

CS 350: Homework #4

Due 3PM, Tuesday, March 1

Kristel Tan (ktan@bu.edu)

LATE SUBMISSION

Question 3

Please refer to the file `hw4_q3.py` submitted with this write-up to view the source code this problem. It has been heavily commented so you can follow along.

The code attempts to implement the system described in Question 4 of Assignment #3. The main function simulates the system, passing events to the CPU, disk, and network data structures according to the probabilities given. If you run the program you will generate a log of the events moving throughout the system until it is complete.

Unfortunately, I don't believe I have implemented the simulation 100% correctly. If you look at the log, you will find that the only resource that ever really gets busy is the disk. I was unable to debug this problem further, but I suspect it may be because the simulated service time for each process is not quite right. I am particularly referring to the part of the `cpu()` function where I generate a new birth event and move the event in the CPU server to another part of the system:

```
91  # Generate a random new event with random arrival time
92  # and service time uniformly distributed between 1 and 39 msec
93  arrival_time = list(sched)[-1].arrival_time + exponential(40)
94  serv_start = max(arrival_time, list(sched)[-1].end_time)
95  serv_time = random.uniform(0.001, 0.0039)
96  new_event = Event(arrival_time, serv_start, serv_time, EventType.birth, Resource.cpu)
97  sched.append(new_event)
98
99  if (arrival_time > list(cpu)[0].arrival_time):
100     # Generate random probability to see where next CPU process should go
101     prob = random.uniform(0, 1)
102     # Move first process of CPU to Disk with probability of 0.1
103     if (prob <= 0.1):
104         e = cpu.pop()
105         e.res = Resource.disk
106         clock += e.serv_time
107         sim_disk(clock, e, arrival_time)
108     # Move first process of CPU to Network with probability of 0.4
109     elif (prob > 0.1 and prob < 0.5):
110         e = cpu.pop()
111         e.res = Resource.net
112         e.serv_time = 0.025
113         clock += e.serv_time
114         sim_net(clock, e, arrival_time)
115     # Process is finished with probability of 0.5
116     else:
117         cpu.pop()
118
```

I believe the 'if' statement in line 99 seems to be passing every time, which always clears the CPU's queue. Something similar may be happening in the network's resource data structure.

Without meaningful information about each resource's queue, I am unable to analyze parts (a) - (d) for this problem, but hopefully the logic of the simulation can receive partial credit.