Unidad 1:
Gestión de Procesos

Tema 2, Concurrencia:
Interbloqueo e inanición.

2.1 Caracterización del interbloqueo y grafo de asignación de recursos.

2.2 Estrategias de tratamiento del interbloqueo:

- Prevención y Predicción (Algoritmo del banquero).
- Detección y Recuperación del interbloqueo.

2.3 Problema de la cena de los filósofos.

2.1 Caracterización del interbloqueo.

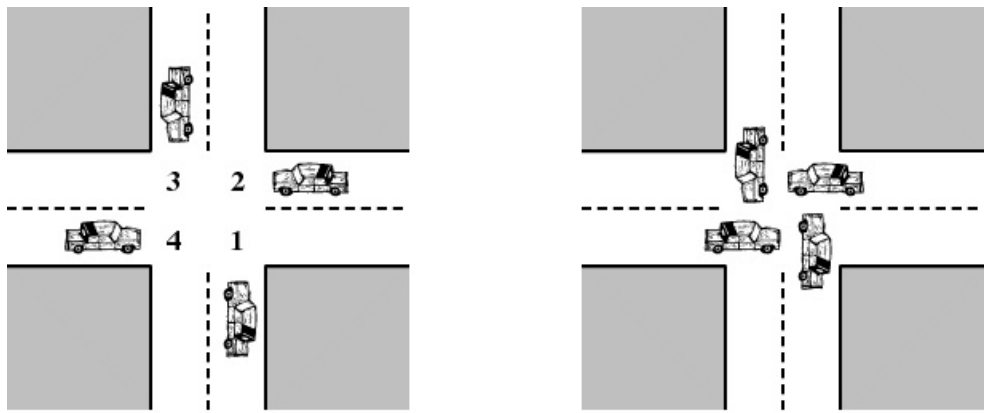
- Interbloqueo:
  - **Problema que afecta a procesos concurrentes** que utilizan recursos en un sistema.
  - Esquema de **interacción procesos-recursos**:
    - Un proceso, a través de una llamada al sistema, solicita recursos.
    - Usa esos recursos, en caso de que estén disponibles.
    - A través de una llamada al sistema, el proceso libera los recursos cuando ya no los necesita.
  - **Ejemplo:** Considera dos procesos y dos recursos. Supón que cada proceso necesita acceder a ambos recursos para llevar a cabo una parte de su función. Puede suceder que el sistema operativo asigne R1 a P1 y R2 a P2. **Cada proceso está esperando uno de los dos recursos. Ninguno liberará el recurso que posee hasta que adquiera el otro y realice su tarea.**

2.1 Caracterización del interbloqueo.

- Modelo del sistema:
  - Un sistema consta de un **número finito de recursos** que son distribuidos entre un número de procesos que compiten por ellos.
  - Puede haber **varios ejemplares de un mismo recurso**, en ese 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.**
  - Una **tabla del sistema** registra cuando un recurso está libre o asignado y en ese caso, a qué proceso se le asignó.

2.1 Caracterización del interbloqueo.

- El problema del interbloqueo:
  - Situación de 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).



(a) Deadlock possible (b) Deadlock

2.1 Caracterización del interbloqueo.

- El problema del interbloqueo:
  - La **gestión del interbloqueo no es responsabilidad de las aplicaciones**, sino del sistema de gestión de recursos.
  - **Ejemplo:** Los procesos A y B se pueden interbloquear, aunque estén escritos correctamente.

Proceso A	Proceso B
- Solicita (escáner)	- Solicita (impresora)
- Solicita (impresora)	- Solicita (escáner)
- Usa impresora y escáner	- Usa impresora y escáner
- Libera (impresora)	- Libera (escáner)
- Libera (escáner)	- Libera (impresora)

2.1 Caracterización del interbloqueo.

- Tipos de recursos:
  - **Recursos reutilizables:**
    - Pueden ser usados por un proceso y no se agotan con el uso.
    - Los procesos solicitan, usan y liberan los recursos para que posteriormente otros procesos los reutilicen.
    - Ejemplos: Procesadores, impresoras, dispositivos de E/S, memorias principal y secundaria, archivos, DDBB y semáforos.
  - **Recursos consumibles:**
    - Pueden ser producidos (creados) y consumidos (destruidos) por un proceso.
    - Cuando un proceso adquiere un recurso éste deja de existir.
    - Ejemplos: Interrupciones, señales, mensajes e información en buffers de E/S.

