

SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Approved by AICTE, New Delhi, Affiliated to Anna University,
Chennai Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &
Accredited by NBA (B.E - CSE, EEE, ECE, Mech & B.Tech.IT)
COIMBATORE-641 035, TAMIL NADU



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

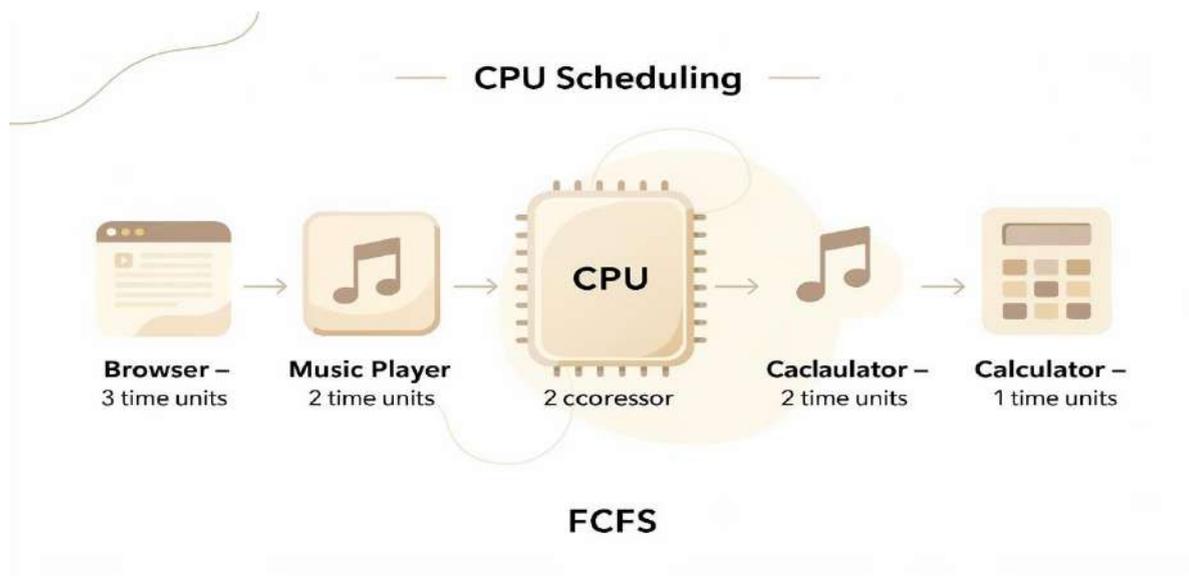
Course Code & Name : 23CST206-OPERATING SYSTEMS AND VIRTUALIZATION
Class : II CSE
Course Faculty : Ms.R.SWATHIRAMYA

Question Bank

UNIT-II PROCESS MANAGEMENT

🌟 PUZZLE 1: "Who Runs First?"

🔍 Problem Statement



Answer / Solution

Answer 1

Browser finishes last (needs 3 time units, runs first under FCFS)

Answer 2

OS switches between programs to ensure fair CPU time allocation and enable multitasking.

Puzzle 2 : “The Confused Computer Lab”

In a computer lab, **three processes** are running at the same time:

- **Process A** – A student typing code
- **Process B** – A student playing music
- **Process C** – A student printing notes

There is **only one CPU** and **one printer**.

What happens in the system:

1. The CPU starts executing **Process A**.
2. Suddenly, **Process B** needs immediate CPU time to keep the music smooth.
3. While this is happening, **Process C** sends data to the printer.
4. At the same time, **Process A** and **Process C** both try to update a shared log file.

After a few seconds:

- The music plays smoothly
- Printing happens one job at a time
- The shared log file is not corrupted
- All three processes eventually finish successfully

Puzzle Solution

- The CPU switched from Process A to Process B using **preemptive scheduling** to keep the system responsive.
- Printing occurred one job at a time because the printer was a **shared resource protected by synchronization**.
- The shared log file was protected using **IPC with mutual exclusion**, preventing data corruption.

