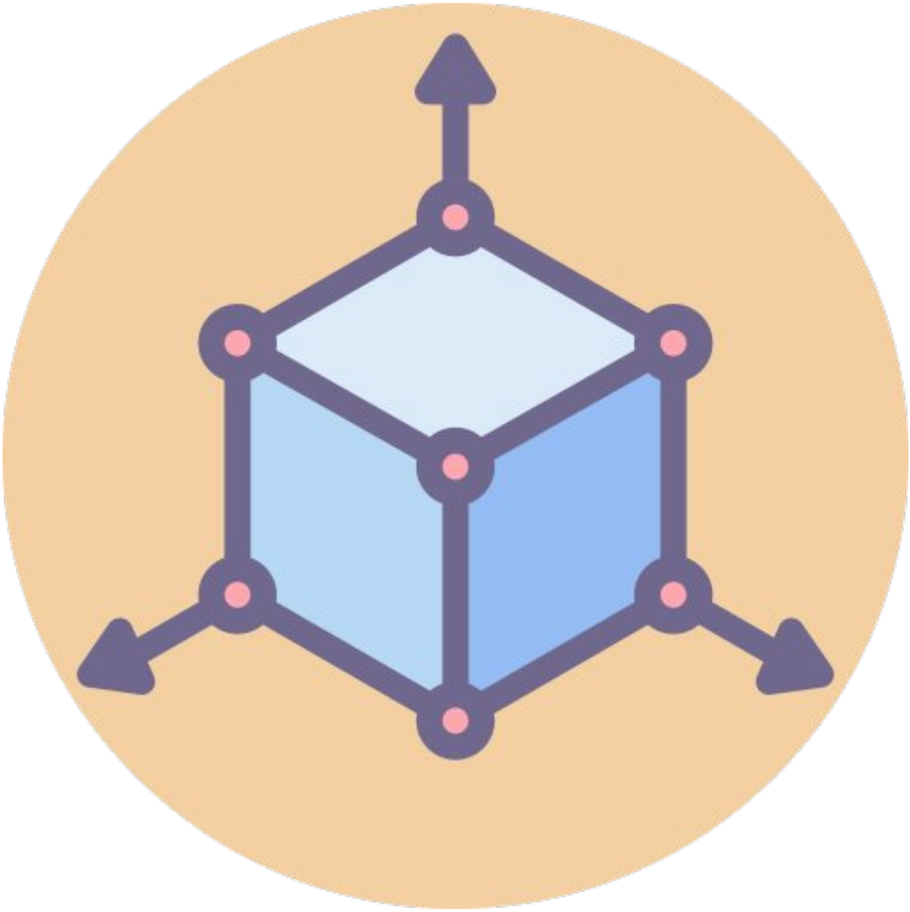
Bandarban University

Department of Computer Science and Engineering

Course Title: Simulation and Modeling

Course Code:CSE 801

Lab Report



Submitted To:

Mr. Mohammad Shadikul Islam
Adjunct Faculty (Guest Lecturer)
Computer Science and Engineering
Bandarban University

Submitted By:

Talimul Islam
ID: 15052201
Semester: 8th
Batch: 5th
Bandarban University

:INDEX:

Experiment Number : 05

Experiment Number : Simulates a multi-server M/M/2 bank queue system

Objective:

- To simulate a multi-server queue system with two parallel servers (M/M/2) in a bank environment. To model customer arrivals using a Poisson process and service times using an exponential distribution.
- To assign each arriving customer to the first available server (Server 1 or Server 2).
- To compute key performance metrics including waiting time, service time, server utilization, and queue length.
- To analyze how parallel service channels affect system efficiency and customer experience.
- To evaluate the probability of customer waiting and total time spent in the system.
- To visualize the timeline of service allocation and server idle periods.

