Câu Hỏi Về Deadlock

1. **What is a race condition? Give an example**
2. **What is deadlock? What is startvation? How do they differ from each other?**
3. **What are the four condtions required for deadlock to occur?**
4. **Describe four general strategies for dealing with deadlocks.**
5. **For single unit resources, we can model resource allocation and requests as a directed graph connecting processes and resources. Given such a graph, what is involved in deadlock detection.**
6. **Is the following system of four processes with 2 resources deadlocked?**

Current allocation matrix
P1 1 3
P2 4 1
P3 1 2
P4 2 0

Current request matrix
P1 1 2
P2 4 3
P3 1 7
P4 5 1

Availability Vector
1 4

If the availability vector is as below, is the system above still deadlocked?
2 3

7. **What must the banker's algorithm know a priori in order to prevent deadlock?**
8. **Describe the general strategy behind dealock prevention, and give an example of a practical deadlock prevention method.**
9. Describe four ways to *prevent* deadlock by attacking the conditions required for deadlock.
10. Answer the following questions about the tables.

    a. Compute what each process still might request and display in the columns labeled "still needs".
    b. Is the system in a safe or unsafe state? Why?
    c. Is the system deadlocked? Why or why not?
    d. Which processes, if any, are or may become deadlocked?
    e. Assume a request from p3 arrives for (0,1,0,0)
        1. Can the request be safely granted immediately?
        2. In what state (deadlocked, safe, unsafe) would immediately granting the request leave the system?
        3. Which processes, if any, are or may become deadlocked if the request is granted immediately?

available

r1 r2 r3 r4

2 1 0 0

	current allocation				**maximum demand**				**still needs**			
process	r1	r2	r3	r4	r1	r2	r3	r4	r1	r2	r3	r4
p1	0	0	1	2	0	0	1	2				
p2	2	0	0	0	2	7	5	0				
p3	0	0	3	4	6	6	5	6				
p4	2	3	5	4	4	3	5	6				
p5	0	3	3	2	0	6	5	2				

	current allocation				**maximum demand**				**still needs**			
process	r1	r2	r3	r4	r1	r2	r3	r4	r1	r2	r3	r4
p1	0	0	1	2	0	0	1	2	0	0	0	0
p2	2	0	0	0	2	7	5	0	0	7	5	0
p3	0	0	3	4	6	6	5	6	6	6	2	2
p4	2	3	5	4	4	3	5	6	2	0	0	2
p5	0	3	3	2	0	6	5	2	0	3	2	0

11. **When Is A System In Safe State?**