CSC 139

Summer 2024, 2nd Session

Professor Syed

Final Project:

Producer and Consumer Problem

Group Members:

Name	ID	Email	Role
Mansoor Ali	219321922	mansoorali2@csus.edu	Project Lead
Christian Ramirez	220105016	christianaramirez@csus.edu	Document Editor
Illya Gordyy	302682939	igordyy@csus.edu	Document Editor
Bilal Baloch	218756812	bbaloch2@csus.edu	Technical Expert

Objective:

The objective of the project is to 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. We are required to create three types of threads producer, First level Consumer/Producer, and Second-level Consumer each having a responsibility regarding placing or taking away from the buffer. Our primary goal is to understand and demonstrate synchronization mechanisms to ensure safe and efficient access to shared resources in a concurrent environment.

Organization:

In our project we have various roles including Technical Expert, Document Editor and Project Leader. The technical expert was heavily involved in developing the code and ensuring it is correct. The document editor is responsible for understanding code, explaining it and designing our project document. The group leader is responsible for coordinating meetings, facilitating group discussion, structuring the problem, ensuring each assigned task is completed, and leading the project's completion within the appropriate timeline.

Description of the code:

In main we declare threads for all types of producers and consumers. There are three types of producers/consumers. There is the producerThread, firstlevelConsumerProducerThread, and secondLevelConsumerThread. After that we declare variables for 2 producer entities, 2 producer/consumer entities, and 2 consumers. These are the id's for each of the producers/consumers. Next we initialize the semaphores for each buffer, buffer 1 and 2. There is a semaphore for the full and empty. After that we create threads for each number of producers and consumers for each level. There are 2 producers for buffer 1. 2

producer/consumers. And 2 second level consumers. The total threads created are 6. After that

we join the 3 different types of threads to wait for them to complete using pthread_join() on each

type of thread created (there are three types). Next we destroy the threads, using

pthread_mutex_destory, which destroys the mutex buffer1_mutex and buffer2_mutex.

