Computer System Modeling & Simulation – Project

System #2:

Your system consists of a single CPU with finite buffer capacity. Jobs arrive according to a Poisson process with rate λ jobs/sec. The job sizes are exponentially distributed with mean $1/\mu$ seconds. Jobs are serviced in FCFS order. Let $N - 1$ denotes the maximum number of jobs that your system can hold in the queue. Thus, including the job serving, there are a maximum of N jobs in the system at any one time. If a job arrives when there are already N jobs in the system, then the arriving job is rejected.

Give a detailed description of:

Q1, The system model

The System model components:

Entity: Customers
: Server
: Queue

State: Number of jobs in the system
: Server state (Idle, Busy)

Event: Customer arrival
: Customer departure
: Simulation ending condition

- Logical relationship between model components:

  a) **Queue length (Customers in the queue):** as the arrival rate of customers increase the number of customers in the system increase and the server state goes to busy.
  b) **Jobs in the Server:** as the arrival rate of customers increase, the number of customers in the server increases.

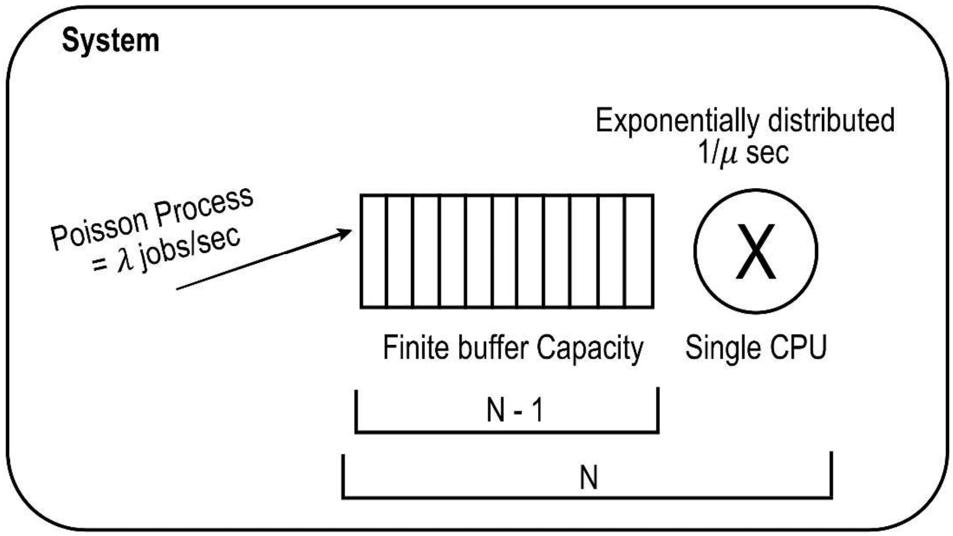
- Essential characteristics for each component:

  a) **Customers:** Arrival rate
  : Arrival pattern

  b) **Server:** Service rate

  c) **Queue:** Queue discipline (FCFS)
  : Queue size (N – 1)

- Structural description:



- Input models/assumptions:

    a) Arrival rate (λ)
    b) Service rate (1/μ)
    c) Queue capacity = (N – 1)
    d) System capacity = N

Q2, Pseudo code/flow chart for your DES algorithm (you have to apply the event scheduling approach)

```
Initialization
Schedule the first arrival
Insert the scheduled arrival in the FES
While the CurrentTime <= SimuTime
    Remove the first event from FES
    Advance the Current time=the event time
    If the event == Arrival event
        Destroy the record (from FES)
        Check if the Number of customers in the Queue is less than N - 1
          Number of customers in the queue++
          Schedule the next arrival
          Insert the scheduled arrival in the FES and Sort FES
          If the server state==idle
```

```
            Server state=busy
            Schedule departure event
            Insert the departure event in the FES and Sort FES
            Number of customers in the queue--
        else:
            Queue ++
    else if the Number of customers in the Queue is equal to  N - 1
        Drop the Arrival event

    Else If the event == Departure event
        Destroy the record (from FES)
        if Number of customers in the queue != 0
                Server state=busy
                Schedule departure event
                Insert the departure event in the FES and Sort FES
                Number of customers in the queue --
        Else
          Server state=idle
```

Q3, Random number and random variable generation methods you employ?

Ans: random.expovariate()

Random: module is used to generate random number in Python. It generates pseudo-random numbers. That implies that these randomly generated numbers can be determined.

