**Question 1:**

Write C++/C code for a program that takes as command line argument a source file name. The program creates a producer thread which reads each time 20 character from the file and writes the characters to a shared memory buffer. The shared memory buffer can store at most 20 characters. Now there is another thread which acts as a consumer, and reads those 20 characters from shared memory, prints the characters on the screen, and waits for the user to press enter key. The producer can only write the next 20 characters of file to shared memory after the consumer has read the previous 20 characters from the shared memory. So, you need to synchronize the two threads using semaphores. When the file has been complete written to the shared memory, the producer writes a $ to shared memory which indicates to the consumer that file has been finished. (Assume the file’s size is always multiple of 20, such as 20, 40, 60, etc.)

**For example:** Suppose there is a file of 40 characters.

* Producer writes 20 characters and waits for the consumer to read these 20 characters.
* Consumer reads 20 characters, prints on the screen and waits for user to press enter key.
* Producer now writes the next 20 characters and waits for the consumer to reads the 20 characters.
* The consumer reads the 20 characters, prints the data on the screen, and waits for the user to press enter key.
* The end-of-file has been reached, so the producer writes $ to the shared memory.
* The consumer reads $, which indicates that the file’s data has completely been read.

**Question 2:**

There are exactly 3 threads generate sa, b and c in arbitrary order. In the absence of any synchronization mechanism there will be no order in the generation of a, b and c. In the form of regular expression the string (a | b | c)\* {\* means many times a character an occur, | means or, so different combinations can be aaaaaaa… , bbbbbbbbbbb… , ccccc… }. Synchronize threads using semaphore in such a way that your printed string will be (cbbba)\* {\* means many times cbbba can occur, so different combinations will be cbbbacbbbacbbbacbbba….}.

**Note:** you are not allowed to add or delete any cout statement

|  |  |  |
| --- | --- | --- |
| //thread 1  While(1)  {    Cout << ‘a’;  } | //thread 1  While(1)  {  Cout << ‘b’;  } | //thread 1  While(1)  {  Cout << ‘c’;  } |