Output:

```
bbaloch2@ecs-pa-coding1:~/csc139

Consumer ID: 4 , Consumed: 215 from Buffer 1
Producer ID: 4 , Produced: 2154 into Buffer 2
Consumer_ID: 6 ,Consumed: 2154 from Buffer 2
Consumer ID: 3 , Consumed: 216 from Buffer 1
Producer ID: 3 , Produced: 2163 into Buffer 2
Producer ID: 1, Produced: 114 into Buffer 1
Producer ID: 1, Produced: 115 into Buffer 1
Consumer_ID: 5 ,Consumed: 2163 from Buffer 2
Consumer_ID: 4 , Consumed: 217 from Buffer 1
Producer ID: 4 , Produced: 2174 into Buffer 2
Producer ID: 1, Produced: 116 into Buffer 1
Consumer_ID: 5 ,Consumed: 2174 from Buffer 2
Consumer_ID: 3 , Consumed: 113 from Buffer 1
Producer ID: 3 , Produced: 1133 into Buffer 2
Consumer_ID: 6 ,Consumed: 1133 from Buffer 2
Consumer_ID: 4 , Consumed: 218 from Buffer 1
Producer ID: 4 , Produced: 2184 into Buffer 2
Producer ID: 1, Produced: 117 into Buffer 1
Consumer ID: 3 , Consumed: 219 from Buffer 1
Producer ID: 3 , Produced: 2193 into Buffer 2
Consumer ID: 4 , Consumed: 114 from Buffer 1
Producer ID: 4 , Produced: 1144 into Buffer 2
Consumer_ID: 6 ,Consumed: 2184 from Buffer 2
Consumer ID: 4 , Consumed: 115 from Buffer 1
Producer ID: 4 , Produced: 1154 into Buffer 2
Consumer_ID: 5 ,Consumed: 2193 from Buffer 2
Producer ID: 1, Produced: 118 into Buffer 1
Consumer ID: 3 , Consumed: 116 from Buffer 1
Producer ID: 3 , Produced: 1163 into Buffer 2
Consumer_ID: 5 ,Consumed: 1144 from Buffer 2
```

```
bbaloch2@ecs-pa-coding1:~/csc139

Producer ID: 2, Produced: 214 into Buffer 1
Producer ID: 1, Produced: 111 into Buffer 1
Consumer ID: 3 , Consumed: 109 from Buffer 1
Producer ID: 3 , Produced: 1093 into Buffer 2
Consumer_ID: 6 ,Consumed: 1093 from Buffer 2
Consumer ID: 4 , Consumed: 212 from Buffer 1
Producer ID: 4 , Produced: 2124 into Buffer 2
Consumer_ID: 5 ,Consumed: 2124 from Buffer 2
Producer ID: 1, Produced: 112 into Buffer 1
Producer ID: 2, Produced: 215 into Buffer 1
Consumer ID: 4 , Consumed: 110 from Buffer 1
Producer ID: 4 , Produced: 1104 into Buffer 2
Producer ID: 2, Produced: 216 into Buffer 1
Consumer ID: 3 , Consumed: 213 from Buffer 1
Producer ID: 3 , Produced: 2133 into Buffer 2
Consumer_ID: 6 ,Consumed: 1104 from Buffer 2
Producer ID: 2, Produced: 217 into Buffer 1
Consumer_ID: 5 ,Consumed: 2133 from Buffer 2
Producer ID: 1, Produced: 113 into Buffer 1
Consumer ID: 4 , Consumed: 214 from Buffer 1
Producer ID: 4 , Produced: 2144 into Buffer 2
Consumer_ID: 6 ,Consumed: 2144 from Buffer 2
Consumer ID: 3 , Consumed: 111 from Buffer 1
Producer ID: 3 , Produced: 1113 into Buffer 2
Consumer_ID: 6 ,Consumed: 1113 from Buffer 2
Consumer ID: 3 , Consumed: 112 from Buffer 1
Producer ID: 3 , Produced: 1123 into Buffer 2
Consumer_ID: 5 ,Consumed: 1123 from Buffer 2
Producer ID: 2, Produced: 218 into Buffer 1
Producer ID: 2, Produced: 219 into Buffer 1
```

```
bbaloch2@ecs-pa-coding1:~/csc139

Producer_ID: 3 , Produced: 2073 into Buffer 2
Consumer_ID: 5 ,Consumed: 2073 from Buffer 2
Producer_ID: 1, Produced: 107 into Buffer 1
Consumer_ID: 4 , Consumed: 107 from Buffer 1
Producer_ID: 4 , Produced: 1074 into Buffer 2
Producer_ID: 2, Produced: 208 into Buffer 1
Consumer_ID: 6 ,Consumed: 1074 from Buffer 2
Producer_ID: 1, Produced: 108 into Buffer 1
Producer_ID: 2, Produced: 209 into Buffer 1
Consumer_ID: 4 , Consumed: 208 from Buffer 1
Producer_ID: 4 , Produced: 2084 into Buffer 2
Consumer_ID: 6 ,Consumed: 2084 from Buffer 2
Producer_ID: 2, Produced: 210 into Buffer 1
Consumer_ID: 3 , Consumed: 108 from Buffer 1
Producer_ID: 3 , Produced: 1083 into Buffer 2
Consumer_ID: 5 ,Consumed: 1083 from Buffer 2
Producer_ID: 2, Produced: 211 into Buffer 1
Consumer_ID: 4 , Consumed: 209 from Buffer 1
Producer_ID: 4 , Produced: 2094 into Buffer 2
Consumer_ID: 6 ,Consumed: 2094 from Buffer 2
Producer_ID: 1, Produced: 109 into Buffer 1
Consumer_ID: 4 , Consumed: 210 from Buffer 1
Producer_ID: 4 , Produced: 2104 into Buffer 2
