Tutorial Session N°5 : 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 3 processes P1, P2, P3, and 4 types of resources R1, R2, R3, R4, with a single instance for R1 and R3, 2 instances for R2, and 3 instances for R4. Consider the following resource allocation:

- P1 holds R2 and requests R1;
- P2 holds R1, R2 and requests R3;
- P3 holds R3 and 2 instances of R4, and requests R2;

Construct the resource allocation graph. Is there a deadlock? If so, which processes are involved?

Exercise 2

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

	R0	R1	R2
P0	7	5	3
P1	3	2	2
P2	9	0	2
P3	2	2	2
P4	4	3	3

ALLOCATION

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

AVAILABLE

R0	R1	R2
3	3	2

1. How many instances are there for each resource in this system?
2. Calculate the **NEED** matrix.
3. Determine if the system is in a safe state using the **Banker's algorithm**.
4. 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 3

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

	R0	R1	R2	R3
P0	1	0	1	0
P1	2	0	0	1
P2	0	1	2	0

REQUEST

	R0	R1	R2	R3
P0	2	3	0	1
P1	2	0	1	0
P2	2	1	4	0

AVAILABLE

R0	R1	R2	R4
5	2	3	1

2) System B :

ALLOCATION

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

REQUEST

	R0	R1	R2
P0	0	0	0
P1	2	0	2
P2	0	0	0
P3	1	0	0
P4	0	0	2

AVAILABLE

R0	R1	R2
0	0	0

GOOD LUCK