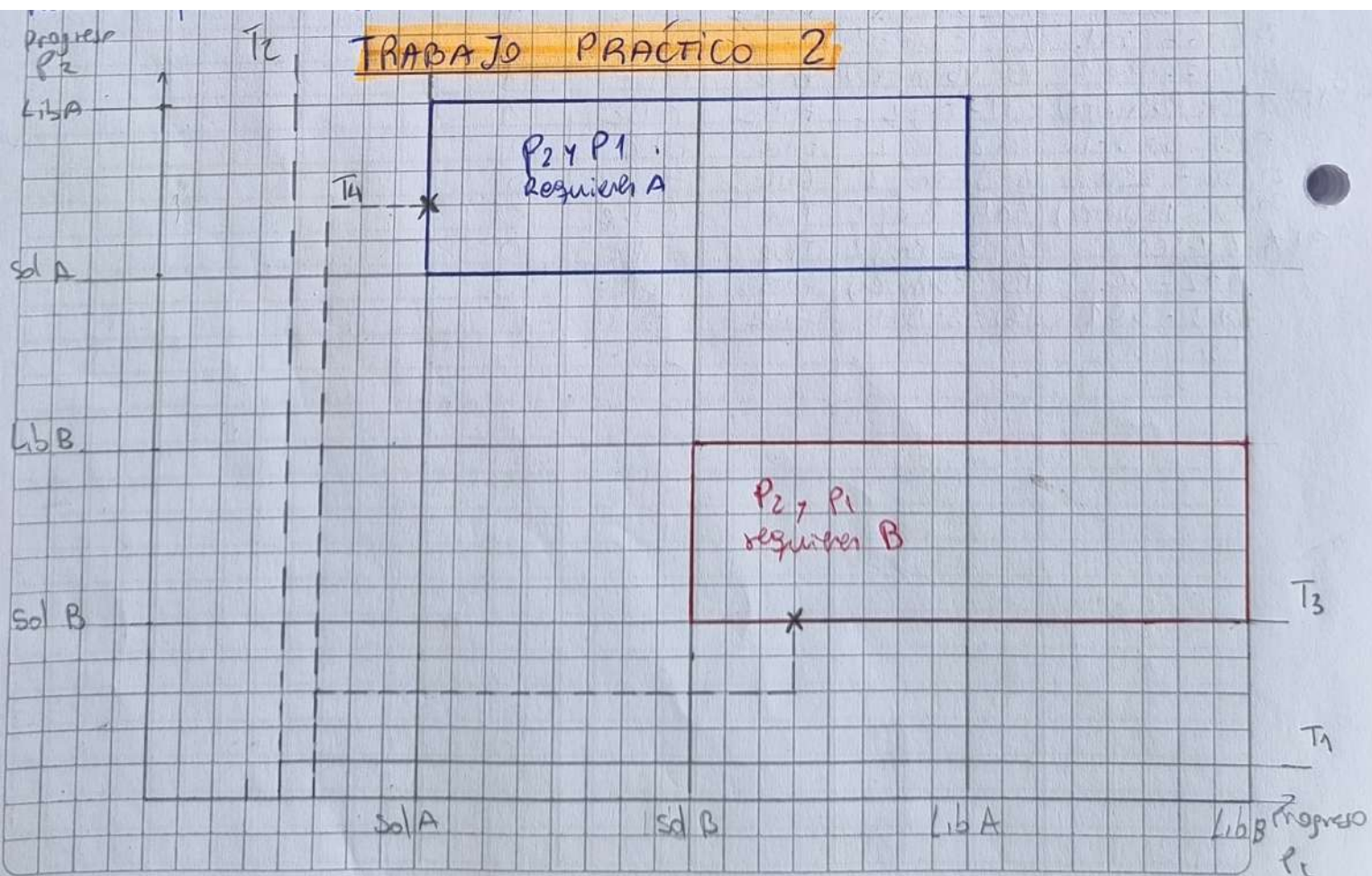


TRABAJO PRACTICO 2

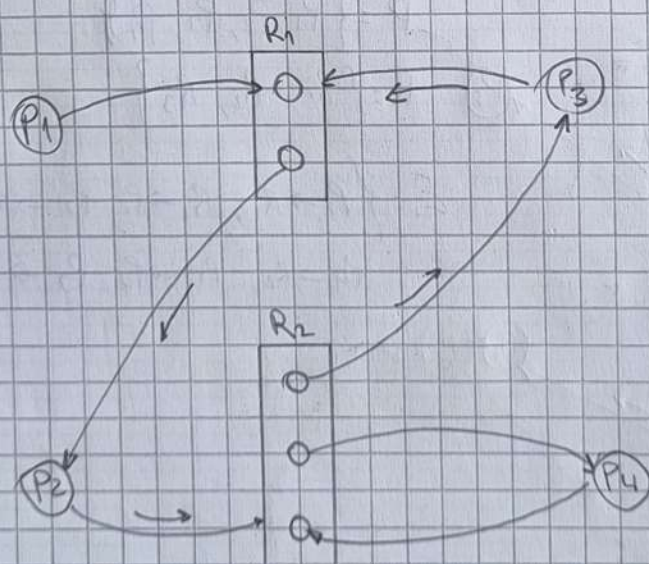


1) a) No existe zona de interbloqueo inevitable. Supongamos q' porque entre dos peticiones de los recursos existe la liberación de uno de ellos, permitiendo seguir la ejecución sin interbloqueo a ambos procesos.

2- T_1 y T_2 son dos trayectorias donde el flujo de ejecución no se ve interrumpido por la solicitud de recursos.

3- T_3 y T_4 son dos trayectorias las cuales se ven interrumpidas en el medio por un cambio de proceso, pero sin embargo, ese cambio de proceso bloquea al proceso entrante por falta de recurso volviendo al proceso anterior para q' termine su ejecución y libere los recursos q' solicite el proceso bloqueado.

2) a)



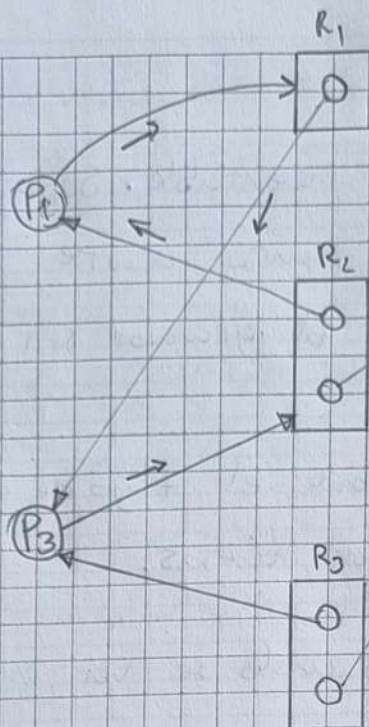
$$P = \{P_1, P_2, P_3, P_4\}$$

$$R = \{R_1, R_2, R_3, R_4, R_5\}$$

$$E = \{R_1 \rightarrow R_1, R_1 \rightarrow R_2, P_3 \rightarrow R_1, R_2 \rightarrow P_2, R_2 \rightarrow P_4, P_1 \rightarrow R_2, P_2 \rightarrow R_2\}$$

Hay un ciclo, el cual es un grafo de asignación de recursos nos indica un posible estado de interbloqueo. Sin embargo suponiendo q' se asigne R_1 a P_1 y R_2 a P_4 y luego los libere se resuelve el posible interbloqueo.

b)



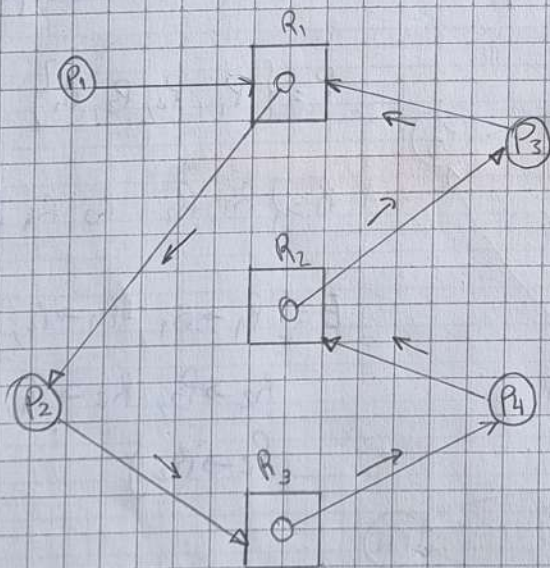
$$P = \{P_1, P_2, P_3\}$$

$$R = \{R_1, R_2, R_2, R_3, R_3, R_3\}$$

$$E = \{P_1 \rightarrow R_1, R_1 \rightarrow P_2, P_2 \rightarrow R_1, \\ P_3 \rightarrow R_2, R_2 \rightarrow P_2, R_3 \rightarrow P_3, \\ R_3 \rightarrow P_2\}$$

tenemos un ciclo dentro de este grafo pero ignorado y P_2 libera R_2 y luego se le asigna R_2 a P_3 este cuando finalice su ejecución liberará los recursos permitiendo a P_1 obtener R_1 solicitado.

c)



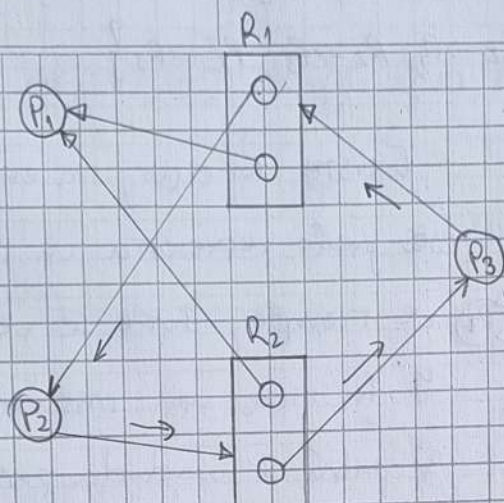
$$P = \{P_1, P_2, P_3, P_4\}$$

$$R = \{R_1, R_2, R_3\}$$

$$E = \{P_1 \rightarrow R_1, R_1 \rightarrow P_2, P_2 \rightarrow R_3, R_3 \rightarrow P_4, \\ P_4 \rightarrow R_2, R_2 \rightarrow P_3, P_3 \rightarrow R_1\}$$

tenemos un ciclo y además una instancia de cada recurso, donde existe una lista cerrada de procesos donde cada proceso solicita el recurso y está ocupando el proceso siguiente, por lo tanto, existe abstracción mortal.

d)



$$P = \{P_1, P_2, P_3\}$$

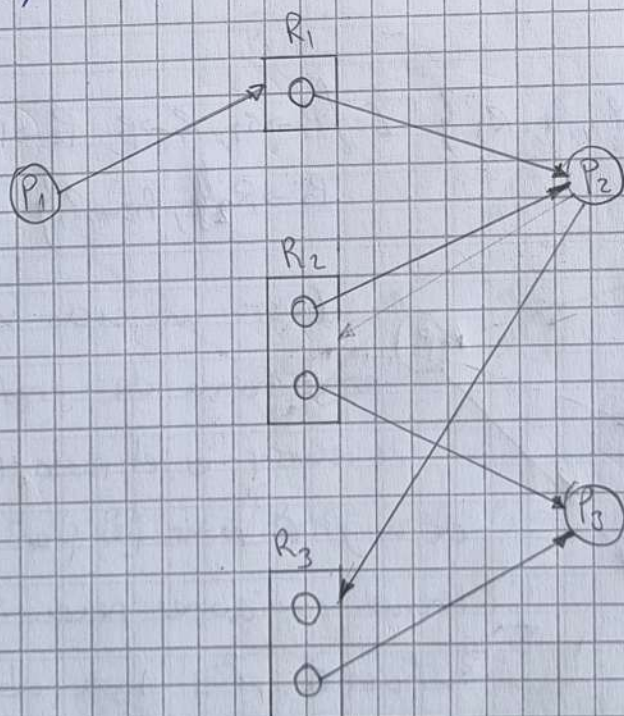
$$R = \{R_1, R_2, R_2, R_2\}$$

$$E = \{R_1 \rightarrow P_2, R_1 \rightarrow P_1, P_2 \rightarrow R_2, R_2 \rightarrow P_1, R_2 \rightarrow P_3, P_3 \rightarrow R_1\}$$

Aquí tenemos un ciclo pero suponiendo lo del primer caso podemos considerar q' P_1 libera los recursos de R_1 y R_2 y así poder ~~asignar~~ obtenerlos tanto P_2 a R_2 como R_1 para P_3 .

$$3) a) P = \{P_1, P_2, P_3\}, R = \{R_1, R_2, R_2, R_2, R_3, R_3\}$$

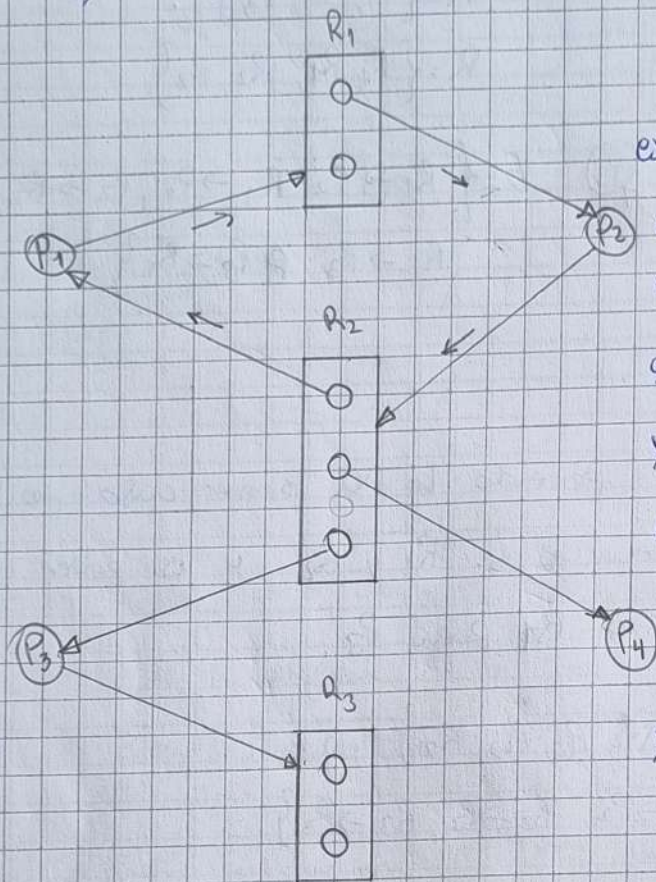
$$E = \{R_1 \rightarrow R_1, R_1 \rightarrow P_2, P_2 \rightarrow R_3, R_2 \rightarrow P_2, R_2 \rightarrow P_3, R_3 \rightarrow P_3\}$$



No hay ciclos ~~se~~ por lo q' no hay posibilidad de abanico mortal

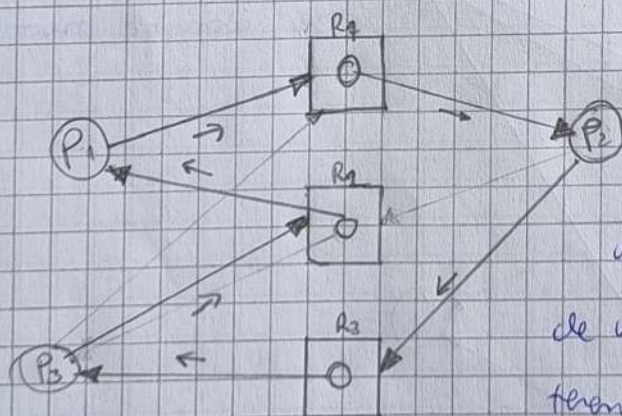
$$b) P = \{P_1, P_2, P_3, P_4\}, R = \{R_1, R_1, R_2, R_2, R_2, R_3, R_3\}$$

$$E = \{P_1 \rightarrow R_1, R_1 \rightarrow P_2, P_2 \rightarrow R_2, R_2 \rightarrow P_1, R_2 \rightarrow P_4, R_2 \rightarrow P_3, P_3 \rightarrow R_3\}$$



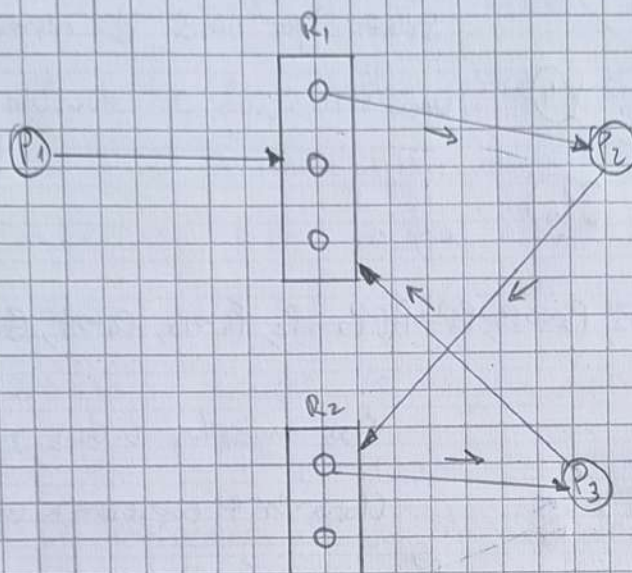
Existe un ciclo, sin embargo, este grafo representa un instante de tiempo, donde se ve claramente q R_1 tiene una instancia más q podrá ser usada por P_1 y luego con la liberación de R_2 por parte de P_4 o P_3 , donde este último también tiene recursos de R_3 disponible, P_2 podrá obtener R_2 y evitando el deadlock.

$$c) P = \{P_1, P_2, P_3\} \quad R = \{R_1, R_2, R_3\} \quad E = \{P_1 \rightarrow R_1, R_1 \rightarrow P_2, P_2 \rightarrow R_3, R_3 \rightarrow P_3, P_3 \rightarrow R_2, R_2 \rightarrow P_1\}$$



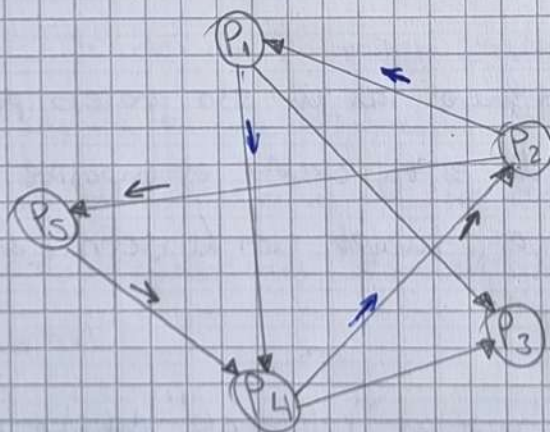
Existe un ciclo y además esta forma de grafo de asignación de recursos es el caso particular de un grafo más fuerte por lo q' tener condiciones necesarias y suficientes p/ q' haya aborto mortal

a) $P = \{P_1, P_2, P_3\}$ $R = \{R_1, R_1, R_1, R_2, R_2, R_2\}$ $E = \{P_1 \rightarrow R_1, R_1 \rightarrow P_2, P_2 \rightarrow R_2, R_2 \rightarrow P_3, P_3 \rightarrow R_1\}$



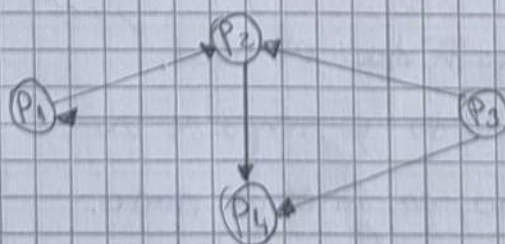
Existe un ciclo, sin embargo, por la reversa posee instrucciones suficientes p/ satisfacer las peticiones de todos los procesos.

4) a) $P = \{P_1, P_2, P_3, P_4, P_5\}$, $E = \{P_1 \rightarrow P_3, P_1 \rightarrow P_4, P_2 \rightarrow P_1, P_2 \rightarrow P_4, P_2 \rightarrow P_5, P_3 \rightarrow P_3, P_3 \rightarrow P_4\}$



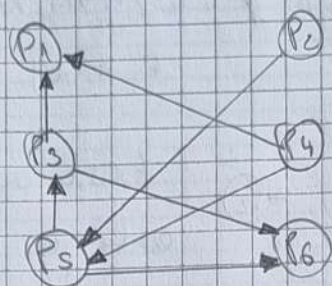
Este grafo wait-for construye dos ciclos por lo q' representa una situación de interbloqueo.

