Student: Niklas Stylianou	Student: Ömer Yildiz	Date: 12-01-2021	Assignment 3: Condition variables
Student ID: 1284037	Student ID: 1644300	Course: 2INCO	Group: 44

Feedback from assignment 2 recommended more than 1 mutex. In this case, the producers use two mutexes:

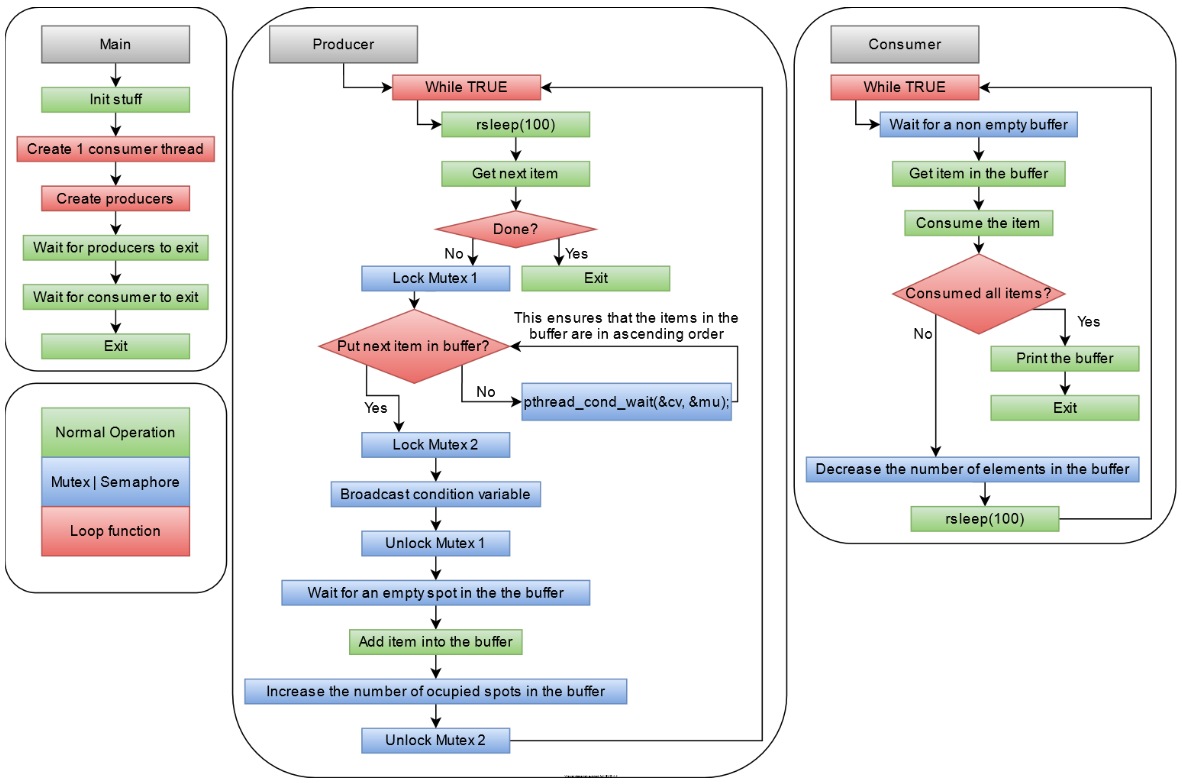
1. The first mutex is used to hand over control between producers to ensure the items in the buffer are in ascending order.
2. The second mutex is used to lock access to the buffer. Only after this mutex has been locked will mutex 1 be unlocked. This parallelism is stimulated, because other producers can continue while the current producer places an item in the buffer.

The consumer does not require are mutex lock even though it accesses the buffer (a shared resource). The producer only notifies the consumer after placing the item. It does this within a mutex lock. Furthermore, once an item has been placed in the buffer, no other thread will change that value. So, there is no condition where the consumer reads a wrong value from the buffer.

Two semaphores to do operations on the buffer:

1. One keeps track of the number of elements in the buffer. To prevent the producer from writing to a full buffer.
2. The other keeps track of the empty spaces. To precent the consumer reading form empty spaces.

The book did a good job in explaining the theory behind the bounded buffer problem. The theory and exercise gave the basis. On the internet (ORACLE, 2012)[1] more info was gathered on how to work with semaphores. It was adapted for our purposes.



[1] ORACLE. (2012, October). *Multithreaded Programming Guide*. Opgeroepen op December 28, 2020, van https://docs.oracle.com/: https://docs.oracle.com/cd/E26502_01/html/E35303/sync-11157.html