

# ***Sistemas Operativos***

***Cursada 2022***

**Comisión S21 y S22**

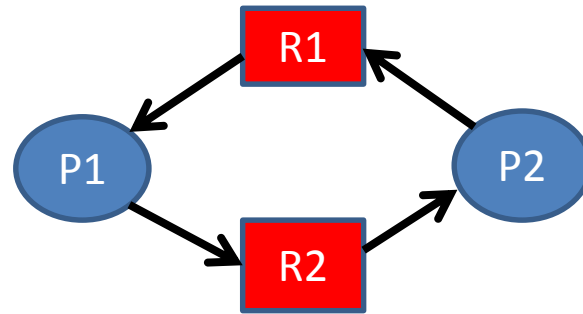
# *Interbloqueo*

Distintas formas de llamarlo

- **Interbloqueo**
- **Abrazo Mortal**
- **Bloqueo Mutuo**
- **Deadlocks**

# *Interbloqueo*

## Ejemplo mas sencillo de interbloqueo



El proceso P1 esta reteniendo el R1

El Proceso P2 esta reteniendo el R2

Se bloquean entre si, por lo tanto decimos que se

**Colgo**

# *Interbloqueo*

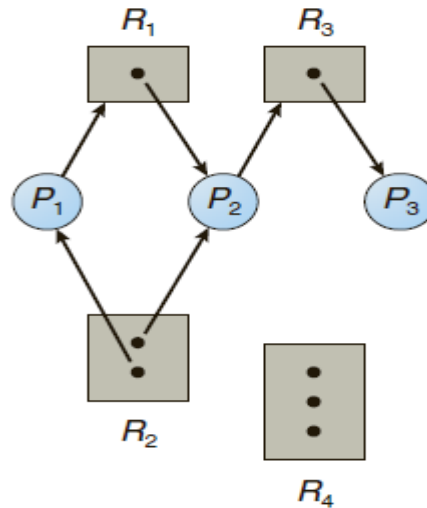
**Condiciones (necesarias) que se deben dar para que haya interbloqueo:**

**(Dado que esto nos va a permitir salir del mismo)**

- **Exclusión Mutua** (Al menos un R debe estar retenido en modo No Compartible)
- **Retención y Espera** (Al menos un P esta reteniendo un R y esperando adquirir otros) El S.O. debe ser permisivo
- **No Apropiación** (El S.O. no atienda solicitudes de expropiación)
- **Espera Circular** (Ejemplo anterior)

# Interbloqueo

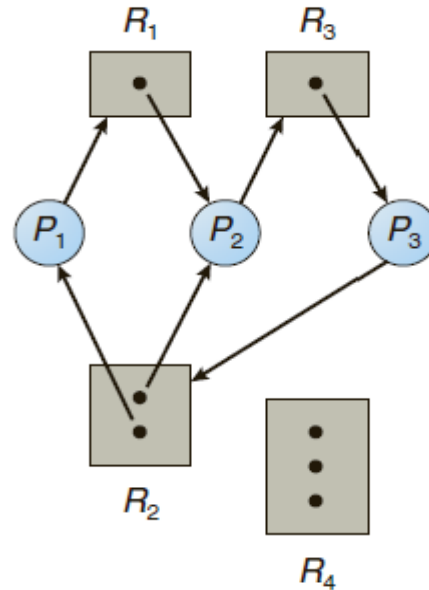
## Otros ejemplos



- Resource instances:
  - One instance of resource type  $R_1$
  - Two instances of resource type  $R_2$
  - One instance of resource type  $R_3$
  - Three instances of resource type  $R_4$
- Process states:
  - Process  $P_1$  is holding an instance of resource type  $R_2$  and is waiting for an instance of resource type  $R_1$ .
  - Process  $P_2$  is holding an instance of  $R_1$  and an instance of  $R_2$  and is waiting for an instance of  $R_3$ .
  - Process  $P_3$  is holding an instance of  $R_3$ .

