SYSC4001 Assignment 3 – Part 2 Response

Eric McFetridge #101310942
Justin Winaford #101199245

We ran both implementations of the TA grading system, the first implementation just uses blind shared memory, while the other uses proper shared memory access management. Both programs successfully ran without any fatal race conditions occurring or deadlock this is because deadlock is extremely unlikely due to the nature of our program. We don't have children waiting on eachother to complete, so they don't get stuck in this way. We have n > 2 TA's grading at any time independently.

The first implementation experienced a bit of live lock, where the output was effected by our children processes accessing the same exams and trying to grade them at the same time. This leads to unpredictable, but non-fatal behavior in the program. Such as random rubric or question changes by two TA's at the same time. This becomes more emphasized when we increase the amount of concurrency, increasing the chance for racing.

When running the modified version of the program using semaphores we should no longer have to worry about potential livelocks. The semaphores enforce unique access per process, meaning no two TA's will have to worry about a race condition from reading or marking the same question at once. This greatly reduces the overall flaws in our program and reduces the chances of it crashing since we safely share objects in memory. We can be much more confident that this program will run without error, while on the other one we just hope to be lucky and that no critical faults occur.