Expovariate(): is an inbuilt method of the **random** module. It is used to return a random floating-point number with exponential distribution**.**

```python
import random

lamda = 3

for i in range(10):
    value = random.expovariate(lamda)
    print(value)
```

Output:

```
In [2]: runfile('C:/Users/Dell-PC/Python/Runner_Simulation_Two.py', wdir='C:/Users/Dell-PC/Python')
0.0718860998003
0.688781696442
0.470857180885
0.126647188503
0.144993996667
0.530108296451
0.0504961510631
0.146068786353
0.220191982057
0.119182638699
```

Q4, The method you employ to verify your operational model?

Verification: determining if the implemented model is consistent with its specifications.

Method:

Comparison of the conceptual or mathematical model to computer representation by long-run measures of performance compare with that of the performance that has been obtained from the simulation or that has been computed analytically.

Q5, Any assumptions you make?

a) Simulation end condition = 100 cycles / iterations
b) Arrival rate (λ): 2
c) Service rate (1/μ): 3
d) Queue capacity: = (50 – 1) = 49
e) System capacity: N = 50

Evaluate the performance of the system using the following parameters:

Q1, Average response time/latency?

$$\text{Average response time / latency} = \frac{\lambda}{\mu}(\mu - \lambda)$$

$$\text{Average response time / latency} = \frac{3}{2}(3 - 2)$$

$$\text{Average response time / latency} = 1.5$$

Q2, Average number of waiting requests/jobs?

$$\text{Server Utilization} = \frac{\lambda}{\mu}$$

$$\text{Server Utilization} = \frac{3}{0.5} = 6$$

$$\text{Number of customer in queue} = \frac{\text{Server Utilization}^2}{1 - \text{Server Utilization}}$$

$$\text{Number of customer in queue} = \frac{36}{5} = 7.2$$

$$\text{Waiting time in the queue} = \frac{Lq}{\mu}$$

$$\text{Waiting time in the queue} = \frac{7.2}{2} = 3.6$$

$$\text{Waiting time in the system} = Wq + \frac{1}{\mu}$$

$$\text{Waiting time in the system} = 3.6 + \frac{1}{0.5}$$

$$\text{Waiting time in the system} = 5.6$$

$$\text{Average Number of waiting Customer} = \lambda\, Ws$$

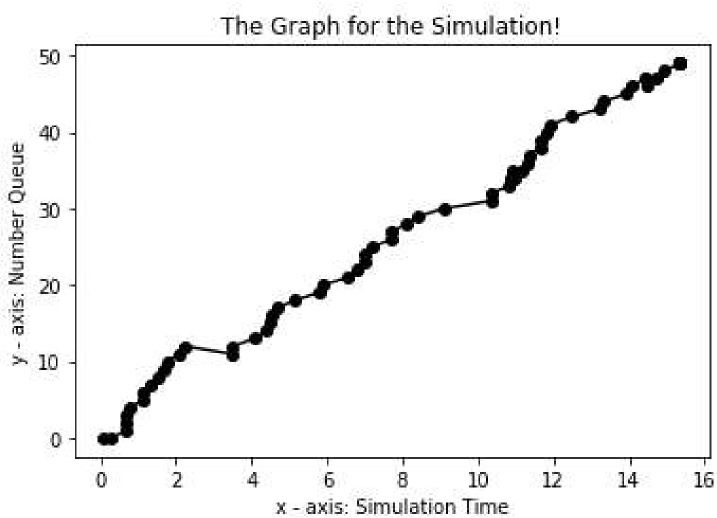
$$\text{Average Number of waiting Customer} = (3)(5.6)$$

$$\text{Average Number of waiting Customer} = 16.8$$

Q3, Blocking probability?

$$\text{Blocking probability} = \frac{\text{Total number of dropped customers}}{\text{Total number of arrived customers}}$$

Q4, Also, you have to show the evolution of the number of jobs/requests in the queue over time, i.e. number of jobs versus simulation time?



('Number of Arrivals : ', 92)
('Number of Departure : ', 9)
('Number of Customers in the Queue', 49)
('Number of Dropped arrivals : ', 32)

Simulation has ended at ==> 17.3795225932
Average number of Waiting time ==> 1.38561468747
Average Response time ==> 0.426067353236
Average number of Waiting Customers in the system ==> 16.5
('Blocking probability ==> ', 0.34782608695652173)
