

OS LAB 5

NAME: Aditya Anand

ROLL NO.: 20124009

BRANCH: IT

S NO.	TITLE	DATE OF IMPLEMENTATION	REMARKS
1	Program to handle the Critical Section Problem using semaphore	16-02-2022	

PRODUCER-CONSUMER PROBLEM SOLUTION USING SEMAPHORE AND MUTEX

The producer-consumer problem is an example of a [multi-process synchronization](#) problem. The problem describes two processes, the producer and the consumer that shares a common fixed-size buffer use it 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.

Problem: Given the common fixed-size buffer, the task is to make sure that the producer can't add data into the buffer when it is full and the consumer can't remove data from an empty buffer.

Solution: 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 manner, the consumer can go to sleep if it finds the buffer to be empty. The next time the producer puts data into the buffer, it wakes up the sleeping consumer.

CODE:

```
// Solution of the Producer Consumer Problem using semaphore
```

```
#include<bits/stdc++.h>
using namespace std;
```

```
// Initially we have an empty buffer with a size 5
int m=1;           // Mutex
int full=0;        // Counting Semaphore
int empty=5;       // Counting Semaphore
int IN=0;
int OUT=0;
```

```
void down(int &s){
    s--;
}
```

```
void up(int &s){
    s++;
}
```

```
void producer(){
    down(m);
    up(full);
    down(empty);

    IN++;
    cout<<"\nProducer produced the item "<<IN;
    up(m);
}
```

```
void consumer(){
    down(m);
    down(full);
    up(empty);
```

```

    OUT++;
    cout<<"\nConsumer consumed the item "<<OUT;
    up(m);
}

int main(){
    cout<<"SOLUTION OF PRODUCER CONSUMER PROBLEM USING SEMAPHORE AND MUTEX C++
IMPLEMENTATION\n";
    cout<<"Name: Aditya Anand\tRoll No.:20124009\t Branch: IT\n\n\n";

    int n=0;
    while(1){
        cout<<"Select the preference: (1, 2, or 3) \n";
        cout<<"1) Producer\n2)Consumer\n3)Exit\n";

        cin>>n;

        if(n==1){
            if(m==1 && empty!=0){
                producer();
                cout<<endl;
            }
            else{
                cout<<"Buffer is full\n";
            }
        }
        else if(n==2){
            if(m==1 && full!=0){
                consumer();
                cout<<endl;
            }
            else{
                cout<<"Buffer is empty\n";
            }
        }
        else{
            break;
        }
    }
    return 0;
}

```

RESULT:

SOLUTION OF PRODUCER CONSUMER PROBLEM USING SEMAPHORE AND MUTEX C++ IMPLEMENTATION

Name: Aditya Anand

Roll No.:20124009

Branch: IT

Select the preference: (1, 2, or 3)

1) Producer

2)Consumer

3)Exit

1

Producer produced the item 1

Select the preference: (1, 2, or 3)

1) Producer

2)Consumer

3)Exit

1

Producer produced the item 2

Select the preference: (1, 2, or 3)

1) Producer

2)Consumer

3)Exit

2

Consumer consumed the item 1

Select the preference: (1, 2, or 3)

1) Producer

2)Consumer

3)Exit

2

Consumer consumed the item 2

Select the preference: (1, 2, or 3)

1) Producer

2)Consumer

3)Exit

2

Buffer is empty