lab1-FIFO

October 13, 2022

[1]:
```
from queue import PriorityQueue
import random
```

1 Defining functions and classes

[2]:
```
def arrival(time, FES, queue):

    global users
    global customer

    # introducing random client arrival
    inter_arrival = random.expovariate(1.0/average_arrival_interval)
    FES.put((time + inter_arrival, 'arrival'))

    # managing the event
    users += 1
    x = 'client' + str(customer)
    customer += 1

    # recording client id and put it in the list
    client = Client(x, time)
    queue.append(client)

    print(f'{client.name} arrived at {client.arrival_time}')

    # start the service in case the server is idle
    if users == 1:
        # scheduling random departure time to the clients
        service_time = random.expovariate(1.0/average_service_time)
        FES.put((time + service_time, 'departure'))



def departure(time, FES, queue):

    global users
```

```
    # manipulating the list of clients to get FIFO orientation
    queue.reverse()
    client = queue.pop()
    queue.reverse()
    users -= 1

    print(f'{client.name} departured at {time}')

    # checking the number of clients in line
    if users > 0:
        # scheduling random departure time to the clients
        service_time = random.expovariate(1.0/average_service_time)
        FES.put((time + service_time, 'departure'))


class Client:
    def __init__(self, name, arrival_time):
        self.name = name
        self.arrival_time = arrival_time
```

2 Implementing the simulation

[3]:
```
# initialization of variables
time = 0
users = 0
customer = 1
queue = []
average_arrival_interval = 3
average_service_time = 6
FES = PriorityQueue()
# the first arrival at time 0
FES.put((0,'arrival'))

# the main loop to give the service to the clients until specific time
while time < 20:
    (time, event_type) = FES.get()
    if event_type == 'arrival':
        arrival(time, FES, queue)
    elif event_type == 'departure':
        departure(time, FES, queue)
```

```
client1 arrived at 0
client2 arrived at 0.06759968795168807
client3 arrived at 0.25936442311660673
```

```
client1 departured at 3.0413531586224263
client4 arrived at 3.563795683093218
client5 arrived at 3.9993820101054447
client2 departured at 5.525833892668124
client6 arrived at 6.992173378331682
client7 arrived at 10.840544673215373
client8 arrived at 12.36040091753929
client3 departured at 16.084712862856385
client4 departured at 17.168117982091776
client9 arrived at 19.60831477481631
client10 arrived at 24.570173094485938
```

[ ]: