CSC 112: Computer Operating Systems
Lecture 4

Deadlocks Exercises

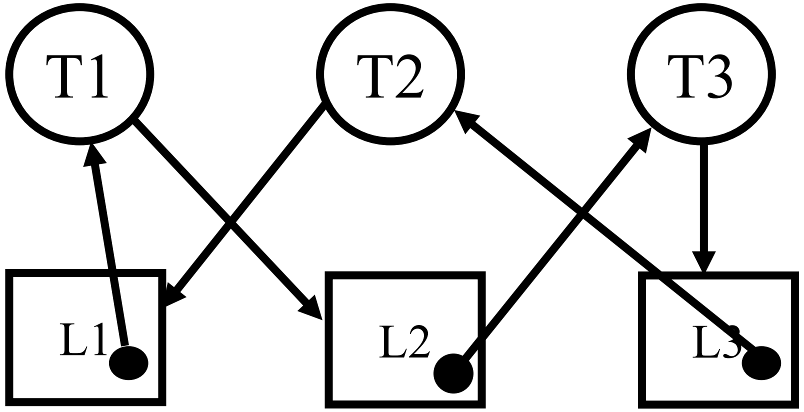
Department of Computer Science,

Hofstra University

Acknowledgement: Lecture slides based on UC Berkeley CS 162: Operating Systems and System Programming

Quiz: Deadlocks II

- Is there a possible deadlock?
- Yes, there is a deadlock. Consider the following interleaving:
  - Thread T1 executes L1.wait(); no blocking
  - Thread T2 executes L3.wait(); no blocking
  - Thread T3 executes L2.wait(); no blocking
- Now there is a circular wait condition:
- T1 waiting for L2 (held by T3) → T2 waiting for L1 (held by T1) → T3 waiting for L3 (held by T2).
- Solution: each thread should acquire locks in the same order, say, L1, L2, L3.




```
Semaphore L1=1, L2=1, L3=1;

// Thread T1:
L1.wait();
L2.wait();
// critical section requiring L1 and L2 locked.
L2.post();
L1.post();

// Thread T2:
L3.wait();
L1.wait();
// critical section requiring L3 and L1 locked.
L1.post();
L3.post();

// Thread T3:
L2.wait();
L3.wait();
// critical section requiring L2 and L3 locked.
L3.post();
L2.post();
```

Quiz: Banker's Algorithm I

- 4 processes P1 through P5; 3 resource types R1, R2, R3 with 7, 3, 6 instances each.
- Run Banker's algorithm to check if the current state is safe. If yes, give a safe sequence of process completions and fill in the table with the sequence of process completions without deadlock, and available resources after the completion of each process.
- (You will be graded on "Need matrix", and "Available resources after completion of each process".)

Max

$$R = \begin{bmatrix} 7 & 5 & 3 \\ 3 & 2 & 2 \\ 9 & 0 & 2 \\ 2 & 2 & 2 \\ 4 & 3 & 3 \end{bmatrix}$$

Allocation

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

Total

$$E = \begin{bmatrix} 7 & 3 & 6 \end{bmatrix}$$

Available resources after completion of each process

	R1	R2	R3
Init			

Quiz Solution: Banker's Algorithm I

- First compute the Need matrix as *Max – Allocation,* and Available vector *A.*
- The state is not safe, the execution sequence P4, P2 leads to a deadlock state, where none of the remaining processes P1, P3, P5 can finish.

Max

$$R = \begin{bmatrix} 7 & 5 & 3 \\ 3 & 2 & 2 \\ 9 & 0 & 2 \\ 2 & 2 & 2 \\ 4 & 3 & 3 \end{bmatrix}$$

Allocation

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \\ 3 & 0 & 2 \\ 2 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

Total

$$E = [7 \quad 3 \quad 6]$$

Available

$$A = [0 \quad 1 \quad 1]$$

Need

$$R - C = \begin{bmatrix} 7 & 4 & 3 \\ 1 & 2 & 2 \\ 6 & 0 & 0 \\ 0 & 1 & 1 \\ 4 & 3 & 1 \end{bmatrix}$$

