Project

Due : 11:59 PM, Friday, August 9, 2024

Project Overview

In this project, you will complete a multi-threaded synchronization problem using semaphores and mutexes in a partially completed project code. The problem involves multiple levels of producers and consumers interacting with two bounded buffers. The primary goal is to understand and demonstrate synchronization mechanisms to ensure safe and efficient access to shared resources in a concurrent environment.

Project Description

- You will require two types of buffer to implement this project. The first buffer (Buffer 1) where initial producer threads producer and place their items . Intermediate consumer/producer threads consume items from Buffer 1 and place their processed items in second buffer (Buffer 2).
- Threads: You need to create three types of threads:
  - **Producers**: These threads will produce items and place them into Buffer 1. Ensure that producers wait if Buffer 1 is full, using semaphores and mutexes for synchronization.
  - **First Level Consumers/Producers**: These threads will consume items from Buffer 1, process them, and then produce new items into Buffer 2. Ensure that these threads wait if Buffer 1 is empty or if Buffer 2 is full, using semaphores and mutexes for synchronization.
  - **Second Level Consumers**: These threads will consume items from Buffer 2. Ensure that they wait if Buffer 2 is empty, using semaphores and mutexes for synchronization.
- Synchronization: To handle the synchronization between threads, you must use semaphores and mutexes. Semaphores will be used to manage the full and empty states of each buffer, while mutexes will ensure mutual exclusion when accessing the shared buffers. This combination will help to avoid race conditions and ensure that the buffers are accessed safely and efficiently by multiple threads.

Additional Requirements

Your program should gracefully terminate after 20 items have been produced, processed, and consumed. This number will be defined by a constant. Each thread should complete its assigned tasks and then terminate, ensuring that all items are handled correctly before the program ends.

Partial Code

The code is provided with all the required function on Canvas assignment page. You need to complete the code in the main() section to demonstrate the output with synchronization requirements. There are functions for produce() and consume() for producing and consuming items.

Furthermore, there functions for producer() entity , first_level_consumer_producer() entity and second_level_consumer() entity. You need to fill out the main() section of the code.

Group Formation

You must create a group of minimum 4 and maximum 5 members to complete the project work. You can sign up for the groups on Canvas in the people section under Project GRP<ID>. You should communicate with your group members to work together on this project through Canvas messages, Emails or other available messaging applications like Discord. The final day to complete group formation is 29th July. The group members can take up the roles as project lead, document creator/editor, technical expert(code) etc.

