**SSN College of Engineering, Kalavakkam**

**Department of Computer Science and Engineering**

**Semester - IV**

**UCS1411 – Operating Systems Lab**

**Academic Year: 2019-2020 Batch: 2018-2022**

**Name : Rahul Ram M**

**Class : CSE – B**

**Register Number : 185001121**

**Lab Exercise 6: Implementation of Producer/Consumer Problem using Semaphores**

**CODE:**

#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <sys/sem.h>  
#include <sys/wait.h>  
#include <sys/errno.h>  
#include <sys/types.h>  
extern int errno;  
#define SIZE 10                      /\*size of the shared buffer\*/  
  
#define VARSIZE 1                 /\*size of shared variable=1byte\*/  
  
#define INPUTSIZE 20  
  
#define SHMPERM 0666          /\*shared memory permissions \*/  
int segid;                                /\* id for shared memory bufer \*/  
int empty\_id;  
int full\_id;  
int mutex\_id;  
int \* buff;  
int input\_string;  
sem\_t \*empty;  
sem\_t \*full;  
sem\_t \*mutex;  
int p=0,c=0;  
  
// Producer function  
  
void produce()  
{  
if(input\_string==0)  
{  
printf("\n Producer %d exited \n",getpid());  
wait(NULL);  
exit(1);  
}  
srand(time(0));  
for(int i=0;i<input\_string;i++)  
{  
printf("\nProducer %d trying to aquire Semaphore Empty \n",getpid());  
sem\_wait(empty);  
printf("\nProducer %d successfully aquired Semaphore Empty \n",getpid());  
printf("\nProducer %d trying to aquire Semaphore Mutex \n",getpid());  
sem\_wait(mutex);  
printf("\nProducer %d successfully aquired Semaphore Mutex \n",getpid());  
buff[p]=random()%input\_string;  
printf("\nProducer %d Produced Item [ %d ] \n",getpid(),buff[p]);  
p++;  
printf("\nItems in Buffer %d \n",input\_string);  
sem\_post(mutex);  
printf("\nProducer %d released Semaphore Mutex \n",getpid());  
sem\_post(full);  
printf("\nProducer %d released Semaphore Full \n",getpid());  
}  
} //producer  
  
int main()  
{  
int i=0;  
int id1;  
pid\_t temp\_pid;  
segid = shmget (1001, SIZE, IPC\_CREAT | IPC\_EXCL | SHMPERM );  
empty\_id=shmget(1002,sizeof(sem\_t),IPC\_CREAT|IPC\_EXCL|  
SHMPERM);  
full\_id=shmget(1003,sizeof(sem\_t),IPC\_CREAT|IPC\_EXCL|  
SHMPERM);  
mutex\_id=shmget(1004,sizeof(sem\_t),IPC\_CREAT|IPC\_EXCL|  
SHMPERM);  
buff = shmat( segid, (int \*)0, 0 );  
empty = shmat(empty\_id,(int \*)0,0);  
full = shmat(full\_id,(int \*)0,0);  
mutex = shmat(mutex\_id,(int \*)0,0);  
// Initializing Semaphores Empty , Full & Mutex  
sem\_init(empty,1,SIZE);  
sem\_init(full,1,0);  
sem\_init(mutex,1,1);  
printf("\n Main Process Started \n");  
printf("\n Enter the no of numbers : ");  
  
id1=shmget(112,50,IPC\_CREAT | 00666);  
input\_string=shmat(id1,NULL,0);  
scanf("%d",&input\_string);  
printf("number of numbers : %d",input\_string);  
produce();  
shmdt(input\_string);  
shmdt(buff);  
shmdt(empty);  
shmdt(full);  
shmdt(mutex);  
return(0);  
} //main  
  
