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 < didio. h >
# include < semaphore.h>
# define BUFFER_SIZE 5
ant buffer[BUFFER_SIZE];
int " = 0, out = 0;
int "stem = 1;
"int empty = BUFFER_SIZE;
int full =0;
void peroducer () {
        privil ("Buffer "is full cannot produce. In").
  if (emply = =0)[
      y return;
      buffer [in] = item;
      pourt ( * produced : % d \n" ; item )
      item ++;
```

```
in = (in+1) % BUFFER_SIZE;
                    o anilio
   empty --;
Jull ++;
 void consumer(){
      2 ( full = = 0) {
           point ("Buffer is empty! cannot
                                 consume !\n");
           nuture:
       "int consumed - item = buffer [out];
        print ("Consermer consumed: %od/n",
                               consumed - tem);
        out = (oud +1) % BUFFER_SIZE;
        full --;
       empty++;
    int main () ?
       print ( "Producer - consumer Problem to In");
       'unt choice;
       3 (1) elister
            privide ("In 1. Produce Item In 2. Consume
                                Item \n 3. Exit \n Enter
                                          choice"):
            scary ("%d", & chince);
            switch (choice) {
                 case 1:
                   54 peroducet().
                      break;
                Case 2:
                       consumer ().
                       break;
```

Case 3: : ("n/ ... margored pritis =") prived : o neutose defautt: pound (" Invalid choice, Please tory agoin \n"). : () Harris my boy 3(0 - : lluf) JE some shall a withing I know · (a/ smorra · control . : Las Trespect and bonevas in printy ("Converment consumed") plant Consumed banning. (out +1) 9 60111 R- STEEL! · · · · · · · · · · · · · (++ jutgin) 3 (ainn tas · (" ri/ en melelarg Herrison - Kembreg) juing white the energy of 11/ mets subort 111") fring Netra 1/tex 3.5 most met · (" wint . (with a "box") freed. I (was) which : 1940) · [Membarg : Yest Cake 2 . · () Kenuscai · woods

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

OUTPUT!

- 1. Produce Tem
- 2. consume Item
- 3. Exit

ENER your choice: 1

Peroducer produced: 1

Enter your choice: 1

Peroducer Peroduced: 2

Enter your choice: 2

Consumer Consumed:

Enter your choice: 1

Producer Produced: 3 Enter Choice: 2

Consumer consumed: 2

Enler choice: 3

Exiting program ...

Hence the producer consumer wing semaphore has been implemented and executed

Successfully