

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

Protected by PDF Anti-Copy Free

Lab Lecture-34 By Vivek Dubey

PDF

COMPUTER SCIENCE AND ENGINEERING

CS403PC

COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CS403PC

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Scheme of Operating System

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



S. No.	Course Code	Course Title	L	T	P	Credits
3	CS403PC	Operating Systems	3	0	0	3
6	CS406PC	Operating Systems Lab	0	0	3	1.5

AIM : To Write a C program to simulate producer - consumer problem using semaphores.

DESCRIPTION

- Producer consumer problem is a synchronization problem.
- There is a fixed size buffer where the producer produces items and that is consumed by a consumer process.
- One solution to the producer-consumer problem uses shared memory.
- To allow producer and consumer processes to run concurrently, there must be available a buffer of items that can be filled by the producer and emptied by the consumer.
- This buffer will reside in a region of memory that is shared by the producer and consumer processes.
- The producer and consumer must be synchronized, so that the consumer does not try to consume an item that has not yet been produced.

Output

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - Buffer is Empty

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 100

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 100

Output

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 100

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 200

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 300

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 400

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 500

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 600

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 700

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 800

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - Enter the value: 900

1. Produce 2. Consume 3. Exit

- Enter your choice: 1
 - **Buffer is Full**

Output

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 100

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 200

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 300

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 400

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 500

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 600

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 700

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 800

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - The consumed value is 900

1. Produce 2. Consume 3. Exit

- Enter your choice: 2
 - **Buffer is Empty**

Coding-1

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- `#include<stdio.h>`
- `void main()`
- `{`
 - `int buffer[10], bufsize, in, out, produce, consume,`
 - `choice=0;`
 - `in = 0;`
 - `out = 0;`
 - `bufsize = 10;`

Coding-2

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- while(choice !=3)
- {
- printf("\n1. Produce \t 2. Consume \t3. Exit");
- printf("\nEnter your choice: ");
- scanf("%d",&choice);
-

Coding-3

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- switch(choice)
- {
- case 1:
- if((in+1)%bufsize==out)
- printf("\nBuffer is Full");
- else
- {
- printf("\nEnter the value: ");
- scanf("%d", &produce);
- buffer[in] = produce;
- in = (in+1)%bufsize;
- }
- break;

- case 2:
- if(in == out)
- printf("\nBuffer is Empty");
- else
- {
- consume = buffer[out];
- printf("\nThe consumed value is %d", consume);
- out = (out+1)%bufsize;
- }
- break;
- }
- }