

## Trabajo Práctico N° 4: Deadlocks y programación concurrente

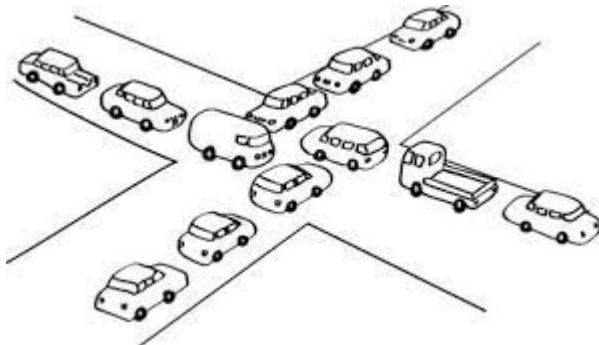
1. ¿Qué se entiende por deadlock? Enumere las condiciones para que ocurra. De un ejemplo de dead-lock fuera de los sistemas computacionales y analícelo/relaciónelo con las condiciones anteriores.

Por deadlock se entiende cuando un proceso se queda esperando por otro procesos y este otro proceso espera por el primero ya sea porque compiten por recursos del sistema o bien se comunican entre ellos (pueden ser dos como en este ejemplo o muchos más procesos).

Un deadlock ocurre si se cumplen las siguientes condiciones:

1. Exclusión mutua: sólo un proceso a la vez puede usar un recurso.
2. Mantener y esperar: un proceso que contiene al menos un recurso está esperando para adquirir recursos adicionales en poder de otros procesos.
3. Sin desalojo: un recurso puede ser liberado solo voluntariamente por el proceso que lo posee, después de que ese proceso haya completado su tarea.
4. Espera circular: existe un conjunto  $\{P_0, P_1, \dots, P_n\}$  de procesos en espera tal que  $P_0$  está esperando un recurso que está en manos de  $P_1$ ,  $P_1$  está esperando un recurso que está en manos de  $P_2$ , ...,  $P_{n-1}$  está esperando un recurso que está en manos de  $P_n$ .

Ejemplo de deadlock fuera de los sistemas computacionales:



2. Responda las siguientes preguntas:

(a) Una computadora tiene 6 drives de cinta, con  $n$  procesos compitiendo por ellos. Cada proceso puede necesitar dos drives. ¿Para qué valor de  $n$  el sistema está libre de deadlocks? Relacione el caso para  $n+1$  con las condiciones de deadlock.

(b) Considere un sistema con 4 recursos del mismo tipo compartidos por tres procesos, cada uno de los cuales requiere a lo sumo dos recursos. ¿En qué estado se encuentra el sistema? ¿Por qué?

(a) Si tenemos  $n$  procesos y tenemos  $n+1$  drives de cinta no hay deadlock ya que al menos un proceso va a poder ejecutar normalmente e ir liberando recursos para los demás. Con un  $n=5$ . Si tuviéramos 6 procesos los 6 podrían tomar un drive de cinta de a uno por vez y ser interrumpidos. Cuando ya no hay más drives de cinta para tomar y si todos los procesos necesitaban 2 drives de cintas para poder ejecutarse se produce un deadlock ya que ninguno va a poder finalizar.

- Exclusión mutua: Al competir por los recursos el proceso que llega primero hace uso del drive de cinta y lo reserva, por lo tanto hay exclusión mutua. Por lo tanto cada uno de los 6 procesos en este caso reserva el drive de cinta que va a usar.
- Hold and wait: Suponiendo que agarra un recurso y después mira si hay otro se da hold and wait porque un proceso agarra un drive de cinta y le falta agarrar el segundo para que pueda terminar con su ejecución.
- Sin desalojo: El drive de cinta va a ser liberado cuando el proceso haya finalizado su ejecución.
- Circular Wait / Espera cíclica: En el caso de  $n+1$  cada proceso que tomó el recurso del drive de cinta está esperando a que otro proceso libere el recurso para poder finalizar, por lo tanto, los 6 procesos se están esperando los unos a los otros en una espera cíclica.

(b) Se encuentra en un estado libre de deadlocks ya que por lo menos uno de los procesos va a poder ejecutarse con normalidad cuando tenga los 2 recursos y, de esta manera, va a poder ir liberando los recursos para los demás procesos.

3. En un sistema bancario existen cientos de procesos idénticos trabajando como sigue: cada proceso lee una línea de entrada especificando una suma de dinero, la cuenta para ser acreditada y la cuenta para ser debitada. Luego bloquea ambas cuentas y transfiere el dinero, liberando el bloqueo cuando termina. ¿Puede existir deadlock? ¿Por qué?. En caso afirmativo, describa posibles esquemas que eviten el mismo.

4. Considere un sistema en el siguiente estado:

	Asignación				Max				Disponible			
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
$P_0$	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	3	2	3	5	6				
$P_3$	0	6	3	2	0	6	5	2				
$P_4$	0	0	1	4	0	6	5	6				

Conteste las siguientes preguntas utilizando el algoritmo del banquero:

(a) ¿Cuál es el contenido de la matriz Need?

(b) ¿El sistema está en un estado seguro?

(c) Si llega un requerimiento de  $P_1$  para (0, 4, 2, 0) ¿puede ser otorgado inmediatamente?