Ex. No.: 8
Date: 23.4.24

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/Consumer the item in critical section.
6. Call sem_post on mutex semaphore followed by full semaphore before exiting critical section.
7. Allow the other thread to enter its critical section.
8. Terminate after looping ten times in producer and consumer threads each.

Program Code: // vi pcsem.c

```
#include <stdio.h>
#include<stdlib.h>
int mutex=1, full=0, empty=3, x=0;
int wait(int s){
   return (--s);
}
int signal(int s){
   return (++s);
}
void producer(){
   mutex=wait(mutex);
   full=signal(full);
   empty=wait(empty);
   x++;
   printf("\n producer produces item %d", x);
   mutex=signal(mutex);
}
void consumer(){
   mutex=wait(mutex);
   full=wait(full);
   empty=signal(empty);
   printf("\n consumer consumes item %d", x);
   x--;
   mutex=signal(mutex);
}
```

```
int main(){
    int n;
    printf("\n1.Produce\n2.consume\n3.EXIT");
    while(1){
        printf("\nEnter your choice: ");
        scanf("%d",&n);
        switch(n){
            case 1: if((mutex==1)&&(empty!=0))
                        producer();
                    else
                        printf("Buffer is full!");
                    break;
            case 2: if((mutex==1)&&(full!=0))
                        consumer();
                    else
                        printf("Buffer is empty!");
                    break;
            case 3:
                    exit(0);
                    break;
        }
    }
    return 0;
}
```

Output: gcc pcsem.c
./a.out

1. Produce
2. Consume
3. Exit
Enter your choice: 1
produced produces item 1

Enter your choice: 1
produced produces item 2
Enter your choice: 1
producer produces item 3

enter your choice: 1
Buffer is full!

Enter your choice: 2
consumer consumes item 3
enter your choice: 2
consumer consumes item 2
Enter your choice: 2

consumer consumes item 1
Enter your choice: 2
Buffer is empty!

enter your choice: 3

— x —
EXIT

RESULT:

The program has been compiled and executed and the output has been verified successfully.