Operating Systems (OS) – Theory

Assignment:03

Multi-Level Feedback Queue(Simulation)

Submitted by:

Tayyaba Asif

2019-CE-5

Submitted to:

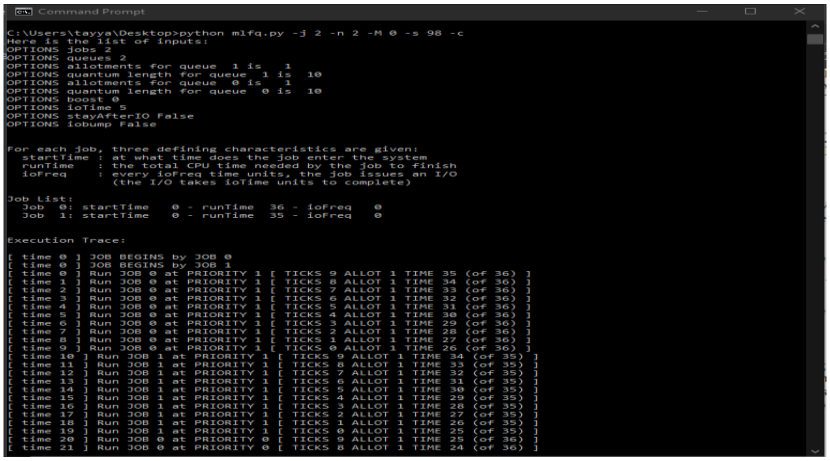
Sir Waqas Ali

Date: 25th February, 2022

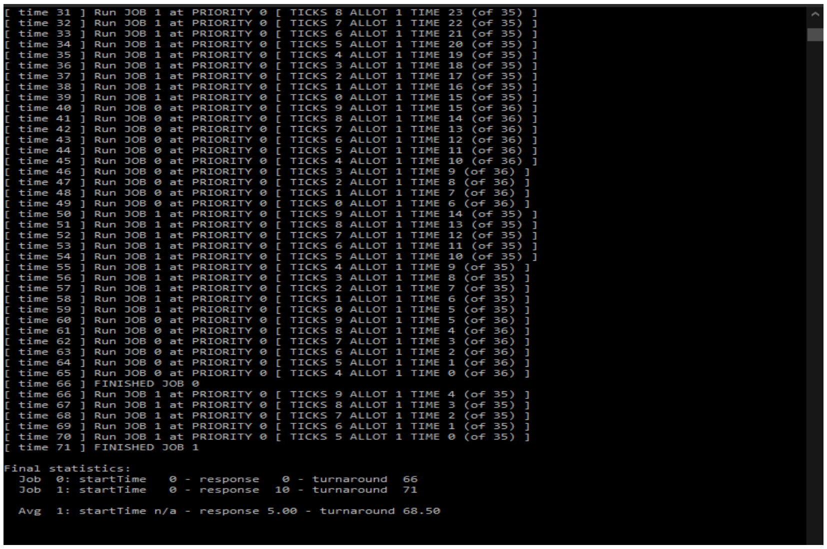
Question No.01:

Run a few randomly-generated problems with just two jobs and two queues; compute the MLFQ execution trace for each. Make your life easier by limiting the length of each job and turning off I/O’s

