



Departamento de Lenguajes y Ciencias de la
Computación Universidad de Málaga

Sistemas Operativos

Curso 2001/2002

TEMA 3: Interbloqueo



E.T.S.I. Informática

Profesor:
Francisco López Valverde

<http://www.lcc.uma.es/~valverde>

Tema 3: Interbloqueo

SISTEMAS OPERATIVOS

Índice de contenidos

- Introducción al interbloqueo
- Recursos
- Condiciones para que se produzca el interbloqueo
- Modelado del interbloqueo
- Métodos para el tratamiento del interbloqueo
 - Prevención
 - Evitación
 - Trayectoria de recursos
 - Estados seguros
 - Detección
 - Recuperación
 - Estrategia combinada para el manejo de interbloques
- Fuentes de información



- La administración de los recursos es una de las principales tareas del sistema operativo.
- Los sistemas operativos tienen que ofrecer mecanismos que permitan a los procesos acceder de forma exclusiva a este tipo de recursos.
- Cuando un proceso solicita ciertos recursos y éstos no están disponibles en ese momento, entra en un estado de espera.
- Requerir el acceso exclusivo no sólo a un recurso, sino a varios.



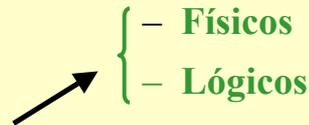
Interbloqueo

- *Deadlock*
- Abrazo mortal
- Bloqueo mutuo

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



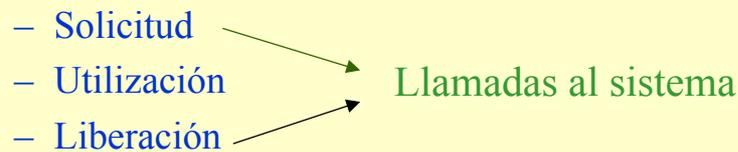
- Un sistema se compone de un número finito de recursos que se distribuyen entre varios procesos que compiten por ellos.



- Recursos



- Modo de operación:



Condiciones para que se produzca interbloqueo

- Puede surgir si y sólo si en un sistema se presentan simultáneamente las condiciones:
 - Exclusión mutua.
 - Retención y espera
 - No expropiación
 - Espera circular
- No son completamente independientes



Modelado del interbloqueo

- Los interbloqueos pueden describirse utilizando un grafo dirigido y bipartito $G(N, A)$ llamado **grafo de asignación de recursos**
- que consta en un conjunto de N nodos (vértices) y E arcos.

- 2 tipos de nodos

}	Procesos
}	Recursos
- 2 tipos de arcos

- **Arco de solicitud.** Es un arco que parte de un proceso P_i hacia un tipo de recurso R_j y se representa por $P_i \rightarrow R_j$ (o (P_i, R_j)). Significa que el proceso P_i solicitó una instancia del recurso R_j y se encuentra esperándolo.
- **Arco de asignación.** Es un arco que sale de un tipo de recurso R_j y se dirige a un proceso P_i (representado por $R_j \rightarrow P_i$ o (R_j, P_i)). Significa que se ha asignado un ejemplar del tipo de recurso R_j al proceso P_i .



Modelado del interbloqueo

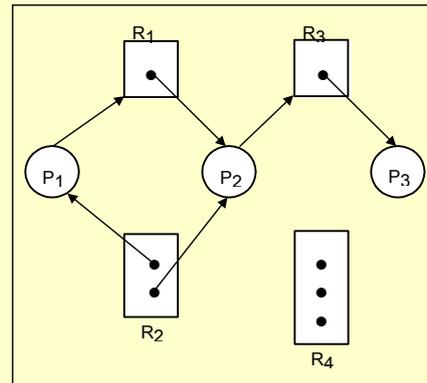
- Gráficamente, se representa cada proceso con un círculo y cada tipo de recurso con un rectángulo.
- Si de algún tipo de recurso existe más de un ejemplar, se representa cada uno con un punto dentro del rectángulo.
- Un arco de solicitud parte entonces de un círculo y apunta a un rectángulo.
- Un arco de asignación parte desde un punto dentro del rectángulo y señala hacia un círculo.

