EX:6 PRODUCER-CONSUMER PROBLEM

-S.Vishakan CSE-C 18 5001 196

SOURCE CODE – (Parent & Child):

```
#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 buffer */
int empty_id;
int full_id;
int mutex_id;
char *buff;
char *input_string;
sem_t *empty;
sem_t *full;
sem_t *mutex;
int p=0,c=0;



void produce(){ // Producer function
```

```
int i=0;
while (1){
if(i>=strlen(input_string)){
printf("\n Producer %d exited. \n",getpid());
wait(NULL);
exit(1);
}

printf("\nProducer %d trying to acquire Semaphore Empty. \n",getpid());
sem_wait(empty);

printf("\nProducer %d successfully acquired Semaphore Empty. \n",getpid());

printf("\nProducer %d trying to acquire Semaphore Mutex. \n",getpid());
sem_wait(mutex);

printf("\nProducer %d successfully acquired Semaphore Mutex. \n",getpid());
buff[p]=input_string[i];

printf("\nProducer %d Produced Item: [ %c ] \n",getpid(),input_string[i]);
i++;
p++;

printf("\nItems in Buffer: %d \n",p);
sem_post(mutex);

printf("\nProducer %d released Semaphore Mutex. \n",getpid());
sem_post(full);

printf("\nProducer %d released Semaphore Full. \n",getpid());
sleep(2/random());

}
}

void consume(){ // Consumer function
int i=0;
while (1){
if(i>=strlen(input_string)){
printf("\n Consumer %d exited. \n",getpid());
exit(1);
}
printf("\nConsumer %d trying to acquire Semaphore Full. \n",getpid());
sem_wait(full);
printf("\nConsumer %d successfully acquired Semaphore Full. \n",getpid());

printf("\nConsumer %d trying to acquire Semaphore Mutex. \n",getpid());
sem_wait(mutex);

printf("\nConsumer %d successfully acquired Semaphore Mutex.\n",getpid());

printf("\nConsumer %d Consumed Item: [ %c ] \n",getpid(),buff[c]);
buff[c]=' ';
c++;

printf("\nItems in Buffer: %d \n",strlen(input_string));
i++;
sem_post(mutex);

printf("\nConsumer %d released Semaphore Mutex. \n",getpid());
sem_post(empty);

printf("\nConsumer %d released Semaphore Empty. \n",getpid());
sleep(1);

}
}

int main(){
int i=0;
pid_t temp_pid;
segid = shmget(IPC_PRIVATE, SIZE, IPC_CREAT | IPC_EXCL | SHMPERM);
empty_id=shmget(IPC_PRIVATE,sizeof(sem_t),IPC_CREAT|IPC_EXCL|
SHMPERM);
full_id=shmget(IPC_PRIVATE,sizeof(sem_t),IPC_CREAT|IPC_EXCL|SHMPERM);
mutex_id=shmget(IPC_PRIVATE,sizeof(sem_t),IPC_CREAT|IPC_EXCL|
SHMPERM);
buff = shmat(segid, (char *)0, 0);
empty = shmat(empty_id,(char *)0,0);
full = shmat(full_id,(char *)0,0);
mutex = shmat(mutex_id,(char *)0,0);

// Initializing Semaphores Empty , Full & Mutex
sem_init(empty,1,SIZE);
sem_init(full,1,0);
sem_init(mutex,1,1);

printf("\nMain Process Started. \n");
printf("\nEnter the input string (20 characters MAX) : ");
```

```
input_string=(char *)malloc(20);
scanf("%s",input_string);
printf("Entered string : %s",input_string);
temp_pid=fork();
if(temp_pid>0){ //parent
produce();
}

else{ //child
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("\nMain process exited. \n\n");
return(0);

}
```

OUTPUT:

```
(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex6 Semaphores$ ./s

Main Process Started.

Enter the input string (20 characters MAX) : Semaphore
Entered string : Semaphore
Producer 2246 trying to acquire Semaphore Empty.

Producer 2246 successfully acquired Semaphore Empty.

Producer 2246 trying to acquire Semaphore Mutex.
Producer 2246 successfully acquired Semaphore Mutex.

Producer 2246 Produced Item: [ S ]

Items in Buffer: 1

Producer 2246 released Semaphore Mutex.

