Comments about homework 3: Discrete Event Simulators Using Queues

8:00 (1) – 12:00 8.05:22 (55) , 8:05:23 (56),

0, 1, 2, 3,

1. 14,400

|  |
| --- |
| Job1: 8.00.01 1 |
| : 8.00.02 2 |
| 8:01:00 60 |
|  |
|  |
|  |
|  |
|  |

Instead of saying a job will take 10 seconds. Simply say 10

Clock = 0:

First job: clock+ Random number generate (0,5) : 3 time increment of first job

Clock 3

Second job: clock + Random number generate (0,5) +

Stuct job

{

Int jobID: //identify each job

Int jobArrivalTime; // at what time did the job arrive: e.g., an integer between 1: 14,400 for four ///hours translated into seconds

//but you can run it for 1 hour = 60\*60 = 360 the top integer value

Int jobLength; //random number may be between 1 and 10

Int JobWaitTime; //calculated based on job arrival time and job service time

Int JobFinishTime; // based service time according to clock value

}

For bonus points: you need to keep information about each cpu

e.g.,

struct cpu{

NumberOfJobsServedByCPU

LengthOfMaximumJob

TotalEmptyIdelTime

Other statistics required by the question.

};

Struct aboutQueue{

//queue statistics

{

e.g., maxQueueLengthAtAnyPointInTime

e.g. total time when queue was empty

e.g., longest Job waited in queue

e.g., LongestWaitingTimeForASingleJob

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Job # | Arrival Time | Length |  |  |  |
| 1 | 1 | 2 |  |  |  |
| 2 | 3 | 5 |  |  |  |
| 3 | 3 | 3 |  |  |  |
| 4 | 10 | 9 |  |  |  |
| 5 | 12 | 7 |  |  |  |
| 6 | 13 | 5 |  |  |  |
| 7 | 14 | 5 |  |  |  |

CPU2 Job2

Clock = 3

CPU1 Job 1

Clock = 1