```
Script started on 2019-03-03 22:39:07+0530
praveen@praveen$ cat singleprogram.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <semaphore.h>
#include <pthread.h>
#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 = 1 byte */
#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;
sem_t *empty;
sem_t *full;
sem_t *mutex;
int p = 0, c = 0;
char *buff, *input_string;
//
// Producer function
//
void produce()
{
    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 produced: %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(random()%5);
    }
```

```
}
//
// Consumer function
//
void consume()
{
    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 consumed: %d \n",i+1);
        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);
    }
}
//--------------------------------------------------------------
//Main function
//---------------------------------------------------------------
int main()
{
    int i=0;
    buff = (char *)malloc(100);
    input_string = (char*)malloc(100);
    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("\n Main Process Started \n");
    printf("\n Enter 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("\n Main process exited \n\n");
    return(0);
}
praveen@praveen$ gcc singleprogram.c -pthread
praveen@praveen$ ./a.out

 Main Process Started

 Enter the input string (20 characters MAX) : COMPUTER
Entered string : COMPUTER
Producer 9278 trying to acquire Semaphore Empty

Producer 9278 successfully acquired Semaphore Empty

Producer 9278 trying to acquire Semaphore Mutex

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ C ]
Entered string : COMPUTER

Consumer 9279 trying to acquire Semaphore Full
Items produced: 1

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Consumer 9279 successfully acquired Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ C ]

Items consumed: 1

Consumer 9279 released Semaphore Mutex

Consumer 9279 released Semaphore Empty

Consumer 9279 trying to acquire Semaphore Full
```

```
Producer 9278 trying to acquire Semaphore Empty

Producer 9278 successfully acquired Semaphore Empty

Producer 9278 trying to acquire Semaphore Mutex

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ O ]

Items produced: 2

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Consumer 9279 successfully acquired Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ O ]

Items consumed: 2

Consumer 9279 released Semaphore Mutex

Consumer 9279 released Semaphore Empty


Consumer 9279 trying to acquire Semaphore Full
Producer 9278 trying to acquire Semaphore Empty

Producer 9278 successfully acquired Semaphore Empty

Producer 9278 trying to acquire Semaphore Mutex

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ M ]

Items produced: 3

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Consumer 9279 successfully acquired Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ M ]

Items consumed: 3

Consumer 9279 released Semaphore Mutex

Consumer 9279 released Semaphore Empty
Consumer 9279 trying to acquire Semaphore Full

Producer 9278 trying to acquire Semaphore Empty

Producer 9278 successfully acquired Semaphore Empty

Producer 9278 trying to acquire Semaphore Mutex

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ P ]

Items produced: 4

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Consumer 9279 successfully acquired Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex


Consumer 9279 Consumed Item [ P ]
Producer 9278 trying to acquire Semaphore Empty


Items consumed: 4
Producer 9278 successfully acquired Semaphore Empty


Consumer 9279 released Semaphore Mutex
Producer 9278 trying to acquire Semaphore Mutex

Consumer 9279 released Semaphore Empty

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ U ]

Items produced: 5

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Consumer 9279 trying to acquire Semaphore Full

Consumer 9279 successfully acquired Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ U ]

Items consumed: 5

Consumer 9279 released Semaphore Mutex
Consumer 9279 released Semaphore Empty

Consumer 9279 trying to acquire Semaphore Full

Producer 9278 trying to acquire Semaphore Empty

Producer 9278 successfully acquired Semaphore Empty

Producer 9278 trying to acquire Semaphore Mutex

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ T ]

Items produced: 6

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Producer 9278 trying to acquire Semaphore Empty

Producer 9278 successfully acquired Semaphore Empty

Producer 9278 trying to acquire Semaphore Mutex

Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ E ]

Items produced: 7

Producer 9278 released Semaphore Mutex


Consumer 9279 successfully acquired Semaphore Full
Producer 9278 released Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ T ]

Items consumed: 6

Consumer 9279 released Semaphore Mutex

Consumer 9279 released Semaphore Empty


Producer 9278 trying to acquire Semaphore Empty
Consumer 9279 trying to acquire Semaphore Full


Producer 9278 successfully acquired Semaphore Empty
Consumer 9279 successfully acquired Semaphore Full


Producer 9278 trying to acquire Semaphore Mutex
Consumer 9279 trying to acquire Semaphore Mutex
```

```
Producer 9278 successfully acquired Semaphore Mutex

Producer 9278 Produced Item [ R ]

Items produced: 8

Producer 9278 released Semaphore Mutex

Producer 9278 released Semaphore Full

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ E ]

Items consumed: 7

Consumer 9279 released Semaphore Mutex

Consumer 9279 released Semaphore Empty

Consumer 9279 trying to acquire Semaphore Full

Consumer 9279 successfully acquired Semaphore Full

Consumer 9279 trying to acquire Semaphore Mutex

Consumer 9279 successfully acquired Semaphore Mutex

Consumer 9279 Consumed Item [ R ]

Items consumed: 8

Consumer 9279 released Semaphore Mutex

Consumer 9279 released Semaphore Empty

 Producer 9278 exited

 Consumer 9279 exited
praveen@praveen$ exit
exit

Script done on 2019-03-03 22:40:15+0530
```