Universidad Nacional de Lanús
Departamento de Desarrollo Productivo y Tecnológico
Licenciatura en Sistemas

Unidad Nº 2:

Manejo de Interbloqueo en Sistemas Operativos

Sistemas Operativos

SINCRONIZACIÓN DE PROCESOS

- Busca que los Procesos no se interfieran entre sí al ejecutarse en forma concurrente.

- Controla *el acceso de los Recursos Compartidos* de los Procesos

Región Crítica

buscando garantizar la ***Exclusión Mutua***.

- Semáforos
- Monitores

SINCRONIZACIÓN DE PROCESOS

- Implementación de Exclusión Mutua mediante **Semáforos**:

  - Ventajas:
    - ✓ No presenta espera activa.
    - ✓ Es confiable.

  - Problemas:
    - ❖ Si se usan múltiples procesos y semáforos, puede ser difícil de implementar y verificar para evitar Deadlock y Starvation.

SINCRONIZACIÓN DE PROCESOS

- Implementación de Exclusión Mutua mediante **Semáforos**:

- Ejemplo : Semáforo A = 1;
  B = 1;

P1	P2	P3
~~Down(A)~~	~~Down(B)~~	~~Down(B)~~
~~Down(B)~~	~~Down(A)~~	~~Down(A)~~
~~Up(A)~~	~~Up(A)~~	~~Up(B)~~
~~Up(B)~~	~~Up(B)~~	~~Up(A)~~
Termina en T5	Termina en T9	Termina en T12

(sin concurrencia)

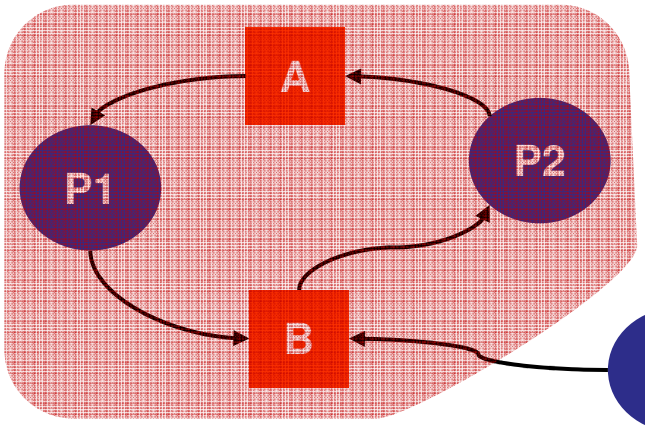
T	A	B
0	1	1
1	0	
2		0
3	1	
4		1
5		0
6	0	
7	1	
8		1

SINCRONIZACIÓN DE PROCESOS

- Implementación de Exclusión Mutua mediante **Semáforos**:

- Ejemplo : Semáforo A = 1;
  B = 1;

P1	P2	P3
~~Down(A)~~	~~Down(B)~~	~~Down(B)~~
~~Down(B)~~	~~Down(A)~~	Down(A)
Up(A)	Up(A)	Up(B)
Up(B)	Up(B)	Up(A)



DeadLock

Starvation

(concurrencia completa)

	P2	P3, P1
T	A	B
0	1	1
1	0	~~0~~ -1
2	-1	-2
3		
4		
5		
6		
7		
8		

INTERBLOQUEO ENTRE PROCESOS

- **DeadLock**

  también denominado Interbloqueo entre Procesos,
  Bloqueo Mutuo o Abrazo Mortal

  - Es cuando un conjunto de procesos (o threads) en un sistema concurrente que compiten por recursos del sistema, están bloqueados en forma *permanente*.

  - Los procesos que participan del deadlock están en estado <Bloqueado> esperando por un evento que *no va a suceder*.

INTERBLOQUEO ENTRE PROCESOS

DeadLock

- Condiciones:

1) *Mutua Exclusión*
2) *Tomar y Esperar*
3) *Recursos No Apropiativos*

Condiciones Necesarias

4) *Espera Circular*

Condición Suficiente

INTERBLOQUEO ENTRE PROCESOS

- **DeadLock**

Belo Horizonte, Brasil

INTERBLOQUEO ENTRE PROCESOS

- **Starvation**

  también denominado Inanición

  Es cuando uno o más procesos (o threads) esperan *indefinidamente* para acceder a un recurso aunque no participan (activamente) de una *espera circular*.

INTERBLOQUEO ENTRE PROCESOS

- Mecanismos para resolver DeadLock & Starvation:
  - Estrategia del Avestruz
  - Prevenir
  - Evitar
  - Detectar & Eliminar

SINCRONIZACIÓN DE PROCESOS

- Estrategia del Avestruz:
  - No hacer nada!