Puzzle 3:

THE OS JUGGLER



THE CHALLENGE:

1. Minimize Total Waiting Time
2. Ensure Fairness (No Starvation)

YOUR TASK:

Decide the Juggling Order! **?**
Which CPU Scheduling Strategy?

FCFS SJF Priority Round Robin

Solution:

Goal: Minimize total waiting time → shortest jobs first.

Given execution times:

- P1 = 10, P2 = 5, P3 = 8, P4 = 6, P5 = 2

Step 1: Sort processes by execution time (Shortest Job First – SJF):

- P5 (2), P2 (5), P4 (6), P3 (8), P1 (10)

Step 2: Order of execution:

- CPU executes: P5 → P2 → P4 → P3 → P1

Step 3: Waiting time calculation (optional):

- P5 waits 0
- P2 waits 2 (P5's time)
- P4 waits 2 + 5 = 7

- P3 waits $7 + 6 = 13$
- P1 waits $13 + 8 = 21$

Result:

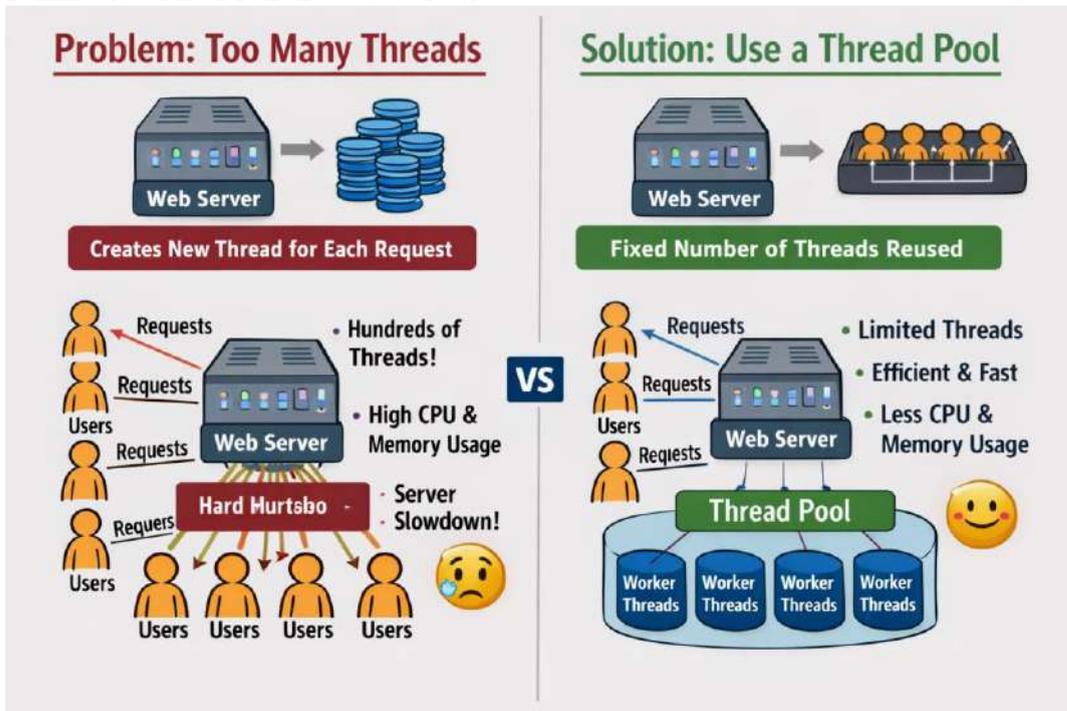
- Total waiting time = $0 + 2 + 7 + 13 + 21 = 43$ units
- This is the minimum total waiting time compared to any other order.

Explanation:

- Using SJF (Shortest Job First) ensures that shorter processes finish earlier, reducing the overall waiting time for all processes.

This is analogous to the juggler throwing the shortest-duration balls first, keeping all balls moving efficiently and preventing long delays.

Puzzle 4: “The Frozen Web Server”



🧠 Puzzle Question:

What threading issue is causing the slowdown, and how can it be solved without reducing the number of users?

💡 *Hint:* Think about thread creation overhead and resource limits.

