

**Karachi Institute of Economics & Technology**

**Project Name:**

**Producer Consumer**

**Submitted by:**

**Amna Iqbal : 12683**

**Muhammad Anus Tauni : 12671**

**Arisha Nisar : 13192**

**Alishba Amin : 13198**

**Advisor:**

**Ms. Umm e Kulsoom**

College of Computing & Information Sciences

**Spring 2023**

# ABSTRACT

The producer-consumer project is concerned with the synchronization, coordination, and management of resources between producers and consumers in a system. The project uses concurrency and parallelism to effectively produce and handle tasks or data. Decoupling and scalability are emphasized in order to respond to shifting system needs. The project's concepts are useful for real-world applications like task queues and data processing pipelines. The project increases throughput and reduces delay by avoiding bottlenecks and doing performance optimization. The producer-consumer concept has extensive ramifications across several areas and applications and is essential for understanding resource management, synchronizations, and system optimizations.

# INTRODUCTION

Producer-Consumer problem is a classical synchronization problem in the operating system. With the presence of more than one process and limited resources in the system the synchronization problem arises. If one resource is shared between more than one process at the same time then it can lead to data inconsistency. Producer-Consumer problem is a classical synchronization problem in the operating system. With the presence of more than one process and limited resources in the system the synchronization problem arises. If one resource is shared between more than one process at the same time then it can lead to data inconsistency. In the producer-consumer problem, the producer produces an item and the consumer consumes the item produced by the producer

# PROBLEM STATEMENT

There is one Producer and one Consumer in the producer-consumer problem.

Producer –

The producer process executes a set of statements int produce to create a data element and stores it in the buffer.

Consumer –

If the buffer has items, a consumer process executes a statement consume with the data element as a parameter.

# PROJECT OBJECTIVES

● 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 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.

# SCOPE OF PROJECT

the producer-consumer problem involves the teacher (producer) generating tasks and the students (consumers) processing them. The teacher puts tasks in a shared resource, and students retrieve them for solving. Synchronization mechanisms like semaphores or mutex locks are used to coordinate access to the shared resource. The goal is to prevent conflicts and ensure data consistency. Deadlock and starvation scenarios should be avoided by implementing appropriate scheduling or prioritization strategies. By addressing these challenges, the producer-consumer problem allows for efficient task generation and processing in the teacher-student context.

# SIGNIFICANCE OF THE PROJECT

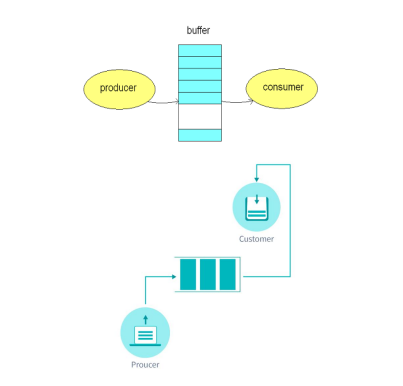
The producer-consumer initiative is important for a number of reasons. It exemplifies synchronisation, coordination, and efficient resource management between producers and consumers. It enables concurrency and parallelism, allowing numerous things to operate at once. As a result of the project's emphasis on decoupling and scalability, it may be modified to meet shifting system demands. It is useful in real-world situations and aids in the creation of effective systems. The project increases throughput and reduces delay by avoiding bottlenecks and doing performance optimisation. The producer-consumer project is vital for many different fields and applications because it handles important issues in resource management, synchronisation, and system optimisation.

# PROJECT METHODOLOGY

The project will be divided into several phases to make sure we are going in the right direction. These phases contain their own deliverables and guide the project members to move on to the next step of phase step by step.

