

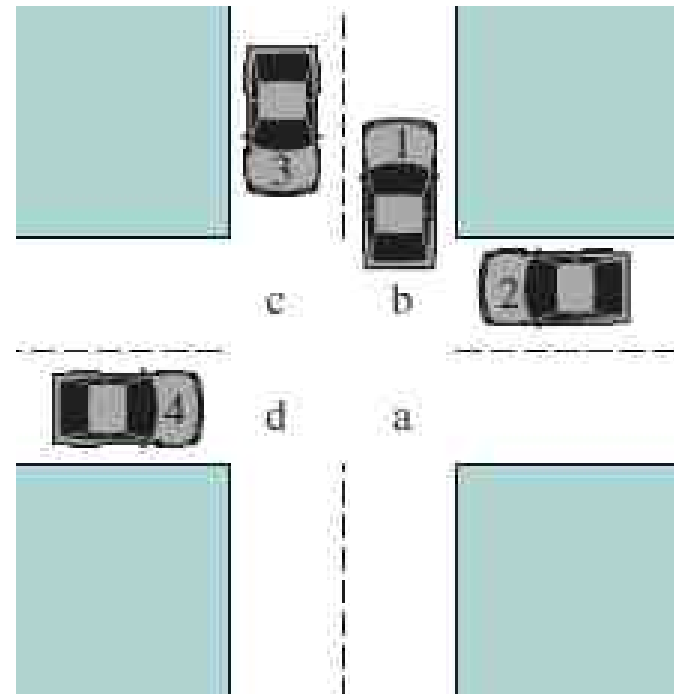
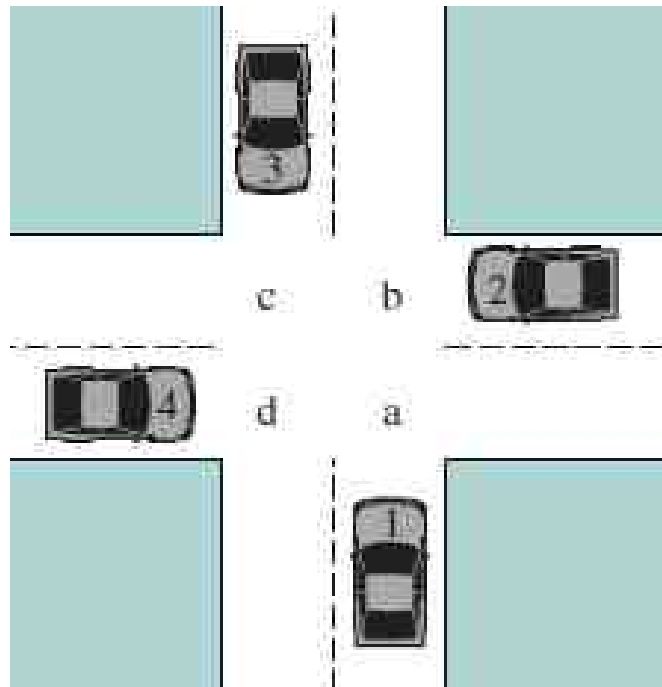
# Interbloqueo / Deadlock