Operación	Efecto en el grafo
P_i solicita un ejemplar del tipo R_j	Se inserta un arco de solicitud $P_i \rightarrow R_j$
Una instancia del recurso R_j se asigna al proceso P_i	El arco de solicitud $P_i \rightarrow R_j$ se transforma instantáneamente en arco de asignación $R_j \rightarrow P_i$
El proceso P_i libera el recurso R_j	Se elimina el arco de asignación $R_j \rightarrow P_i$



Modelado del interbloqueo

Grafo de asignación de recursos



Estados de los procesos:

El proceso P_1 tiene asignado un recurso de tipo R_2 , y espera un recurso de tipo R_1 .

El proceso P_2 tiene asignado un recurso de tipo R_1 y otro de tipo R_2 , y espera un recurso de tipo R_3 .

El proceso P_3 tiene asignado un recurso de tipo R_3 .



Modelado del interbloqueo

- Si el grafo de asignación de recursos no contiene ciclos, entonces ningún proceso del sistema se encuentra en interbloqueo.
- Si existe un ciclo, puede haber interbloqueo.



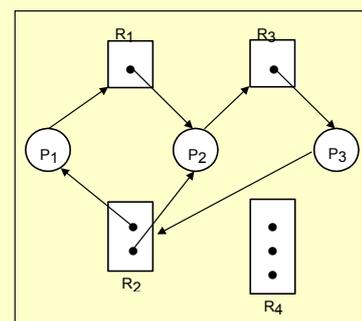
La presencia de un ciclo es condición necesaria pero no suficiente para la existencia de interbloqueo.

- Si de cada tipo de recurso existe un único ejemplar, entonces la presencia de un ciclo determina que existe interbloqueo.

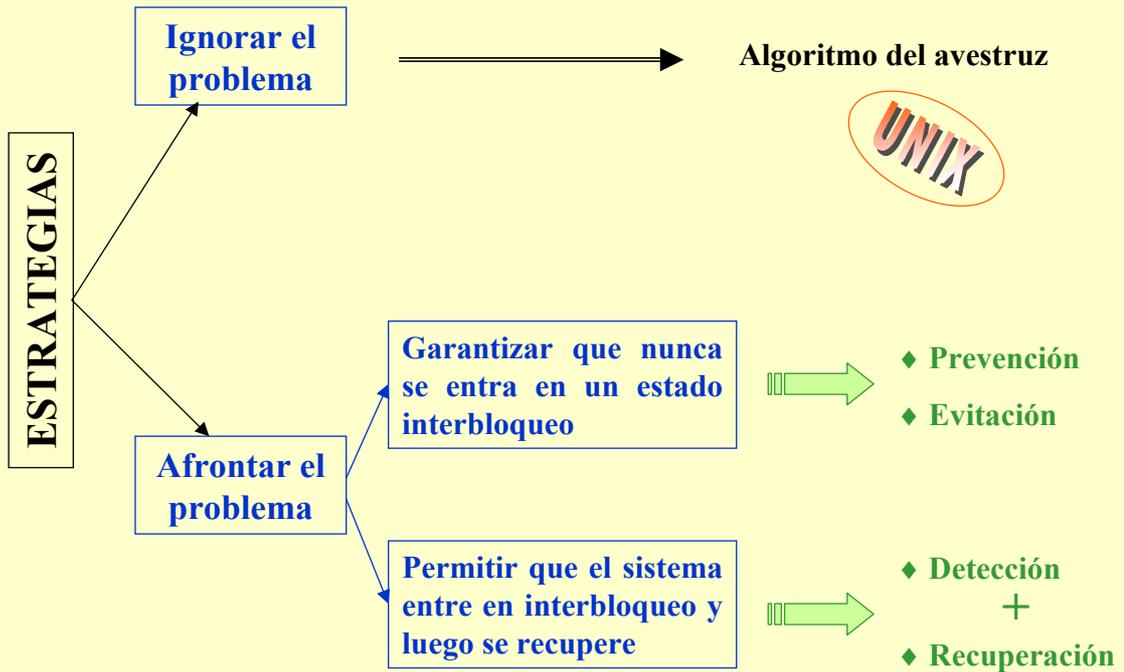


En este caso la existencia de un ciclo si es condición necesaria y suficiente para la existencia de interbloqueo.

