SYSC 4001

Assignment 3: Part 2C Report

Saim Hashmi (101241041)
Abdullah Salman (101282570)
Link to Part 2: click here

Part 2.c: DeadLock, Livelock Analysis and Execution Order:

Overview and Setup

This report examines the behavior of the concurrent exam marking system from part 2, focusing on potential deadlocks, livelocks, and the execution order of Tas.

Number of TAs tested: 2,3 and 4

- For part 2.a (Only shared memory):
  - With 2_TAs_test.log
  - With 3_TAs_test.log
  - With 4_TAs_test.log

- For part 2.b (semaphores and shared memory):
  - With 2_TAs_test.log
  - With 3_TAs_test.log
  - With 4_TAs_test.log

Exam files: It starts from exam01.txt to exam20.txt

The exam20.txt is the termination exam because each file contains a student no. and the systems stop when the student no. 9999 is loaded which is stored in the exam20.txt.The logs captures every parent and TA action done also capturing the rubric update, question marking, exam completion and shutdown message.

Part 2.a Shared memory contains

- It contains the current student no. for each student and the rubric of 5 questions in shared memory which can be read and written concurrently by all TAs.
- The current exam index which shows the status of the current exam in shared memory which one is loaded and which is next.

- An array per question state (e.g. “not started”, ”being marked” and “done”) is also updated without locks.
- It contains a termination flag to terminate all processes.

Behavior of Part 2.a

In part 2.a ,TA operates without synchronization, and each TA has a workflow:

- They review the rubric and optionally make correction.
- They pick an unmarked question and mark it.
- Once all questions are marked, one TA moves to load the next exam.
- Termination occurs when the last exam (for student no.9999) is reached.

Observations (for 2,3 and 4 TAs):

- All exams were eventually processed, including the last exam.
- Race conditions were common such as multiple TAs sometimes attempted the same question simultaneously and also the entire rubric could be overwritten if two TAs updated them at nearly same time.
- Despite these issues, no deadlocks or livelocks occurred. The program always made progress to completion.

Analysis:

- Progress is guaranteed because TAs never block on shared data, they always first read than write and continue.
- There is no circular waiting since no locks exist.
- Inconsistent rubric values and overlapping question marking are the main risks if there are more TAs.

Behavior of Part 2.b

In part 2.b ,to address race conditions there is use of semaphores:

- To protect rubric updated and file writing
- To ensure only one TA claims a question at a time
- Also to prevent log messages from interleaving

Observations (for 2,3 and 4 TAs):

- All exams for all tested TA counts, were marked correctly.
- No two TAs marked the same question simultaneously.
- Rubric was updated and exam loading occurred in a controlled and predictable order.
- No deadlocks or livelocks were observed.

Deadlock and Livelock Analysis

Part 2.a (unsynchronized):

- No deadlocks or livelocks were detected.
- Race conditions were present, leading to occasional overlapping marking and inconsistent rubric states.

Part 2.b (semaphore):

- Semaphores ensure proper access so only one TA can update a rubric and mark a question or load the next exam at a time.
- Lock acquisition order is consistent and held only briefly, preventing circular waiting.
- No situation occurred where TAs waited indefinitely.

Execution Order Observations

- Execution order is inherently non determined due to concurrency.
- Different TAs may mark questions in different orders depending on scheduling.
- Despite variability in timing, all students always receive fully marked exams.
- With semaphores output is consistent and predictable even if execution order is different.

Conclusion

- Part 2.a, the system works without deadlocks, but race conditions make it unsafe for larger scales or more TAs.
- Part 2.b, the semaphore based synchronization prevents race conditions, guarantees correct marking and also maintains consistent execution order.
- Across all test (2,3, and 4 TAs) no deadlock or livelock were observed.
- Semaphores are effective in ensuring safe and predictable concurrent operation.