長庚大學109學年度第一學期 作業系統 第四次小考

系級: 姓名: 學號:

1. (40%) Banker's Algorithm is a deadlock avoidance algorithm. Assume that there are 5 processes {P_0, P_1, P_2, P_3, P_4} and three types of shared resources {A, B, C} in the system, and the details are in the following table. By Banker's Algorithm, is the system in a safe state? If your answer is yes, please provide a safe sequence. If your answer is no, please provide the reason.

	Allocation			Max			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	0	1	2	7	2	5	7	1	3	2	1	2
P1	2	0	0	4	1	4	2	1	4			
P2	3	0	2	7	1	4	4	1	2			
P3	2	1	1	2	3	8	0	2	7			
P4	0	0	2	2	1	4	2	1	2			

Answer:

Yes.

The safe sequence is: P4(need(2,1,2) ≤ available(2,1,2)) ➔ P1(need(2,1,4) ≤ available(2,1,4)) ➔ P2(need(4,1,2) ≤ available(4,1,4)) ➔ P0(need(7,1,3) ≤ available(7,1,6)) ➔ P3(need(0,2,7) ≤ available(7,2,8))

2. (30%) For the situation in question 1, P_4 further has a request (1, 1 , 0) to use 1 more instance of A and 1 more instance of B. Should the request be granted? Again, provide the reason to support your answer.

Answer:

Yes

(1) request(1, 1 , 0) ≤ P4 need(2, 1, 2)

(2) request(1, 1 , 0) ≤ system available(2, 1, 2)

(3) Try to grant the request: P4 need(2,1,2) ➔ need(1,0,2), P4 allocation(0,0,2) ➔ allocation(1,1,2), available(2,1,2) ➔ available(1,0,2).

The safe sequence is: P4(need(1,0,2) ≤ available(1,0,2)) ➔ P1(need(2,1,4) ≤ available(2,1,4)) ➔ P2(need(4,1,2) ≤ available(4,1,4)) ➔ P0(need(7,1,3) ≤ available(7,1,6)) ➔ P3(need(0,2,7) ≤ available(7,2,8))

3. (30%) For another system state in the following table, by Banker's Algorithm, is the system in a safe state? If your answer is yes, please provide a safe sequence. If your answer is no, please provide the reason.

	Allocation			Max			Need			Available		
	A	B	C	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3	3	3	2
P1	1	0	1	2	4	3	1	4	2			
P2	3	0	2	9	0	2	6	0	0			
P3	0	1	1	0	2	2	0	1	1			
P4	2	1	1	6	4	2	4	3	1			

Answer:

Yes.

The safe sequence is: P3(need(0,1,1) ≤ available(3,3,2)) ➔ P1(need(1,4,2) ≤ available(3,4,3)) ➔ P4(need(4,3,1) ≤ available(4,4,4)) ➔ P2(need(6,0,0) ≤ available(6,5,5)) ➔ P0(need(7,4,3) ≤ available(9,5,7))