Grafo de asignación de recursos con un interbloqueo



Métodos para el tratamiento del interbloqueo



Métodos para el tratamiento del interbloqueo

PREVENCIÓN

Objetivo: conseguir que sea imposible la aparición de situaciones de interbloqueo



Impedir que se produzca una de las cuatro condiciones necesarias para producirlo

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



Métodos para el tratamiento del interbloqueo

PREVENCIÓN

EXCLUSIÓN MUTUA

- Existen recursos para los que no es posible negar la condición de exclusión mutua
 - **Ficheros**: permiten múltiples accesos de lectura, pero únicamente un proceso puede escribir a la vez
- Es posible eliminar la condición de exclusión mutua en algunos recursos.
 - **Impresora**  **Spooler**



Métodos para el tratamiento del interbloqueo

PREVENCIÓN

RETENCIÓN Y ESPERA

Forzar a un proceso a que libere todos los recursos que retiene cada vez que solicite un recurso que no esté disponible.

Los procesos que esperan no tienen recursos y los que tienen no esperan.

POSIBILIDAD 1

El proceso solicita todos los recursos necesarios **antes** de comenzar su ejecución.



Pobre utilización de recursos



Reducción del nivel de concurrencia



Métodos para el tratamiento del interbloqueo

PREVENCIÓN

RETENCIÓN Y ESPERA

POSIBILIDAD 2

El proceso solicita los recursos de forma **incremental** a lo largo de su ejecución, pero libera todos los recursos retenidos si se encuentra con una negativa



Menos exigente



Los cambios hechos sobre la **memoria** o sobre **ficheros** pueden corromper el sistema si no se llevan a término

- Inconveniente adicional de ambos métodos

Pueden conducir a la postergación indefinida (o **inanición**) de algunos procesos que solicitan recursos muy utilizados



Métodos para el tratamiento del interbloqueo

PREVENCIÓN

NO EXPROPIACIÓN

- Permitir que el sistema revoque la propiedad de ciertos recursos a los procesos bloqueados.
- La expropiación es involuntaria



El sistema operativo debe encargarse de salvar el estado y restaurarlo



Aún más difícil de llevar a cabo que la liberación voluntaria

- Este esquema se puede aplicar sobre recursos cuyo estado puede ser salvado y restaurado posteriormente (**registros de CPU**, **memoria**).



Métodos para el tratamiento del interbloqueo

PREVENCIÓN

ESPERA CIRCULAR

- Ordenación lineal de los diferentes tipos de recursos del sistema.
- Los recursos del sistema se dividen en clases diferentes C_j ,
- Los interbloques se previenen con 2 restricciones:
 - 1 exigiendo que todos los procesos soliciten y adquieran sus recursos en un orden estrictamente creciente de las clases mencionadas.
 - 2 La adquisición de todos los recursos pertenecientes a una misma clase debe efectuarse con una sola petición (**no incrementalmente**)

- ✓ No se pueden producir ciclos.
- ✓ El respeto de los procesos a la ordenación prescrita puede ser comprobada en tiempo de compilación.

- ✓ No se aprovechan adecuadamente los recursos.



Se adquieren en el orden prescrito y no se solicitan cuando realmente se necesitan

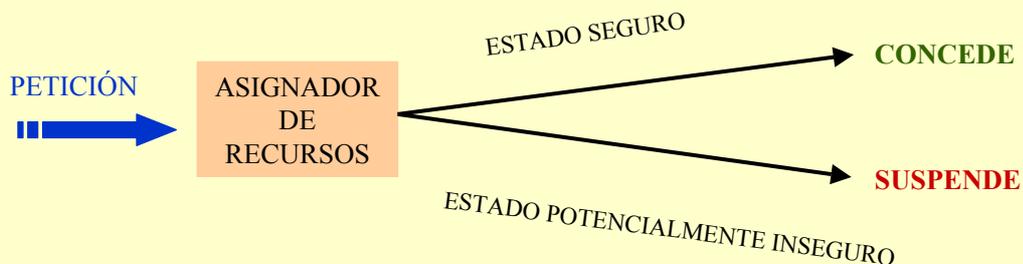


Métodos para el tratamiento del interbloqueo

EVITACIÓN

IDEA BÁSICA

Conceder únicamente las peticiones de recursos disponibles que no conduzcan a estados propensos al interbloqueo.