  
//consumer :  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <semaphore.h>  
#include <pthread.h> // for semaphore operations sem\_init,sem\_wait,sem\_post  
#include <sys/ipc.h>  
#include <sys/shm.h>  
#include <sys/sem.h>  
#include <sys/wait.h>  
#include <sys/errno.h>  
#include <sys/types.h>  
extern int errno;  
#define SIZE 10 /\* size of the shared buffer\*/  
#define VARSIZE 1 /\* size of shared variable=1byte\*/  
#define INPUTSIZE 20  
#define SHMPERM 0666 /\* shared memory permissions \*/  
int segid; /\* id for shared memory bufer \*/  
int empty\_id;  
int full\_id;  
int mutex\_id;  
int \* buff;  
int input\_string;  
sem\_t \*empty;  
sem\_t \*full;  
sem\_t \*mutex;  
int p=0,c=0;  
// Consumer function  
//  
void consume()  
{  
  
if(input\_string==0)  
{  
printf("\n Producer %d exited \n",getpid());  
wait(NULL);  
exit(1);  
}  
for(int j=0;j<input\_string;j++)  
{  
printf("\nConsumer %d trying to aquire Semaphore Full \n",getpid());  
sem\_wait(full);  
printf("\nConsumer %d successfully aquired Semaphore Full \n",getpid());  
printf("\nConsumer %d trying to aquire Semaphore Mutex \n",getpid());  
sem\_wait(mutex);  
printf("\nConsumer %d successfully aquired Semaphore Mutex\n",getpid());  
printf("\nConsumer %d Consumed Item [ %d ] \n",getpid(),buff[c]);  
buff[c]=' ';  
c++;  
sem\_post(mutex);  
printf("\nConsumer %d released Semaphore Mutex \n",getpid());  
sem\_post(empty);  
printf("\nConsumer %d released Semaphore Empty \n",getpid());  
}  
return;  
} //consumer fn  
  
//main function  
  
int main()  
{  
int i=0;  
int id1;  
pid\_t temp\_pid;  
segid = shmget (1001, SIZE,0);  
empty\_id=shmget(1002,sizeof(sem\_t),0);  
full\_id=shmget(1003,sizeof(sem\_t),0);  
mutex\_id=shmget(1004,sizeof(sem\_t),0);  
buff = shmat( segid, (int \*)0, 0 );  
empty = shmat(empty\_id,(int \*)0,0);  
full = shmat(full\_id,(int \*)0,0);  
mutex = shmat(mutex\_id,(int \*)0,0);  
id1=shmget(112,50,0);  
input\_string=shmat(id1,NULL,0);  
// Initializing Semaphores Empty , Full & Mutex  
consume();  
shmdt(buff);  
shmdt(empty);  
shmdt(full);  
shmdt(mutex);  
shmctl(segid, IPC\_RMID, NULL);  
semctl( empty\_id, 0, IPC\_RMID, NULL);  
semctl( full\_id, 0, IPC\_RMID, NULL);  
semctl( mutex\_id, 0, IPC\_RMID, NULL);  
sem\_destroy(empty);  
sem\_destroy(full);  
sem\_destroy(mutex);  
printf("\n Main process exited \n\n");  
return(0);  
} //main  
  
  
OUTPUT :  
  
PRODUCER OUTPUT :  
  
Main Process Started  
  
Enter the no of numbers : 5  
number of numbers : 5  
Producer 3560 trying to aquire Semaphore Empty  
  
Producer 3560 successfully aquired Semaphore Empty  
  
Producer 3560 trying to aquire Semaphore Mutex  
  
Producer 3560 successfully aquired Semaphore Mutex  
  
Producer 3560 Produced Item [ 3 ]  
  
Items in Buffer 5  
  
Producer 3560 released Semaphore Mutex  
  
Producer 3560 released Semaphore Full  
  
Producer 3560 trying to aquire Semaphore Empty  
  
Producer 3560 successfully aquired Semaphore Empty  
  
Producer 3560 trying to aquire Semaphore Mutex  
  
Producer 3560 successfully aquired Semaphore Mutex  
  
Producer 3560 Produced Item [ 1 ]  
  
Items in Buffer 5  
  
Producer 3560 released Semaphore Mutex  
  
Producer 3560 released Semaphore Full  
  
Producer 3560 trying to aquire Semaphore Empty  
  
Producer 3560 successfully aquired Semaphore Empty  
  
Producer 3560 trying to aquire Semaphore Mutex  
  
Producer 3560 successfully aquired Semaphore Mutex  
  
Producer 3560 Produced Item [ 2 ]  
  
Items in Buffer 5  
  
Producer 3560 released Semaphore Mutex  
  
Producer 3560 released Semaphore Full