

1. ¿Qué se entiende por Deadlock y Livelock? De un ejemplo de cada uno fuera de los sistemas computacionales y analícelo.

DeadLock :
Son dos o mas procesos/hilos incapaces de realizar cualquier acción, ya que uno de los Procesos esta esperando que el otro realice una accion y viceversa. Ya que los procesos estan esperando los procesos que necesitan para continuar y siguen en ejecución, impidiendo el avance de la ejecución.

- **Ejemplo:** Dos Carpinteros que estan trabajando en una casa, uno se lleva los clavos y el otro el martillo, como ninguno de los quiere soltar el elemento que tienen porque lo necesita, no terminan de realizar sus trabajos quedando estancados.

Livelock:
Son dos o mas procesos/hilos que dependen del estado del otro para su progreso, llegado al punto terminan bloqueándose esperando la accion del otro.

- **Ejemplo:** Dos personas en sillas de ruedas intentan pasar por un pasillo angosto pero como solo entra uno, uno de ellos esta comenzando a pasar pero en la otra punta ve a la otra también esta intentando pasar asi que la deja, pero el de la otra punta ve que justo el otro esta intentando pasar realiza la misma accion y nunca terminan pasando ninguno de los dos, bloqueándose mutuamente.

2. En un sistema conviven 3 procesos y 2 recursos. Uno de los recursos (R2) es de uso exclusivo y el otro (R1) puede ser compartido por hasta dos procesos.

a) ¿Puede haber deadlock o livelock?

b) ¿Y si ahora R1 puede ser compartido por hasta 3 procesos?

a) Pueden suceder los dos caso

- En el caso de (R2) habría un LiveLock ya que si el planificador no coordina bien el tiempo del paso de los Procesos/Hilos al recurso exclusivo terminan nunca entrando, ya que si dos procesos quieren entrar a este mismo terminan bloqueados.
- En el caso de (R1) si uno de los Procesos/Hilos ocupa el Recurso impide que el otro Proceso/Hilo pueda acceder.

b) Si puede ser compartido por 3 Procesos/Hilos pero dependen del Planificador haga bien su trabajo para que todos los Procesos/hilos puedan acceder en los tiempos correctos al recuso sin que se bloqueen mutuamente aplicando Exclusión Mutua.

3. 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? ¿Porqué?

En el caso que el Sistema no tenga bien asignada la Planificación de los Recursos se encontraría bloqueado, ya que los Procesos/Hilos estarían en un carrera (**RaceCondition**) por lo Recursos, los Procesos/Hilos se encontrarían con la situación que dos de ellos obtuvieron solo uno de Recursos y esperan que el otro Proceso que lo tiene, lo deje (**DeadLock**) y que dos Procesos/Hilos intenten acceder al mismo Recurso (**LiveLock**).

4. En un sistema hay tres procesos (P1, P2 y P3) y tres recursos (R1, R2 y R3). Los tres recursos son de uso exclusivo. Se sabe que P1 requiere los tres recursos, P2 requiere de R1 y R2 y P3 solo requiere R3.

a) ¿El sistema está libre de deadlock?

b) ¿P3 influye en que el sistema está o no libre deadlock?

c) Si me aseguro que P2 no podrá pedir ningún recurso hasta que P1 haya liberado todos sus recursos. ¿El sistema está libre de deadlock? ¿Porque?

a) El sistema no estaría libre.

b) No, porque no elimina la situacion de P1 y P2

c) No estaría libre, porque el P3 y el P1 depende del R3 y si en el hipotético caso que P3 allá accedido antes que el P1, este quedaría bloqueado hasta que P3 lo libere o viceversa.

RECURSOS		ESPERA											
R1 - R2 - R3				1	Proceso1()			1	Proceso2()			1	Proceso3()
R2 - R3				2	R1			2				2	
R3				3				3	R2			3	
				4				4				4	R3
				5				5				5	SC
R3		P1		6	BLOQUERADO			6				6	
		P1 - P2		7				7	BLOQUERADO			7	
				8				8				8	TEMINADO
		P1-P2		9	R1-R3			9				9	
		P1-P2		10				10	BLOQUERADO			10	
		P1-P2		11	BLOQUERADO			11				11	
				12				12				12	
				13				13				13	

5. Considere el siguiente programa (pseudocódigo). Note que el scheduler irá ejecutando estos procesos de manera concurrente, mezclando la ejecución P1 y P2.

int x = 10;	consola	1	Proceso1()	1	Proceso2()
		2	while(true)	2	
		3		3	while(true)
x = 9		4	x--	4	
x = 10		5	x++	5	
x = 9		6		6	x--
x = 9		7	if(x != 10)	7	x++
x=10		8		8	
	"10"	9	IMPRIME	9	if(x != 10)
		10		10	
		11	x--	11	
		12	x++	12	

6. Considera que tiene un programa con dos procesos. Un proceso se encarga de imprimir y el otro se encarga de introducir ficheros para su impresión.

a) ¿Presenta algún problema el código anterior si se ejecutara de manera concurrente? Analice el código.

b) Si la respuesta es negativa, explique el porqué. Si la respuesta es positiva, muestre la secuencia donde se genera el problema (Trace del programa).

a) Si presenta problemas.

b) Secuencia donde se encuentra el error.

VARIABLE	PROCESO IMPRESORA	VALOR R/ ESTADOS	PROCESO FICHERO	VALOR R/ ESTADOS
int peticiones = 0;	0 R1		0	
	1		1 R2	
	2 while		2	
	3		3 while	
	4 if (peticion>0)	BLOQUEADO	4	
	5		5 InsertarFicheroCola()	
	6 if (peticion>0)	BLOQUEADO	6	
	7		7 R2 = peticiones	
	8 if (peticion>0)	BLOQUEADO	8	
	9		9 R2 = R2 +1	R2 = 1
peticiones = 1	10 if (peticion>0)	BLOQUEADO	10	
	11		11 peticiones = R2	
	12 if (peticion>0)	SI	12	
	13		13 InsertarFicheroCola()	
	14 extraerColaPeticion()		14	
	15		15 R2 = peticiones	R2 = 1
	16 R1 = peticiones	R1 = 1	16	
	17		17 R2 = R2 +1	
	18 R1 = R1 - 1	R1 = 0	18	
peticiones = 2	19		19 peticiones = R2	
peticiones = 0	20 peticiones = R1		20	
	21		21 InsertarFicheroCola()	
	22 if (peticion>0)	BLOQUEADO	22	
	23		23 R2 = peticiones	R2 = 0
	24 if (peticion>0)	BLOQUEADO	24	
	25		25 R2 = R2 +1	R2 = 1
	26 if (peticion>0)	BLOQUEADO	26	
peticiones = 1	27		27 peticiones = R2	
	28 if (peticion>0)	SI	28	
	29		29 InsertarFicheroCola()	
	30 extraerColaPeticion()		30	
	31		31 R2 = peticiones	R2 = 1
	32 R1 = peticiones	R1 = 1	32	
	33		33 R2 = R2 +1	R2 = 2
	34 R1 = R1 - 1	R1 = 0	34	
peticiones = 2	35		35 peticiones = R2	
peticiones = 0	36 peticiones = R1		36	