Producer 2246 released Semaphore Full.
Entered string : Semaphore
Consumer 2247 trying to acquire Semaphore Full.

Producer 2246 trying to acquire Semaphore Empty.

Producer 2246 successfully acquired Semaphore Empty.
Consumer 2247 successfully acquired Semaphore Full.

Producer 2246 trying to acquire Semaphore Mutex.
Consumer 2247 trying to acquire Semaphore Mutex.

Producer 2246 successfully acquired Semaphore Mutex.

Producer 2246 Produced Item: [ e ]

Items in Buffer: 2

Producer 2246 released Semaphore Mutex.

Producer 2246 released Semaphore Full.

Consumer 2247 successfully acquired Semaphore Mutex.
Producer 2246 trying to acquire Semaphore Empty.

Producer 2246 successfully acquired Semaphore Empty.

Consumer 2247 Consumed Item: [ S ]
Producer 2246 trying to acquire Semaphore Mutex.

Items in Buffer: 9

Consumer 2247 released Semaphore Mutex.

Consumer 2247 released Semaphore Empty.
Producer 2246 successfully acquired Semaphore Mutex.
Producer 2246 Produced Item: [ m ]
Items in Buffer: 3
Producer 2246 released Semaphore Mutex.
Producer 2246 released Semaphore Full.
Producer 2246 trying to acquire Semaphore Empty.
Producer 2246 successfully acquired Semaphore Empty.
Producer 2246 trying to acquire Semaphore Mutex.
Producer 2246 successfully acquired Semaphore Mutex.
Producer 2246 Produced Item: [ a ]
Items in Buffer: 4
Producer 2246 released Semaphore Mutex.
Producer 2246 released Semaphore Full.
Producer 2246 trying to acquire Semaphore Empty.
Producer 2246 successfully acquired Semaphore Empty.
Producer 2246 trying to acquire Semaphore Mutex.
Producer 2246 successfully acquired Semaphore Mutex.
Producer 2246 Produced Item: [ p ]
Items in Buffer: 5
Producer 2246 released Semaphore Mutex.
Producer 2246 released Semaphore Full.
Producer 2246 trying to acquire Semaphore Empty.
Producer 2246 successfully acquired Semaphore Empty.
Producer 2246 trying to acquire Semaphore Mutex.

Producer 2246 successfully acquired Semaphore Mutex.

Producer 2246 Produced Item: [ h ]

Items in Buffer: 6

Producer 2246 released Semaphore Mutex.

Producer 2246 released Semaphore Full.

Producer 2246 trying to acquire Semaphore Empty.

Producer 2246 successfully acquired Semaphore Empty.

Producer 2246 trying to acquire Semaphore Mutex.

Producer 2246 successfully acquired Semaphore Mutex.

Producer 2246 Produced Item: [ o ]

Items in Buffer: 7

Producer 2246 released Semaphore Mutex.

Producer 2246 released Semaphore Full.

Producer 2246 trying to acquire Semaphore Empty.

Producer 2246 successfully acquired Semaphore Empty.

Producer 2246 trying to acquire Semaphore Mutex.

Producer 2246 successfully acquired Semaphore Mutex.

Producer 2246 Produced Item: [ r ]

Items in Buffer: 8

Producer 2246 released Semaphore Mutex.

Producer 2246 released Semaphore Full.

Producer 2246 trying to acquire Semaphore Empty.
Producer 2246 successfully acquired Semaphore Empty.
Producer 2246 trying to acquire Semaphore Mutex.
Producer 2246 successfully acquired Semaphore Mutex.
Producer 2246 Produced Item: [ e ]
Items in Buffer: 9
Producer 2246 released Semaphore Mutex.
Producer 2246 released Semaphore Full.
Producer 2246 exited.
Consumer 2247 trying to acquire Semaphore Full.
Consumer 2247 successfully acquired Semaphore Full.
Consumer 2247 trying to acquire Semaphore Mutex.
Consumer 2247 successfully acquired Semaphore Mutex.
Consumer 2247 Consumed Item: [ e ]
Items in Buffer: 9
Consumer 2247 released Semaphore Mutex.
Consumer 2247 released Semaphore Empty.
Consumer 2247 trying to acquire Semaphore Full.
