***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.***