Sistemas Operativos y Distribuidos

Práctico 5
Interbloqueos

5.1. Describir qué debería suceder para que un determinado conjunto de procesos alcance el estado de interbloqueo (*deadlock*).

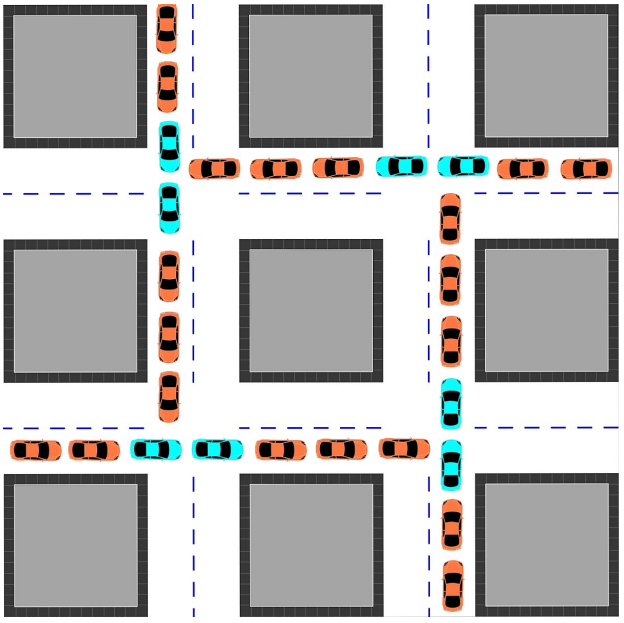
5.2. Describir las cuatro condiciones que se deben cumplir simultáneamente en un sistema para que se de un interbloqueo.

5.3. ¿Cuáles son los tres métodos generales que pueden implementarse para manejar interbloqueos?

5.4. ¿Cuál es la diferencia entre prevención de interbloqueos y evasión de interbloqueos?

5.5. Describir como un estado seguro garantiza que se puede evitar un interbloqueo.

5.6. El término *gridlock* alude a un problema de tráfico en el cual todo vehículo participante es incapaz de moverse.



a) Mostrar que en este caso se verifican las cuatro condiciones necesarias para la ocurrencia de un interbloqueo.
b) Describir una regla simple que permita evitar los interbloqueos en este problema.

5.7. ¿Es posible que exista un interbloqueo que involucre a un proceso con un solo hilo?

5.8. Considerar un sistema que posee doce recursos, asignados a los procesos P_0, P_1 y P_2 según se indica en la siguiente tabla:

Procesos	Máxima Necesidad	Mantiene
P_0	10	5
P_1	4	2
P_2	9	2

a) Partiendo del estado en el que se encuentra el sistema, determinar si los procesos se encuentran interbloqueados.

5.9. Considerar el siguiente estado de un sistema en un determinado instante de tiempo:

Procesos	Asignación				Máximo				Disponible			
	A	*B*	*C*	*D*	*A*	*B*	*C*	*D*	*A*	*B*	*C*	*D*
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

a) Escribir la tabla total de recursos.
b) Escribir la tabla de necesidad.
c) Determinar si el sistema se encuentra en un estado seguro.
d) Suponer que el proceso P_1 solicita (0,4,2,0). ¿Puede concederse inmediatamente dicha solicitud?

5.10. Considerar el siguiente estado de un sistema en un determinado instante de tiempo:

Procesos	Asignación			Máximo			Disponible		
	A	*B*	*C*	*A*	*B*	*C*	*A*	*B*	*C*
P_0	0	1	0	7	5	3	3	3	2
P_1	2	0	0	3	2	2			
P_2	3	0	2	9	0	2			
P_3	2	1	1	2	2	2			
P_4	0	0	2	4	3	3			

a) Escribir la tabla de necesidad.
b) Determinar si el sistema se encuentra en un estado seguro.
c) Suponer que el proceso P_1 solicita (1,0,2). ¿Puede concederse inmediatamente dicha solicitud?

5.11. A partir de los datos del ejercicio anterior y utilizando el algoritmo del banquero calcular el valor mínimo de *x* e *y* correspondiente a recursos de tipo *A* y *B* en disponible, de modo que el sistema esté libre de *deadlock*.

Disponible		
A	*B*	*C*
x	***y***	2

5.12. Considerar un sistema que posee 150 bloques de memoria, asignados de acuerdo a la siguiente tabla:

Procesos	Max	Mantiene
P_0	70	45
P_1	60	40
P_2	60	15

Aplicar el algoritmo del banquero con el objetivo de determinar si resulta seguro garantizar cada uno de los siguientes requerimientos:

a) Arriba un cuarto proceso, con una necesidad máxima de 60 bloques de memoria y una necesidad inicial de 25 bloques de memoria.
b) Arriba un cuarto proceso, con una necesidad máxima de 60 bloques de memoria y una necesidad inicial de 35 bloques de memoria.

5.13. Evaluar si la implementación del algoritmo del banquero resultaría útil en un sistema operativo real. Justificar adecuadamente.

5.14. Nombrar tres cuestiones que deben abordarse si se utiliza apropiación de recursos para la recuperación de un interbloqueo.

5.15. Explique las diferencias entre *deadlock*, *livelock*, y *starvation*.