# Deadlock

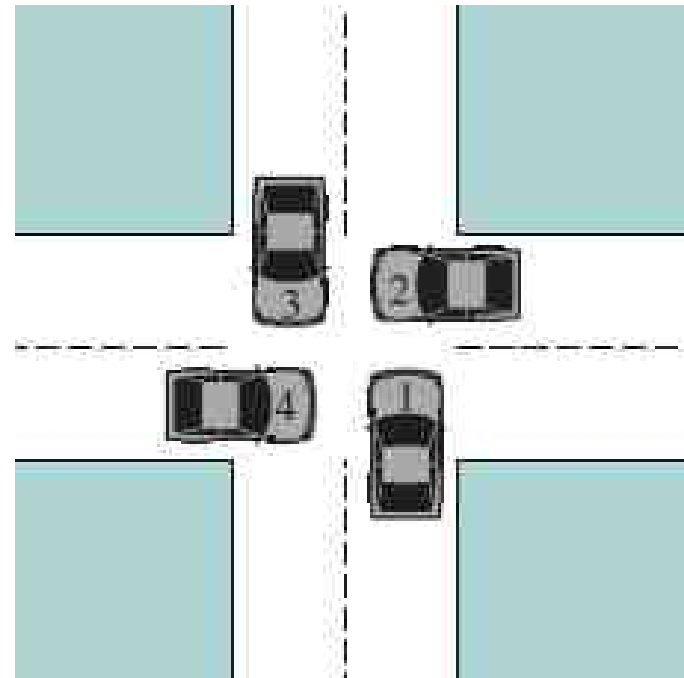
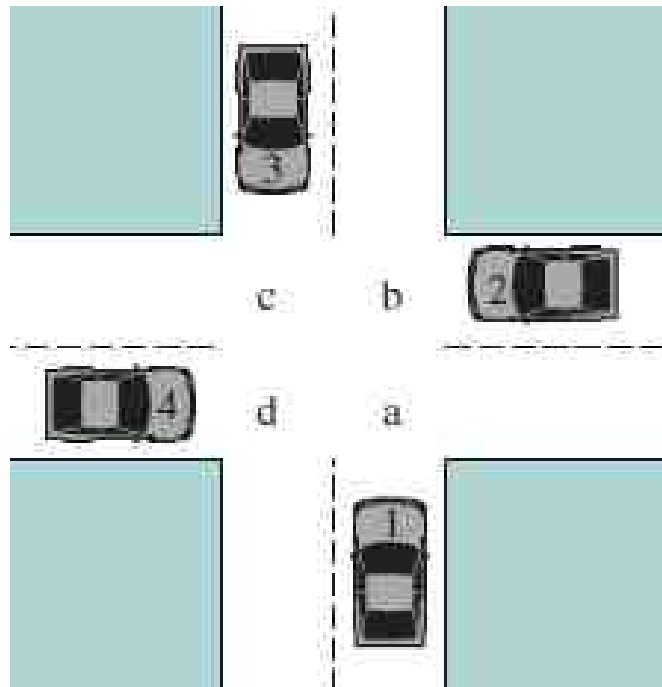
## Introducción

- Recursos Limitados (CPU, Memoria, Dispositivos, etc)
- Procesos
  - 1.Solicitan
  - 2.Usan
  - 3.Liberan
- Si un recurso no está disponible, un proceso puede bloquearse.

# Deadlock



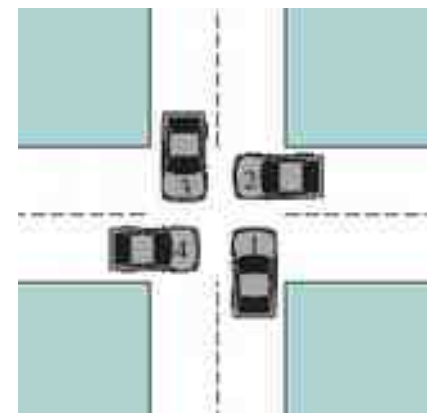
## Deadlock



# Deadlock

<u>Auto 1</u>	<u>Auto 2</u>	<u>Auto 3</u>	<u>Auto 4</u>
wait(a) ; wait(b) ;	wait(b) ; wait(c) ;	wait(c) ; wait(d) ;	wait(d) ; wait(a) ;
avanzar() ;	avanzar() ;	avanzar() ;	avanzar() ;
signal(a) ; signal(b) ;	signal(b) ; signal(c) ;	signal(c) ; signal(d) ;	signal(d) ; signal(a) ;

a = 1  
 b = 1  
 c = 1  
 d = 1



# Deadlock

Proceso 1	Proceso 2
<pre>wait (sem_a); wait (sem_b); * SECCIÓN CRÍTICA * signal (sem_a); signal (sem_b);</pre>	<pre>wait (sem_b); wait (sem_a); * SECCIÓN CRÍTICA * signal (sem_b); signal (sem_a);</pre>

