

Operating System Practicals

Assignment 9

Krunal Rank
U18C0081

Implement Producer Consumer Problem solution using Semaphore.

```
#include <cstdlib>
#include <iostream>
#include <mutex>
#include <thread>
#include <chrono>
using namespace std;

std::mutex g_mutex;
bool g_ready = false;
int g_data = 0;

int produceData()
{
    int randomNumber = rand() % 5;
    cout << "Produced Data: " << randomNumber << endl;
    return randomNumber;
}

void consumeData(int data)
{
    cout << "Received Data: " << data << endl;
    this_thread::sleep_for(chrono::seconds(data));
}

void consumer()
{
    while (true)
    {
        while (!g_ready)
        {
            cout << "Consumer can't access Critical Zone!" << endl;
            this_thread::sleep_for(chrono::seconds(1));
        }
        cout << "Locking Mutex for Consuming Data!" << endl;
        unique_lock<mutex> ul(g_mutex);
        consumeData(g_data);
        g_ready = false;
    }
}
```

```

        cout<<"Unlocking Mutex after Consuming Data!"<<endl;
    }
}

void producer()
{
    while (true)
    {
        unique_lock<mutex> ul(g_mutex);

        cout<<"Locking Mutex for Producing Data!"<<endl;
        g_data = produceData();
        g_ready = true;
        ul.unlock();
        cout<<"Unlocking Mutex after Producing Data!"<<endl;
        while (g_ready)
        {
            cout << "Producer can't access Critical Zone!" << endl;
            this_thread::sleep_for(chrono::seconds(1));
        }
    }
}

void consumerThread() { consumer(); }

void producerThread() { producer(); }

int main()
{
    thread t1(consumerThread);
    thread t2(producerThread);
    t1.join();
    t2.join();
    return 0;
}

```

Output:

```
Consumer can't access Critical Zone!
Locking Mutex for Producing Data!
Produced Data: 3
Unlocking Mutex after Producing Data!
Producer can't access Critical Zone!
Locking Mutex for Consuming Data!
Received Data: 3
Producer can't access Critical Zone!
Producer can't access Critical Zone!
Producer can't access Critical Zone!
Unlocking Mutex after Consuming Data!
Consumer can't access Critical Zone!
Locking Mutex for Producing Data!
Produced Data: 1
Unlocking Mutex after Producing Data!
Producer can't access Critical Zone!
Locking Mutex for Consuming Data!
Received Data: 1
Producer can't access Critical Zone!
Unlocking Mutex after Consuming Data!
Consumer can't access Critical Zone!
Locking Mutex for Producing Data!
Produced Data: 2
Unlocking Mutex after Producing Data!
Producer can't access Critical Zone!
Locking Mutex for Consuming Data!
Received Data: 2
Producer can't access Critical Zone!
Producer can't access Critical Zone!
Unlocking Mutex after Consuming Data!
Consumer can't access Critical Zone!
Locking Mutex for Producing Data!
Produced Data: 0
Unlocking Mutex after Producing Data!
Producer can't access Critical Zone!
Locking Mutex for Consuming Data!
Received Data: 0
Unlocking Mutex after Consuming Data!
Consumer can't access Critical Zone!
Locking Mutex for Producing Data!
Produced Data: 3
Unlocking Mutex after Producing Data!
Producer can't access Critical Zone!
```

```
Locking Mutex for Consuming Data!  
Received Data: 3  
Producer can't access Critical Zone!  
Producer can't access Critical Zone!  
Producer can't access Critical Zone!  
Unlocking Mutex after Consuming Data!  
Consumer can't access Critical Zone!  
Locking Mutex for Producing Data!  
Produced Data: 0  
Unlocking Mutex after Producing Data!  
Producer can't access Critical Zone!
```