Consumer 2247 successfully acquired Semaphore Full.
Consumer 2247 trying to acquire Semaphore Mutex.
Consumer 2247 successfully acquired Semaphore Mutex.
Consumer 2247 Consumed Item: [ m ]
Items in Buffer: 9
```

```
Consumer 2247 released Semaphore Mutex.
Consumer 2247 released Semaphore Empty.
Consumer 2247 trying to acquire Semaphore Full.
Consumer 2247 successfully acquired Semaphore Full.
Consumer 2247 trying to acquire Semaphore Mutex.
Consumer 2247 successfully acquired Semaphore Mutex.
Consumer 2247 Consumed Item: [ a ]
Items in Buffer: 9
Consumer 2247 released Semaphore Mutex.
Consumer 2247 released Semaphore Empty.
Consumer 2247 trying to acquire Semaphore Full.
Consumer 2247 successfully acquired Semaphore Full.
Consumer 2247 trying to acquire Semaphore Mutex.
Consumer 2247 successfully acquired Semaphore Mutex.
Consumer 2247 Consumed Item: [ p ]
Items in Buffer: 9
Consumer 2247 released Semaphore Mutex.
Consumer 2247 released Semaphore Empty.
Consumer 2247 trying to acquire Semaphore Full.
Consumer 2247 successfully acquired Semaphore Full.
Consumer 2247 trying to acquire Semaphore Mutex.
Consumer 2247 successfully acquired Semaphore Mutex.
Consumer 2247 Consumed Item: [ h ]
```

```
Items in Buffer: 9

Consumer 2247 released Semaphore Mutex.

Consumer 2247 released Semaphore Empty.

Consumer 2247 trying to acquire Semaphore Full.

Consumer 2247 successfully acquired Semaphore Full.

Consumer 2247 trying to acquire Semaphore Mutex.

Consumer 2247 successfully acquired Semaphore Mutex.

Consumer 2247 Consumed Item: [ o ]

Items in Buffer: 9

Consumer 2247 released Semaphore Mutex.

Consumer 2247 released Semaphore Empty.

Consumer 2247 trying to acquire Semaphore Full.

Consumer 2247 successfully acquired Semaphore Full.

Consumer 2247 trying to acquire Semaphore Mutex.

Consumer 2247 successfully acquired Semaphore Mutex.

Consumer 2247 Consumed Item: [ r ]

Items in Buffer: 9

Consumer 2247 released Semaphore Mutex.

Consumer 2247 released Semaphore Empty.

Consumer 2247 trying to acquire Semaphore Full.

Consumer 2247 successfully acquired Semaphore Full.

Consumer 2247 trying to acquire Semaphore Mutex.
Consumer 2247 successfully acquired Semaphore Mutex.

Consumer 2247 Consumed Item: [ e ]

Items in Buffer: 9

Consumer 2247 released Semaphore Mutex.

Consumer 2247 released Semaphore Empty.

Consumer 2247 exited.
```

***SOURCE CODE* – (Server/Producer Program):**

```
#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 VARSIZE 1 /* size of shared variable=1byte*/
#define INPUTSIZE 20
#define SHMPERM 0666 /* shared memory permissions */

int segid; /* id for shared memory buffer */
int empty_id;
int full_id;
int mutex_id;
char *buff;
sem_t *empty;
sem_t *full;
sem_t *mutex;
int p = 0, c = 0, n = 0;

void produceRandomNumbers()
{ // Producer function to generate random numbers
int i = 0, random_num = 0;
while (i < n)
{
random_num = random() % n;

printf("\nProducer %d trying to acquire Semaphore Empty. \n", getpid());
sem_wait(empty);

printf("\nProducer %d successfully acquired Semaphore Empty. \n", getpid());

printf("\nProducer %d trying to acquire Semaphore Mutex. \n", getpid());
sem_wait(mutex);
