

Assignment (Shared Memory and Semaphore)

In this program, you'll write a program to solve the m -producer n -consumer problem, $m, n \geq 1$. You have a shared circular buffer that can hold 20 integers. Each of the producer processes stores the numbers 1 to 50 in the buffer one by one (in a for loop with 50 iterations) and then exits. Each of the consumer processes reads the numbers from the buffer and adds them to a shared variable SUM (initialized to 0). Any consumer process can read any of the numbers in the buffer. The only constrain is that every number written by some producer should be read exactly once by exactly one of the consumers. Of course, a producer should not write when the buffer is full and a consumer should not read when the buffer is empty.

Write a program that first creates the shared circular buffer and the shared variable SUM using the `shm*()` calls in Linux. You can create any other shared variable that you think you may need. The program then reads in the value of m and n from the user, and forks m producers and n consumers. The producer and consumer codes can be written as functions that are called by the child processes. After all the producers and consumers have finished (the consumers exit after all the data produced by all the producers have been read. How does a consumer know this?), the parent process prints the value of SUM. Note that the value of SUM should be $m \times 25 \times 51$ if your program is correct. Test your program with at least (a) $m=1, n=1$, (b) $m=1, n=2$, (c) $m=2, n=1$, and (d) $m=2, n=2$.