2.1 Caracterización del interbloqueo.

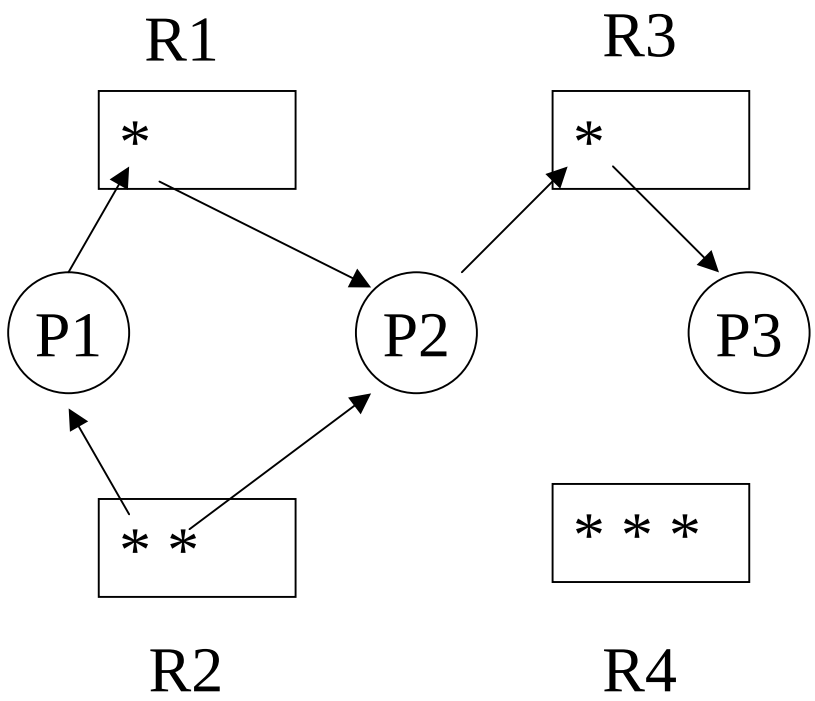
- Grafo de asignación de recursos:
  - Sirve para describir el interbloqueo, consta de un conjunto de vértices y un conjunto de flechas.
  - **Vértices:** procesos y recursos.
  - **Flechas:**
    - De asignación ($R_j \rightarrow P_i$), indican que el recurso j se asignó al proceso i.
    - De solicitud ($P_i \rightarrow R_j$), indican que el proceso i solicita el recurso j.
  - **Instancias:** Número de ejemplares que hay de cada recurso.

2.1 Caracterización del interbloqueo.

- Grafo de asignación de recursos:
  - **Ejemplo:**
    - P = {P1, P2, P3}
    - R = {R1, R2, R3, R4}
    - F = {P1R1, P2R3, R1P2, R2P2, R2P1, R3P3}
    - Instancias:
      - 1 de R1
      - 2 de R2
      - 1 de R3
      - 3 de R4.

2.1 Caracterización del interbloqueo.

- Grafo de asignación de recursos:

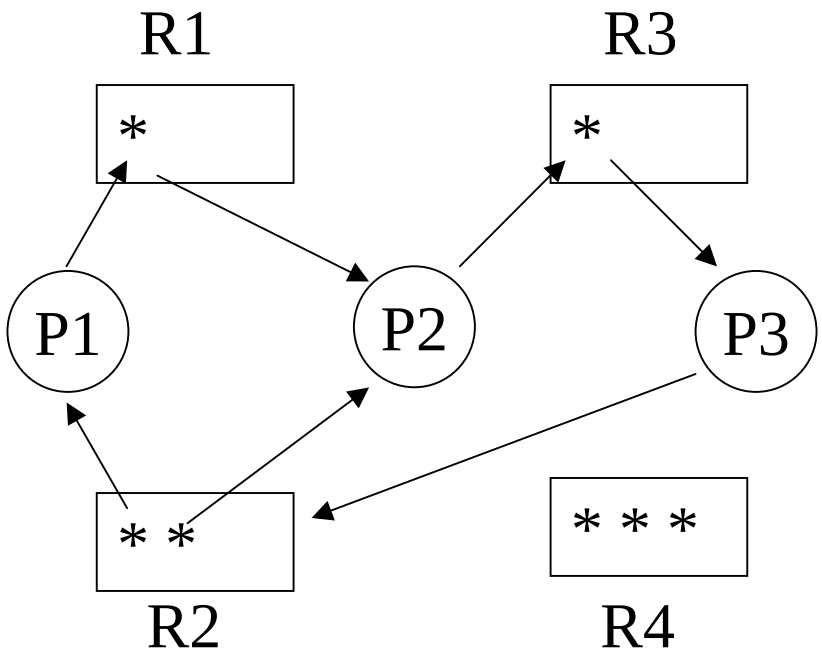


2.1 Caracterización del interbloqueo.

- Grafo de asignación de recursos:
  - **Estado de los procesos:**
    - P1 posee una instancia del recurso R2 y espera por una de R1.
    - P2 tiene asignada una instancia de R1, una instancia de R2 y espera por una de R3.
    - P3 tiene asignada una instancia de R3.
  - **Si en el grafo no existen ciclos, ningún proceso del sistema está en bloqueo mutuo.**
  - En el ejemplo anterior no hay interbloqueo.

2.1 Caracterización del interbloqueo.

- Grafo de asignación de recursos:
  - P1, P2 y P3 están en interbloqueo.



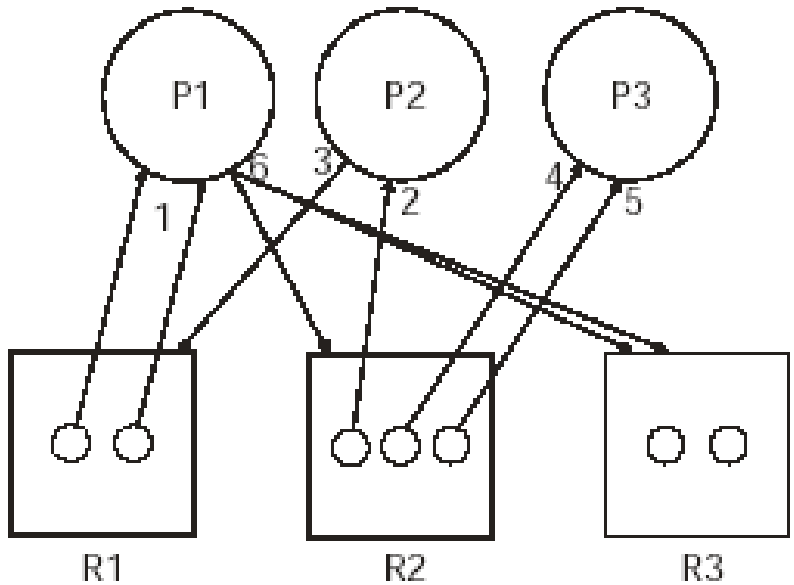
2.1 Caracterización del interbloqueo.

- Ejemplo 1 de grafo de asignación de recursos:

3 proc y 3 rec

$R_1(2), R_2(3), R_3(2)$

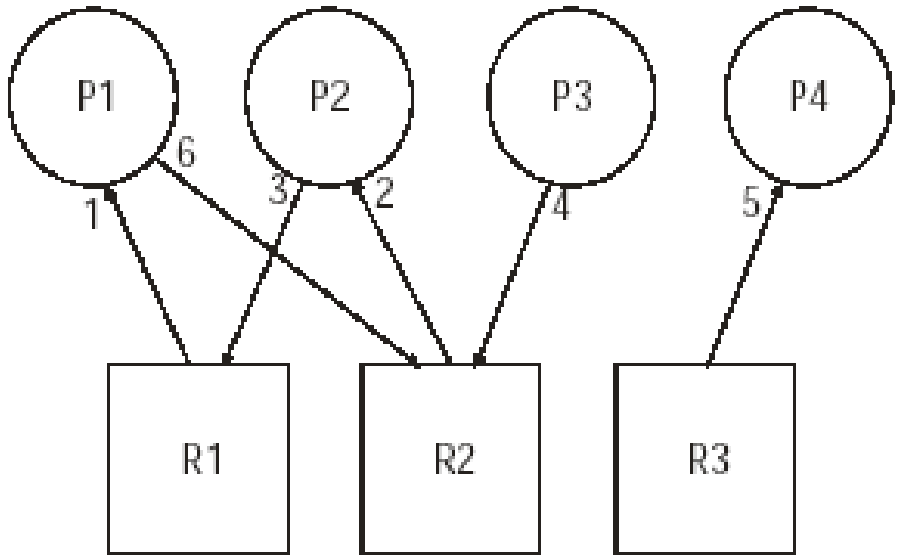
1. P_1: solicita($R_1[2]$) → solicita 2
2. P_2: solicita($R_2[1]$)
3. P_2: solicita($R_1[1]$) → bloq
4. P_3: solicita($R_2[1]$)
5. P_3: solicita($R_2[1]$)
6. P_1: solicita($R_2[1]$,$R_3[2]$) → bloq



2.1 Caracterización del interbloqueo.

- Ejemplo 2 de grafo de asignación de recursos:

1. P_1: solicita(R_1)
2. P_2: solicita(R_2)
3. P_2: solicita(R_1) → bl.
4. P_3: solicita(R_2) → bl.
5. P_4: solicita(R_3)
6. P_1: solicita(R_2)..



2.1 Caracterización del interbloqueo.

- Condiciones de Interbloqueo:
  - El interbloqueo se da si se cumplen las cuatro condiciones:
    - 1. **Exclusión mutua:** Sólo un proceso puede usar un recurso cada vez. Si otro proceso solicita ese recurso deberá esperar a que se libere el recurso.
    - 2. **Retención y espera:** Un proceso retiene unos recursos mientras espera que se le asignen otros.
    - 3. **No expropiación:** Un proceso no puede ser forzado a abandonar un recurso que retiene.
    - 4. **Espera circular:** Existe una cadena cerrada de procesos {P0, P1, ..., Pn}, cada uno de los cuales retiene, al menos, un recurso que necesita el siguiente proceso de la cadena (P0 espera por P1, P1 por P2, P2 por P3, ..., Pn por P0).

2.2 Estrategias de tratamiento del interbloqueo.

- Tratamiento del Interbloqueo:
  - **Garantizando que no ocurra nunca en el sistema:**
    - **Prevención:** Asegura que no ocurre fijando reglas.
      - Infrautiliza recursos: se deben pedir antes de necesitarlos.
    - **Predicción:** Asegura que no ocurre basándose en conocimiento de necesidades futuras de los procesos.
      - Dificultad de conocimiento de evolución futura.
      - Coste de algoritmo + infrautilización de recursos.
  - **Detección y recuperación:** Se detecta y se recupera.
    - Coste de algoritmo + pérdida de trabajo realizado.
  - **Ignorar el problema:** Utilizado por la mayoría de los SO.
    - Dada la baja probabilidad de que ocurra y el coste que conlleva evitarlo (infrautilización y/o coste de algoritmos).

2.2 Estrategias de tratamiento del interbloqueo.

- Prevención del Interbloqueo (1):
  - Tratando de eliminar la aparición de alguna de las cuatro condiciones necesarias para el interbloqueo.
    - **Exclusión mutua:** Depende de la naturaleza del recurso, así que no se puede eliminar.

2.2 Estrategias de tratamiento del interbloqueo.

- 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 pide todos los 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 casos puede darse inanición (un proceso no se ejecuta nunca).

2.2 Estrategias de tratamiento del interbloqueo.

- Prevención del Interbloqueo(3):
  - **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.