Code:

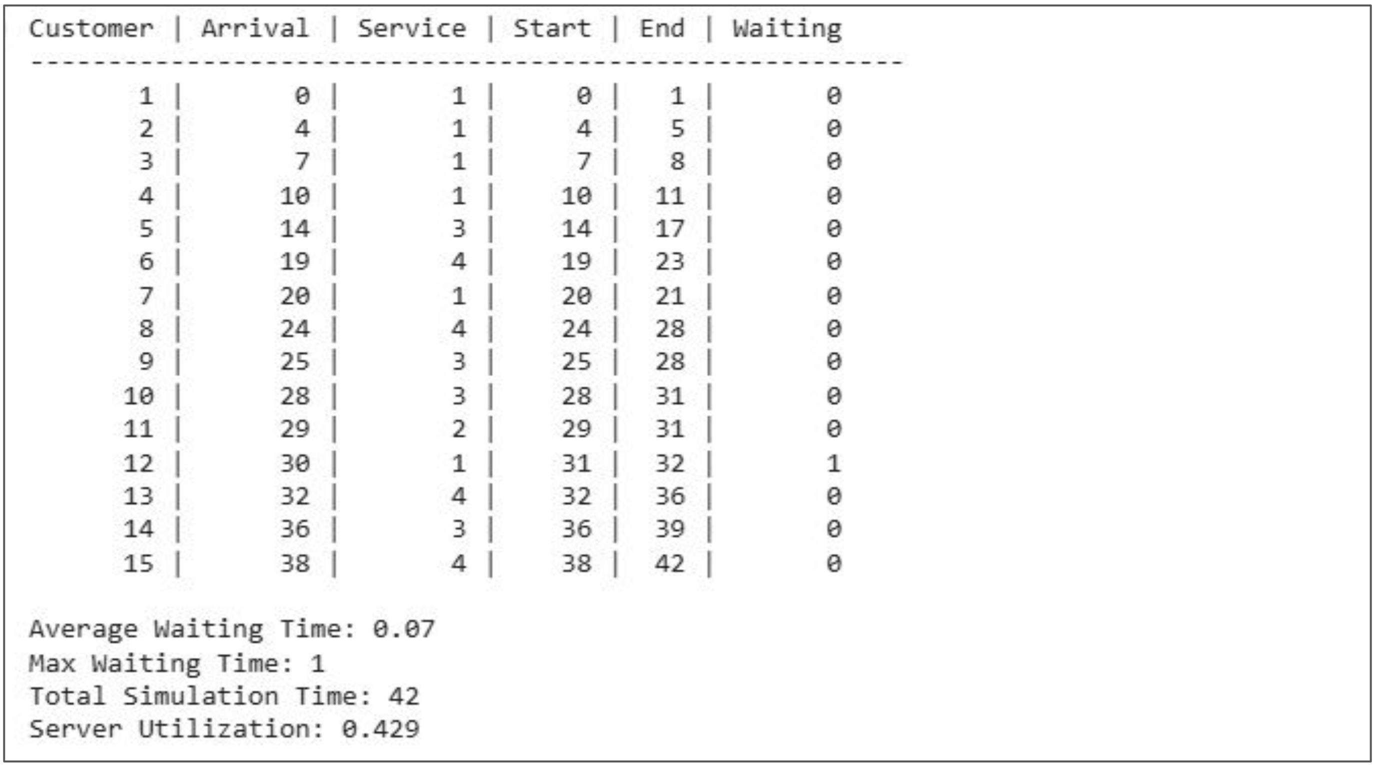
```
n = int(input("Enter the number of customers: "))
print(f"\nEnter {n} arrival times separated by spaces:")
arrival_times = list(map(int, input().split()))
print(f"\nEnter {n} service times separated by spaces:")
service_times = list(map(int, input().split()))
if len(arrival_times) != n or len(service_times) != n:
    print("Error: Number of arrival and service times must match the specified customer count.")
else:
    start_times = [0] * n
    finish_times = [0] * n
    waiting_times = [0] * n
    server_idle_time=[0]*n
    for i in range(n):
        if i == 0:
            start_times[i] = arrival_times[i]
            server_idle_time[i] = 0
        else:
            start_times[i] = max(arrival_times[i], finish_times[i - 1])
            server_idle_time[i] = max(0, arrival_times[i] - finish_times[i - 1])
        finish_times[i] = start_times[i] + service_times[i]
        waiting_times[i] = start_times[i] - arrival_times[i]

    total_service_time = sum(service_times)
    total_server_idle_time = sum(server_idle_time)
```

```
avg_service_time = total_service_time / n
  avg_waiting_time = sum(waiting_times) / n
  probability_waiting_time = sum(waiting_times) / 100
  avg_server_idle_time = sum(server_idle_time) / n

  print("\nCustomer | Arrival | Service | Start | Finish | Wait | Server Idle")
  for i in range(n):
     print(f"{i+1:^8} | {arrival_times[i]:^7} | {service_times[i]:^7} | {start_times[i]:^5} |
{finish_times[i]:^6} | {waiting_times[i]:^4} | {server_idle_time[i]:^7}")
  print("\n Summary:")
  print(f"Total Service Time: {total_service_time} minutes")
  print(f"Total server Idle Time: {total_server_idle_time} minutes")
  print(f"Average Service Time: {avg_service_time:.2f} minutes")
  print(f"Average Waiting Time: {avg_waiting_time:.2f} minutes")
  print(f"probability Waiting Time: {probability_waiting_time:.2f} minutes")
  print(f"Average server Idle Time: {avg_server_idle_time:.2f} minutes")
```