Available resources after completion of each process

	R1	R2	R3
Init	0	1	1
P4	2	2	2
P2	4	2	2
	Deadlock		

Your task

Quiz: Banker's algorithm II

- 4 processes P1, P2, P3; 3 resource types R1, R2, R3 with 8, 6, 4 instances each.
- 1) Run Banker's algorithm to check if the current state is safe. If yes, give a safe sequence of process completions and fill in the table with the sequence of process completions without deadlock, and available resources after the completion of each process.
- 2) Starting from the initial state, if P1 makes request for 2 more instances of resource 3, should we grant it?
- 3) Starting from the initial state, if P2 makes request for 2 more instances of resource 1, should we grant it?

Max

$$\begin{bmatrix} 8 & 4 & 3 \\ 6 & 2 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

Allocation

$$\begin{bmatrix} 0 & 0 & 1 \\ 3 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

Total

$$E = \begin{bmatrix} 8 & 6 & 4 \end{bmatrix}$$

Available resources after completion of each process

	R1	R2	R3
Init			

Quiz Solution: Banker's algorithm II

- The initial state is safe, with safe sequences of P2, P3, P1 or P2, P2, P1

Max

$$\begin{bmatrix} 8 & 4 & 3 \\ 6 & 2 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

Allocation

$$\begin{bmatrix} 0 & 0 & 1 \\ 3 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

Need

$$\begin{bmatrix} 8 & 4 & 2 \\ 3 & 0 & 0 \\ 1 & 1 & 2 \end{bmatrix}$$

Total

$$E = \begin{bmatrix} 8 & 6 & 4 \end{bmatrix}$$

Available

$$E = \begin{bmatrix} 3 & 2 & 2 \end{bmatrix}$$

Available resources after completion of each process

	R1	R2	R3
Init	3	2	2
P2	6	4	2
P3	8	6	3
P1	8	6	4

Or

Available resources after completion of each process

	R1	R2	R3
Init	3	2	2
P3	5	4	3
P2	8	6	3
P1	8	6	4

Quiz Solution: Banker's algorithm II

- Starting from the initial state, if P1 makes request for 2 more instances of resource 3, then we calculate the state of the system if this request is fulfilled.
- The state is unsafe, so we deny this request.

Max

$$\begin{bmatrix} 8 & 4 & 3 \\ 6 & 2 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

Allocation

$$\begin{bmatrix} 0 & 0 & 3 \\ 3 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

Need

$$\begin{bmatrix} 8 & 4 & 0 \\ 3 & 0 & 0 \\ 1 & 1 & 2 \end{bmatrix}$$

Total

$$E = \begin{bmatrix} 8 & 6 & 4 \end{bmatrix}$$

Available

$$E = \begin{bmatrix} 3 & 2 & 0 \end{bmatrix}$$

Available resources after completion of each process

	R1	R2	R3
Init	3	2	0
P2	6	4	0
	Deadlocked		

Quiz Solution: Banker's algorithm II

- Starting from the initial state, if P2 makes request for 2 more instances of resource 1, then we calculate the state of the system if this request is fulfilled.
- The state is safe, with safe sequences of P2, P3, P1 or P2, P2, P1, so we can grant this request.

Max

$$\begin{bmatrix} 8 & 4 & 3 \\ 6 & 2 & 0 \\ 3 & 3 & 3 \end{bmatrix}$$

Allocation

$$\begin{bmatrix} 0 & 0 & 1 \\ 5 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix}$$

Need

$$\begin{bmatrix} 8 & 4 & 2 \\ 1 & 0 & 0 \\ 1 & 1 & 2 \end{bmatrix}$$

Total

$$E = [8 \quad 6 \quad 4]$$

Available

$$E = [1 \quad 2 \quad 2]$$

Available resources after completion of each process

	R1	R2	R3
Init	1	2	2
P2	6	4	2
P3	8	6	3
P1	8	6	4

Or

Available resources after completion of each process

	R1	R2	R3
Init	1	2	2
P3	3	4	3
P2	8	6	3
P1	8	6	4

Banker's Algorithm: 4 philosophers each holding his left fork

Max

$$R = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Allocation

$$C = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Need

$$R - C = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Total

$$E = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \end{vmatrix}$$

Available

$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Suppose we have 5 philosophers P1-P5, and 5 forks R1-R5; philosopher Pi has left fork Ri, and right fork R(i+1)%5. Philosophers P1-P4 each is holding his left fork.