`sem_a = 1`

`sem_b = 1`

# Deadlock

## Definición

**Bloqueo permanente de un conjunto procesos donde cada uno de estos procesos está esperando un evento que sólo puede generar un proceso del conjunto.**

# Deadlock

## **Tipos de recursos:**

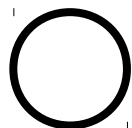
- Reutilizables
- Consumibles



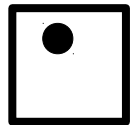
# Deadlock

## Grafo de asignación de recursos:

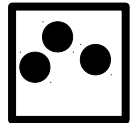
- Permite representar el estado del sistema con respecto a la asignación de los recursos a cada proceso en un momento determinado.



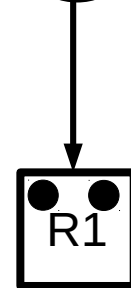
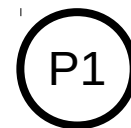
PROCESO



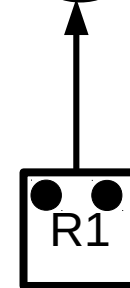
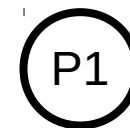
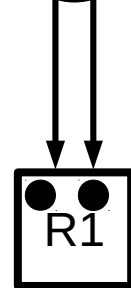
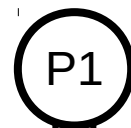
RECURSO  
CON 1 INSTANCIA



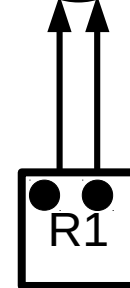
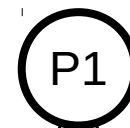
RECURSO  
CON 3 INSTANCIAS



El proceso solicita  
el recurso

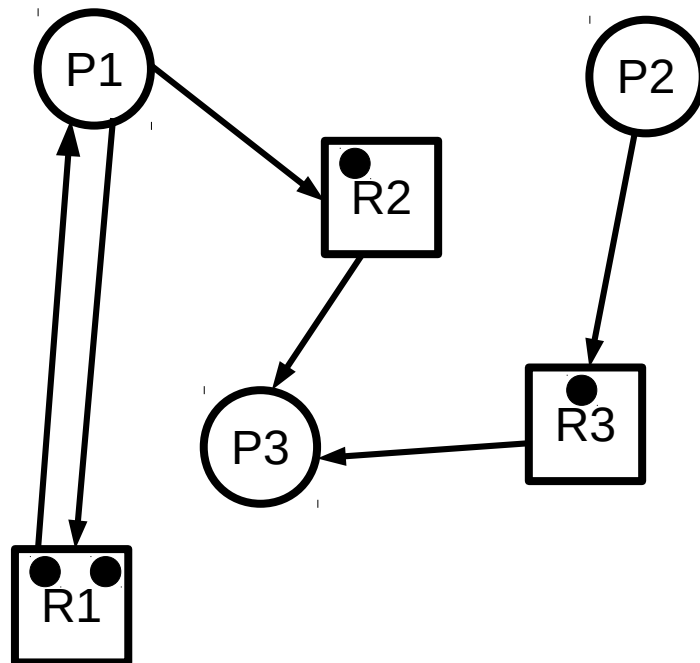


El recurso está  
asignado al proceso



# Deadlock

## Grafo de asignación de recursos:



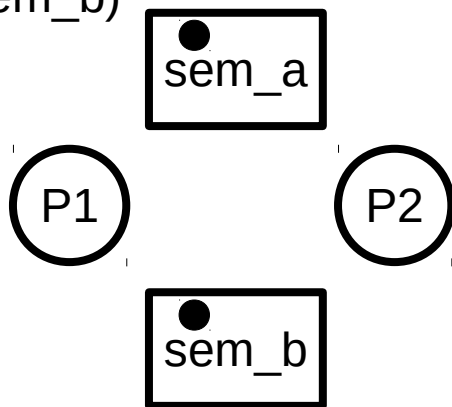
- Hay 3 procesos.
- Hay 1 recurso con dos instancias.
- Hay 2 recursos con una instancia.
- El *P1* solicita 1 de *R1* y 1 de *R2*.
- El *P1* tiene asignado 1 de *R1*.
- El *P2* solicita 1 de *R3*.
- El *P2* no tiene recursos asignados.
- El *P3* no solicita recursos.
- El *P3* tiene asignado 1 de *R2* y 1 de *R3*.

# Deadlock

## Grafo de asignación de recursos:

	Proceso 1	Proceso 2	
1	<code>wait (sem_a);</code>	<code>wait (sem_b);</code>	3
2	<code>wait (sem_b);</code>	<code>wait (sem_a);</code>	7
4	<b>* SECCIÓN CRÍTICA *</b>	<b>* SECCIÓN CRÍTICA *</b>	8
5	<code>signal (sem_a);</code>	<code>signal (sem_b);</code>	9
6	<code>signal (sem_b);</code>	<code>signal (sem_a);</code>	10

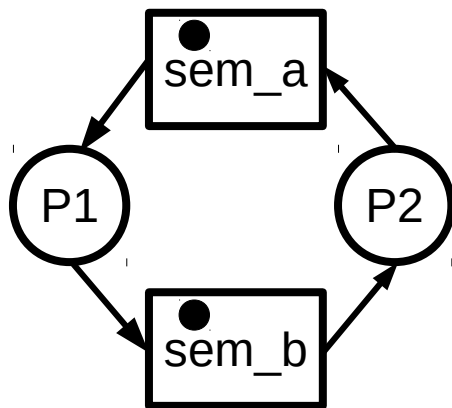
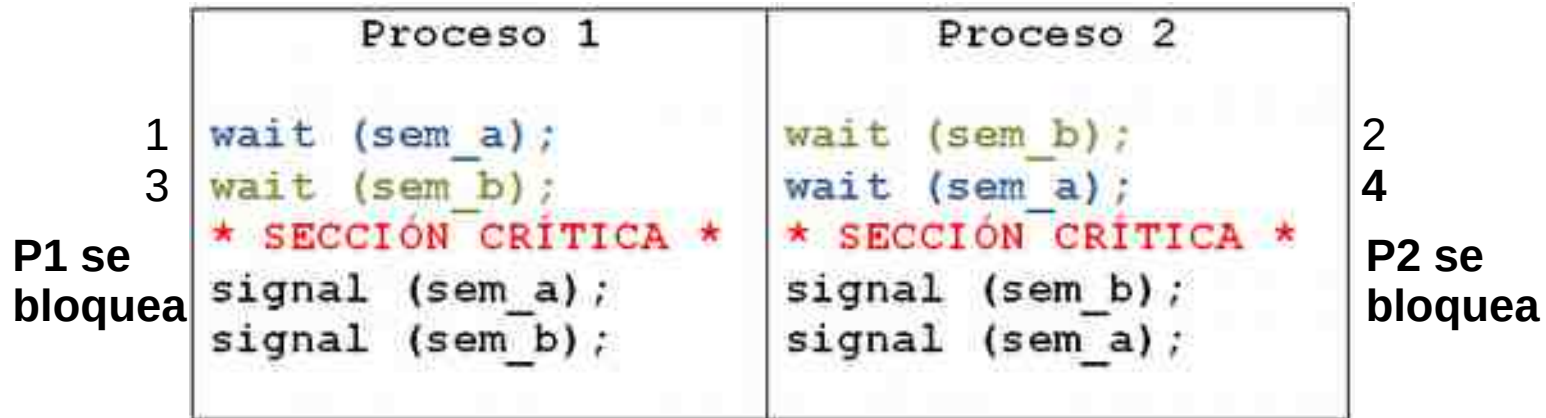
WakeUp  
(cola\_sem\_b)