2.2 Estrategias de tratamiento del interbloqueo.

- Prevención del Interbloqueo(4):
  - **Espera circular:** Se puede evitar forzando un orden en la petición de recursos.
    - Cada recurso tiene asignado un número de orden.
    - Los recursos se deben pedir en orden ascendente.
    - Aconsejable: que el orden de petición de los recursos se establezca según el orden de uso normal de los recursos de un sistema.
    - Efecto negativo:
      - Se limita la libertad de escritura de código.
      - Se puede inducir a una mala utilización de los recursos.

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo:
  - Se trata de conceder los recursos sólo cuando no representen un riesgo potencial de interbloqueo.
  - Se necesita conocer las peticiones futuras de recursos: Los procesos han de declarar por anticipado la cantidad máxima de recursos que van a utilizar a lo largo de su vida.
  - Los procesos son independientes, no hay condiciones de sincronización.
  - Dos enfoques para la predicción del interbloqueo:
    - No iniciar un proceso si sus demandas pueden llevar al interbloqueo.
    - No conceder una solicitud de incrementar los recursos de un proceso si esta asignación puede llevar al interbloqueo.

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo:
  - **Concepto de estado seguro:**
    - El estado de un sistema viene dado por la asignación actual de recursos a los procesos.
    - **Estado seguro: estado en el que existe, al menos, una secuencia de asignación de recursos a los procesos que no lleva al interbloqueo.**
    - **Estado inseguro: estado que no es seguro.**
    - Sólo concedemos recursos si el estado resultante tras la petición es seguro.

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Basado en la **negativa de asignación de recursos**.
  - {P1, P2, ..., Pn} es una secuencia segura si los recursos que Pi puede pedir, en el peor caso se pueden atender con lo que hay disponibles más los recursos poseídos por todos los procesos $P_{j,j<i}$.
  - **Funcionamiento:**
    - El peor caso: que todos los procesos soliciten a la vez el máximo de recursos a los que tienen derecho.
    - El primer proceso de la secuencia podría finalizar con los recursos disponibles en el sistema.
    - El segundo podría finalizar con los disponibles más los que libera el proceso 1 y así sucesivamente.
    - **Si todos los procesos finalizan ⇒ no hay interbloqueo.**

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Determinación de un estado seguro: estado inicial

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Matriz demanda

	R1	R2	R3
P1	1	0	0
P2	6	1	2
P3	2	1	1
P4	0	0	2

Matriz asignación

R1	R2	R3
9	3	6

Vector recursos

R1	R2	R3
0	1	1

Vector disponible

(a) Estado inicial

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Determinación de un estado seguro: P2 terminado.

	R1	R2	R3
P1	3	2	2
P2	0	0	0
P3	3	1	4
P4	4	2	2

Matriz demanda

	R1	R2	R3
P1	1	0	0
P2	0	0	0
P3	2	1	1
P4	0	0	2

Matriz asignación

R1	R2	R3
6	2	3

Vector disponible

(b) P2 terminado

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Determinación de un estado seguro: P1 terminado.

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	3	1	4
P4	4	2	2

Matriz demanda

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	2	1	1
P4	0	0	2

Matriz asignación

R1	R2	R3
7	2	3

Vector disponible

(c) P1 terminado

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Determinación de un estado seguro: P3 terminado.

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	0	0	0
P4	4	2	2

Matriz demanda

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	0	0	0
P4	0	0	2

Matriz asignación

R1	R2	R3
9	3	4

Vector disponible

(d) P3 terminado

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Determinación de un estado inseguro: estado inicial.

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Matriz demanda

	R1	R2	R3
P1	1	0	0
P2	6	1	2
P3	2	1	1
P4	0	0	2

Matriz asignación

R1	R2	R3
9	3	6

Vector recursos

R1	R2	R3
0	1	1

Vector disponible

(a) Estado inicial

2.2 Estrategias de tratamiento del interbloqueo.

- Predicción del Interbloqueo: Algoritmo del banquero.
  - Determinación de un estado inseguro: estado interbloqueado.