* Phase 1: Project Planning
* Phase 2: Identify the constant, manipulated and responding variables.
* Phase 3: Application development
* Phase 4: Data analysis and discussion
* Phase 5: Conclusion

***DESCRIPTION OF PROJECT:***



# CODE & IMPLEMENTATION

# #include<stdio.h>

# #include<stdlib.h>

# #include<string.h>

# #include<unistd.h>

# #include<conio.h>

# struct Assignment

# {

# char ChapNumber[256];

# char SubmissionDate[256];

# };

# struct Quiz

# {

# char TopicName[256];

# char DateOfQuiz[256];

# };

# struct ClassTask

# {

# char TopicName[256];

# char SessionNumber[256];

# };

# int assign\_count=0,quiz\_count=0,ClassTask\_count=0;

# int N\_Assignment=5,N\_Quiz=2,N\_CT=0;

# int main()

# {

# 

# 

# int in\_Ass=0,in\_Quiz=0,in\_CT=0;

# int out\_Ass=0,out\_Quiz=0,out\_CT=0;

# 

# struct Assignment assign[N\_Assignment];

# struct Quiz quiz\_[N\_Quiz];

# struct ClassTask \*classtask;

# classtask = (struct ClassTask \*)malloc(100\*sizeof(struct ClassTask));

# int ch1,choice;

# char buffer[512];

# 

# 

# while(ch1!=3)

# {

# 

# // system("cls");

# printf("\t \t \t =============================\t \t \t");

# printf("\n \t \t \t PRODUCER AND CONSUMER PROBLEM \t \t \t \n \n");

# printf("");

# printf("\t \t \t 1) 11304 FURQAN ZAHID \n");

# printf("\t \t \t 2) 11295 MOHAMMAD AKRAM \n");

# printf("\t \t \t 3) 11125 MOHAMMAD HAMZA \n");

# printf("\t \t \t =============================\t \t \t");

# printf("\n");

# printf("\n");

# 

# printf("\t1. Teacher \n\t2. Std \n\t3. EXIT \n \n \tYOUR CHOICE HERE : ");

# scanf("%d",&ch1);

# 

# 

# if(ch1==1)

# {

# 

# choice=0;

# 

# /////////////////////Teacher Start////////////////////

# 

# while(choice != 4)

# {

# // system("cls");

# printf("\n");

# printf("\t\t1. Assignment \n\t\t2. Quiz \n\t\t3. Class Task \n\t\t4. EXIT \n \n \t \t YOUR CHOICE HERE : ");

# scanf("%d",&choice);

# 

# switch(choice)

# {

# 

# 

# case 1:

# if( assign\_count == N\_Assignment)

# {

# printf("\n");

# printf("\nYou cannot Add more Assignments untill pervious one are utilized\n");

# break;

# }

# 

# printf("Enter Chap Number :");

# scanf("%s",&buffer);

# strcpy(assign[in\_Ass].ChapNumber,buffer);

# printf("Enter submission date :");

# scanf("%s",&buffer);

# strcpy(assign[in\_Ass].SubmissionDate,buffer);

# 

# in\_Ass=(in\_Ass+1)%N\_Assignment;

# assign\_count = assign\_count +1;

# break;

# 

# 

# 

# case 2:

# 

# if( quiz\_count == N\_Quiz)

# {

# printf("\n");

# printf("\nYou cannot give more Quizs untill pervious one are Taken\n");

# break;

# }

# printf("Enter Topic Name :");

# scanf("%s",&buffer);

# strcpy(quiz\_[in\_Quiz].TopicName,buffer);

# printf("Enter Quiz date :");

# scanf("%s",&buffer);

# strcpy(quiz\_[in\_Quiz].DateOfQuiz,buffer);

# 

# in\_Quiz=(in\_Quiz+1)%N\_Assignment;

# quiz\_count = quiz\_count +1;

# break;

# 

# 

# 

# case 3:

# 

# 

# printf("Enter Topic Name :");

# scanf("%s",&buffer);

# strcpy((classtask + in\_CT)->TopicName,buffer);

# 

# printf("Enter Session number :");

# scanf("%s",&buffer);

# strcpy((classtask + in\_CT)->SessionNumber,buffer);

# 

# in\_CT++;

# ClassTask\_count = ClassTask\_count +1;

# N\_CT++;

# break;

# 

# 

# case 4:

# 

# 

# break;

# 

# 

# default:

# printf("\nSelect valid option\n");

# break;

# }

# }

# 

# /////////////////////Teacher