sem\_a = 1 0 0 1  
sem\_b = 1 0 -1 0 1

# Deadlock

## Grafo de asignación de recursos:



sem\_a = 1 0 -1  
 sem\_b = 1 0 -1

# Deadlock

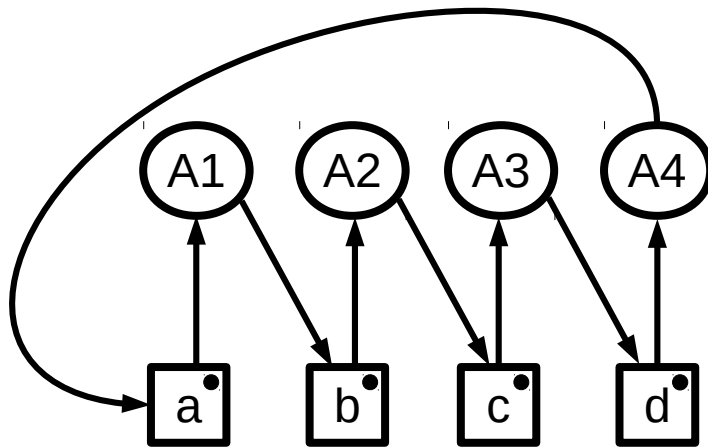
## **Grafo de asignación de recursos:**

- Ciclos:
  - Si no hay ciclos no hay Deadlock.
  - Si hay un ciclo en el grafo puede implicar la existencia de Deadlock.
  - Si hay un ciclo y además todos los recursos son de una sola instancia, entonces hay Deadlock.

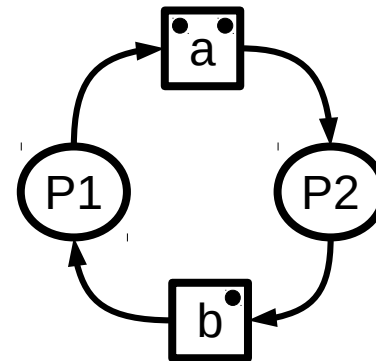
# Deadlock

## Grafo de asignación de recursos:

- Ciclos:



Ciclo CON Deadlock



Ciclo SIN Deadlock

# Deadlock

## Condiciones para la existencia de Deadlock

- Mutua Exclusión.
  - Retención y Espera.
  - Sin desalojo de recursos.
- } Condiciones Necesarias

Proceso 1	Proceso 2
<pre>wait (sem_a); wait (sem_b); * SECCIÓN CRÍTICA * signal (sem_a); signal (sem_b);</pre>	<pre>wait (sem_b); wait (sem_a); * SECCIÓN CRÍTICA * signal (sem_b); signal (sem_a);</pre>

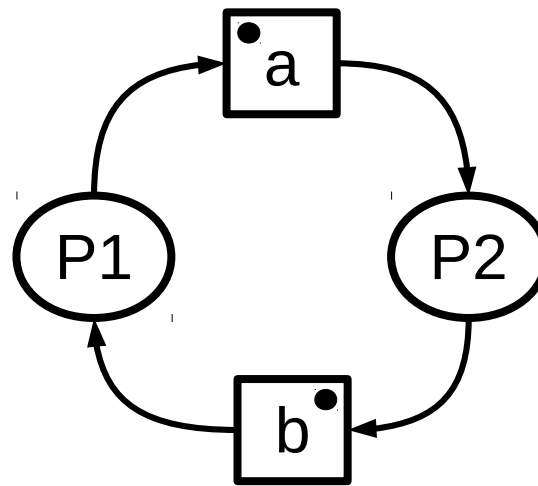
# Deadlock

## Condiciones para la existencia de Deadlock

- **Mutua Exclusión.**
- **Retención y Espera.**
- **Sin desalojo de recursos.**
- **Espera Circular.**

Condiciones  
Necesarias

Condiciones  
Necesarias y  
Suficientes





# Deadlock

## Tratamiento del Deadlock:

- Estrategias para que nunca ocurra Deadlock.
- Estrategias en la que puede ocurrir Deadlock.

