Southern Methodist University
Bobby B. Lyle School of Engineering Department of Computer Science
Homework 6

Operating System and Software System

Name: Bingying Liang
ID: 48999397
Email: bingyingl@smu.edu

CS7343 Distance

Apr 29 2023

- **All students who are signed for this course at the CS 7343 must answer both questions.**
- **All students who are signed for this course at the CS 5343 must answer exactly one question.**

1. Consider the following snapshot of a system (P=Process, R=Resource) :

Available			
RA	RB	RC	RD
8	5	9	7

Maximum Demand				
	RA	RB	RC	RD
P0	3	2	1	4
P1	0	2	5	2
P2	5	1	0	5
P3	1	5	3	0
P4	3	0	3	3

Current Allocation				
	RA	RB	RC	RD
P0	1	0	1	1
P1	0	1	2	1
P2	4	0	0	3
P3	1	2	1	0
P4	1	0	3	0

Answer the following questions using banker's algorithm:

(a) Calculate the Needs matrix:

Solution:

Needs				
	RA	RB	RC	RD
P0	2	2	0	3
P1	0	1	3	1
P2	1	1	0	2
P3	0	3	2	0
P4	2	0	0	3

(b) Is the system in a safe state? If so, show a safe order in which the processes can run.

Solution:
Yes. Because the available after allocation is in the following as the question mentioned which are satisfy the processes.

Available			
RA	RB	RC	RD
8	5	9	7

For this moment it satisfy the P0, P1, P2, P3, P4 needs. Then we can go P0 first.

- P0 allocation and finish, the available:

Available			
RA	RB	RC	RD
9	5	10	8

- P1 allocation and finish, the available:

Available			
RA	RB	RC	RD
9	6	12	9

- P2 allocation and finish, the available:

Available			
RA	RB	RC	RD
13	6	12	12

- P3 allocation and finish, the available:

Available			
RA	RB	RC	RD
14	8	13	12

- P4 allocation and finish, the available:

Available			
RA	RB	RC	RD
15	8	16	12

All processes finshes.

Therefore, A safe order can be P0, P1, P2, P3, P4.

(c) Can a request of one instance of RA by Process P0 be granted safely according to Banker's algorithm? Why/Why not?

Solution: Yes.
When one instance of RA by process P0 be granted safely will become in the following:

Needs				
	RA	RB	RC	RD
P0	1	2	0	3
P1	0	1	3	1
P2	1	1	0	2
P3	0	3	2	0
P4	2	0	0	3

Current Allocation				
	RA	RB	RC	RD
P0	2	0	1	1
P1	0	1	2	1
P2	4	0	0	3
P3	1	2	1	0
P4	1	0	3	0

Available			
RA	RB	RC	RD
7	5	9	7

For this moment it satisfy the P0, P1, P2, P3, P4 needs. Then we can go P0 first.

- P0 allocation and finish, the available:

Available			
RA	RB	RC	RD
9	5	10	8

- P1 allocation and finish, the available:

Available			
RA	RB	RC	RD
9	6	12	9

- P2 allocation and finish, the available:

Available			
RA	RB	RC	RD
13	6	12	12

- P3 allocation and finish, the available:

Available			
RA	RB	RC	RD
14	8	13	12

- P4 allocation and finish, the available:

Available			
RA	RB	RC	RD
15	8	16	12

Therefore, a request of one instance of RA by Process P0 can be granted safely according to Banker's algorithm.

2. At an instant, the resource allocation state in a system is as follows: 4 processes P1–P4
   4 resource types: R1–R4
   R1 (5 instances), R2 (3 instances), R3 (3 instances), R4 (3 instance)
   Snapshot at time T0:

	Allocation				Request				Available			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P1	0	0	1	0	2	0	0	2	2	1	1	2
P2	2	0	0	1	1	3	0	1				
P3	0	1	1	0	2	1	1	0				
P4	1	1	0	0	4	0	3	1				

Run the deadlock detection algorithm and test whether the system is deadlocked or not. If it is, identify the processes that are deadlocked.

Solution:

(1) Mark each process that has a row in the Allocation matrix of all zeros. Mark = (0, 0 , 0 , 0)

(2) Initialize a temporary vector W to equal the Available vector. W = (2, 1, 1, 2)

(3)
- The request of process P1 (2, 0, 0, 2) is less than or equal to W, so mark P1 Mark = (1, 0, 0, 0) and set W = W +(0, 0, 1, 0) = (2, 1, 2, 2).
- The request of process P3 (2, 1, 1, 0) is less than or equal to W, so mark P3 Mark = (1,0,1,0) and set W = W + (0, 1, 1, 0) = (2, 2, 3, 2).

(4) P2, P4 unmarked. P2 request (1,3,0,1), P4 request (4, 0, 3, 1) which are more than W. Therefore, terminate the algorithm.

The algorithm concludes with P2 and P4 unmarked, indicating that these processes are deadlocked.