Examinar dinámicamente el estado de asignación de recursos para asegurar que no pueda presentarse la condición de espera circular



ESTADO SEGURO



Métodos para el tratamiento del interbloqueo

EVITACIÓN

TRAYECTORIAS DE RECURSOS

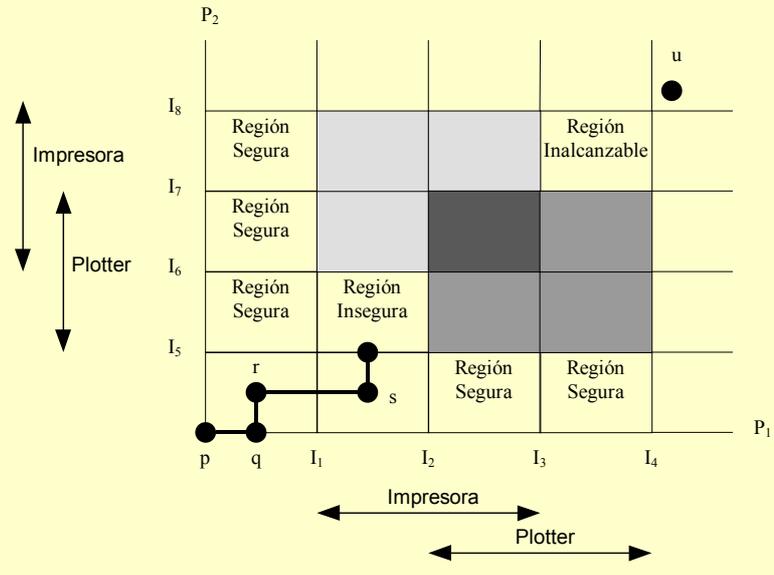


Gráfico de asignación de recursos con un interbloqueo



Métodos para el tratamiento del interbloqueo

EVITACIÓN

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 aún así evitar el interbloqueo.

Un sistema se encuentra en estado **seguro** sólo si existe una secuencia segura.

Una secuencia de ejecución de procesos $\langle P_1, P_2, \dots, P_n \rangle$ es segura para el estado actual de asignación si, para cada proceso P_i , los recursos que aún puede solicitar P_i pueden satisfacerse con los recursos actualmente disponibles más los retenidos por todos los P_j , tales que $j < i$.

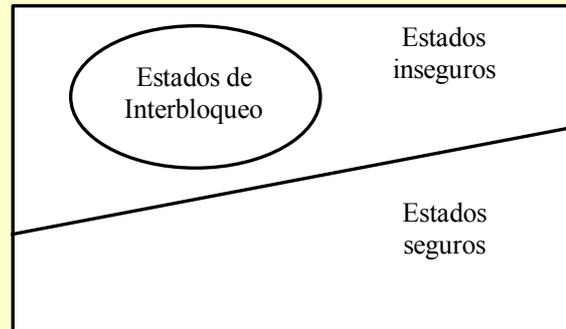
Si no existe esta secuencia, se dice que el estado es **inseguro**.



Métodos para el tratamiento del interbloqueo

EVITACIÓN

ESTADO INSEGURO \neq ESTADO DE INTERBLOQUEO



El algoritmo de evitación más famoso se debe a **Dijkstra** y se conoce con el nombre de **algoritmo del banquero**.



Métodos para el tratamiento del interbloqueo

EVITACIÓN

Problema

Sean tres procesos P_1 , P_2 y P_3 y un único tipo de recurso del que existe un total de **10** unidades. En el estado inicial P_1 tiene **3** instancias del recurso, pero podría necesitar en algún instante hasta **9**; P_2 tiene **2** pero podría necesitar **4** más adelante y P_3 tiene **2** pero podría necesitar **7**. En ese instante quedan libres **3** unidades.

