sea seguro, es necesario que exista al menos una secuencia segura



Evitación de interbloqueos

Secuencia segura

Es una cierta ordenación de los procesos que asegura se podrá completar la ejecución de todos ellos

- ✓ Al menos uno puede ejecutarse, y al acabar liberará recursos, que utilizará otro proceso... y así sucesivamente
- ✓ Es una visión pesimista, asume que todos los procesos necesitarán de todos los recursos al mismo tiempo
- ✓ Si dicha secuencia existe, como mucho cada proceso tendrá que esperar a que los procesos que van antes que él en la secuencia, liberen sus recursos

Algoritmo del Banquero

Cuando un proceso realice una petición de recursos, el sistema se los concederá sólo en el caso de que la petición mantenga al sistema en un estado seguro

En dicha petición se comprueba que...

1. El proceso no pida más recursos de los que originalmente necesitaba
2. Que el recurso pedido está disponible
3. Que tras la asignación el sistema queda en estado seguro

¿Hay algún proceso que pueda terminar su ejecución con los recursos restantes?
Tras acabar dicho proceso, y liberar por tanto sus recursos, ¿habría otro proceso que pudiera terminar su ejecución?
Repetir hasta que se encuentre una forma en la que todos los procesos puedan terminar su ejecución

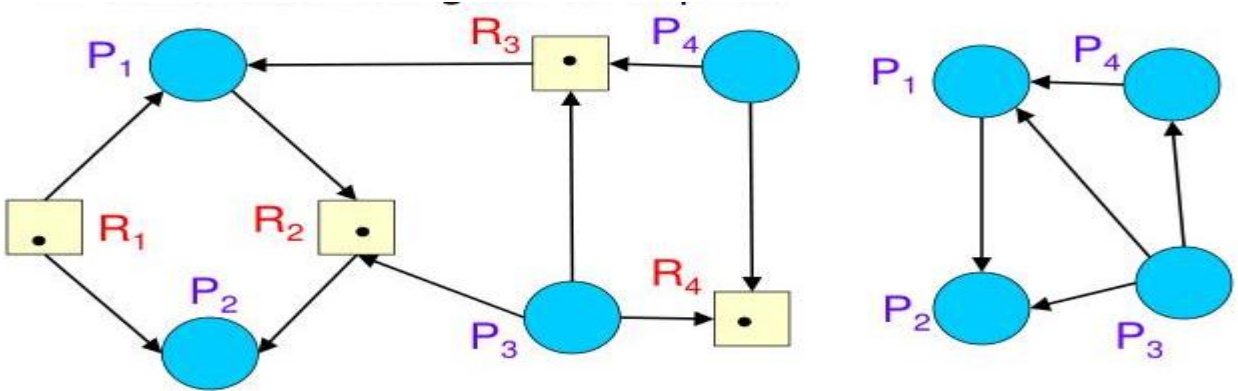
https://www.youtube.com/watch?v=kbXjKKaMUAI&t=36s

Problemas de la Evitación

- Es necesario conocer información privilegiada con antelación
- Los procesos deben ser independientes
  - Si el orden de ejecución está condicionado a los requisitos de sincronización, el sistema no es libre de elegir una secuencia segura
- El número de recursos e instancias es fijo e inalterable
  - No se pueden añadir recursos nuevos...
  - ...ni eliminarlos

Detección de interbloqueos

- Caso más sencillo
- Grafo de “espera-a” (wait-for)
- Se basa en el grafo de asignación de recursos
- Se eliminan los nodos correspondientes a recursos, y se ajustan los arcos de forma que habrá un arco del proceso Pi al proceso Pj si Pj posee un recurso que Pi ha solicitado
- Existirá interbloqueo si y sólo si hay un ciclo en el grafo



Detección de interbloqueos

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 de interbloqueos

Alternativas

- **Terminación de procesos**
  - Terminación de todos los procesos interbloqueados
  - Terminación iterativa de procesos, hasta que el interbloqueo desaparece
- **Expropiación de recursos**: El sistema expropia de recursos de los procesos interbloqueados, hasta que desaparece el interbloqueo

Problemas a resolver

- **Selección de una víctima**: A quién se elige para apropiarse de sus recursos
- **Vuelta atrás**: El proceso al que se le quitan los recursos ha de ser devuelto a un estado seguro. La solución trivial es abortar dicho proceso
- **Inanición**: Hay que considerar que no se debería quitar siempre los recursos al mismo proceso, sobre todo si la vuelta atrás supone abortarlo y obligarle a empezar desde el principio

Implantación en Sistemas Operativos

- ✓ Mayoría lo ignora o no da una solución general
- ✓ Distinción entre dos tipos de recursos:
  - **Recursos internos (propios del SO)**
    - Usados por un proceso en modo sistema
    - Uso restringido a ejecución de una llamada
    - Ej. semáforo interno para acceder a t. de procesos o buffer
    - Interbloqueo puede causar colapso del sistema
  - **Recursos de usuario**
    - Usados por un proceso en modo usuario
    - Uso durante tiempo impredecible
    - Ej. dispositivo dedicado o semáforo de aplicación
    - Interbloqueo afecta a procesos y recursos involucrados

Implantación en Sistemas Operativos

- ✓ **Tratamiento de recursos internos**
  - Código del S.O. es algo que apenas se modifica
    - Se puede estudiar a priori uso de recursos
      - ❖Interbloqueo →error de programación de S.O.
  - Uso de estrategias de prevención es adecuado
    - Dado que tiempo de uso es breve y acotado
- ✓ **Tratamiento de recursos de usuario**
  - Código de procesos que usan recursos es impredecible
  - No hay tratamiento general para todos los recursos
  - Prevención →Infrautilización
  - Predicción →Dificultad de conocer información a priori
  - Detección y recuperación →Demasiada sobrecarga
    - Puede usarse para un único recurso
    - P.ej. En 4.4BSD se usa para cerrojos sobre archivos

Resumen

Técnica	Polítca de asignación de recursos	Esquemas	Ventajas principales	Desventajas principales
Prevención	Conservadora: los recursos están poco ocupados.	Solicitud de todos los recursos a la vez.	Funciona bien con procesos que realizan una sola ráfaga de actividad. No es necesaria la expropiación.	Ineficiencia Retrasos en el inicio de los procesos Se deben conocer las necesidades futuras de recursos.
		Expropiación.	Conveniente cuando se aplica a recursos cuyo estado se puede salvar y restaurar fácilmente.	Expulsa más habitualmente de lo necesario.
		Ordenación de recursos.	Aplicación factible por medio de chequeos durante la compilación. No hace falta procesamiento durante la ejecución, ya que el problema se resuelve durante el diseño del sistema.	No permite las solicitudes incrementales de recursos.
Evasión (Predicción)	A medio camino entre la detección y la prevención.	Manipulación para encontrar al menos un camino seguro.	No es necesaria la expropiación.	Deben conocerse las demandas futuras de recursos. Los procesos pueden bloquearse durante largos periodos.
Detección	Muy liberal; los recursos solicitados se conceden cuando es posible.	Comprobación periódica del interbloqueo.	Nunca retrasa el inicio de un proceso. De fácil adaptación al entorno.	Pérdidas inherentes a la expropiación.