printf("\nProducer %d successfully acquired Semaphore Mutex. \n", getpid());
buff[p] = (char)random_num + 48; //converting it to char to store in array

printf("\nProducer %d Produced Item: [ %d ] \n", getpid(), random_num);
i++;
p++;

printf("\nItems in Buffer: %d \n", p);
sem_post(mutex);

printf("\nProducer %d released Semaphore Mutex. \n", getpid());
sem_post(full);

printf("\nProducer %d released Semaphore Full. \n", getpid());
sleep(2);
}
}

int main()
{
int i = 0;
segid = shmget(100, 5, IPC_CREAT | IPC_EXCL | SHMPERM);
empty_id = shmget(101, sizeof(sem_t), IPC_CREAT | IPC_EXCL | SHMPERM);
full_id = shmget(102, sizeof(sem_t), IPC_CREAT | IPC_EXCL | SHMPERM);
mutex_id = shmget(103, sizeof(sem_t), IPC_CREAT | IPC_EXCL | SHMPERM);
buff = shmat(segid, (char *)0, 0);
empty = shmat(empty_id, (char *)0, 0);
full = shmat(full_id, (char *)0, 0);
mutex = shmat(mutex_id, (char *)0, 0);

printf("\nEnter the number of random numbers to generate: ");
scanf("%d", &n);

// Initializing Semaphores Empty , Full & Mutex
sem_init(empty, 1, n);
sem_init(full, 1, 0);
sem_init(mutex, 1, 1);

printf("\nProducer Process Started To Produce Random Numbers. \n");
produceRandomNumbers();

shmdt(buff);
shmdt(empty);
shmdt(full);
shmdt(mutex);

printf("\nProducer Process Exited. \n\n");
return (0);
}
```

OUTPUT:

```
(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex6 Semaphores$ ./p

Enter the number of random numbers to generate: 5

Producer Process Started To Produce Random Numbers.

Producer 2446 trying to acquire Semaphore Empty.

Producer 2446 successfully acquired Semaphore Empty.

Producer 2446 trying to acquire Semaphore Mutex.

Producer 2446 successfully acquired Semaphore Mutex.

Producer 2446 Produced Item: [ 3 ]

Items in Buffer: 1

Producer 2446 released Semaphore Mutex.

Producer 2446 released Semaphore Full.

Producer 2446 trying to acquire Semaphore Empty.

Producer 2446 successfully acquired Semaphore Empty.

Producer 2446 trying to acquire Semaphore Mutex.

Producer 2446 successfully acquired Semaphore Mutex.

Producer 2446 Produced Item: [ 6 ]

Items in Buffer: 2

Producer 2446 released Semaphore Mutex.
Producer 2446 released Semaphore Full.
Producer 2446 trying to acquire Semaphore Empty.
Producer 2446 successfully acquired Semaphore Empty.
Producer 2446 trying to acquire Semaphore Mutex.
Producer 2446 successfully acquired Semaphore Mutex.
Producer 2446 Produced Item: [ 7 ]
Items in Buffer: 3
Producer 2446 released Semaphore Mutex.
Producer 2446 released Semaphore Full.
Producer 2446 trying to acquire Semaphore Empty.
Producer 2446 successfully acquired Semaphore Empty.
Producer 2446 trying to acquire Semaphore Mutex.
Producer 2446 successfully acquired Semaphore Mutex.
Producer 2446 Produced Item: [ 5 ]
Items in Buffer: 4
Producer 2446 released Semaphore Mutex.
Producer 2446 released Semaphore Full.
Producer 2446 trying to acquire Semaphore Empty.
Producer 2446 successfully acquired Semaphore Empty.
Producer 2446 trying to acquire Semaphore Mutex.
Producer 2446 successfully acquired Semaphore Mutex.
Producer 2446 Produced Item: [ 3 ]
```

```
Items in Buffer: 5

Producer 2446 released Semaphore Mutex.

Producer 2446 released Semaphore Full.

Producer Process Exited.
