**OSP programming assignment 1:**

**S3540567 – Timothy Tran**

**Github:**

**Notes, Issues, and limitations:**

Both for problem A and E, the Valgrind input indicates that all blocks are freed on exit as indicated here:  
Text

Description automatically generated  
figure (1): Valgrind output for A

Text

Description automatically generated  
figure(2): Valgrind output for E

As pointed out in the code, solution for Problem A was based on possible solution provided by lecturers and referenced from <https://shivammitra.com/c/producer-consumer-problem-in-c/#>. For problem A, Semaphores were used over pthread conditional signaling to achieve process synchronization and avoid busy-waiting in terms of checking the size of the buffer in order to indicate whether a process should be sleeping or not to signal the use of a thread. The use of semaphores provides a simple elegant solution to the sleeping condition of a producer consumer problem of which indicates whether there are empty spots in the buffer/bucket to fill and to sleep if there aren’t for the producer, and whether there a filled slot in the buffer/bucket and to sleep if there aren’t for the consumer.

In the Smoker-Agent problem there may be potential busy waiting solutions that could have been implemented, however I am unclear about what constitutes such issues in the context of my work at this time. The implementation of “making it fair” may have been in itself a potential busy waiting issue.

**Producer and consumer problem: A**

Concessions made