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

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

Multi-Armed Lawyers

- Consider a large table with IDENTICAL multi-armed alien lawyers. In the center is a pile of chopsticks. In order to eat, a lawyer must have one chopstick in each hand. The lawyers are so busy talking that they can only grab one chopstick at a time. Design a deadlock-free algorithm using monitors and Bankers algorithm. Assume total number of chopsticks >= number of hands of each lawyer, so at least one lawyer can eat.
- It is not a generalization of the 2-armed Dining Philosophers problem. Since the chopsticks are in a pile at center of the table, we should model them as a single resource with multiple instances, instead of multiple resources for the Dining Philosophers, where each fork (chopstick) has a fixed position in-between two philosophers. Hence the R and C matrices have a single column.

Ack: this example is taken from UC Berkeley CS162 course.

Example: 5 Lawyers, each with 2 arms, 5 chopsticks

Total Request matrix R (NumArms=2)
Current allocation matrix C
Resources in existence E
Resources available A

$$R = \begin{vmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{vmatrix}, C = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}, E = |5|, A = |5|$$

- Initially, all chopsticks are free.

$$R = \begin{vmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{vmatrix}, C = \begin{vmatrix} 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{vmatrix}, E = |5|, A = |1|$$

- Two lawyers grab two chopsticks each and start eating. No other lawyers can eat.

Quiz: Dining Lawyers I

- Is it possible for the system to get into deadlock?

Quiz: Dining Lawyers I

- If each lawyer has 2 arms, and there is a pile of knives and forks at center of the table. Assume there are at least 1 knife and 1 fork, so at least one lawyer can eat. (There is no other constraint on the numbers of knives, forks, or lawyers.) Each lawyer follows the following steps:
- (1) Pick up a knife
- (2) Pick up a fork
- (3) Eat
- (4) Return the knife and fork to the pile
- Q: Can the system be deadlocked?

Quiz: Dining Lawyers II

- If each lawyer has 4 arms, and there is a pile of knives and forks at center of the table. Assume there are at least 2 knives and 2 forks, so at least one lawyer can eat. Each lawyer follows the following steps:
- (1) Pick up 2 knives atomically
- (2) Pick up 2 forks atomically
- (3) Eat
- (4) Return the knives and forks to the pile
- Q: Can the system be deadlocked?

Quiz: Dining Lawyers III

- If each lawyer has 4 arms, and there is a pile of knives and forks at center of the table. Assume there are at least 2 knives and 2 forks, so at least one lawyer can eat. Each lawyer follows the following steps:
- (1) Pick up a knife
- (2) Pick up another knife
- (3) Pick up a fork
- (4) Pick up another fork
- (5) Eat
- (6) Return the knife and fork to the pile
- Q1: Can the system be deadlocked?
- Q2: What if each lawyer may have a different number of arms, and may request a different ratio of knives vs. forks?

Quiz: Banker's algorithm

- 4 threads P0 through P4; 4 resource types with 10, 5, 6, 5 instances each.
- Current system state encoded in matrices *R, C* and vector *E.*

Total Request matrix

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

Current allocation matrix

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

Resources in existence

$$E = [10 \quad 5 \quad 7]$$

Resources available

$$A = [3 \quad 3 \quad 2]$$

Quiz: Deadlocks II

- Is there a possible deadlock?

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

// Thread 1:
L1.wait();
L2.wait();
// critical section requiring L1 and L2 locked.
L2.post();
L1.post();

// Thread 2:
L3.wait();
L1.wait();
// critical section requiring L3 and L1 locked.
L1.post();
L3.post();

// Thread 3:
L2.wait();
L3.wait();
// critical section requiring L2 and L3 locked.
L3.post();
L2.post();
```