- ◆ Demostrar que el estado inicial es seguro.
- ◆ Demostrar que si P_1 obtiene un ejemplar del recurso el estado resultante es inseguro.





Métodos para el tratamiento del interbloqueo

EVITACIÓN

ALGORITMO DEL BANQUERO

- Cuando un proceso entra al sistema declara el número máximo de ejemplares de cada recurso que va a necesitar (no podrá exceder del total de recursos del sistema).
- Cuando un proceso solicita un conjunto de recursos, el sistema debe determinar si su asignación dejará al sistema en un estado seguro. Si es así, los recursos se asignan; si no, el proceso esperará hasta que otro libere suficientes recursos.
- Se necesitan varias estructuras de datos para representar el estado de asignación de recursos del sistema.



Métodos para el tratamiento del interbloqueo

EVITACIÓN

ALGORITMO DEL BANQUERO

Sea n el número de procesos en el sistema y m el número de tipos de recursos.

Las estructuras de datos necesarias son:

- *Disponible*: Vector de longitud m que indica el número de recursos de cada tipo que hay disponible.
- *Max*: Matriz de $n \times m$ que define la demanda máxima de cada proceso para cada tipo de recurso.
- *Asignación*: Matriz $n \times m$ que define el número de recursos de cada tipo asignados a cada proceso en ese instante.
- *Necesidad*: Matriz $n \times m$ que indica los recursos que le pueden hacer falta a cada proceso.

$$\text{Necesidad}[i, j] = \text{Max}[i, j] - \text{Asignación}[i, j].$$

Para simplificar, usaremos la siguiente notación:

Dados X e Y dos vectores de longitud n , se dice que $X \leq Y$ si y sólo si

$$X[i] \leq Y[i] \quad \forall i = 1, 2, \dots, n. \quad X < Y \text{ si } X \leq Y \text{ y } X \neq Y.$$

Métodos para el tratamiento del interbloqueo

EVITACIÓN

ALGORITMO DEL BANQUERO

Sea $Solicitud_i$ el vector de solicitudes para el proceso P_i . P_i efectúa una solicitud de recursos y el asignador de recursos toma las siguientes acciones:

- 1 Si $Solicitud_i \leq Necesidad_i$, continuar con el paso 2. De lo contrario, devolver error, puesto que el proceso se ha excedido de su demanda máxima.
- 2 Si $Solicitud_i \leq Disponible$, continuar en el paso 3. De lo contrario, P_i deberá esperar, pues los recursos no están disponibles.
- 3 El sistema calcula el nuevo estado de asignación. Es decir,

$$Disponible := Disponible - Solicitud_i$$

$$Asignación_i := Asignación_i + Solicitud_i$$

$$Necesidad_i := Necesidad_i - Solicitud_i$$
- 4 Analizar la seguridad del nuevo estado (**algoritmo de seguridad**). Si el estado resultante **es seguro**, se confirma la transacción y los recursos son definitivamente asignados a P_i . Si el nuevo estado **no es seguro**, se restablece el estado anterior y P_i deberá esperar para obtener los recursos solicitados.

Cuando un recurso es liberado, el asignador de recursos actualiza la estructura de datos $Disponible$ y reconsidera las peticiones pendientes, de ese tipo de recurso.



Métodos para el tratamiento del interbloqueo

EVITACIÓN

ALGORITMO DE SEGURIDAD

Averiguar si un sistema está o no en un estado seguro

- 1 Sean $Trabajo$ y Fin dos vectores auxiliares de longitud m y n , respectivamente. Ambos vectores toman los siguientes valores iniciales:

$$Trabajo := Disponible$$

$$Fin[i] := falso, \text{ para todo } i=1, 2, \dots, n$$
- 2 Encontrar un i tal que se cumplan las dos proposiciones siguientes:
 - a) $Fin[i] = falso$
 - b) $Necesidad_i \leq Trabajo$
 Si no existe tal i , continuar con el paso 4.
- 3 Hacer

$$Trabajo := Trabajo + Asignación_i$$

$$Fin[i] := verdadero$$
 Continuar en el paso 2.
- 4 Si $Fin[i] = verdadero$ para todo i , entonces el sistema está en estado seguro. Si no, el estado es inseguro.