SINCRONIZACIÓN DE PROCESOS

- Estrategia del Avestruz:

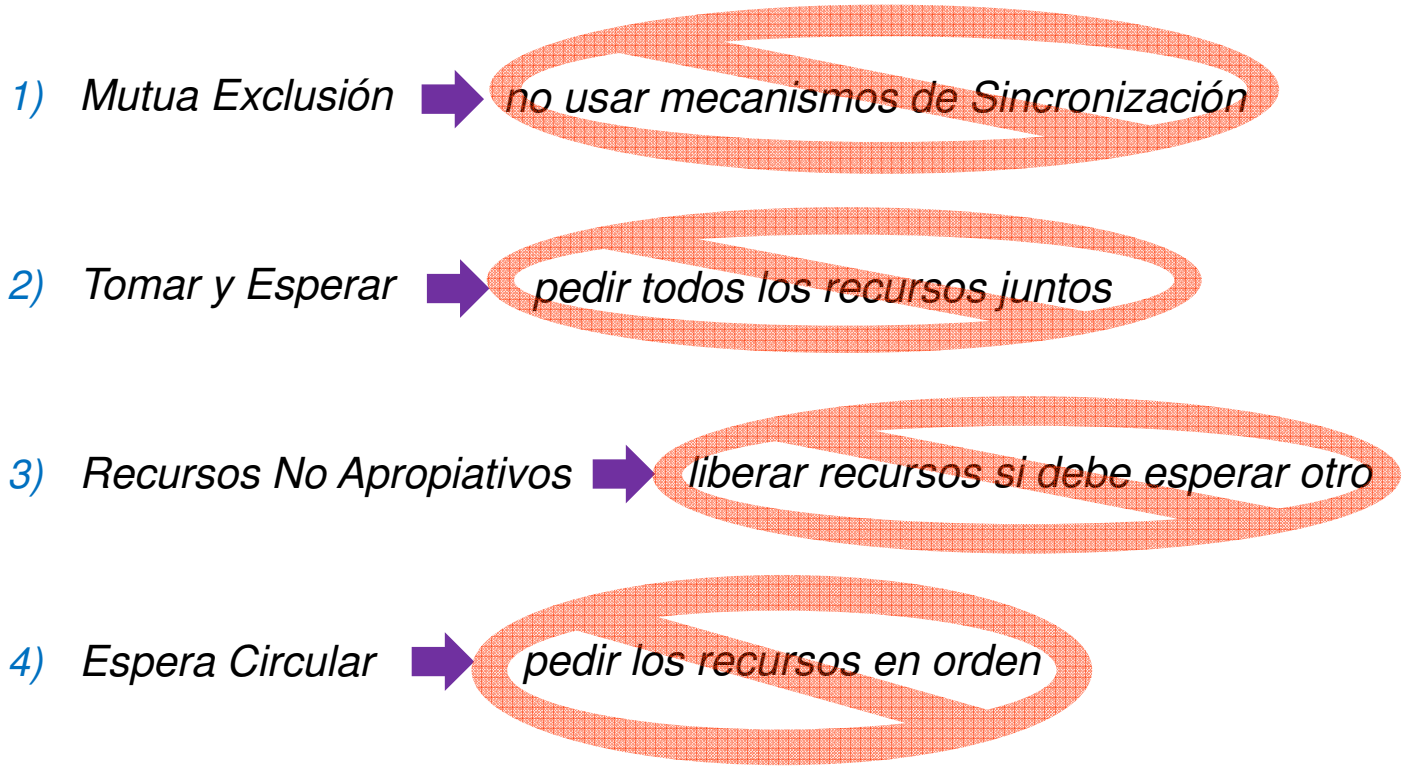
SINCRONIZACIÓN DE PROCESOS

- Estrategia del Avestruz:

SINCRONIZACIÓN DE PROCESOS

- Prevenir el DeadLock:

  Eliminar las Condiciones del DeadLock:



SINCRONIZACIÓN DE PROCESOS

- Evitar el DeadLock:

Asignar los recursos a medida de que son solicitados

siempre que no produzcan una *Espera Circular.*

➡ **Algoritmo del Banquero** [Dijkstra]

Ventajas:

- ✓ Nunca se va a producir un DeadLock.

Problemas:

- ❖ Es costoso para el sistema.
- ❖ Se necesita conocer todos los recursos que va a solicitar cada proceso por adelantado.

SINCRONIZACIÓN DE PROCESOS

- Detectar y Eliminar el DeadLock:

En forma *recurrente* se analiza si hay *Espera/s Circular/es* entre los procesos

(se simula la ejecución de los procesos para determinar si terminan o no)

- → si no hay, no hace nada
- → si hay, se elimina
  - liberar recurso/s tomado/s
  - rollback de proceso/s
  - matar proceso/s

Ventajas:

- ✓ Menos costoso para el sistema.

Problemas:

- ❖ Se puede generar DeadLocks.
- ❖ Se necesita saber los recursos que podría solicitar cada proceso por adelantado.

Bibliografía

- Guía de Estudio Nº 2: *Manejo de Interbloqueo en sistemas operativos* http://sistemas.unla.edu.ar/sistemas/sls/ls-4-sistemas-operativos/pdf/SO-GE2-Interbloqueo.pdf

- Stallings, W. (2011). *Sistemas Operativos - Aspectos Internos y Principios de Diseño*, 7ma Edición Prentice Hall. Capítulo 6.

- Tanenbaum, A.S. (2009). *Sistemas Operativos Modernos*, 3ra Edición Prentice Hall. Capítulo 6.