DVA315 - LAB 2

Olle Olofsson, Felix Sjöqvist

February 19, 2019

1. The producer related issues were that they were trying to produce when the buffer was full, and accessing the shared resources (buffer) at the same time. And similarly, the consumer related issues were that they were trying consume from an empty buffer as well as accessing it at the same time.

2. The implementation of a mutex made sure that no two processes could access the shared resources at the same time. So now all access to the buffer is managed sufficiently enough to hinder errornous reads and writes. What remains to be solved is producer trying to produce, and consumers trying to consume, when the buffer is either full or empty, respectively.

3. The mutex makes sure that access to buffer is controlled, and the counting semaphore is used to let the actors of the system "know" wheter they should act or not.

4. Errorproducing sequence: Producer 1 reads buffer to check whether it is full, determines the buffer has one slot left, proceeds to write to the buffer. Between the read/write process Producer 2 reads the buffer, also concludes 1 slot is empty. Now we have two producers, both trying to add one item each to the buffer with only one free slot. This might not lead to any error messages, since these only occur when a process finds that the buffer is full/empty, but it can cause fatal behaviour since the first item to be added will be overwriten by the latter.
   Another way errornous behaviour could occur, which also produces an error message, is where a producer/consumer tries to produce/consume when the buffer is full/empty, this is because they have no way of "knowing" beforehand wheter or not the buffer is in a state where their respective actions are suited.