EX.NO: 4 **IMPLEMENTATION OF SEMAPHORES**

Date : 12.08.2024

AIM:

To write a C program to implement PCP (Producer Consumer Problem) using semaphores.

ALGORITHM:

Step1: Read size of buffer
Step2: Producer process produces and buffers the items using shmget() and shmctl()
Step3: Consumer process consumes item from buffer using semop() and semrel()
Step4: Producer process waits if buffer is full and consumer process waits if buffer is empty.

PROGRAM:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/shm.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <stdlib.h>

union semun
{
   int val;
   struct semid_ds *buf;
   unsigned short *array;
};
void semaphoreOperation(int semid, int sem_num, int op)
{
   struct sembuf sb;
   sb.sem_num = sem_num;
   sb.sem_op = op;
   sb.sem_flg = 0;
   semop(semid, &sb, 1);
}
int main() {
   int n, sid, k;
   int *buffer;
   int in = 0, out = 0;
   printf("Enter the size of the buffer: ");
   scanf("%d", &n);
   sid = semget(IPC_PRIVATE, 3, IPC_CREAT | 0666);
   if (sid == -1) {
      perror("Semaphore creation failed");
      exit(1);
   }
```

```
    union semun arg;
    arg.array = (unsigned short *)malloc(3 * sizeof(unsigned short));
    arg.array[0] = n; // Semaphore for empty slots
    arg.array[1] = 0; // Semaphore for full slots
    arg.array[2] = 1; // Mutex semaphore
    semctl(sid, 0, SETALL, arg);
    k = shmget(IPC_PRIVATE, n * sizeof(int), IPC_CREAT | 0666);
    buffer = (int *)shmat(k, NULL, 0);
    if (fork() > 0) { // Producer Process
        for (int i = 0; i < n; i++) {
            int item;
            printf("Enter an item to produce: ");
            scanf("%d", &item);
            semaphoreOperation(sid, 0, -1);
            semaphoreOperation(sid, 2, -1);
            buffer[in] = item;
            in = (in + 1) % n;
            semaphoreOperation(sid, 2, 1);
            semaphoreOperation(sid, 1, 1);
        }
        shmdt(buffer);
        shmctl(k, IPC_RMID, NULL);
        semctl(sid, 0, IPC_RMID, 0);
    }
else
{
        for (int j = 0; j < n; j++)
{
            int item;
            semaphoreOperation(sid, 1, -1);
            semaphoreOperation(sid, 2, -1);
            item = buffer[out];
            out = (out + 1) % n;
            semaphoreOperation(sid, 2, 1);
            semaphoreOperation(sid, 0, 1);
            printf("Consumer consumed %d\n", item);
        }
        shmdt(buffer);
        shmctl(k, IPC_RMID, NULL);
        semctl(sid, 0, IPC_RMID, 0);
    }

    free(arg.array);
    return 0;
}
```

OUTPUT:

```
Enter the size of the buffer: 3
Enter an item to produce: 5
Enter an item to produce: 10
Enter an item to produce: 15
Consumer consumed 5
Consumer consumed 10
Consumer consumed 15
```

RESULT:

Thus the program to implement PCP (Producer Consumer Problem) using semaphores was executed successfully.