Output Requirements

The output of the program should display the sequence of produced and consumed items, indicating which buffer they were placed into or taken from. The project considers that there are two producer entities to produce items and place them in Buffer 1, two producer/consumer entities who consume from Buffer1 and produce items in buffer 2 and lastly, there are 2 consumer entities who consume items from buffer 2. Each operation should be printed to the console with clear identification of the thread performing the action and the buffer involved. For example:

```
(base) SacStar:SMR24 Syed$ ./producer-consumer-project
Consumer ID: 3 , Consumed: 0 from Buffer 1
Producer ID: 3 , Produced: 3 into Buffer 2
Producer ID: 1, Produced: 100 into Buffer 1
Producer ID: 2, Produced: 200 into Buffer 1
Consumer ID: 4 , Consumed: 200 from Buffer 1
Producer ID: 4 , Produced: 2004 into Buffer 2
Consumer_ID: 5 ,Consumed: 0 from Buffer 2
Consumer_ID: 6 ,Consumed: 0 from Buffer 2
Consumer_ID: 5 ,Consumed: 0 from Buffer 2
Consumer ID: 4 , Consumed: 0 from Buffer 1
Producer ID: 4 , Produced: 4 into Buffer 2
Consumer_ID: 5 ,Consumed: 0 from Buffer 2
Consumer ID: 3 , Consumed: 0 from Buffer 1
Producer ID: 3 , Produced: 3 into Buffer 2
Producer ID: 2, Produced: 201 into Buffer 1
Producer ID: 1, Produced: 101 into Buffer 1
Consumer_ID: 6 ,Consumed: 0 from Buffer 2
Producer ID: 2, Produced: 202 into Buffer 1
Consumer_ID: 5 ,Consumed: 0 from Buffer 2
Consumer ID: 4 , Consumed: 202 from Buffer 1
Producer ID: 4 , Produced: 2024 into Buffer 2
Consumer ID: 4 , Consumed: 0 from Buffer 1
Producer ID: 4 , Produced: 4 into Buffer 2
Consumer_ID: 6 ,Consumed: 0 from Buffer 2
Producer ID: 2, Produced: 203 into Buffer 1
Consumer ID: 3 , Consumed: 0 from Buffer 1
```

```
Producer ID: 3 , Produced: 3 into Buffer 2
Producer ID: 1, Produced: 102 into Buffer 1
Producer ID: 2, Produced: 204 into Buffer 1
Producer ID: 1, Produced: 103 into Buffer 1
Consumer ID: 3 , Consumed: 204 from Buffer 1
………………………………………………
………………………………………………
………………………………………………
………………………………………………
………………………………………………
………………………………………………
………………………………………………

Consumer_ID: 6 ,Consumed: 2154 from Buffer 2
Producer ID: 1, Produced: 117 into Buffer 1
Consumer_ID: 5 ,Consumed: 2053 from Buffer 2
Consumer ID: 4 , Consumed: 114 from Buffer 1
Producer ID: 4 , Produced: 1144 into Buffer 2
Consumer ID: 3 , Consumed: 115 from Buffer 1
Producer ID: 3 , Produced: 1153 into Buffer 2
Consumer_ID: 6 ,Consumed: 1153 from Buffer 2
Consumer_ID: 5 ,Consumed: 2103 from Buffer 2
Consumer_ID: 5 ,Consumed: 2114 from Buffer 2
Producer ID: 2, Produced: 218 into Buffer 1
Consumer ID: 3 , Consumed: 216 from Buffer 1
Producer ID: 3 , Produced: 2163 into Buffer 2
Consumer ID: 3 , Consumed: 116 from Buffer 1
Producer ID: 3 , Produced: 1163 into Buffer 2
Producer ID: 1, Produced: 118 into Buffer 1
Consumer ID: 4 , Consumed: 217 from Buffer 1
Producer ID: 4 , Produced: 2174 into Buffer 2
Producer ID: 1, Produced: 119 into Buffer 1
Consumer_ID: 6 ,Consumed: 2174 from Buffer 2
Consumer ID: 4 , Consumed: 117 from Buffer 1
Producer ID: 4 , Produced: 1174 into Buffer 2
Consumer ID: 3 , Consumed: 218 from Buffer 1
Producer ID: 3 , Produced: 2183 into Buffer 2
Consumer_ID: 6 ,Consumed: 1174 from Buffer 2
Consumer ID: 4 , Consumed: 118 from Buffer 1
Producer ID: 4 , Produced: 1184 into Buffer 2
Producer ID: 2, Produced: 219 into Buffer 1
Consumer ID: 3 , Consumed: 119 from Buffer 1
Producer ID: 3 , Produced: 1193 into Buffer 2
Consumer_ID: 6 ,Consumed: 2183 from Buffer 2
Consumer ID: 3 , Consumed: 219 from Buffer 1
Producer ID: 3 , Produced: 2193 into Buffer 2
Consumer_ID: 6 ,Consumed: 1184 from Buffer 2
Consumer_ID: 6 ,Consumed: 1193 from Buffer 2
Consumer_ID: 6 ,Consumed: 2193 from Buffer 2
All producers and consumers have finished
```

Submission Requirements:

1. **Code**: You must submit a complete functional code on Canvas.
2. **Final report**: You should submit a report on Canvas which will include the following
    a. A cover page with project title and group members along with the student ID and email address.
    b. Objective of the project.
    c. Organization of the project members and their roles and functions they performed in the project. There are groupwork points assigned for this task and coordination.
    d. Description of the code on how each function execute and successfully perform the task of production and consumption.
    e. Output: The report must consist of a output section for showing the output screenshot of your code. For example, see the output section in previous page
    f. Observation: You musy provide observation and comments on the behavior of the code. Does it match what you intended to accomplish
    g. Challenges and Limitation: Report challenges and limitation of your project and code.

Rubric:

- Successful code compilation and output: 30 Points
- Code comments : 10 Points
- Final report : 40 points
- Group Work : 20 points

Note: You must follow the submission guidelines and rubric to score maximum points in this project.