

Tutorial Session N°4: Deadlock

Objective: To understand the concept of deadlock in operating systems, identify the conditions under which it occurs, and explore the main techniques used for deadlock prevention, avoidance, detection, and recovery.

Exercise 1

Consider a system with 5 processes (P0, P1, P2, P3, P4) and 3 resource types (R0, R1, R2). The state of the system is:

MAX				ALLOCATION				AVAILABLE		
	R0	R1	R2		R0	R1	R2	R0	R1	R2
P0	7	5	3	P0	0	1	0	3	3	2
P1	3	2	2	P1	2	0	0			
P2	9	0	2	P2	3	0	2			
P3	2	2	2	P3	2	1	1			
P4	4	3	3	P4	0	0	2			

- How many instances are there for each resource in this system?
- Calculate the **NEED** matrix.
- Determine if the system is in a safe state using the **Banker's algorithm**.
- Assuming the system receives the following requests sequentially, determine at each moment whether the request can be granted or not (**Banker's algorithm**):
 - At time t1, P1 makes the following request: (1, 0, 2).
 - At time t2, P4 makes the following request: (3, 3, 0).
 - At time t3, P0 makes the following request: (0, 2, 0).

Exercise 2

Consider the following systems. Use the **Deadlock Detection Algorithm** to determine whether the processes in each system are in deadlock or not:

1) System A :

ALLOCATION					REQUEST					AVAILABLE			
	R0	R1	R2	R3		R0	R1	R2	R3	R0	R1	R2	R4
P0	1	0	1	0	P0	2	3	0	1	5	2	3	1
P1	2	0	0	1	P1	2	0	1	0				
P2	0	1	2	0	P2	2	1	4	0				

2) System B :

ALLOCATION				REQUEST				AVAILABLE		
	R0	R1	R2		R0	R1	R2	R0	R1	R2
P0	0	1	0	P0	0	0	0	0	0	0
P1	2	0	0	P1	2	0	2			
P2	3	0	2	P2	0	0	0			
P3	2	1	1	P3	1	0	0			
P4	0	0	2	P4	0	0	2			

GOOD LUCK