CS 331: System Software Lab- Documentation

Exp6: Producer consumer problem using semaphores

Date of submission: 27st September 2020

Submitted by,

Shree Lakshmi Lanish

S5 CSE

Roll no 55

TCR18CS055

Aim of the experiment

Implement the producer-consumer problem using semaphores.

THEORY

In computing, the producer–consumer problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue. The producer's job is to generate data, put it into the buffer, and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer), one piece at a time. The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer. The solution for the producer is to either go to sleep or discard data if the buffer is full. The next time the consumer removes an item from the buffer, it notifies the producer, who starts to fill the buffer again. In the same way, the consumer can go to sleep if it finds the buffer empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer. The solution can be reached by means of inter-process communication, typically using semaphores. An inadequate solution could result in a deadlock where both processes are waiting to be awakened. The problem can also be generalized to have multiple producers and consumers.

Compilation of code

The code is provided in the **program.c** file. Open command prompt and then run the commands:

gcc program.c

.\a.exe

The content of **input.txt** file is displayed in the program.

Note: There should not be a new line.

Screenshots of the output

The output obtained upon execution of program.c is as follows: