Ejemplo Algoritmo del Banquero y consideraciones para algoritmo de Detección.

Dado un sistema cuya situación inicial en cuanto a recursos del sistema son 3 unidades para cada tipo de recursos que son R1, R2, R3.

Suponga que se tiene como estrategia para tratar el deadlock la evasión mediante el algoritmo del banquero. Siendo el estado de los recursos el siguiente:

Maximo				
	R1	R2	R3	R4
PA	2	1	1	2
PB	2	2	0	1
PC	1	0	0	2

Allocation				
	R1	R2	R3	R4
PA	1	1	1	0
PB	0	0	0	0
PC	0	1	1	1

a) Determine al Matriz Necesidad
b) Se encuentra en estado Seguro
c) Si llega una solicitud del PA por una unidad más del R1 (1000), ¿se le puede conceder inmediatamente?
d) Luego llega otra solicitud de PC por una unidad de R4 (0001), ¿podrá ser concedida inmediatamente?
e) Luego llega otra solicitud de PB por una unidad de R3 (0010), ¿podrá ser concedida inmediatamente?

Recordar:

- Que la Matriz Max = a la *demanda máxima* que puede solicitar el o los procesos. En esta Matriz se encuentran tanto los recursos que ya están asignados como los que todavía no les fue asignados, pero si ya está su solicitud máxima, no pudiendo ser una nueva petición de solicitud mayor de lo que ya se pidió.
- Matriz Allocation (Asignación), son los recursos que los procesos del Sistema ya tienen asignado.
- Matriz Necesidad (Need) resulta de la diferencia del total que demanda Max menos lo que ya le fue asignado [Max (i,j) – Allocation(i,j)] esta diferencia no puede tener valores negativos sino estaría demostrando que tienen asignados más recursos de lo máximo permitido.

  La matriz Need es necesaria determinarla para poder continuar con el Algoritmo del Banquero.
- Vector Disponibilidad (Avaible): Es lo que queda del total de los recursos del Sistema menos los recursos que tienen asignados (es una diferencia escalar) o sea si observamos la matriz Asignación podemos determinar la totalidad asignada de cada recurso la cual será restada

del total de recursos del Sistema dando como resultado lo Disponible en eses instante. Este vector Disponibilidad es también necesario para poder continuar con el Algoritmo del Banquero.

Tener en cuenta que cuando llega una nueva petición de solicitud por parte de un proceso sobre un recurso, esta tampoco deberá ser mayor que lo que está disponible.

Resolución:

Maximo				
	R1	R2	R3	R4
PA	2	1	1	2
PB	2	2	0	1
PC	1	1	1	2

Allocation				
	R1	R2	R3	R4
PA	1	1	1	0
PB	0	0	0	0
PC	0	1	1	1

a) Para determinar la Matriz Need = Max – Asignación

Necesidad				
	R1	**R2**	**R3**	**R4**
PA	1	0	0	2
PB	2	2	0	1
PC	1	0	0	1

b) Para poder determinar si el Sistema está en estado Seguro necesitamos saber la Disponibilidad para este instante de tiempo.
   - Pueden ocurrir dos posibilidades, que se les de la Disponibilidad o bien que deban calcularla (para esto necesitan saber la totalidad de recursos del Sistema) y si prestan atención si le llegaríamos a preguntar cuántos recursos en total cuenta el Sistema y tendrían como dato la Disponibilidad también no sería difícil determinarlo.

En este caso debemos calcular la Disponibilidad y como dato tenemos los recursos totales del sistema.

Rtotales			
R1	R2	R3	R4
3	3	3	3

Allocation				
	R1	R2	R3	R4
PA	1	1	1	0
PB	0	0	0	0
PC	0	1	1	1
Ta	1	2	2	1

Ta=total de recursos asignados a todos los procesos

>> D [2 1 1 2] → (R tot – Ta).

Obtenida la disponibilidad y la matriz necesidad comenzaremos a comparar:

Necesidad				
	R1	**R2**	**R3**	**R4**
PA	1	0	0	2
PB	2	2	0	1
PC	1	0	0	1

Siempre comenzamos en orden FiFO para ir en orden. Comparamos si PA_N[1 0 0 2] <= D[2 1 1 2] (comparamos componente a componente) y si se cumple entonces:
La secuencia de procesos comienza con PA, recuerden que tienen que estar todos los procesos para que el Sistema esté en Estado Seguro.

- ✓ Ahora debemos devolver lo que tenía asignado PA, esos recursos deberán pasar a ser parte de la nueva disponibilidad.
  D[2 1 1 2] + PA_A[1 1 1 0] →D[3 2 2 2]

  PB_N[2 2 0 1] <= D[3 2 2 2]

  D[3 2 2 2]+PB_A[0 0 0 0] →D[3 2 2 2]

  PC_N[1 0 0 1] <= D[3 2 2 2]

  D[3 2 2 2]+PC_A[0 1 1 1] →D[3 3 3 3] (Si observan una vez que terminamos todos los procesos el vector D[] es igual a los recursos totales del sistema.)

- ✓ S = {PA, PB, PC) Secuencia que muestra que todos los procesos podrán ejecutarse por lo tanto el Sistema se encuentra en estado seguro.
- ✓ Si algún proceso no estaría en el conjunto significaría que el estado no es seguro pero que no implica que exista bloqueo indefinido.

c) Si llega una solicitud del PA por una unidad más del R1 (1000), ¿se le puede conceder inmediatamente?

