

---

# Sistemas Operativos I

**“A computer is a state machine. Threads are for people who can't program state machines.”**

**Alan Cox**

Clase: planificación de procesos - sincronización

Rafael Ignacio Zurita <[rafa@fi.uncoma.edu.ar](mailto:rafa@fi.uncoma.edu.ar)>

---

**Advertencia:** Estos slides traen ejemplos.

**No copiar (ctrl+c) y pegar en un shell o terminal los comandos aquí presentes.**

**Algunos no funcionarán, porque al copiar y pegar también van caracteres “ocultos” (no visibles pero que están en el pdf) que luego interfieren en el shell.**

**Sucedió en vivo :)**

**Conviene “escribirlos” manualmente al trabajar.**

## Contenido

## Deadlocks

## Deadlock

Una situación en donde existe un **conjunto bloqueado de procesos** porque cada proceso en el conjunto está *reteniendo un recurso* y además cada proceso en el conjunto está *esperando por un recurso* que fue asignado a otro proceso.

## Deadlock

Es una situación donde existe un **conjunto de procesos bloqueados**.

- Cada proceso en el conjunto está *reteniendo un recurso* y además
- Cada proceso en el conjunto está *esperando un recurso* que fue asignado a otro proceso.

# Sistemas Operativos I - Interbloqueos

---

## Deadlock

Es una situación donde existe un **conjunto de procesos bloqueados**.

- Cada proceso en el conjunto está *reteniendo un recurso* y además
- Cada proceso en el conjunto está *esperando un recurso* que fue asignado a otro proceso.

### Ejemplo:

*Para conseguir trabajo en algún empleo hay que tener experiencia,  
pero la experiencia sólo puede ser obtenida si se trabaja en algún empleo.  
(habría que explicarle a las empresas el concepto de deadlock)*

# Sistemas Operativos I - Interbloqueos

---

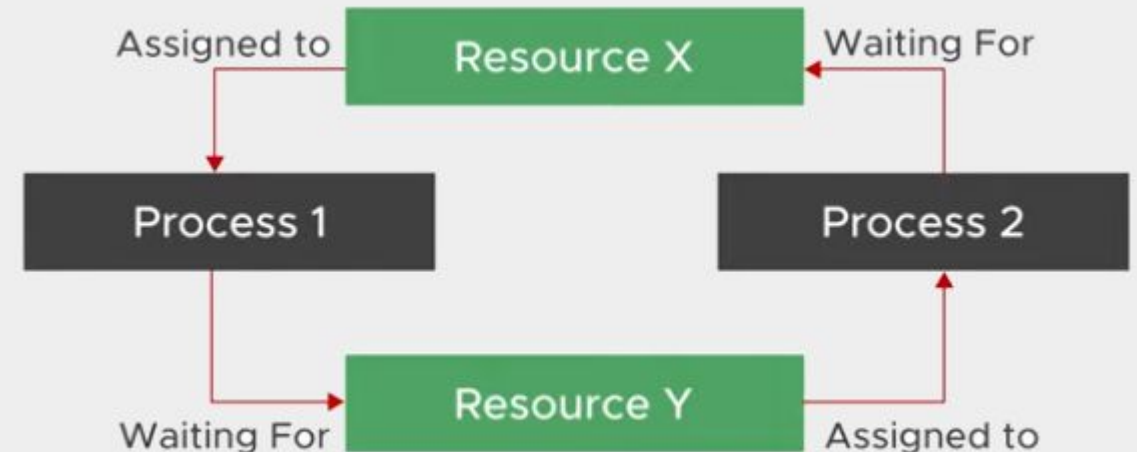
## Deadlock

*Dos o mas procesos están esperando indefinidamente por un evento que solo puede ser causado por uno de los procesos en espera.*

**Ejemplo:**

**P1 y P2 son 2 procesos.**

**X e Y son dos recursos del sistema.**



# Sistemas Operativos I - Interbloqueos

---

## Deadlock

**Dos o mas procesos están esperando indefinidamente por un evento que solo puede ser causado por uno de los procesos en espera.**

**Ejemplo:**

**P0 y P1 son 2 procesos.**

**S y Q son dos semáforos inicializados con el valor 1.**

$P_0$	$P_1$
<code>wait(S);</code>	<code>wait(Q);</code>
<code>wait(Q);</code>	<code>wait(S);</code>
<code>.</code>	<code>.</code>
<code>.</code>	<code>.</code>
<code>signal(S);</code>	<code>signal(Q);</code>
<code>signal(Q);</code>	<code>signal(S);</code>

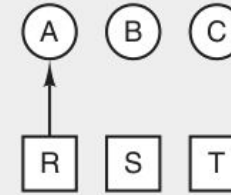


# Sistemas Operativos I - Interbloqueos

## Deadlock : Condiciones - Coffman et al. (1971)

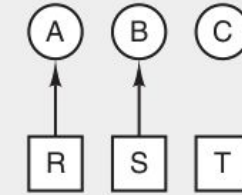
- Exclusión Mutua
- Retención y espera (Hold and wait)
- No desalojo (no preemption)
- Espera circular

1. A requests R  
2. B requests S  
3. C requests T  
4. A requests S  
5. B requests T  
6. C requests R  
deadlock

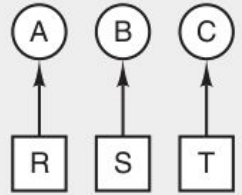


(d)

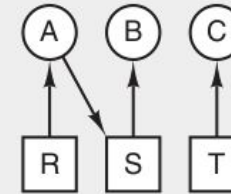
(e)



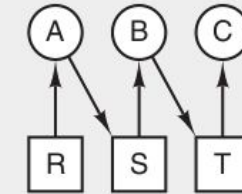
(f)



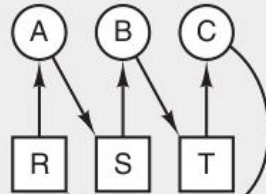
(g)



(h)



(i)



(j)

# Sistemas Operativos I - Interbloqueos

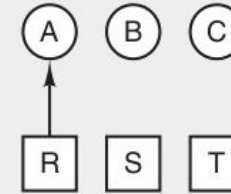
## Deadlock : Condiciones - Coffman et al. (1971)

- Exclusión Mutua
- Retención y espera (Hold and wait)
- No desalojo (no preemption)
- Espera circular

Cómo gestionar deadlocks:

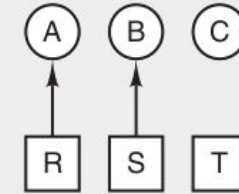
- Prevenir que ocurra una de las cuatro condiciones.  
En la práctica, la última.
- Mismo orden en la obtención de recursos.
- Con un algoritmo de detección (analizar procesos y recursos),  
y algoritmo de recuperación (abortar un proceso, o apropiarse de un recurso).

1. A requests R
  2. B requests S
  3. C requests T
  4. A requests S
  5. B requests T
  6. C requests R
- deadlock

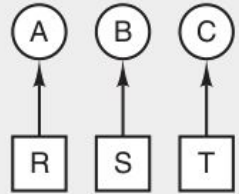


(d)

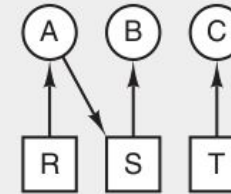
(e)



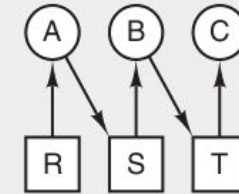
(f)



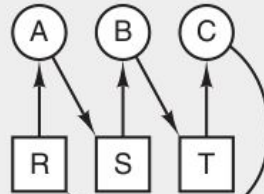
(g)



(h)



(i)



(j)

# Sistemas Operativos I - Sincronización entre procesos

---

## Inversión de Prioridades : Ejemplo Mars Pathfinder (1997)

- Al menos 3 procesos en un sistema con planificación por prioridades.
- Un proceso con baja prioridad retiene un recurso
- El proceso de más alta prioridad requiere el recurso
- Un proceso con prioridad intermedia utiliza la CPU

