Exp 8:

Aim: To write a program to implement solution to producer consumer problem using semaphores.

Program code:

```c
int semid = semget(IPC_PRIVATE, 3, 0666 | IPC_CREAT);
semctl(semid, 0, SETVAL, 1); // mutex = 1
semctl(semid, 1, SETVAL, 0); // full = 0
semctl(semid, 2, SETVAL, SIZE); // empty = SIZE

int pid = fork();

if (pid < 0) {
    printf("Fork failed!\n");
    return 1;
}

// Producer Process
else if (pid == 0) {
    for (i = 0; i < SIZE; i++) {
        wait(semid, 2); // wait(empty)
        wait(semid, 0); // wait(mutex)

        buffer[i] = i + 1;
        printf("Producer produced: %d\n", buffer[i]);

        signal(semid, 0); // signal(mutex)
        signal(semid, 1); // signal(full)
        sleep(1);
    }
}

// Consumer Process
else {
    sleep(1);   // Give producer time to produce
    for (i = 0; i < SIZE; i++) {
        wait(semid, 1); // wait(full)
        wait(semid, 0); // wait(mutex)

        printf("Consumer consumed: %d\n", buffer[i]);
        buffer[i] = 0; // Clear the slot

        signal(semid, 0); // signal(mutex)
        signal(semid, 2); // signal(empty)
        sleep(2);
    }

    // Detach & destroy
    shmdt(buffer);
    shmctl(shmid, IPC_RMID, NULL);
    semctl(semid, 0, IPC_RMID);
}

return 0;
}
```

Output:

```
Producer produced: 1
Consumer consumed: 1
Producer produced: 2
Producer produced: 3
Consumer consumed: 2
Producer produced: 4
1Producer produced: 5
Consumer consumed: 3
Consumer consumed: 4
Consumer consumed: 5
```