Output



```
Customer | Arrival | Service | Start | End | Waiting
--------------------------------------------------------
       1 |       0 |       1 |     0 |   1 |       0
       2 |       4 |       1 |     4 |   5 |       0
       3 |       7 |       1 |     7 |   8 |       0
       4 |      10 |       1 |    10 |  11 |       0
       5 |      14 |       3 |    14 |  17 |       0
       6 |      19 |       4 |    19 |  23 |       0
       7 |      20 |       1 |    20 |  21 |       0
       8 |      24 |       4 |    24 |  28 |       0
       9 |      25 |       3 |    25 |  28 |       0
      10 |      28 |       3 |    28 |  31 |       0
      11 |      29 |       2 |    29 |  31 |       0
      12 |      30 |       1 |    31 |  32 |       1
      13 |      32 |       4 |    32 |  36 |       0
      14 |      36 |       3 |    36 |  39 |       0
      15 |      38 |       4 |    38 |  42 |       0

Average Waiting Time: 0.07
Max Waiting Time: 1
Total Simulation Time: 42
Server Utilization: 0.429
```

Discussion

The M/M/2 queue system represents a multi-server model where:

- Arrivals follow a Poisson distribution with rate \lambda .
- Service times follow an exponential distribution with rate \mu .
- Two servers operate in parallel, serving customers on a first-available basis.

Each customer is assigned to the server that becomes free first. The simulation tracks:

1. Start Time of Service

   $$ST_i = \max(A_i, \min(FT_1, FT_2))$$

2. Finish Time of Service

   $$FT_i = ST_i + S_i$$

3. Waiting Time

   $$W_i = ST_i - A_i$$

4. Total Service Time

   $$T_s = \sum_{i=1}^{n} S_i$$

5. Average Service Time

   $$\bar{S} = \frac{T_s}{n}$$

6. Average Waiting Time

   $$\bar{W} = \frac{1}{n}\sum_{i=1}^{n} W_i$$

7. Server Utilization

$$U_j = \frac{\text{Busy Time of Server } j}{\text{Total Simulation Time}}, \quad j = 1, 2$$

This simulation helps evaluate how adding a second server reduces average waiting time and improves throughput. It also highlights the trade-off between server utilization and customer satisfaction. The M/M/2 model is widely used in banking, call centers, and service desks where parallel processing is essential.