Ex. No.: 8 **231901039**

Date:19/3/2025

PRODUCER CONSUMER USING SEMAPHORES

Aim: To write a program to implement solution to producer consumer problem using semaphores.

Algorithm:

1. Initialize semaphore empty, full and mutex.
2. Create two threads- producer thread and consumer thread.
3. Wait for target thread termination.
4. Call sem_wait on empty semaphore followed by mutex semaphore before entry into critical section.
5. Produce/Consume the item in critical section.
6. Call sem_post on mutex semaphore followed by full semaphore 7. before exiting critical section.
8. Allow the other thread to enter its critical section.
9. Terminate after looping ten times in producer and consumer Threads each.

Program Code:

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
#define SIZE 5

int buffer[SIZE];
int in = 0, out = 0;

sem_t empty, full, mutex;

void* producer(void* arg) {
    for (int i = 0; i < 10; i++) {
        int item = i + 1;
        sem_wait(&empty);
        sem_wait(&mutex);

        buffer[in] = item;
        printf("Producer produced item %d at
index %d\n", item, in);
        in = (in + 1) % SIZE;

        sem_post(&mutex);
        sem_post(&full);
        sleep(1); // Simulate time taken to
produce
    }
    return NULL;
}

void* consumer(void* arg) {
    for (int i = 0; i < 10; i++) {
        sem_wait(&full);
        sem_wait(&mutex);

        int item = buffer[out];
        printf("Consumer consumed item %d
from index %d\n", item, out);
        out = (out + 1) % SIZE;

        sem_post(&mutex);
        sem_post(&empty);

        sleep(2); // Simulate time taken to
consume
    }
    return NULL;
}

int main() {
    pthread_t prod, cons;

    // Initialize semaphores
    sem_init(&empty, 0, SIZE);
    sem_init(&full, 0, 0);
    sem_init(&mutex, 0, 1);

    // Create threads
    pthread_create(&prod, NULL, producer,
NULL);
    pthread_create(&cons, NULL, consumer,
NULL);

    // Wait for threads to finish
    pthread_join(prod, NULL);
    pthread_join(cons, NULL);

    // Destroy semaphores
    sem_destroy(&empty);
    sem_destroy(&full);
    sem_destroy(&mutex);

    return 0;
}
```

Sample Output:

```
        1. Producer
        2.Consumer
        3.Exit
        Enter your choice:1
        Producer produces the item 1
Enter your choice:2
Consumer consumes
item 1 Enter your
choice:2  Buffer is
empty!!
        Enter your choice:1
        Producer produces the item 1
        Enter your choice:1
        Producer    produces
the    item    2    Enter    your
choice:1
        Producer
produces   the   item   3
Enter    your    choice:1
Buffer is full!!
        Enter your choice:3
```

Result:

solution to producer consumer problem using semaphores has been implemented using C.