Shared Circular Buffer Homework problem (assignment #3) for ICS 462 Operating Systems

This assignment is worth 15 points and is due by 11:59 pm June 11, 2019.

This is an extension of assignment #2; make a C / C++, Java, or Python program with two processes, a producer and a consumer.

The producer and consumer will share 4 items. An integer array 5 long which is a circular buffer. A producer index (an integer initialized to 0) that points to the last item placed in the buffer by the producer. A consumer index (an integer initialized to 0) points to the last item removed from the shared circular buffer. The last shared variable is an “I’m done” flag, which initialized to 0 and then set to 1 when the producer places the last item in the buffer. Note, when the producer index and consumer index are equal, the circular buffer is empty.

The producer process consists of a loop that writes the loop count (a value from 0 to 99) into the shared buffer. On each pass through the loop, before the producer writes into the shared variable, it does a random wait of from one to five seconds (compute a new random wait value on each pass through the loop). The loop is to be executed 100 times. Each time through the loop, it places the loop count into the shared circular buffer, if there is an available slot in the buffer, and updates the producer index to point to the last item entered into the buffer (note that this will have to rap around as the array is only 5 items long). The producer must not clobber any item in the buffer that has not been read by the consumer. If the list is full, the producer must wait for the consumer to remove one or more items from the list before placing the next item in it. This wait is inside the main loop and must have a 1 second sleep in it. The buffer may get full because the producer may be faster than the consumer at times.

The consumer process consists of a loop that reads from the shared circular buffer until the “I’m done” flag is set. On each pass through the loop, before it reads from the shared buffer, the consumer does a random wait of from two to four seconds (compute a new random value on each pass through the loop). It will only read a value from the shared circular buffer if there is one in the buffer it hasn’t read. If there is nothing to read, it must wait until there is something to read. When it has to wait it should write “consumer waiting” into the output file. When waiting it should do so in a loop that has a 1 second sleep in it and write the “consumer waiting” message into the file each pass through this wait loop. It has to compare the consumer index and the producer index to determine if there is anything to read. After the consumer reads the value from the shared buffer, it writes it into an output file and updates the consumer index it shares with the producer (again note that this has to take into consideration the fact that this index will rap abound). When it completes, the consumer writes the phrase “Consumer done” into the output file.

Place your code, a copy of the output file from your run for in the D2L Dropbox for the 3rd assignment. Also include a word or text file with your observations on the assignment: What if anything did you learn doing this assignment? Roughly how long (in hours) did the assignment take?

Put comments at the top of your program with your name, the date, the assignment number, and a brief description of the program. I also want to see comments within the program. The first items in your data file and observations file must be your name and the assignment number (have your program output this information into your output file). You may zip your files together, but do not send me a tar file or another type of compression file or place the files out on the Internet or the Cloud for me to pick up.