



Government Engineering College Thrissur

System Software Lab

Navaneeth D

TCR18CS043

S5, CSE

Producer-Consumer Problem

AIM

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, proposed by [Edsger W. Dijkstra](#).^[3] 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.

RESULT

Producer-consumer problem was implemented with successful output.

Output Screenshots

The screenshot displays the Visual Studio Code interface with the 'input.txt - Cprog - Visual Studio Code' window. The Explorer sidebar on the left shows a project structure with folders 'Exp2', 'Exp3', 'Exp4', 'Exp5', and 'Exp6'. The 'Exp4' folder is expanded, showing files 'a.out', 'bankers.c', 'input.txt', 'output.txt', and 'Readme.pdf'. The 'Exp5' folder is also expanded, showing 'a.out', 'input.txt', 'output.txt', and 'ReadmeE5.pdf'. The 'Exp6' folder is expanded, showing 'a.out', 'input.txt', 'output.txt', 'PC.c', 'screenshots', 'a.out', 'Exp1.tar.gz', 'filetest.c', and 'test.c'. The 'input.txt' file is selected. The Terminal window at the bottom shows the following output:

```
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp6$ gcc -lrt -pthread PC.c
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp6$ ./a.out
Enter the Buffer size: 20

-----Output-----

Buffer space: 0
Producer produced 3 items

Buffer space: 3
Producer produced 7 items

Buffer space: 10
Consumer consumed 4 items

Buffer space: 6
Consumer consumed 3 items

Buffer space: 20
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp6$
```

The status bar at the bottom indicates the current line and column as 'Ln 3, Col 6', the number of spaces as 'Spaces: 4', the encoding as 'UTF-8', the line ending as 'LF', and the text format as 'Plain Text'. The system clock shows '13:10'.