Ex. No.: 8 Date: 29 3 25

PRODUCER CONSUMER USING SEMAPHORES

Cal They si readed Jakerd

restured muters, various extra handly

JIVER OF WALLE

1 11111 market

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

Algorithm:

- 1. Initialize semaphore empty, full and mutex.
- Create two threads- producer thread and consumer thread.

 West for the second sec
- 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 (stallib. h)
# include 4 thread . hs
# include ( semaphore . h)
# include ( unistal his
# define BUPFER SIZE 3
 int buffer [BUFFER_SIZE];
 int count =0;
 sem't empty;
 sem t full;
 Ptrovead matex t mentex
  void * producer (void * ang)
    Static int items; comments and warm handy
    pthread_mutex_lock (&mutex);
    if (count == BUFFER_SIZE)
```

```
print ("Butter is full! "\n"),
    pthouad_muter_unlock(& mutex);
    netwon NULL;
  pthread_mutex_unlock(& mutex);
 sem watt ( & empty);
 pthread_mutex_lock (& mutex);
 buffer[count] = item.
 printe ("Producer Produces then item %d\n", item);
 item++;
 count++;
 pthread_mutex unlock (& mutex);
 sem_port(2 full);
 netwon NULL:
· void * consumor (void * arg) $
  if (count ==0) $
      printf (" Buffer is empty !! \n"); howard & encount
     network NULL:
Sem_walt (& full);
pthread_mutex_lock (& mutex);
if (count >0) s
     int item = butter [count-1];
     Printf ("Consumer consumes "tem % d/n", "tem).
                           GOOD LION ROUBERT & JOICH
    court --;
pthorad_mutex_unlock (omutex);
                         Admin's to Fathery bowers
sem_port(& empty);
return NULL;
```

```
int main ()
    pthread_t prodthread, cons Thread;
    int choice;
    sem_inct ( wempty, O, BUFFER_SIZE);
   sem_ net ( & full, 0,0);
   pthoreochmutex intit (umutex, NULL);
   while(1)
     prints ( 'In 1. Producer In 2. Consumos In 3. Exit In Enter your choice)
     scanf ("% d' & choice).
    switch (choice)
      case 1:
       Pthoread_create (& prodThoread, NULL, producer, NULL);
       pthread_join (proditioned, NULL);
       break ?
      Case 2:
        Pthread_create (& consthuad, NULL, consumor, NULL).
        pthoread_join (consTroved, NULL); break;
     case 3:
         Printf (" Exiting . . . \n");
         sem_destroy (& empty);
        sem destroy (& full);
        pthoroad_mutex_destroy (& mutex);
        exit (0);
    defauat:
        prints ("Invalid choice!");
outwon o;
```

Output:

1. Producer 2-consumor 3- Frat Producer produces item! 4 reducers related to the reducer producers Enter your choice:1 Enter your choice: 2 consumon consumes item! Enter your choice: 2 Bugger is empty !! Enter your choice: 1 Produces produces item 2 Enter your choice: 1 Producer produces item ? Enter your choice:1 Producos Produces Item 4 Enter your choice:1 Bugger is fuel Enter your choice: 3 Exiting ...

Sample Output:

solodo Jui

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:

Hence program to implement solution to produced using semaphores has been executed consumer problem Successfully.