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	3	1	4
P4	4	2	2

Matriz demanda

	R1	R2	R3
P1	0	0	0
P2	0	0	0
P3	2	1	1
P4	0	0	2

Matriz asignación

R1	R2	R3
7	2	3

Vector disponible

(b) P1 solicita una unidad de R1 y otra de R3

2.2 Estrategias de tratamiento del interbloqueo.

- Detección del Interbloqueo:
  - La detección del interbloqueo **no limita el acceso a los recursos** ni restringen las acciones de los procesos.
  - Ahora **se conceden los recursos que los procesos necesiten siempre que sea posible.**
  - El interbloqueo se puede detectar comprobando si existe una secuencia de terminación de procesos:
    - Sean C los procesos del sistema y R los recursos disponibles.
      - 1. Buscar en C un proceso que puede continuar con los R.
      - 2. Si no se encuentra ningún proceso ir al paso 5.
      - 3. Suponer que un proceso termina (lo retiramos de C) y que libera los recursos que retiene (los añadimos a R).
      - 4. Volver al paso 1.
      - 5. Si C está vacío, hay interbloqueo.

2.2 Estrategias de tratamiento del interbloqueo.

- Recuperación del Interbloqueo:
  - Un sistema que pretenda recuperarse del interbloqueo, debe **invocar a un algoritmo de detección** cuando lo considere oportuno (de manera más o menos periódica).
  - **Formas de intentar la recuperación:**
    - Terminación de procesos.
    - Expropiación de recursos.

2.2 Estrategias de tratamiento del interbloqueo.

- Recuperación del Interbloqueo:
  - **Terminación de procesos:**
    - **Abortando a todos los procesos implicados** (muy drástico).
      - Efecto negativo: se pierde el trabajo previamente realizado.
    - **Abortando a uno de los procesos** ¿cuál?
      - El que más recursos libere.
      - El que menos tiempo lleve en ejecución.
    - **Retrocediendo la ejecución de algún proceso**.
      - Efecto negativo: muy complicado de implementar y necesita que el programa esté diseñado para que pueda retroceder.

2.2 Estrategias de tratamiento del interbloqueo.

- Recuperación del Interbloqueo:
  - **Expropiación de recursos:**
    - **Selección de la víctima:**
      - ¿Qué recursos y de qué procesos se expropian?
    - **Retroceso:**
      - Si expropiamos a un recurso de un proceso ¿qué hacemos con ese proceso?.
    - Orden de los criterios de selección para expropiación de recursos:
      - 1. Menor tiempo ya consumido en el procesador.
      - 2. Menor número de salidas.
      - 3. Menor tiempo restante estimado.
      - 4. Menor número total de recursos ya asignados.
      - 5. Prioridad más baja.

2.2 Estrategias de tratamiento del interbloqueo.

- Inanición:
  - Para los dos casos de intento de recuperación del interbloqueo (terminación de procesos o expropiación de recursos) hay que tener cuidado de no provocar la inanición de procesos.
  - **¿Qué es la inanición?**
    - Supón tres procesos que acceden periódicamente a un recurso. Considera que P1 posee el recurso y que P2 y P3 están esperando. Cuando P1 haya ejecutado su sección crítica tanto P2 como P3 podrán solicitar el recurso. Supón que le le concede el acceso a P3 y que, antes de que termine su sección crítica, P1 solicita acceso de nuevo y así sucesivamente, **se puede llegar a una situación en la que P2 nunca accede al recurso.**

2.3 Problema de la Cena de los Filósofos.

- Problema de coordinación de procesos concurrentes.
  - **Enunciado:**
    - Cinco filósofos viven juntos y su única actividad es pensar y comer. Tras años de mucho pensar deciden que la única comida que les ayuda a pensar son los espaguetis. Se sientan en torno a una mesa redonda en la cual hay una fuente con espaguetis, cinco platos y cinco tenedores, uno para cada filósofo. Un filósofo que quiera comer se sentaría en su lugar y usando los dos tenedores de cada lado del plato, cogería los espaguetis y se los comería. El problema viene cuando todos quieren comer a la vez puesto que no disponen de tenedores suficientes.