**SVKM’S NMIMS**

**MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT& ENGINEERING**

**(Campus Name)**

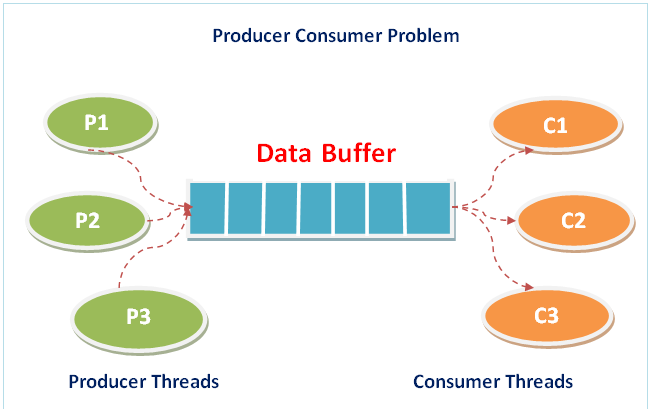
Academic Year: 2022-2023

**Practical 5 –Program to demonstrate synchronization through Producer/Consumer problem.**

|  |  |  |
| --- | --- | --- |
| Name | Roll\_No | SAP-ID |
| Jal Bafana | K005 | 70102300054 |

Dear all,

Kindly complete the following task with your name in output file also attach the C/Java program with the file.



import threading

import time

from queue import Queue

# Buffer size

BUFFER\_SIZE = 10

# Create a shared buffer using Queue

buffer = Queue(BUFFER\_SIZE)

# Shared variable for item count

x = 0

# Function to produce an item and add it to the buffer

def producer():

global x

time.sleep(1) # Simulating some processing delay

if not buffer.full():

x += 1

buffer.put(x) # Add item to the buffer

print(f"Producer produces item {x}")

else:

print("Buffer is full!")

# Function to consume an item and remove it from buffer

def consumer():

global x

time.sleep(1) # Simulating some processing delay

if not buffer.empty():

consumed\_item = buffer.get() # Get item from the buffer

x -= 1

print(f"Consumer consumes item {consumed\_item}")

else:

print("Buffer is empty!")

# Function to simulate user input for a given action

def simulate\_action(choice):

if choice == 1:

producer()

elif choice == 2:

consumer()

elif choice == 3:

print("Exiting...")

return False

return True

# Main simulation loop

def main():

print("\n1. Press 1 for Producer")

print("2. Press 2 for Consumer")

print("3. Press 3 for Exit")

running = True

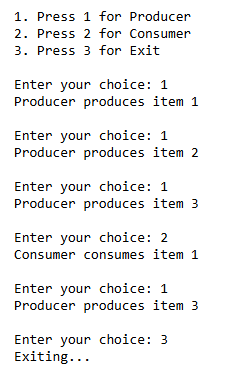
while running:

choice = int(input("\nEnter your choice: "))

running = simulate\_action(choice)

if \_\_name\_\_ == "\_\_main\_\_":

main()



## **Conclusion: -**

Write your observation about Producer- consumer problem. How it is more useful in modern operating systems.

The **Producer-Consumer problem** ensures synchronization between processes producing and consuming data. It's crucial for efficient resource management and preventing deadlocks. This concept applies to areas like **CPU scheduling** and **networking buffers**.

References:

studocu.com/row/document/hamdard-university/legal-system/lab-8-producer-consumer-problem/29445188

<https://www.geeksforgeeks.org/producer-consumer-problem-in-c/?ref=rp>