```

SOURCE CODE – (Client/Consumer Program):

```
#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 VARSIZE 1 /* size of shared variable=1byte*/
#define INPUTSIZE 20
#define SHMPERM 0666 /* shared memory permissions */

int segid; /* id for shared memory buffer */
int empty_id;
int full_id;
int mutex_id;
char *buff;
sem_t *empty;
sem_t *full;
sem_t *mutex;
int p = 0, c = 0, buff_size = 0;

void consumeRandomNumbers()
{ // Consumer function to consume random numbers generated
int i = 0;
buff_size = strlen(buff);
```

```
while (i < buff_size)
{
printf("\nConsumer %d trying to acquire Semaphore Full. \n", getpid());
sem_wait(full);

printf("\nConsumer %d successfully acquired Semaphore Full. \n", getpid());

printf("\nConsumer %d trying to acquire Semaphore Mutex. \n", getpid());
sem_wait(mutex);

printf("\nConsumer %d successfully acquired Semaphore Mutex.\n", getpid());

printf("\nConsumer %d Consumed Item: [ %c ] \n", getpid(), buff[c]);
buff[c] = ' ';
c++;

printf("\nItems in Buffer: %d \n", (buff_size - (++i)));
sem_post(mutex);

printf("\nConsumer %d released Semaphore Mutex. \n", getpid());
sem_post(empty);

printf("\nConsumer %d released Semaphore Empty. \n", getpid());
sleep(1);
}
}

int main()
{
int i = 0;

segid = shmget(100, 5, IPC_EXCL | SHMPERM);
empty_id = shmget(101, sizeof(sem_t), IPC_EXCL | SHMPERM);
full_id = shmget(102, sizeof(sem_t), IPC_EXCL | SHMPERM);
mutex_id = shmget(103, sizeof(sem_t), IPC_EXCL | SHMPERM);
buff = shmat(segid, (char *)0, 0);
empty = shmat(empty_id, (char *)0, 0);
full = shmat(full_id, (char *)0, 0);
mutex = shmat(mutex_id, (char *)0, 0);

printf("\nConsumer Process Started To Consume Random Numbers. \n");

consumeRandomNumbers();

shmdt(buff);
shmdt(empty);
shmdt(full);
shmdt(mutex);
shmctl(segid, IPC_RMID, NULL);
shmctl(empty_id, IPC_RMID, NULL);
shmctl(full_id, IPC_RMID, NULL);
shmctl(mutex_id, 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);
//NOTE: semctl() or sem_destroy() causes seg.faults after one successful execution.

printf("\nConsumer process exited. \n\n");
return (0);
}
```

OUTPUT:

```
(base) vishakan@Legion:~/Desktop/Operating-Systems/Ex6 Semaphores$ ./c

Consumer Process Started To Consume Random Numbers.

Consumer 2507 trying to acquire Semaphore Full.

Consumer 2507 successfully acquired Semaphore Full.

Consumer 2507 trying to acquire Semaphore Mutex.

Consumer 2507 successfully acquired Semaphore Mutex.

Consumer 2507 Consumed Item: [ 3 ]

Items in Buffer: 9

Consumer 2507 released Semaphore Mutex.

Consumer 2507 released Semaphore Empty.

Consumer 2507 trying to acquire Semaphore Full.

Consumer 2507 successfully acquired Semaphore Full.
Consumer 2507 trying to acquire Semaphore Mutex.
Consumer 2507 successfully acquired Semaphore Mutex.
Consumer 2507 Consumed Item: [ 6 ]
Items in Buffer: 8
Consumer 2507 released Semaphore Mutex.
Consumer 2507 released Semaphore Empty.
Consumer 2507 trying to acquire Semaphore Full.
Consumer 2507 successfully acquired Semaphore Full.
Consumer 2507 trying to acquire Semaphore Mutex.
Consumer 2507 successfully acquired Semaphore Mutex.
Consumer 2507 Consumed Item: [ 7 ]
Items in Buffer: 7
Consumer 2507 released Semaphore Mutex.
Consumer 2507 released Semaphore Empty.
Consumer 2507 trying to acquire Semaphore Full.
Consumer 2507 successfully acquired Semaphore Full.
Consumer 2507 trying to acquire Semaphore Mutex.
Consumer 2507 successfully acquired Semaphore Mutex.
Consumer 2507 Consumed Item: [ 5 ]
Items in Buffer: 6
Consumer 2507 released Semaphore Mutex.
Consumer 2507 released Semaphore Empty.
Consumer 2507 trying to acquire Semaphore Full.

Consumer 2507 successfully acquired Semaphore Full.

Consumer 2507 trying to acquire Semaphore Mutex.

Consumer 2507 successfully acquired Semaphore Mutex.

Consumer 2507 Consumed Item: [ 3 ]

Items in Buffer: 5

Consumer 2507 released Semaphore Mutex.

Consumer 2507 released Semaphore Empty.

Consumer process exited.
```