# Deadlock

## Tratamiento del Deadlock:

- **1) Prevención de Deadlock:**
  - Garantiza que no ocurrirá Deadlock.
  - Impedir que se produzca alguna de las cuatro condiciones.
    - Mutua-Exclusión.
    - Retención y Espera.
    - Sin desalojo de recursos.
    - Espera Circular.

# Deadlock

## Tratamiento del Deadlock:

- **1) Prevención de Deadlock:**
  - Condición 1: Mutua-Exclusión
    - Si hay recursos que no se pueden compartir no puede evitarse.

# Deadlock

## Tratamiento del Deadlock:

- **1) Prevención de Deadlock:**
  - Condición 2: Retención y Espera
    - Solicitar todos los recursos juntos.
    - Solicitar los recursos de a uno o varios, utilizarlos y liberarlos.

# Deadlock

## Tratamiento del Deadlock:

- **1) Prevención de Deadlock:**
  - Condición 3: Sin desalojo de recursos
    - Si un proceso que tiene recursos asignados solicita uno que no está disponible, debe liberar sus recursos.
    - Si un proceso A solicita un recurso que está asignado a otro proceso B que está a la espera de más de recursos. El recurso asignado al proceso B puede asignarse al proceso A.

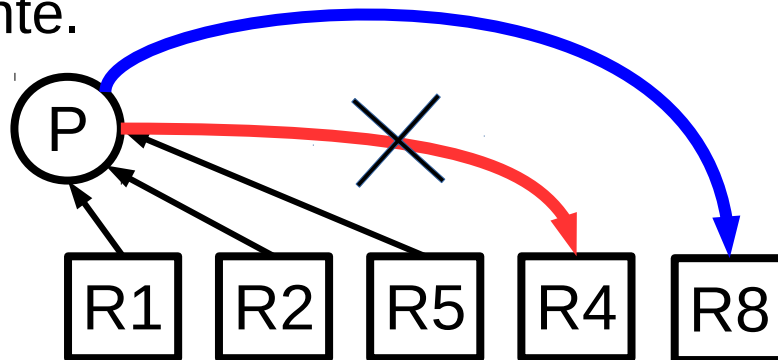
# Deadlock

## Tratamiento del Deadlock:

### ▪ 1) Prevención de Deadlock:

#### ▪ Condición 4: Espera Circular

- Asignar un número de orden a los recursos. Los recursos sólo pueden solicitarse en orden creciente.



# Deadlock

## Tratamiento del Deadlock:

- **2) Evasión o Predicción de Deadlock:**
  - Garantiza que no ocurrirá Deadlock.
  - Técnicas:
    - 1) Denegar el inicio de un proceso.
    - 2) Denegar la asignación de un recurso.
      - Algoritmo del Banquero.

# Deadlock

## Tratamiento del Deadlock:

- 2) Evasión o Predicción de Deadlock:

- 1) Denegar el inicio de un proceso.

$$M_{n+1} + \sum_{i=1}^n M_i \leq RT$$

$M_i$  = Necesidades Máximas declaradas por el proceso  $i$ .

$n$  = Cantidad actual de procesos

$RT$  = Recursos Totales del sistema.



# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero

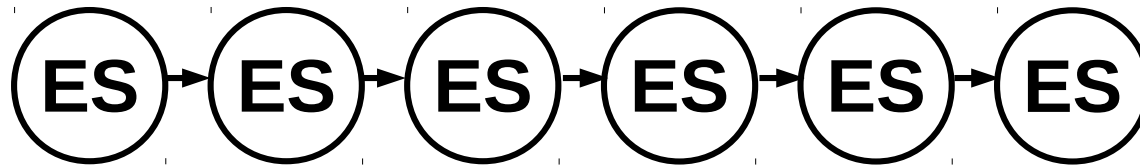
- Estado Seguro ==> No habrá Deadlock
  - Se asigna el recurso al proceso.
- Estado Inseguro ==> Podría existir Deadlock
  - No se asigna el recurso al proceso.

# Deadlock

## Tratamiento del Deadlock:

- 2) Evasión o Predicción de Deadlock:

### 2) Algoritmo del Banquero



# Deadlock

## Tratamiento del Deadlock:

- **2) Evasión o Predicción de Deadlock:**

- 2) Algoritmo del Banquero

- Estado del sistema:

- Vector de Recursos Totales del sistema.
      - Vector de Recursos Disponibles del sistema.

# Deadlock

## Tratamiento del Deadlock:

- **2) Evasión o Predicción de Deadlock:**

- 2) Algoritmo del Banquero

Estado del sistema:

- Matriz de necesidades máximas declaradas por el proceso.
- Matriz de recursos asignados a los procesos.

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero

#### Estado del sistema: Ejemplo

Necesidades Máximas

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

Recursos Asignados

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

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero Algoritmo de seguridad

Necesidades Máximas

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

Recursos Asignados

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

Necesidades pendientes

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

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

R1	R2	R3
0	1	1

## 2) Algoritmo del Banquero Algoritmo de seguridad

Necesidades Máximas

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

Recursos Asignados

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

Necesidades pendientes

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

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero Algoritmo de seguridad

R1	R2	R3
0	1	1

Finaliza P2

R1	R2	R3
2	1	2

Necesidades Máximas

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

Recursos Asignados

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

Necesidades pendientes

	R1	R2	R3
P1	2	1	0
P2	<del>0</del>	<del>0</del>	<del>1</del>
P3	1	2	0

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1



# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero Algoritmo de seguridad

Necesidades Máximas

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

Recursos Asignados

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

Necesidades pendientes

	R1	R2	R3
<del>P1</del>	<del>2</del>	<del>1</del>	<del>0</del>
<del>P2</del>	<del>0</del>	<del>0</del>	<del>1</del>
P3	1	2	0

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1

R1	R2	R3
0	1	1

Finaliza P2

R1	R2	R3
2	1	2

Finaliza P1

R1	R2	R3
2	2	2

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero Algoritmo de seguridad

Necesidades Máximas

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

Recursos Asignados

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

Necesidades pendientes

	R1	R2	R3
P1	<del>2</del>	<del>1</del>	<del>0</del>
P2	<del>0</del>	<del>0</del>	<del>1</del>
P3	<del>1</del>	<del>2</del>	<del>0</del>

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1

R1	R2	R3
0	1	1

Finaliza P2

R1	R2	R3
2	1	2

Finaliza P1

R1	R2	R3
2	2	2

Finaliza P3

R1	R2	R3
3	2	3

Es Estado Seguro. La secuencia Segura es P2 → P1 → P3

# Deadlock

## Tratamiento del Deadlock:

- **2) Evasión o Predicción de Deadlock:**

- 2) Algoritmo del Banquero

- Algoritmo de solicitud de recursos

- Se simula la asignación del recurso y se aplica el algoritmo de seguridad.

