

ISMAEL FERNÁNDEZ HERRERUELA

ACTIVIDAD 1

1.No existe interbloqueo ya que se rompe la espera circular. Una secuencia segura seria P5 ya que tiene recursos asignados y no tiene que esperar a nada.

2.Se cumplen:

- Retencion y espera
- Exclusion mutua
- No expulsión

Y no se cumple la espera circular

ACTIVIDAD 2

1. Es cierto que siempre ocurrirá un interbloqueo cuando tengamos un ciclo dirigido en un GAR. No se necesita realizar ningún cambio.
2. Los interbloques aún son posibles con solo dos subprocesos si ambos están esperando recursos que posee el otro subproceso.
Por lo tanto, la oración se cambiaría para que diga: "Los interbloques son posibles incluso con solo 2 subprocesos".
3. Los monitores pueden ayudar a prevenir algunos interbloques, pero no pueden garantizar que no sucedan. Esto cambiaría la declaración original para que diga: "Los monitores pueden reducir el riesgo de interbloques, pero no son una forma infalible de prevenirlos".
4. Podría surgir una situación en la que un proceso tiene recursos que no está utilizando pero otros procesos no pueden avanzar porque están esperando los mismos recursos porque cada proceso solicita todos los recursos que necesitará al comienzo de su ejecución. La oración entonces se cambiaría para quedar de la siguiente manera: "Una estrategia de solución para evitar interbloques es que cada proceso solicite los recursos que necesita cuando los necesita y los libera cuando ya no los necesita".
5. La mayoría de los procesos que pueden solicitar una instancia de cada recurso sin causar un interbloqueo son seis si hay dos tipos de recursos (R1, R2) y tres instancias de cada uno. Por lo tanto, no se requieren ajustes.
6. El gráfico de asignación de recursos tiene un bucle dirigido si al menos un proceso está esperando un recurso propiedad de otro proceso que también está esperando un recurso propiedad del primer proceso. Como resultado, no se requieren ajustes.

ACTIVIDAD 3

1. Los procesos siempre deben solicitar recursos en el mismo orden para evitar interbloques. En este caso, par procesa la primera solicitud R_i , seguida de $R_{((i+1)\%8)}$, mientras que impar procesa la primera solicitud R_i , seguida de $R_{((i+1)\%8)}$. Como resultado, es necesario alterar el perfil de ejecución de los procesos pares para que soliciten recursos en el mismo orden que los procesos impares: primero R_i y luego $R_{((i+1)\%8)}$, para evitar interbloques en un sistema con un número par de procesos.

2. Todas las condiciones de Coffman se cumplen si cada proceso completa su declaración A antes de que cualquiera de ellos complete su declaración B. Esto se debe a que, en esa situación, ningún proceso puede apoderarse de un recurso que requiere otro proceso porque todos están esperando un recurso que se requiere para que otro proceso termine su ejecución. Como resultado, se evita la condición de interbloqueo de Coffman.
3. La condición de espera circular de Coffman se viola si las declaraciones A y B se intercambian en los perfiles de ejecución de los procesos del mismo nivel. En esta situación, si el proceso P0 adquiere el recurso R1 y luego intenta adquirir el recurso R0, ocurrirá un interbloqueo porque el proceso P1 ya tiene el recurso R0 y está esperando que el proceso P0 libere el recurso R1. Esta afirmación es precisa como resultado.
4. Un punto muerto no sucederá si un proceso cambia su perfil de ejecución para que ejecute su declaración B primero y luego es una declaración, y los otros procesos mantienen el perfil mencionado anteriormente. Esto se debe a que el proceso en cuestión adquirirá con éxito el recurso R1 antes de intentar adquirir el recurso R ((i + 1)% 8), lo que significa que ningún proceso esperará el recurso R1 que el proceso en cuestión tiene. Además, debido a que los otros procesos no realizan solicitudes de recursos en el mismo orden que el proceso que liberó el recurso R1, ningún otro proceso puede reclamarlo. Por lo tanto, la afirmación es precisa.

