

Name: Taimoor

Registration no: BSCS-20F-0004

Semester: Fifth

Lab Assignment: Operating Systems

**Task 1: Run the following Code and prepare the report**

#include <stdio.h>

#include <sys/ipc.h>

#include <sys/shm.h>

#include <sys/types.h>

#include <unistd.h>

#include<sys/wait.h>

int main()

{

int shmid, \*a, \*b, i, lockid, \*la, \*lb;

shmid = shmget(IPC\_PRIVATE, 2\*sizeof(int), 0777|IPC\_CREAT);

lockid = shmget(IPC\_PRIVATE, sizeof(int), 0777|IPC\_CREAT);

if (fork() == 0)

{

b = (int \*) shmat(shmid, 0, 0);

lb = (int \*) shmat(lockid, 0, 0);

\*lb = 0;

for( i=0; i< 10; i++)

{

while(\*lb != 1);

\*lb = 1;

printf("\t\t\t Child reads: %d,%d\n",b[0],b[1]);

\*lb = 0;

}

shmdt(b);

}

else

{

a = (int \*) shmat(shmid, 0, 0);

la = (int \*) shmat(lockid, 0, 0);

\*la = 1;

a[0] = 0;

a[1] = 1;

for( i=0; i< 10; i++)

{

while(\*la != 0);

\*la = 0;

a[0] = a[0] + a[1];

a[1] = a[0] + a[1];

printf("Parent writes: %d,%d\n",a[0],a[1]);

\*la = 1;

}

wait(0);

shmdt(a);

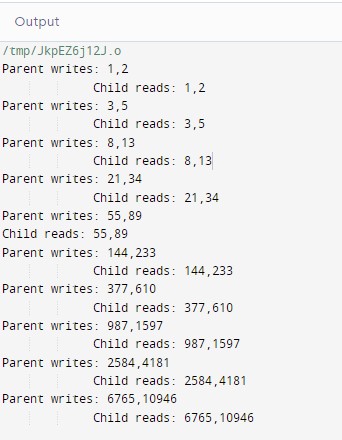
shmctl(shmid, IPC\_RMID, 0);

}

return 0;

}

**Output:**



Share memory provides an efficient way of communication via the sharing the data that resides in memory.

The Shared memory can be accessed in non-serial (random) manner. It means that semaphore or other synchronization method must be used to coordinate access to shared memory segment.

shmget is used the create and gain access to an existing segment.

shmctl is used to obtain the status of a memory segment, set permissions, and remove a shared memory segment.

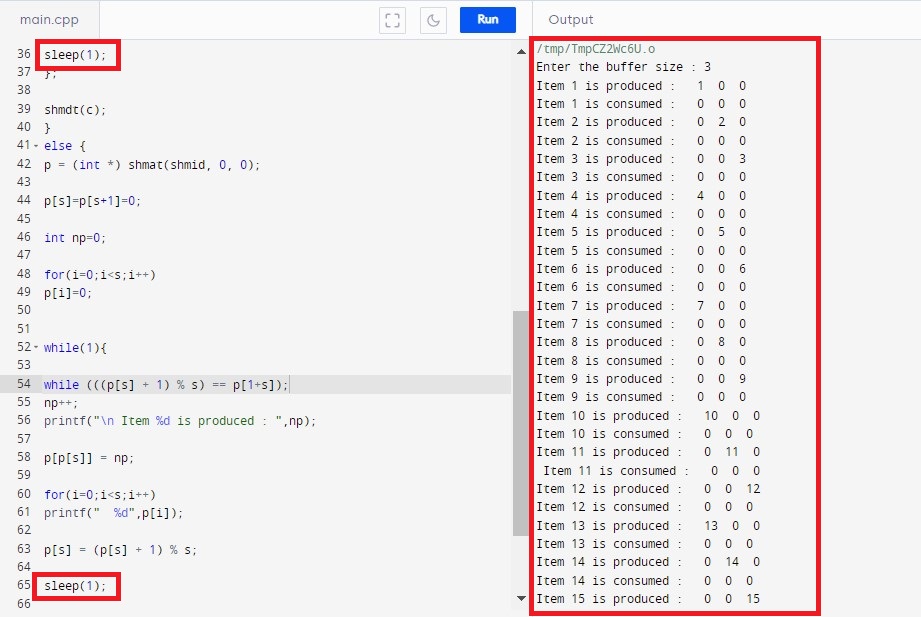
shmat and shmdt are used to attach and detach shared memory segments.

mmap system call is used to map the virtual memory space of a process to a file. It is very useful since the data in a file, and it won’t lose after process exists.