Puzzle 1 – Simple Answer

The problem occurs because the server creates too many threads, one for each request. This consumes a lot of CPU and memory, making the server slow.

Solution:

Use a thread pool, where a fixed number of threads are reused to handle requests. This reduces overhead and improves performance.

Puzzle 5: The Shared Notebook

Two students, A and B, are writing marks in the same notebook.

- A reads the current total as 50
- B reads the same total as 50
- A adds 10 and writes 60
- B adds 20 and writes 70

✦ The correct total should be 80, but the notebook shows 70.

? **Puzzle Question:**

1. What OS problem is illustrated here?
2. Which concept can prevent this issue?

👉 **Hint:** Think in terms of critical section and synchronization.

Puzzle 6:

PUZZLE: The Shared Online Classroom

An online classroom system supports:

- Many students viewing lecture notes
- Only one instructor updating the notes

System



Rules

- 1 Students can read simultaneously.
- 2 While the instructor updates, no student can read.
- 3 Instructor should not wait forever.

Tasks

- ✓ Identify which classical synchronization problem this represents.
- ✓ Specify the monitor condition variables required.
- ✓ Explain how the monitor ensures mutual exclusion.
- ✓ Discuss how starvation is avoided.

SOLUTION: The Shared Online Classroom

Problem Identified:



Readers–Writers Problem

Many students can read; **only one** instructor can write at a time.

Monitor Condition Variables:



okToRead

Signals waiting students that reading is allowed.



okToWrite

Signals the waiting instructor that writing is allowed.

Mutual Exclusion Ensured:



Mutual exclusion inside the monitor allows many students but only one instructor.

Starvation Avoided:



Priority to Writing:

- okToWrite signaled if waiting and no students are reading.
- Gives precedence to the instructor for writing next.

Solution Implement



Priority to Writing:

- okToWrite signaled if waiting and no students are reading.
- Gives precedence to the instructor for writing next.

Puzzle 7: The Shared Online Classroom

An online classroom system supports:

- Many students **viewing lecture notes**
- Only one instructor **updating the notes**

Rules:

1. Students can read simultaneously.
2. While the instructor updates, no student can read.
3. Instructor should not wait forever.

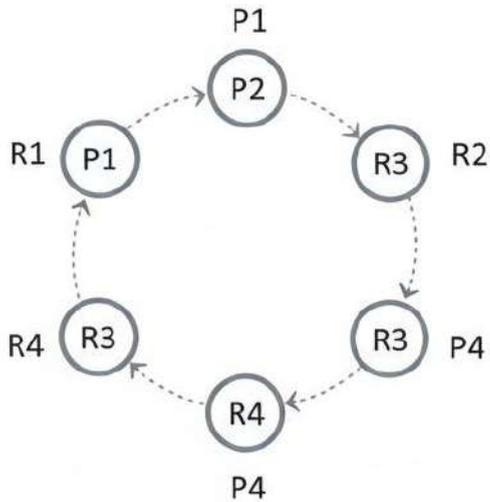
Tasks:

1. Identify which **classical synchronization problem** this represents.
2. Specify the **monitor condition variables** required.
3. Explain how the monitor ensures **mutual exclusion**.
4. Discuss how starvation is avoided.

DEADLOCK PUZZLE (REASONING BASED)

The Four Resource Lab

Current Situation



Tasks (Answer in Words / Diagrams)

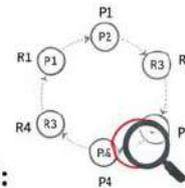
1. DEADLOCK STATE?

 YES, a circular wait dependency exists.

2. NECESSARY CONDITIONS:

- **Mutual Exclusion:** Resources are non-sharable.
- **Hold & Wait:** Processes hold resources while waiting for others.
- **No Preemption:** Resources cannot be taken.
- **Circular Wait:** $P1 > R2, P2 > R3, P3 > R4, P4 > R1$

3. RESOURCE ALLOCATION GRAPH:



4. PREVENTION TECHNIQUE:

 (e.g., impose resource ordering)

5. RECOVERY METHOD



 Process Termination (e.g. Abort P1 to break cycle. Victim choice depends on cost/priority).