# Interbloqueo

## Otros ejemplos



$$\begin{aligned} P_1 &\rightarrow R_1 \rightarrow P_2 \rightarrow R_3 \rightarrow P_3 \rightarrow R_2 \rightarrow P_1 \\ P_2 &\rightarrow R_3 \rightarrow P_3 \rightarrow R_2 \rightarrow P_2 \end{aligned}$$

Processes  $P_1$ ,  $P_2$ , and  $P_3$  are deadlocked. Process  $P_2$  is waiting for the resource  $R_3$ , which is held by process  $P_3$ . Process  $P_3$  is waiting for either process  $P_1$  or process  $P_2$  to release resource  $R_2$ . In addition, process  $P_1$  is waiting for process  $P_2$  to release resource  $R_1$ .

# ***Interbloqueo***

**Tenemos situaciones no deseadas**

**Por lo tanto dada la situación de que existan interbloqueos, vemos que hacen lo sistemas operativos??**

**Políticas por parte de los planificadores ante estas situaciones no deseadas??**

# *Interbloqueo*

**Tenemos 3 políticas generales para evitar que esto pase. Son políticas generales de la Ingeniería.**

**Que no ocurra:** tomar las medidas necesarias para evitarlos

**Tener un plan:** detectarlo tempranamente y resolverlo.

**No hacer nada:** puede pasar? Y bueno que pase



# ***Interbloqueo***

**Empezamos por la ultima**

**No Hacer Nada**

**No seria buena idea pero ocurre**

**Que no ocurra:**

➤ **Prevenir**

➤ **Evitarlo**

# *Interbloqueo*

Vamos a trabajar sobre las condiciones antes mencionada

## ➤ **Prevenir**

**Exclusión Mutua:** La característica de los recursos es que no son compartibles. (Ej. Impresora)

**Retención y Espera:** Podemos determinar que:

- Se espera un tiempo determinado
- Si esta en wait mucho tiempo le quito todo
- El sistema debe ser transaccional
- Se retrotrae todo al principio

# *Interbloqueo*

## **Espera Circular:**

- Puedo evitar que se formen ciclos?
- Podría hacerlo implementando un algoritmo que recorra el grafo.
- Anotar que proceso tiene cada recurso
- Correr el algoritmo detectando en el grafo donde se produce un interbloqueo
- Son algoritmos muy complejos para trabajar en una prevención

# ***Interbloqueo***

## **No Apropiación:**

- Podría permitir Expropiar
- Procesos con mayor prioridad
- Puede ser que un proceso expropie (Tiempo Real)
- En sistemas de T.R. debemos garantizar que no ocurra, por eso permitimos la expropiación
- Debemos tener en cuenta en algunos sistemas que el mismo sea transaccional.

# *Interbloqueo*

## ➤ Evitarlo ( Política de que no pase )

Una forma de resolver esto es determinar si:

- Proceso esta en estado **seguro**
- Proceso esta en estado **inseguro**
- Estando en estado inseguro no quiere decir que caída en Deadlocks
- Para estar **seguro** debo asignar todos los recursos al momento de solicitarlos
- No empiezo hasta que tenga todo
- Si otro necesita algún recurso deberá ir a lista espera

# *Interbloqueo*

## ➤ Evitarlo ( Política de tener u Plan)

- Detectar
- Recuperar
  - La detección no es algo simple.
  - Algoritmos matemáticos que trabajen con grafos
  - Recurso tiene una sola instancia es mas fácil

Una política es sacar una foto de todos los procesos que están en wait esperando recursos, esto es específico de algunos.

# ***Interbloqueo***

## ➤ **Evitarlo ( Política de tener u Plan)**

Luego hago el grafo de asignación de recursos

En definitiva dejo que pase

Que puedo hacer?

- Mato a todos ( que empiecen de nuevo)
- Matar de a uno y ver que pasa
- Cual mato, ahí puedo aplicar prioridad
- Una política muy usada es modificar las condiciones del S.O. para que permita la expropiación de recursos.
- Queda en manos del usuario en ciertas ocasiones

***Fin del Tema***