Part 2: Concurrent Processes in Unix

SYSC 4001: Operating Systems

Group Member 1: Amreen Shahid (101306199)
Group Member 2: Celina Yang (101299938)

Carleton University
Dec 1st, 2025

Introduction: Explanation of Code

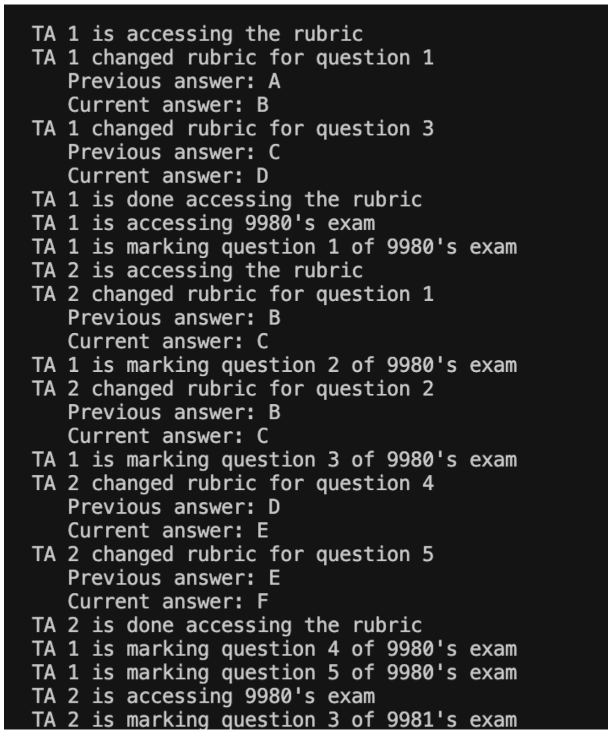
In this part of the assignment, we implemented a TA marking simulation. The program first loads the rubric and first exam using the load_rubric() and load_exam() method. Each TA will then iterate through the rubric and modify if needed using the iterate_rubric() method. Once the rubric is reviewed and updated, the TA begins to mark the exam using the mark() method. The marking method iterates through each question of the loaded exam. If the TA finds an unmarked question, it will be marked. After all questions are marked the TA will load the next exam. The program terminates once student number 9999 is reached.

To avoid race conditions, two semaphores were added, sem_rubric and sem_exam. These ensure that only one TA can access the rubric at a time. It also prevents multiple TAs from marking the same question of the one exam.

Execution Order

1. TA 1 accesses the rubric, changes questions and finishes
2. TA 1 accesses the loaded exam and begins marking questions
3. While TA 1 is marking, TA 2 accesses the rubric, changes questions and finishes
4. TA 2 accesses the loaded exam and marks questions

Steps 1-4 repeat for the remaining exams until student number 9999 is reached. This process can be seen in the output below.



```
TA 1 is accessing the rubric
TA 1 changed rubric for question 1
   Previous answer: A
   Current answer: B
TA 1 changed rubric for question 3
   Previous answer: C
   Current answer: D
TA 1 is done accessing the rubric
TA 1 is accessing 9980's exam
TA 1 is marking question 1 of 9980's exam
TA 2 is accessing the rubric
TA 2 changed rubric for question 1
   Previous answer: B
   Current answer: C
TA 1 is marking question 2 of 9980's exam
TA 2 changed rubric for question 2
   Previous answer: B
   Current answer: C
TA 1 is marking question 3 of 9980's exam
TA 2 changed rubric for question 4
   Previous answer: D
   Current answer: E
TA 2 changed rubric for question 5
   Previous answer: E
   Current answer: F
TA 2 is done accessing the rubric
TA 1 is marking question 4 of 9980's exam
TA 1 is marking question 5 of 9980's exam
TA 2 is accessing 9980's exam
TA 2 is marking question 3 of 9981's exam
```

Rubric Changes

The following table shows the contents of the rubric file before and after marking the exams

Question	Before Marking	After Marking
1	A	V
2	B	X
3	C	[
4	D	T
5	E	V

Critical Section Requirements

1. **Mutual Exclusion**
   Mutual exclusion ensures that shared resources are only accessed by one process at a time. The shared resources in our program are the rubric and exams. We did this by introducing semaphores sem_rubric and sem_exam. Before entering a critical section, each TA performs a sem_wait(). Once the TA is done, it will perform a sem_post signalling to the other TAs that the critical section is now “open”.

2. **Progress**
   Progress ensures that if no process is in a critical section, another process that is waiting must enter. Since we added semaphores progress is bound to happen because once sem_post() is called another TA that is waiting, immediately has access to the critical section.

3. **Bounded Waiting**
   Bounded waiting prevents ‘starvation’, ensuring that all processes will eventually enter the critical sections resulting in no infinite waiting. All TAs who are waiting to enter a critical section are added into the wait queue. Once one TA is finished, the next TA in the wait queue gets access.

Conclusion

Since no deadlocks or livelocks were observed, we can conclude that the semaphores work correctly to allow concurrent marking without any conflicts.