Consumer_ID: 5 ,Consumed: 2104 from Buffer 2
Producer_ID: 2, Produced: 212 into Buffer 1
Consumer_ID: 3 , Consumed: 211 from Buffer 1
Producer_ID: 3 , Produced: 2113 into Buffer 2
Consumer_ID: 5 ,Consumed: 2113 from Buffer 2
Producer_ID: 1, Produced: 110 into Buffer 1
Producer_ID: 2, Produced: 213 into Buffer 1
```

```
bbaloch2@ecs-pa-coding1:~/csc139

Producer_ID: 2, Produced: 204 into Buffer 1
Consumer_ID: 3 , Consumed: 204 from Buffer 1
Producer_ID: 3 , Produced: 2043 into Buffer 2
Consumer_ID: 6 ,Consumed: 2043 from Buffer 2
Producer_ID: 1, Produced: 103 into Buffer 1
Producer_ID: 1, Produced: 104 into Buffer 1
Consumer_ID: 4 , Consumed: 103 from Buffer 1
Producer_ID: 4 , Produced: 1034 into Buffer 2
Consumer_ID: 6 ,Consumed: 1034 from Buffer 2
Producer_ID: 2, Produced: 205 into Buffer 1
Consumer_ID: 3 , Consumed: 104 from Buffer 1
Producer_ID: 3 , Produced: 1043 into Buffer 2
Consumer_ID: 5 ,Consumed: 1043 from Buffer 2
Consumer_ID: 4 , Consumed: 205 from Buffer 1
Producer_ID: 4 , Produced: 2054 into Buffer 2
Consumer_ID: 6 ,Consumed: 2054 from Buffer 2
Producer_ID: 1, Produced: 105 into Buffer 1
Consumer_ID: 4 , Consumed: 105 from Buffer 1
Producer_ID: 4 , Produced: 1054 into Buffer 2
Producer_ID: 2, Produced: 206 into Buffer 1
Consumer_ID: 3 , Consumed: 206 from Buffer 1
Producer_ID: 3 , Produced: 2063 into Buffer 2
Consumer_ID: 5 ,Consumed: 1054 from Buffer 2
Consumer_ID: 5 ,Consumed: 2063 from Buffer 2
Producer_ID: 1, Produced: 106 into Buffer 1
Consumer_ID: 4 , Consumed: 106 from Buffer 1
Producer_ID: 4 , Produced: 1064 into Buffer 2
Consumer_ID: 6 ,Consumed: 1064 from Buffer 2
Producer_ID: 2, Produced: 207 into Buffer 1
Consumer_ID: 3 , Consumed: 207 from Buffer 1
```

```
Consumer_ID: 6 ,Consumed: 1154 from Buffer 2
Producer ID: 1, Produced: 119 into Buffer 1
Consumer ID: 4 , Consumed: 117 from Buffer 1
Producer ID: 4 , Produced: 1174 into Buffer 2
Consumer ID: 3 , Consumed: 118 from Buffer 1
Producer ID: 3 , Produced: 1183 into Buffer 2
Consumer_ID: 5 ,Consumed: 1163 from Buffer 2
Consumer_ID: 6 ,Consumed: 1174 from Buffer 2
Consumer_ID: 3 , Consumed: 119 from Buffer 1
Producer ID: 3 , Produced: 1193 into Buffer 2
All producers and consumers have finished.
[bbaloch2@ecs-pa-coding1 csc139]$
```

Observation:

After a few runs, we noticed that each producer was able to produce 20 items successfully, and each consumer was able to consume 20 items. Specifically, we noticed that for each ID, there were 20 lines of activity that corresponded to that ID. What this indicated was that each of the producers/consumers had 20 activities they did before the program gracefully terminated. This ended with a total of 160 lines.The output showed correct synchronization for the actions done. Also, there is a single ID associated with both a producer and consumer. For example, ID 3 has lines of consuming from buffer 1 and producing to buffer 2. ID 4 also does the same thing. This observation does match what I intended to accomplish, as I see that 20 items are produced and consumed and the behavior of each ID is performing what it is described to do in the buffers 1 and 2.

Challenges and Limitations:

Some of the challenges that we faced were properly synchronizing interaction between threads and managing resources. The race condition was one of such synchronization problems, since we had to prevent data corruption using synchronization mechanisms such as mutex locks and semaphores which are generally tricky to keep track of with bigger data. Another challenge was managing the buffer correctly, ensuring that consumers wait if Buffer 1 is empty or if Buffer 2 is

full. Another challenge was understanding the parameters that go into the functions that we were given in the project code. By reading through the given functions, it was then understandable for what values to give to parameters for the given functions.