b) $P = \{P_1, P_2, P_3, P_4\}$, $E = \{P_1 \rightarrow P_2, P_2 \rightarrow P_4, P_3 \rightarrow P_1, P_3 \rightarrow P_2, P_3 \rightarrow P_4\}$



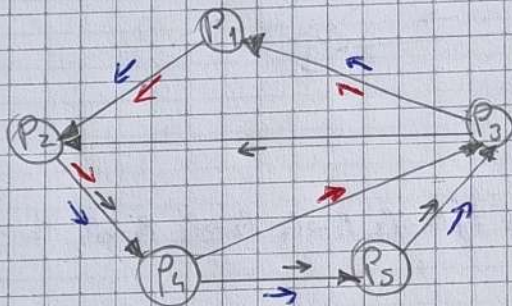
No existen ciclos por lo q' no representa una situación de bloqueo mutuo.

c) $P = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ $E = \{P_2 \rightarrow P_5, P_3 \rightarrow P_1, P_3 \rightarrow P_6, P_4 \rightarrow P_1, P_4 \rightarrow P_5, P_5 \rightarrow P_3, P_5 \rightarrow P_6\}$



No existe ciclo alguno en el grafo por lo que no representa una situación de interbloqueo.

d) $P = \{P_1, P_2, P_3, P_4, P_5\}$ $E = \{P_1 \rightarrow P_2, P_2 \rightarrow P_3, P_3 \rightarrow P_1, P_2 \rightarrow P_4, P_4 \rightarrow P_3, P_4 \rightarrow P_5, P_5 \rightarrow P_3\}$



Hay múltiples ciclos y claramente representa una situación de abruzo mortal.

5) No es posible un interbloqueo con un solo proceso participando, pues la condición suficiente de espera circular es imposible q' se dé y además con un solo proceso no se cumple con la definición de abruzo mortal.

6) Entre las dificultades por hacer un "rollback" debido a un interbloqueo se enumeraron:

los recursos extras q' ocupa el SO para mantener los checkpoints de cada proceso. Esto se denomina overhead.

Pérdida de trabajo realizado y tiempo de uso de la CPU, pues todo el trabajo realizado no sirve de nada.

Posible destrucción de recursos utilizada o requerido justo en el ^{momento} del rollback, por lo q' puede bloquear el proceso hasta la disponibilidad del recurso.

7) Para detectar la inanición no existe algún método o algoritmo directo, por lo que el SO utiliza algunas variables como el tiempo de espera para cada proceso donde si se supera determinado umbral sin q' el proceso haya recibido el recurso se determina q' este es inanición. Otra forma es el uso del concepto de prioridad de un proceso, donde el SO lleva la noción de la prioridad de cada proceso y donde le asigna mayor prioridad a los más urgentes garantizando la recepción de los recursos solicitados.

Para la detección de un aborto mortal, a diferencia de la inanición, es un poco más simple determinar una posible situación de interbloqueo y es a través de un grafo de asignación de recursos; donde se analiza si existen ciclos ~~entre de los~~ ^{formados por los recursos} y procesos. Si de cada recurso existe ^{1 instancia} ~~seguir~~ ^{y existen} existe un ciclo, seguro es una situación de interbloqueo, mientras q' si hay más de una instancia de los recursos hay q' analizar y tener en cuenta otras cuestiones para determinar si es una situación de aborto mortal.

8) Exclusión mutua Sin expropiación, retención y espera.

a) No puede ocurrir interbloqueo pues no se cumple la condición necesaria de ~~retención y espera~~ sin expropiación.

b) Si, justamente el hecho de entrar el algoritmo mortal con esta técnica genera otro problema, el del bloqueo indefinido o inanición.

Imaginando P_i A es un proceso largo y cada vez P_i solicita un recurso no disponible este se bloquea reteniendo los recursos permanentemente o P_i muere y luego viene otro proceso recurrente, B, P_i siempre utiliza los recursos P_i utiliza A y además es más corto entonces cada vez P_i A se bloquea, por culpa de B pierde todos sus recursos expropiados B, logrando así P_i A no termine su ejecución.