End////////////////////

# 

# 

# }

# else if(ch1==2)

# {

# choice=0;

# system("cls");

# /////////////////////Student Start////////////////////

# 

# while(choice != 4)

# {

# // system("cls");

# printf("\n");

# printf("\t\t1. Assignment \n\t\t2. Quiz \n\t\t3. Class Task \n\t\t4. EXIT \n\t\tYOUR CHOICE HERE : ");

# scanf("%d",&choice);

# 

# switch(choice)

# {

# 

# case 1:

# 

# if( assign\_count == 0)

# {

# printf("\n");

# printf("\nGood News No more Assignments\n");

# break;

# }

# 

# 

# printf("\nChap Number : %s, \t\tSubmission Date: %s\n",assign[out\_Ass].ChapNumber,assign[out\_Ass].SubmissionDate);

# out\_Ass=(out\_Ass+1)%N\_Assignment;

# assign\_count = assign\_count -1;

# break;

# 

# 

# case 2:

# 

# if( quiz\_count == 0)

# {

# printf("\n");

# printf("\nGood News Not Any Quiz\n");

# break;

# }

# printf("\n");

# printf("\nTopic Name : %s, \t\tQuiz date : %s\n",quiz\_[out\_Quiz].TopicName,quiz\_[out\_Quiz].DateOfQuiz);

# out\_Quiz=(out\_Quiz+1)%N\_Assignment;

# quiz\_count = quiz\_count -1;

# break;

# 

# 

# 

# case 3:

# 

# if( ClassTask\_count == 0)

# {

# printf("\n");

# printf("\nGood News No Class Task pending now\n");

# break;

# }

# printf("\n");

# printf("\nTopic Name : %s, \t\t Quiz date : %s\n",(classtask + out\_CT)->TopicName,(classtask + out\_CT)->SessionNumber);

# out\_CT++;

# ClassTask\_count = ClassTask\_count -1;

# break;

# 

# case 4:

# 

# break;

# 

# default:

# 

# printf("\nSelect valid option\n");

# break;

# }

# }

# 

# 

# 

# /////////////////////student end////////////////////

# }

# }

# 

# return 0;

# 

# }

# RESULTS

# 

# 

# 

# CONCLUSION

● Producer Process produces data item and consumer process consumes data item.

● Both producer and consumer processes share a common memory buffer.

● Producer should not produce any item if the buffer is full.

● Consumer should not consume any item if the buffer is empty.

● Not more than one process should access the buffer at a time i.e mutual exclusion should be there.

● Full, Empty and mutex semaphore help to solve Producer-consumer problem.

● Full semaphore checks for the number of filled space in the buffer by the producer process

● Empty semaphore checks for the number of empty spaces in the buffer.

● mutex checks for the mutual exclusion

# REFERENCES

1. Guru99. (2021). *Priority Scheduling Algorithm: Preemptive, Non-Preemptive EXAMPLE*. Guru99.com.Retrieved 17June2021, from <https://www.guru99.com/priority-scheduling-program.html>.
2. Studytonight. (2021). *Shortest Remaining Time First Scheduling Algorithm - Studytonight*. Studytonight.com.Retrieved 17 June2021, from [https://www.studytonight.com/operating-system/shortest-remaining-time-first-scheduling-alg](https://www.studytonight.com/operating-system/shortest-remaining-time-first-scheduling-algorithm) [orithm](https://www.studytonight.com/operating-system/shortest-remaining-time-first-scheduling-algorithm).
3. Techopedia. (2021). *What is Shortest Job First (SJF)? - Definition from Techopedia*. Techopedia.com. Retrieved17June 2021,from <https://www.techopedia.com/definition/22053/shortest-job-first-sjf>.
4. Tutorialspoint. (2021). *Operating System - Process Scheduling - Tutorialspoint*. Tutorialspoint.com. Retrieved17June2021,from <https://www.tutorialspoint.com/operating_system/os_process_scheduling.htm>.