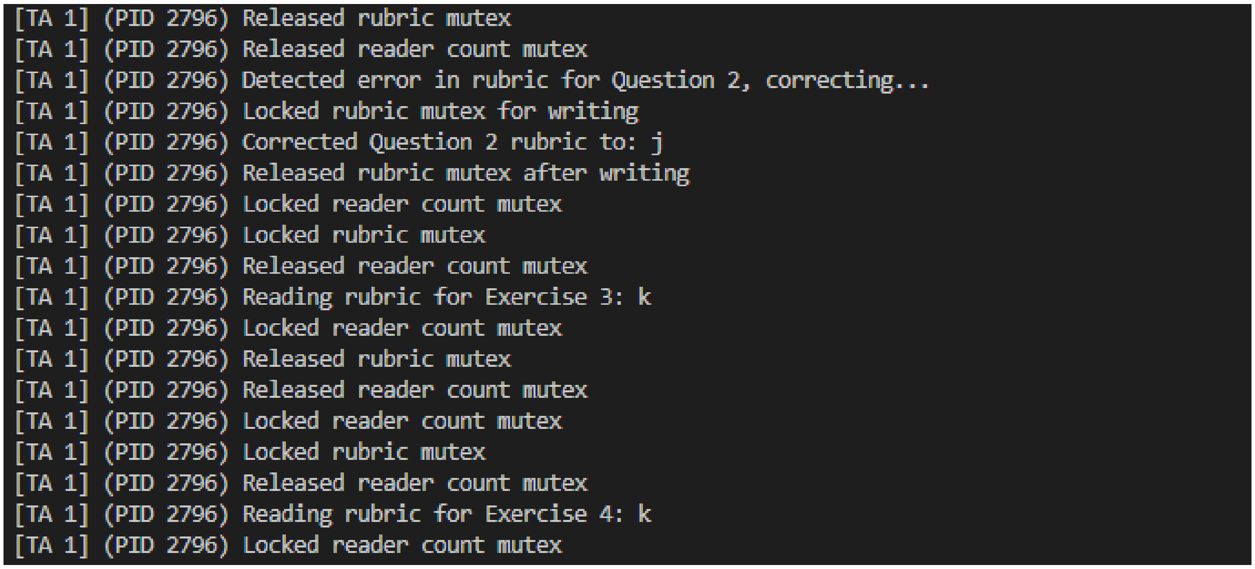
Student 1: Manoj Kuppuswamy Thyagarajan, **ID:** 101166589
Student 2: Shabesa Kohilavani Arunprakash. **ID:** 101258619

Assignment 3 Report - Concurrency, Shared Memory, Virtual Memory, Files

Part 2C: Deadlock and Livelock

Test Scenario: ./TA_marking 3 30

- No. of TAs: 3
- Correction chance for the rubric: 30%
- Result: Execution was successful with no Deadlock or Livelock



```
[TA 1] (PID 2796) Released rubric mutex
[TA 1] (PID 2796) Released reader count mutex
[TA 1] (PID 2796) Detected error in rubric for Question 2, correcting...
[TA 1] (PID 2796) Locked rubric mutex for writing
[TA 1] (PID 2796) Corrected Question 2 rubric to: j
[TA 1] (PID 2796) Released rubric mutex after writing
[TA 1] (PID 2796) Locked reader count mutex
[TA 1] (PID 2796) Locked rubric mutex
[TA 1] (PID 2796) Released reader count mutex
[TA 1] (PID 2796) Reading rubric for Exercise 3: k
[TA 1] (PID 2796) Locked reader count mutex
[TA 1] (PID 2796) Released rubric mutex
[TA 1] (PID 2796) Released reader count mutex
[TA 1] (PID 2796) Locked reader count mutex
[TA 1] (PID 2796) Locked rubric mutex
[TA 1] (PID 2796) Released reader count mutex
[TA 1] (PID 2796) Reading rubric for Exercise 4: k
[TA 1] (PID 2796) Locked reader count mutex
```

Sample execution screenshot

Deadlock Analysis:

1. Mutual Exclusion: Rubric writing, question marking uses exclusive locks.
2. Hold and Wait: Prevented by making TAs release the reader count mutex before acquiring other locks.

   Example from output:

   [TA 1] (PID 2796) Released reader count mutex

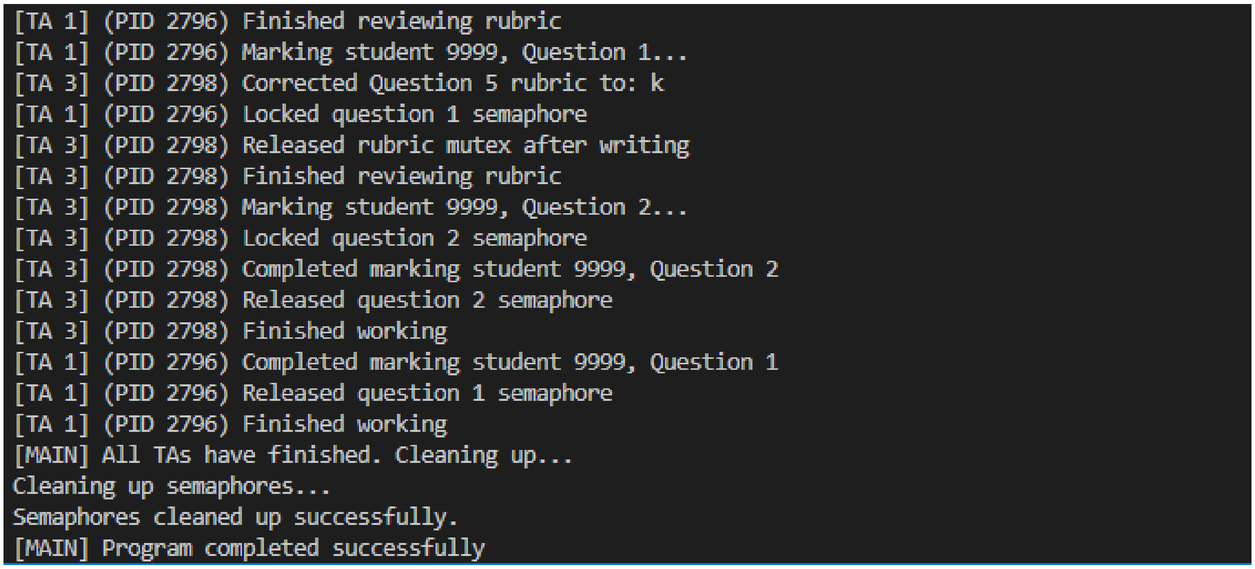
   [TA 1] (PID 2796) Reading rubric for Exercise 2: i

3. No Preemption, as each TA process waited to access the shared memory.

These checks did not allow deadlocks to occur during execution.

Student 1: Manoj Kuppuswamy Thyagarajan, **ID:** 101166589
Student 2: Shabesa Kohilavani Arunprakash. **ID:** 101258619

Livelock Analysis:



```
[TA 1] (PID 2796) Finished reviewing rubric
[TA 1] (PID 2796) Marking student 9999, Question 1...
[TA 3] (PID 2798) Corrected Question 5 rubric to: k
[TA 1] (PID 2796) Locked question 1 semaphore
[TA 3] (PID 2798) Released rubric mutex after writing
[TA 3] (PID 2798) Finished reviewing rubric
[TA 3] (PID 2798) Marking student 9999, Question 2...
[TA 3] (PID 2798) Locked question 2 semaphore
[TA 3] (PID 2798) Completed marking student 9999, Question 2
[TA 3] (PID 2798) Released question 2 semaphore
[TA 3] (PID 2798) Finished working
[TA 1] (PID 2796) Completed marking student 9999, Question 1
[TA 1] (PID 2796) Released question 1 semaphore
[TA 1] (PID 2796) Finished working
[MAIN] All TAs have finished. Cleaning up...
Cleaning up semaphores...
Semaphores cleaned up successfully.
[MAIN] Program completed successfully
```

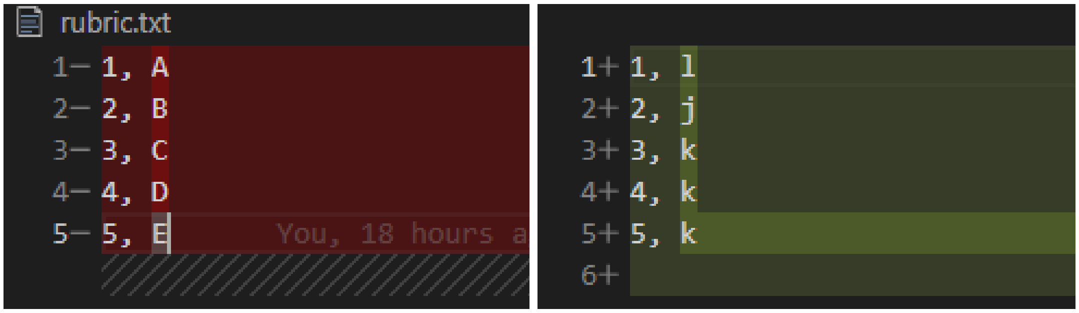
Observations:

- All 3 TAs completed their work successfully.
- Exams process from 0001 to 0020 and 9999.
- The process terminates when the file reaches 9999.
- The code did not get stuck in an infinite loop and have any failures.
- The code cleanly shut down after meeting the exit condition.

Semaphores make sure that only one person can access something at a time, and there are no retry or backoff techniques that could generate conflicts all the time, thus preventing Livelocks.

File Status Verification

Rubric.txt Changes



```
rubric.txt
- 1, A
- 2, B
- 3, C
- 4, D
- 5, E          You, 18 hours a
```

```
+ 1, l
+ 2, j
+ 3, k
+ 4, k
+ 5, k
+
```

Multiple corrections successfully applied by different TAs with 30% probability

Student 1: Manoj Kuppuswamy Thyagarajan, **ID:** 101166589
Student 2: Shabesa Kohilavani Arunprakash. **ID:** 101258619

Conclusion:

The semaphore-based method with three TAs and a 30% chance of rectification works well to stop both deadlock and livelock. The design satisfies all three important requirements:

- Mutual exclusion through dedicated semaphores
- Progress via the reader-writer pattern, enabling concurrent reads
- Bounded waiting through FIFO semaphore ordering

The execution shows safe and efficient concurrent access to shared resources for the different TA processes (rubric, exams, questions) without conflicts, race conditions, deadlocks, and live locks. The system completed successfully with proper cleanup of all the resources that it created.