Run Banker's algorithm to check if the current state is safe. If yes, give a safe sequence of process completions and fill in the table with the sequence of process completions without deadlock, and available resources after the completion of each process.

Available resources after completion of each process

	R1	R2	R3	R4	R5
	0	0	0	0	1

Banker's Algorithm: 4 philosophers each holding his left fork ANS

- Yes, current state is safe, and the only safe sequence is P4, P3, P2, P1, P5

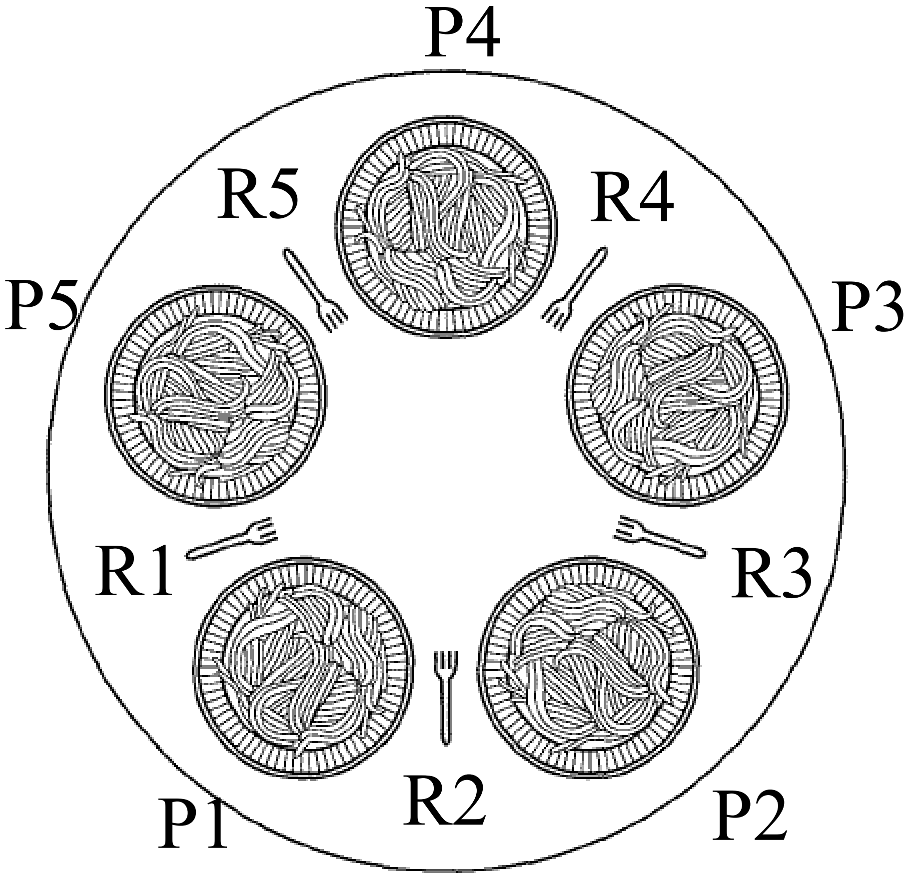


Figure 2-44. Lunch time in the Philosophy Department.

Available resources after completion of each process

	R1	R2	R3	R4	R5
	0	0	0	0	1
P4	0	0	0	1	1
P3	0	0	1	1	1
P2	0	1	1	1	1
P1	1	1	1	1	1
P5	1	1	1	1	1

Banker's Algorithm: 5 philosophers each holding his left fork

Max | Allocation | Need

$$R = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{vmatrix} \quad C = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \boxed{1} \end{vmatrix} \quad C = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Total | Available

$$E = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \end{vmatrix} \qquad A = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Run Banker's algorithm to check if the current state is safe.

ANS: It is not safe, as no process can run to completion based on Need matrix and Available vector.