Métodos para el tratamiento del interbloqueo

EVITACIÓN

Problema

Sea un sistema con cinco procesos, P_1 a P_5 , y tres tipos de recursos, A, B y C, de los que existen 10, 5 y 7 ejemplares, respectivamente. Supongamos que en el instante actual tenemos la siguiente situación del sistema:

	Asignación			Max			Disponible			Necesidad		
	A	B	C	A	B	C	A	B	C	A	B	C
P_1	0	1	0	7	5	3	3	3	2	7	4	3
P_2	2	0	0	3	2	2				1	2	2
P_3	3	0	2	9	0	2				6	0	0
P_4	2	1	1	2	2	2				0	1	1
P_5	0	0	2	4	3	3				4	3	1

- ♦ Determinar si este estado es seguro o inseguro.
- ♦ Supongamos ahora que P_2 hace una petición $Solicitud_2 = (1, 0, 2)$. ¿Cómo actuaría el algoritmo del banquero para decidir si esta petición puede atenderse?
- ♦ Determinar si las siguientes solicitudes pueden atenderse o no:
 - Solicitud de $(3, 3, 1)$ por parte de P_5 .
 - Solicitud de $(0, 2, 0)$ por parte de P_1 .



Métodos para el tratamiento del interbloqueo

EVITACIÓN

RESUMEN

- ✓ No requiere la adquisición de todos los recursos de una vez.
- ✓ No aparece el problema de la infrautilización de recursos debido a la adquisición prematura
- ✓ Requiere que se haga la pre-reclamación de las necesidades máximas de recursos.

Si la pre-reclamación de recursos se realiza de forma conservadora, se limita el **grado de concurrencia** al hacer que el asignador detecte como inseguros un mayor número de estados del sistema. Debido a que los estados inseguros constituyen un conjunto grande de estados dentro del cual los estados de interbloqueo son sólo un subconjunto, se puede **posponer** innecesariamente la asignación de recursos disponibles a procesos que lo solicitan.



Métodos para el tratamiento del interbloqueo

EVITACIÓN

RESUMEN

- ✓ No necesita la expropiación de recursos.
- ✓ Impone costes de almac. y computación para detectar los estados seguros.
- ✓ El algoritmo del banquero, aunque en teoría funciona perfectamente, en la práctica no es útil

- Los procesos rara vez conocen de antemano sus necesidades máximas de recursos.
- El número de procesos no es fijo, como precisa el algoritmo (por ejemplo, cuando se conectan y desconectan los usuarios).



Métodos para el tratamiento del interbloqueo

DETECCIÓN

IDEA BÁSICA

- 1 Conceder libremente los recursos disponibles a los procesos solicitantes
- 2 Examinar el sistema para determinar si se ha producido un interbloqueo
- 3 Intentar romperlo



Métodos para el tratamiento del interbloqueo

DETECCIÓN

Método práctico:

1 Intentar reducir el grafo de asignación de recursos suprimiendo todas las retenciones y peticiones de todo proceso cuyas solicitudes puedan ser concedidas

2 ¿El grafo queda completamente reducido?

no



el sistema está interbloqueado y los procesos que quedan son los afectados

si

el sistema **no** está interbloqueado

La capacidad que tiene el algoritmo para identificar los procesos interbloqueados es también importante, pues facilita la recuperación posterior.



Métodos para el tratamiento del interbloqueo

DETECCIÓN

ALGORITMO DE DETECCIÓN

Es invocado para determinar si se ha producido interbloqueo.