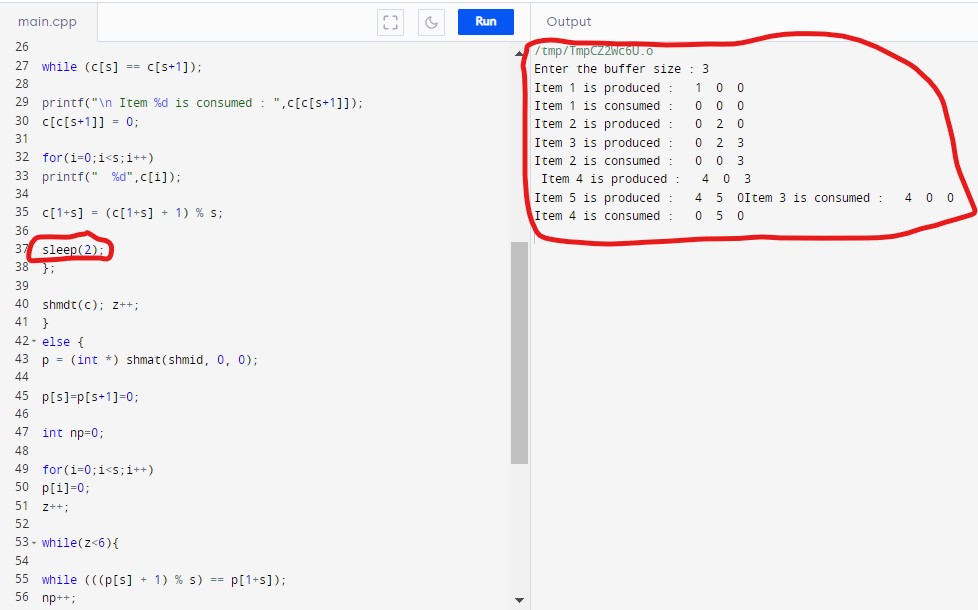
**Task 2:**

**Instructions:**

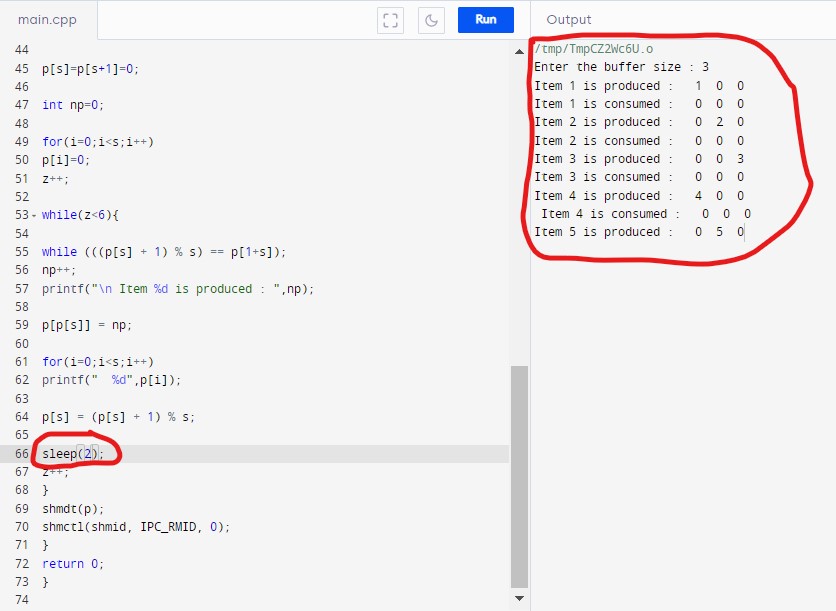
1. **Compiler and run the above code and notice the result.**

****

**2. Modify the sleep in the child process to sleep(2). What happens now?**

****

1. **Modify the sleep in the parent process to sleep(2). What happens now?**

****