In part 2A, the various TA processes accessing the shared data at the same time caused race conditions. In part 2B, this was fixed with the use of a semaphore. The three requirements associated with the solution to the critical section problem are mutual exclusion, progress, and bounded waiting. These three solutions were implemented in the design for part 2B.

The use of the semaphore ensures mutual exclusion since only one TA process can access the shared memory at a time. While the rubric or exam index is being updated by one TA, the others must wait.

When the critical section is free, the TA that reaches the semaphore first is able to continue immediately. The system guarantees that if the critical section is available, it will let a waiting TA process enter so work progresses.

Each TA waits in a queue by the semaphore in FIFO order, so no process is postponed forever. Because of this, every TA will get a chance to enter the critical section. The system doesn’t allow one TA to be skipped indefinitely while others keep going in front of them, satisfying bounded waiting.