- Sean *Trabajo* y *Fin* dos vectores de trabajo de longitud *m* y *n*, respectivamente. Toman inicialmente los siguientes valores:
 $Trabajo := Disponible$
 $Fin[i] := falso$, para todo $i=1, 2, \dots, n$ si $Asignación_i \neq 0$
 $Fin[i] = verdadero$, en caso contrario.
- Encontrar un valor de índice, *i*, tal que se cumplan las dos proposiciones:
 - $Fin[i] = falso$
 - $Solicitud_i \leq Trabajo$
 Si no existe tal *i*, continuar con el paso 4.
- Hacer
 $Trabajo := Trabajo + Asignación_i$
 $Fin[i] := verdadero$
 Continuar en el paso 2.
- Si $Fin[i] = verdadero \forall i$, entonces el sistema no está en estado interbloqueado. Si no, el estado es de interbloqueo. Además, si $Fin[i] = falso$, entonces el proceso P_i está implicado en el interbloqueo detectado.



Métodos para el tratamiento del interbloqueo

DETECCIÓN

Problema

Sean cinco procesos, P_1 a P_5 , y tres tipos de recursos, **A**, **B** y **C** con **7**, **2** y **6** unidades respectivamente. En el instante considerado, tenemos el siguiente estado de asignación de recursos:

	Asignación			Solicitud			Disponible		
	A	B	C	A	B	C	A	B	C
P1	0	1	0	0	0	0	0	0	0
P2	2	0	0	2	0	2			
P3	3	0	3	0	0	0			
P4	2	1	1	1	0	0			
P5	0	0	2	0	0	2			

- 1 Determinar si este estado es o no de interbloqueo.
- 2 Si el proceso P_3 solicita una unidad adicional de **C**, determinar si el estado resultante es de interbloqueo.



Métodos para el tratamiento del interbloqueo

DETECCIÓN

Un aspecto importante es la **frecuencia** con la que se debe ejecutar el algoritmo de detección de interbloqueos, que suele ser un parámetro del sistema.

- Una posibilidad extrema es comprobar el estado cada vez que se solicita un recurso y éste no puede ser asignado.

Inconveniente: Tiempo de CPU que se utiliza

- Otra alternativa es activar el algoritmo ocasionalmente, a **intervalos regulares** o cuando uno o más procesos queden bloqueados durante un **tiempo sospechosamente largo**.



Métodos para el tratamiento del interbloqueo

DETECCIÓN

	VENTAJAS	INCONVENIENTES
F ↑	<ul style="list-style-type: none"> • Descubrir con prontitud los interbloques • Liberación rápida de recursos 	<ul style="list-style-type: none"> • Consumir mucho tº de CPU
F ↓	<ul style="list-style-type: none"> • Consumir poco tº de CPU 	<ul style="list-style-type: none"> • Peligro de dejar el proceso mucho tº interbloqueado • Empobrecimiento en la utilización de recursos



Métodos para el tratamiento del interbloqueo

RECUPERACIÓN

Cuando un algoritmo de detección determina que existe un interbloqueo, existen varias **alternativas** para tratar de eliminarlo:

- 1 Informar al operador del sistema → Resolución manual
 - **Abortar** uno o más **procesos** para romper la espera circular
- 2 El sistema rompe el interbloqueo y se recupera automáticamente
 - **Expropiar** algunos **recursos** de uno o más de los procesos implicados



Métodos para el tratamiento del interbloqueo

RECUPERACIÓN

TERMINACIÓN DE PROCESOS

Abortar todos los procesos interbloqueados

✓ Simple

✓ Coste muy elevado



Trabajo realizado por los procesos

Abortar procesos de uno en uno hasta eliminar el ciclo

✓ Después de abortar cada proceso, hay que invocar al algoritmo de detección para saber si aún se mantiene el interbloqueo



Mucho tiempo de proceso adicional



Métodos para el tratamiento del interbloqueo

RECUPERACIÓN

TERMINACIÓN DE PROCESOS

- En cualquiera de los casos, puede que no sea fácil terminar un proceso (p.e. si se encuentra actualizando un fichero).
- Cuando se utiliza el método *incremental*, se presenta un nuevo problema de política o decisión: **selección de la víctima**
- Desde un punto de vista económico, se debe seleccionar aquel proceso cuya terminación represente el **coste mínimo**.
- Existen muchos factores que pueden determinar el proceso a eliminar, incluyendo:
 - La prioridad del proceso.
 - Tiempo de ejecución ya consumido y tiempo de ejecución previsto.
 - Cuántos recursos y de qué tipo está usando el proceso.
 - Cuántos recursos más necesita el proceso para concluir.
 - Cuántos procesos es preciso terminar.
 - Tipo del proceso (por lotes o interactivo).



