

Simon Fraser University
School of Computing Science
The Producer-Consumer model
CMPT-300: Operating System I
Date: November 4, 2014
Prepared for: Steven Pearce

Prepared by: Worawee Janpongsri, 301205826

In this program, the class “queue” is defined to implement the fixed size buffer, output queue, require queue(to contain material) and tools queue(to contain tools). Inside the class queue, there is another class “node”. Each node contains both data and the “next” pointer to form a simple linked list.

Generator: In the function “generators”, before the while loop there is a variable “tid” to store the thread ID of the generator thread naming 1, 2, and 3. In the while loop, there is a condition check if the user wants to pause the program. If so, all generator threads wait for the user to press key to resume. Else the mutex M of the fixed size buffer (global linked list queue) is locked before entering the critical section. The material that each generator produces is equal to thread ID. Suppose generator with thread ID 1 produce material 1, thread ID 2 produces material 2 and similarly with thread ID 3. Before putting the material in the fixed size buffer, there is a condition to check if the size of the fixed buffer is full. If so, then a particular thread has to wait till the operator thread takes out the material. Second condition to check if there too many of that material (if there is more than 4) and if the difference of the materials are more than 3, then the particular thread waits. Else, enqueue the material inside the fixed size buffer and count how many of that material are produced. Then unlock the mutex M of fixed size buffer.

Deadlock occurs:

- 1) When the fixed size buffer is full. Solution is to wait till one operator takes out one material then wakes up one of the generator that has been waiting.
- 2) When there are too many of that material in the buffer. Solution is to wait till the number of the material of the kind is less than 4 and the difference of the materials in the buffer is less than 3, then wakes up that generator in order to prevent too many of the same products.

Operator: Same as the function “generators”, there is a variable “tid” to store the thread ID of the generators’ thread naming 1, 2, 3 and so on (depend on the user). Define two linked list

queues for each operator thread namely “require” to contain two different materials and “tools” to contain two tools. Then in the while loop, there is a condition check if the user wants to pause the program. If so, all generator threads wait for the user to press key to resume. Else the mutex M of the fixed size buffer (global linked list queue) is locked before entering the critical section. There is a condition to check if the fixed size buffer is empty, if so, then wait till generator thread puts one material in the buffer. Check what is the size of the require queue. If it is zero then get first material. If it is one then get second material. If the second material is same as the first material, put back in to the buffer. If they are different then check if product that the operator is going to produce (not yet produced) is similar with the previous product that just enqueued into the output queue (linked list queue). If they are similar, enqueue the first material back to the buffer. If they are different get the two random tools, if the mutex M1 of the tool is still unlocked. After getting both two different materials and two different tools produces the product. The product is 3 if it is made of material 1 and 2. The product is 4 if it is made of material 1 and 3. The product is 5 if it is made of material 2 and 3. Then enqueue into the output queue and unlock the mutex M and mutex M1 of the tools. Count a particular product that is enqueued.

Deadlock occurs:

- 1) When the fixed size buffer is empty. Solution is to wait till one operator puts one material in the buffer then wakes up one of the operator that has been waiting.
- 2) When the recent product is same as new product. Solution is to wait till different product is insert in the output queue then insert particular product.
- 3) When the number of the difference between products are more than 10. Solution is not to produce that product by putting back the material into the buffer.
- 4) When two materials are of the same kind. Solution is to put back the first material back into the buffer.

In main function: Create three generators and user defined operators. Then enter the while loop to check if the user press any on the keyboard. If the user presses “p” and “enter”, pause all the threads. If the user presses “r” and “enter”, resume whatever all the threads are doing before they got paused. If the user presses “q” and “enter”, quit the program.