

Part 2 Report

In shared memory, we store an array containing the rubric marks, an int that is the student number of the exam being marked, an array that is used to keep track of which questions has been marked, a boolean that indicates whether the current exam has finished being marked, and another boolean that indicates whether all the exams have been marked.

When running the program, a value is passed through the argv parameter to indicate the number of TAs marking the exams. This number is stored in the variable “numTA”. The parent process loops through a for loop numTA times, forking a child process each loop. It then waits for each child process to terminate before ending the program.

Each child process enters an infinite while loop. Inside the while loop they go through two for loop. In the first for loop, the process loops through each of the entries in the rubric marks array in shared memory, randomly deciding each loop whether to change the entry or not. If it does decide to change the entry, the process adjusts the entry in the array and updates the rubric.txt file. Semaphores are used to prevent other child processes from adjusting the rubric at the same time.

In the second for loop, the process loops through each of the entries in the marked array in shared memory. If an entry contains a 0, that indicated that the associated question has not been marked yet. As the process proceeds to mark the question, the associated entry in the array is updated to a 1, alerting other processes that this question is in the process of being marked/already marked. Semaphores are used again to prevent multiple processes from claiming the same question at the same time.

When each process finishes checking all 5 questions, they enter a while loop until the exam has finished being marked. After that, the boolean value in shared memory “finishedExam”, is updated to end the while loop, allowing the processes to restart the loop. The next exam is loaded into shared memory, and the entire process repeats until exam 9999. After exam 9999 is finished being marked, the boolean value in shared memory “finished” is set to true, allowing the processes to reach a break statement, freeing them from the while loop and allowing them to exit.

Once all the child processes have been terminated, the parent process terminates the shared memory and semaphores, closes the rubric.txt file, and terminates.