Maximo				
	R1	R2	R3	R4
PA	2	1	1	2
PB	2	2	0	1
PC	1	1	1	2

Allocation				
	R1	R2	R3	R4
PA	2	1	1	0
PB	0	0	0	0
PC	0	1	1	1

Disponib.			
R1	R2	R3	R4
1	1	1	2

Maximo				
	R1	R2	R3	R4
PA	2	1	1	2
PB	2	2	0	1
PC	1	1	1	2

Allocation				
	R1	R2	R3	R4
PA	1	1	1	0
PB	0	0	0	0
PC	0	1	1	1

D [2 1 1 2]

Necesidad				
	R1	**R2**	**R3**	**R4**
PA	1	0	0	2
PB	2	2	0	1
PC	1	0	0	1

- ✓ Evaluamos la solicitud antes de ponernos a resolver a ciegas:
  1° que la nueva solicitud no sea mayor a lo que tiene el proceso PA en Max por cada recurso o sea que, si es posible porque es una unidad de R1 para el PA y en Max R1 de PA = 2.

- ✓ 2° Evaluamos que esa unidad de R1 sea menor que las unidades de R1 en el vector Disponibilidad que también se cumple ya que D tiene 2u de R1
- ✓ Recién ahora nos ponemos a trabajar con el algoritmo de Estado Seguro

Max queda como está.

Se modifica las matrices Asignación, Need y Disponibilidad.
La solicitud de PA se la debe sumar en la matriz Asignación, y restar del vector Disponibilidad y volver a calcular Need.

Necesidad				
	R1	**R2**	**R3**	**R4**
PA	0	0	0	2
PB	2	2	0	1
PC	1	0	0	1

Aplicando el procedimiento se puede conceder inmediatamente la solicitud y el estado del Sistema es seguro. S{ PA, PB, PC}

d) Luego llega otra solicitud de PC por una unidad de R4 (0001), ¿podrá ser concedida inmediatamente?
Luego de realizar los controles anteriormente mencionados estas son nuestras nuevas matrices.

Maximo				
	R1	R2	R3	R4
PA	2	1	1	2
PB	2	2	0	1
PC	1	1	1	2

Allocation				
	R1	R2	R3	R4
PA	2	1	1	0
PB	0	0	0	0
PC	0	1	1	2

Disponib.			
R1	R2	R3	R4
1	1	1	1

Necesidad				
	R1	**R2**	**R3**	**R4**
PA	0	0	0	2
PB	2	2	0	1
PC	1	0	0	0

Aplicando el procedimiento se puede conceder inmediatamente la solicitud y el estado del Sistema es seguro. S{ PC, PA, PB}

e) Luego llega otra solicitud de PB por una unidad de R3 (0010), ¿podrá ser concedida inmediatamente?
- ✓ Si analizamos no es posible ya que el primer control era que la solicitud del PB por una unidad de R3 no debe ser mayor que lo máximo que tenía permitido.
Solicitud de PB[0 0 1 0] > Max PB [2 2 0 1] recuerden que la comparación siempre se realiza de recurso a recurso.

No se puede atender esa solicitud.

En cambio las solicitudes de este ejemplo todavía los procesos no terminaron su ejecución por eso es que los vamos comparando y modificando sus respectivas Asignaciones (Allocation), Need y disponibilidad. Son simulaciones si hubiese modificación en sus pedidos.

Los recursos totales del sistema no varían, son los mismos desde que comienza hasta que terminan de ejecutarse todos sus procesos.-

Algoritmo de Detección de Bloqueo

Funciona prácticamente igual que el Algoritmo del Banquero

La diferencia es que no tenemos la Matriz Max, solo tenemos la Matriz Asignación, Requerimiento (demanda) y Disponibilidad.

No determinamos la matriz Need (necesidad) sino hacemos la comparación con la matriz Requerimiento con respecto al vector Disponibilidad ------ →Req. <= Disp.

→Luego vamos actualizando Disponibilidad agregándole los recursos de los procesos que se van ejecutando pero los "recursos que tienen asignados".

→Cuando encontramos una secuencia donde están presentes todos los procesos del Sistema podemos decir que no hay Deadlock.

→Si algún proceso queda fuera de esa secuencia, sabemos que son los que están en bloqueo y es fácil determinarlo haciendo el grafo de asignación de recursos para saber cuál es la causa del bloqueo.

→Si llegan solicitudes realizamos la evaluación con demanda y diponibilidad y luego procedemos a aumentar esa unidad de recurso en la Matriz Asignación del proceso correspondiente, restar esa unidad de recurso del proceso correspondiente de la Matriz Demanda (requerimiento) y restar la unidad de recurso del vector Disponibilidad.

→Aquí también los recursos totales del Sistema será = todos los recursos asignados a los proceso + los recursos del vector disponibilidad.

Por lo tanto podemos no darle el vector Disponibilidad de dato pero si darles los recursos totales del sistema y ustedes determinar disponibilidad (RT – Asig. = Disp), o bien solicitarle la cantidad de recursos totales del Sistema y si necesitan el dato de Disponibilidad.