CS444 HW3

Sawyer Bowerman

Prof. Deblois

03/26/2025

1) Hard Disk Drive Arm Motion Problem:

Starting at cylinder 51

q = [60,82,97,41,17,23,83]

a. FCFS (First Come First Served)

   - Start at 51, move to 60: |60-51| = 9 cylinders
   - 60 to 82: |82-60| = 22 cylinders
   - 82 to 97: |97-82| = 15 cylinders
   - 97 to 41: |41-97| = 56 cylinders
   - 41 to 17: |17-41| = 24 cylinders
   - 17 to 23: |23-17| = 6 cylinders
   - 23 to 83: |83-23| = 60 cylinders

   So, $9 + 22 + 15 + 56 + 24 + 6 + 60 = 192\ Cylinders$

b. SSF (Shortest Seek First)

   - Start at 51, closest is 60: |60-51| = 9 cylinders
   - 60, closest is 41: |41-60| = 19 cylinders
   - 41, closest is 23: |23-41| = 18 cylinders
   - 23, closest is 17: |17-23| = 6 cylinders
   - 17, closest is 82: |82-17| = 65 cylinders
   - 82, closest is 83: |83-82| = 1 cylinder
   - 83, closest is 97: |97-83| = 14 cylinders

   So, $9 + 19 + 18 + 6 + 65 + 1 + 14 = 132\ Cylinders$

c. Elevator Problem

   - Start at 51 moving upward, serve 60: |60-51| = 9 cylinders
   - Continue upward to 82: |82-60| = 22 cylinders
   - Continue upward to 83: |83-82| = 1 cylinder
   - Continue upward to 97: |97-83| = 14 cylinders
   - Reverse direction, serve 41: |41-97| = 56 cylinders
   - Continue downward to 23: |23-41| = 18 cylinders
   - Continue downward to 17: |17-23| = 6 cylinders

   So, $9 + 22 + 1 + 14 + 56 + 18 + 6 = 126\ Cylinders$

2) Avoiding Lost Printer Job
   Assuming before the list starts that slots 0-3 are empty and slots 4-6 are full:

   a)

   a. Process A reads nextFileToPrint = 4
   b. Process A is interrupted before it can print the file
   c. Process B reads nextFileToPrint = 4
   d. Process B prints the file at slot 4
   e. Process B increments nextFileToPrint to 5
   f. Process B reads nextFileToPrint = 5
   g. Process B prints the file at slot 5
   h. Process B increments nextFileToPrint to 6
   i. Process B reads nextFileToPrint = 6
   j. Process B prints the file at slot 6
   k. Process B increments nextFileToPrint to 7
   l. Process A resumes and tries to print file at slot 4, which has already been printed
   m. Process A increments nextFileToPrint to 5, which has already been printed
   n. The result is duplicated printing of files and process A never gets to print its intended job

   b)

   a. No two processes may be simultaneously inside their critical regions
   b. No assumptions may be made about speeds or number of CPUs
   c. No process running outside its critical region may block other processes
   d. No process should have to wait forever to enter its critical region

   c)

   a. Process 0 finishes its critical section and sets turn = 1
   b. Process 0 continues to non-critical section
   c. Process 0 quickly completes non-critical section and wants to enter critical section again
   d. Process 0 must wait for Process 1 to set turn = 0
   e. If Process 1 is not interested in entering its critical section, Process 0 remains blocked

   This violates condition 3 because Process 1, running outside its critical section, is blocking Process 0.

   d) Peterson Solution with Timeline

Peterson's solution uses:

1. **The interested array**: This is a boolean array where interested[i] indicates that process i wants to enter its critical section. Initially, all values are FALSE.
2. **The turn variable**: This indicates which process should yield if both processes want to enter their critical sections simultaneously. If both processes try to enter at the same time, the value of turn determines which one waits.

Here's how they work together:

- When process i wants to enter its critical section, it sets interested[i] = TRUE to express interest
- It then sets turn = i, which means "I'll wait if there's a conflict"
- The process then checks: if the other process is interested AND it's my turn to wait (turn == i), then wait
- When exiting the critical section, the process sets interested[i] = FALSE to indicate it's no longer interested

Process 0	Process 1
interested [0] = True	
turn = 0	
# check: other not interested	
# enter critical section	
	interested [1] = True
	turn = 1
	# check: Process 0 in critical section
# critical section works	# Process 1 waits
interested [0] = False	
	# check again process 0 done
	# enter critical section
# process 0 continues	# critical section works
	interested[1] = False

3) Resource Allocation Graph

(a) Are there blocked processes?

Yes, there are blocked processes:

- P1 is blocked because it is requesting a unit of R2, and both units of R2 are currently held by other processes
- P4 is blocked because it is requesting a unit of R3, which is currently held by P3

(b) Is the system in a deadlock?

No, the system is not in a deadlock. While P1 and P4 are blocked, there is no cycle of requests and allocations that prevents all processes from making progress. P2 and P3 can complete and release their resources, which would allow P1 and P4 to proceed.

(c) If the two units of R2 are allocated to P2 and P3, which processes will be blocked?

Processes that will be blocked:

- P1 will be blocked because it is requesting R2, which is fully allocated to P2 and P3
- P4 will remain blocked because it is requesting R3, which is held by P3

(d) If the two units of R2 are allocated to P1 and P2, which processes will be blocked?

Processes that will be blocked:

- P3 will be blocked because it is requesting R2, which is fully allocated to P1 and P2
- P4 will remain blocked because it is requesting R3, which is held by P3

This creates a deadlock between P3 and P4 since P3 is holding R3 which P4 needs, and P4 is holding R4 which P3 needs.

4) Queueing System

Goal: prove the expected number of customers N = r/(1-r) where r is the ratio of arrival rate to service rate.

We know:

$$p0 = 1 - r \ (for\ 0 < r < 1)$$
$$pk = p0 * r^k \ (for\ k > 0)$$

1. Express N in terms of definition of expectation

$$N = \sum_{k=0}^{\infty} (k * pk)$$

2. Sub out pk

$$N = \sum_{k=0}^{\infty} (k * p_0 * r^k)$$
$$N = p_0 \sum_{k=0}^{\infty} (k * r^k)$$

3. And $p_0 = 1 - r$:

$$N = (1 - r) \sum_{k=0}^{\infty} (k * r^k)$$

4.

$$\sum_{k=0}^{\infty} (k * r^k) = \frac{r}{(1 - r)^2}$$

5. Substitute into expression

$$N = 1 - r \left(\frac{r}{(1 - r)^2} \right)$$
$$and\ therefore,$$
$$N = \frac{r}{1 - r}$$