ACTIVIDAD 4

1. Tiempo de respuesta máximo de la tarea A:

$$R_A = 2 + \max(0, \text{ceil}((0 - 4 + 10) / 10) * 2) = 2 + \max(0, \text{ceil}(-0.2) * 2) = 2 \text{ segundos}$$

Tiempo de respuesta máximo de la tarea B:

$$R_B = 5 + \max(0, \text{ceil}((2 - 10 + 20) / 20) * 5) = 5 + \max(0, \text{ceil}(-0.15) * 5) = 5 \text{ segundos}$$

Tiempo de respuesta máximo de la tarea C:

$$R_C = 14 + \max(0, \text{ceil}((5 - 30 + 40) / 40) * 14) = 14 + \max(0, \text{ceil}(-0.275) * 14) = 14 \text{ segundos}$$

Para que el sistema sea planificable, es necesario que el tiempo de respuesta máximo de cada tarea sea menor o igual que su plazo correspondiente. En este caso, como $R_A \leq 4$, $R_B \leq 10$ y $R_C \leq 30$, se puede concluir que el sistema es planificable.

- 2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	A		s1		n		s1															
3	B			s2			n		s2		n		s1			s2	n		s1			
4	C	s1	s1																			

3. Debemos identificar la tarea con la prioridad más alta que necesita acceso a cada semáforo para calcular el techo de prioridad para los semáforos S1 y S2 utilizando el protocolo de techo de prioridad inmediata.

La tarea con la prioridad más alta que necesita el semáforo S1 es la tarea C, que tiene una prioridad de 3. Para el semáforo S1, la prioridad máxima es 3, como resultado.

La tarea B, que tiene una prioridad de 2, es la más importante y la que más necesita el semáforo S2. Debido a esto, el techo de prioridad del semáforo S2 se establece en 2.

4. Tarea A: $20 - 1 = 19$ unidades de tiempo

Tarea B: $21 - 1 = 20$ unidades de tiempo

Tarea C: $15 - 0 = 15$ unidades de tiempo

Siempre que cada tarea se complete antes de su fecha de vencimiento, el sistema se puede programar. El problema de inversión de prioridad también se evita porque las tareas de mayor prioridad tienen acceso primero a los semáforos que necesitan, y ninguna tarea de menor prioridad puede obstruir la ejecución de una tarea de mayor prioridad utilizando un semáforo que pertenezca a este último.

ACTIVIDAD 5

A.) -Tarea A:

Tiempo de llegada: 0

Tiempo de finalización: 1

Tiempo de respuesta = $1 - 0 = 1$

-Tarea B:

Tiempo de llegada: 5

Tiempo de finalización: 8

Tarea C:

Tiempo de llegada: 12

Tiempo de finalización: 20

Tiempo de respuesta = $20 - 12 = 8$.

Para que el sistema sea planificable, es necesario que se cumpla la siguiente condición:

$$\sum C_i/T_i \leq n(2^{1/n} - 1)$$

En este caso, $n = 3$. Reemplazando los valores de las tareas en la fórmula, obtenemos:

$$(1/5) + (3/7) + (3/20) \leq 3(2^{1/3} - 1)$$

$$0.56285714 \leq 0.77976314$$

Como la condición se cumple, podemos decir que el sistema es planificable.

B.) Tres tareas, A, B y C, en el ejemplo dado, comparten dos semáforos, S1 y S2, que están presentes en el caso. El IPCP establece que los límites máximos de prioridad de los semáforos deben coincidir con la prioridad más alta de cualquier tarea que pueda interferir con el semáforo. En consecuencia, estos son los techos de prioridad semafórica:.

Límite máximo de prioridad del semáforo S1: dado que la tarea A es la prioridad más alta de las tres tareas y utiliza el semáforo S1, el semáforo S1 tiene una prioridad máxima de 1.

Límite máximo de prioridad del semáforo S2: la tarea B utiliza el semáforo S2 y es la más importante de las tres tareas, por lo que su prioridad es igual al valor máximo del semáforo S2, que es 2.

C.) Las prioridades están asignadas de la siguiente manera:

Tarea A: prioridad 1

Tarea B: prioridad 2

Tarea C: prioridad 3

Por lo tanto, las tareas A y C pueden ser bloqueadas por la tarea B, ya que tiene una prioridad más alta. El intervalo de bloqueo de la tarea A es de 3 segundos, que es el tiempo que la tarea B utiliza el semáforo S2. El intervalo de bloqueo de la tarea C es de 3 segundos, que es el tiempo que la tarea B utiliza el semáforo S2 después de que la tarea C adquiere el semáforo S1. Los intervalos de bloqueo para cada tarea son los siguientes:

Tarea A: 3 segundos

Tarea B: no se bloquea

Tarea C: 3 segundos