OPERATING SYSTEMS LAB.	NAME:- STEPHEN. B. MHETRE YEAR:- II SEM:-04
EXPERIMENT NO: 06	ROLL NO:- 04.

AIM:- Implementation of Producer-Consumer Problem.

THEORY:-

The producer-consumer problem is a classic example of a process synchronization problem. It deals with how to coordinate access to a shared resource; in this case, a buffer of limited size, between two processes: a producer and a consumer.

Components:

- Producer:
  This process continuously generates items (data) and adds them to the shared buffer.
- Consumer:
  This process removes items from the shared buffer and consumes them (processes them further).
- Shared Buffer:
  This is a memory location of fixed size that acts as a temporary storage area for the items exchanged between the producer and consumer.

Synchronization Techniques

To ensure smooth operation, we need synchronization mechanisms. Here are some common approaches:

- Semaphores:

→ These are integer variables used for signaling between processes.

→ We can use two semaphores:

- Empty: counts the number of empty slots in the buffer.
- Full: counts the number of items in the buffer.

→ The producer decrements the empty semaphore before adding an item and increments the full semaphore. Conversely, the consumer increments the empty semaphore before removing an item and decrements the full semaphore. conversely, the consumer increments the empty semaphore before removing an item and decrements the full semaphore. This ensures mutual exclusion and proper buffer utilization.

- Monitors:

A higher-level construct that provides mutual exclusion and allows for signaling between processes within a critical section of code. The producer & consumer would acquire the monitor lock before accessing the buffer, ensuring exclusive access.