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero

#### Algoritmo de solicitud de recursos

Necesidades Máximas

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

Recursos Asignados

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

Necesidades pendientes

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

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	1	1

P3 solicita una instancia de R2

# Deadlock

## Tratamiento del Deadlock:

### ▪ 2) Evasión o Predicción de Deadlock:

#### 2) Algoritmo del Banquero

#### Algoritmo de solicitud de recursos

Necesidades Máximas

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

Recursos Asignados

	R1	R2	R3
P1	0	1	0
P2	2	0	1
P3	1	<del>0</del> <b>1</b>	1

Necesidades pendientes

	R1	R2	R3
P1	2	1	0
P2	0	0	1
P3	1	<del>2</del> <b>1</b>	0

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	<del>1</del> <b>0</b>	1

P3 solicita una instancia de R2

# Deadlock

## Tratamiento del Deadlock:

- **3) Detección y Recuperación de Deadlock:**
  - Puede ocurrir Deadlock.
  - No hay restricciones para asignar recursos disponibles.
  - Periódicamente se ejecuta el Algoritmo de Detección para determinar la existencia de Deadlock.

# Deadlock

## Tratamiento del Deadlock:

- **3) Detección y Recuperación de Deadlock:**
  - Opciones de Recuperación:
    - Terminar procesos involucrados.
    - Retroceder el proceso a un estado anterior.
    - Terminar algún proceso involucrado hasta que deje de existir Deadlock.
    - Expropiar Recursos hasta que no exista Deadlock.

# Deadlock

## Tratamiento del Deadlock:

- **3) Detección y Recuperación de Deadlock:**
  - Criterios de selección de procesos para terminar o expropiar:
    - Menor tiempo de procesador consumido.
    - Menor cantidad de salida producida.
    - Mayor tiempo restante estimado.
    - Menor número total de recursos asignados.
    - Menor prioridad



# Deadlock

## Tratamiento del Deadlock:

### ▪ 3) Detección y Recuperación de Deadlock:

Recursos Asignados

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

Solicitudes Actuales

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

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	0	1

# Deadlock

## Tratamiento del Deadlock:

### ▪ 3) Detección y Recuperación de Deadlock:

R1	R2	R3
0	0	1

Recursos Asignados

	R1	R2	R3
P1	0	1	0
P2	2	0	1
P3	1	1	1
<del>P4</del>	<del>0</del>	<del>0</del>	<del>0</del>

Solicitudes Actuales

	R1	R2	R3
P1	2	1	0
P2	0	0	1
P3	1	1	0
<del>P4</del>	<del>0</del>	<del>1</del>	<del>1</del>

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	0	1

# Deadlock

## Tratamiento del Deadlock:

### ▪ 3) Detección y Recuperación de Deadlock:

R1	R2	R3
0	0	1

Finaliza P2

R1	R2	R3
2	0	2

Recursos Asignados

	R1	R2	R3
P1	0	1	0
<del>P2</del>	<del>2</del>	<del>0</del>	<del>1</del>
P3	1	1	1
<del>P4</del>	<del>0</del>	<del>0</del>	<del>0</del>

Solicitudes Actuales

	R1	R2	R3
P1	2	1	0
<del>P2</del>	<del>0</del>	<del>0</del>	<del>1</del>
P3	1	1	0
<del>P4</del>	<del>0</del>	<del>1</del>	<del>1</del>

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	0	1

# Deadlock

## Tratamiento del Deadlock:

### ▪ 3) Detección y Recuperación de Deadlock:

R1	R2	R3
0	0	1

Finaliza P2

R1	R2	R3
2	0	2

Recursos Asignados

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

Solicitudes Actuales

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

Recursos Totales

R1	R2	R3
3	2	3

Recursos Disponibles

R1	R2	R3
0	0	1

Los recursos no alcanzan para finalizar P1 y P3, por lo tanto estos procesos están en Deadlock. P4 está en inanición.

# Deadlock

## Tratamiento del Deadlock:

- 4) No tratarlo!

# Deadlock

## **Tratamiento del Deadlock:**

### **Estrategia Integrada**

- Agrupación de recursos y a cada grupo se le aplica alguna de las técnicas.