Métodos para el tratamiento del interbloqueo

RECUPERACIÓN

EXPROPIACIÓN DE RECURSOS

Quitar sucesivamente los recursos de los procesos que los tienen y asignarlos a otros que los solicitan hasta conseguir romper el interbloqueo.

Hay que considerar tres aspectos:

Selección de la víctima

Minimizar el costo

Se debe hacer retroceder al proceso hasta llegar a un estado anterior conocido.

Retroceso (*rollback*)

Asegurar que un mismo proceso sólo es elegido un número finito de veces.



Incluir el número de retrocesos como factor de costo.

Postergación indefinida

Si siempre se elige como víctima al mismo proceso espera activa o *livelock*



Métodos para el tratamiento del interbloqueo

RECUPERACIÓN

RESUMEN

- La detección y recuperación de interbloques proporciona un mayor grado potencial de concurrencia que las técnicas de prevención o de evitación.
- La sobrecarga en tiempo de ejecución de la detección puede ser un parámetro ajustable en el sistema.
- La recuperación de interbloques puede ser atractiva en sistemas con una baja probabilidad de interbloques.
- En sistemas con elevada carga, la concesión sin restricciones de peticiones de recursos puede conducir a frecuentes interbloques



Métodos para el tratamiento del interbloqueo

ESTRATEGIA COMBINADA PARA EL MANEJO DE INTERBLOQUEOS

- Ninguno de los métodos presentados es adecuado para ser utilizado como estrategia exclusiva de manejo de interbloques en un sistema complejo.
- Dividir los recursos del sistema en una colección de clases disjuntas y aplicando el método más adecuado de manejo de interbloqueo a los recursos de cada clase particular.
- La ordenación de recursos puede servir para prevenir interbloques entre clases.



Métodos para el tratamiento del interbloqueo

Considérese un sistema que posee las siguientes clases de recursos:

- 1 **Recursos internos:** recursos del núcleo del sistema (tabla de procesos)
- 2 **Memoria principal:** memoria usada por los procesos de usuario.
- 3 **Dispositivos y ficheros:** dispositivos asignables, tales como impresoras o unidades con soporte extraíble (cintas, discos, floppies), y ficheros.
- 4 **Espacio de intercambio:** espacio para cada proceso de usuario en almacenamiento secundario.

- 1 Utilizar la prevención de interbloqueo entre las cuatro clases de recursos mediante la ordenación lineal de las peticiones en el orden presentado.
- 2 Dentro de cada clase se selecciona como estrategia local la más adecuada para las características específicas de los recursos que la componen.



Métodos para el tratamiento del interbloqueo

Pueden aplicarse las siguientes estrategias:

1 Recursos internos

Frecuentes peticiones y liberaciones de recursos y frecuentes cambios de estado

La **prevención** mediante ordenación de recursos es probablemente la mejor alternativa

2 Memoria principal

La **prevención** mediante expropiación es una elección razonable.

3 Dispositivos y ficheros

La **evitación** se puede llevar a cabo si los procesos son capaces de pre-reclamar las necesidades de recursos.

La **prevención** también es posible mediante la ordenación de recursos

4 Espacio de intercambio

Una posibilidad es la adquisición anticipada de todo el espacio necesario (**prevención**)



Fuentes de información

- "Operating System Concepts". A. Silberschatz, P. B. Galvin. Addison-Wesley. 1994.
- "Sistemas Operativos Modernos". A.S. Tanenbaum. Prentice-Hall. 1992.
- "Sistemas Operativos, Segunda Edición". Deitel, H. M., Addison-Wesley, 1992.
- "Sistemas Operativos, Segunda Edición". W. Stallings. Prentice Hall, 1992.