Answer:



```
C:\Users\tayya\Desktop>python mlfq.py -j 2 -n 2 -M 0 -s 98 -c
Here is the list of inputs:
OPTIONS jobs 2
OPTIONS queues 2
OPTIONS allotments for queue  1 is   1
OPTIONS quantum length for queue  1 is  10
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  0 is  10
OPTIONS boost 0
OPTIONS ioTime 5
OPTIONS stayAfterIO False
OPTIONS iobump False


For each job, three defining characteristics are given:
  startTime : at what time does the job enter the system
  runTime   : the total CPU time needed by the job to finish
  ioFreq    : every ioFreq time units, the job issues an I/O
              (the I/O takes ioTime units to complete)

Job List:
  Job  0: startTime   0 - runTime  36 - ioFreq   0
  Job  1: startTime   0 - runTime  35 - ioFreq   0


Execution Trace:

[ time 0 ] JOB BEGINS by JOB 0
[ time 0 ] JOB BEGINS by JOB 1
[ time 0 ] Run JOB 0 at PRIORITY 1 [ TICKS 9 ALLOT 1 TIME 35 (of 36) ]
[ time 1 ] Run JOB 0 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 34 (of 36) ]
[ time 2 ] Run JOB 0 at PRIORITY 1 [ TICKS 7 ALLOT 1 TIME 33 (of 36) ]
[ time 3 ] Run JOB 0 at PRIORITY 1 [ TICKS 6 ALLOT 1 TIME 32 (of 36) ]
[ time 4 ] Run JOB 0 at PRIORITY 1 [ TICKS 5 ALLOT 1 TIME 31 (of 36) ]
[ time 5 ] Run JOB 0 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 30 (of 36) ]
[ time 6 ] Run JOB 0 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 29 (of 36) ]
[ time 7 ] Run JOB 0 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 28 (of 36) ]
[ time 8 ] Run JOB 0 at PRIORITY 1 [ TICKS 1 ALLOT 1 TIME 27 (of 36) ]
[ time 9 ] Run JOB 0 at PRIORITY 1 [ TICKS 0 ALLOT 1 TIME 26 (of 36) ]
[ time 10 ] Run JOB 1 at PRIORITY 1 [ TICKS 9 ALLOT 1 TIME 34 (of 35) ]
[ time 11 ] Run JOB 1 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 33 (of 35) ]
[ time 12 ] Run JOB 1 at PRIORITY 1 [ TICKS 7 ALLOT 1 TIME 32 (of 35) ]
[ time 13 ] Run JOB 1 at PRIORITY 1 [ TICKS 6 ALLOT 1 TIME 31 (of 35) ]
[ time 14 ] Run JOB 1 at PRIORITY 1 [ TICKS 5 ALLOT 1 TIME 30 (of 35) ]
[ time 15 ] Run JOB 1 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 29 (of 35) ]
[ time 16 ] Run JOB 1 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 28 (of 35) ]
[ time 17 ] Run JOB 1 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 27 (of 35) ]
[ time 18 ] Run JOB 1 at PRIORITY 1 [ TICKS 1 ALLOT 1 TIME 26 (of 35) ]
[ time 19 ] Run JOB 1 at PRIORITY 1 [ TICKS 0 ALLOT 1 TIME 25 (of 35) ]
[ time 20 ] Run JOB 0 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 25 (of 36) ]
[ time 21 ] Run JOB 0 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 24 (of 36) ]
```



```
[ time 31 ] Run JOB 1 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 23 (of 35) ]
[ time 32 ] Run JOB 1 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 22 (of 35) ]
[ time 33 ] Run JOB 1 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 21 (of 35) ]
[ time 34 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 20 (of 35) ]
[ time 35 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 19 (of 35) ]
[ time 36 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 18 (of 35) ]
[ time 37 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 17 (of 35) ]
[ time 38 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 16 (of 35) ]
[ time 39 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 15 (of 35) ]
[ time 40 ] Run JOB 0 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 15 (of 36) ]
[ time 41 ] Run JOB 0 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 14 (of 36) ]
[ time 42 ] Run JOB 0 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 13 (of 36) ]
[ time 43 ] Run JOB 0 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 12 (of 36) ]
[ time 44 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 11 (of 36) ]
[ time 45 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 10 (of 36) ]
[ time 46 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 9 (of 36) ]
[ time 47 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 8 (of 36) ]
[ time 48 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 7 (of 36) ]
[ time 49 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 6 (of 36) ]
[ time 50 ] Run JOB 1 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 14 (of 35) ]
[ time 51 ] Run JOB 1 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 13 (of 35) ]
[ time 52 ] Run JOB 1 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 12 (of 35) ]
[ time 53 ] Run JOB 1 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 11 (of 35) ]
[ time 54 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 10 (of 35) ]
[ time 55 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 9 (of 35) ]
[ time 56 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 8 (of 35) ]
[ time 57 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 7 (of 35) ]
[ time 58 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 6 (of 35) ]
[ time 59 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 5 (of 35) ]
[ time 60 ] Run JOB 0 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 5 (of 36) ]
[ time 61 ] Run JOB 0 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 4 (of 36) ]
[ time 62 ] Run JOB 0 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 3 (of 36) ]
[ time 63 ] Run JOB 0 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 2 (of 36) ]
[ time 64 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 1 (of 36) ]
[ time 65 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 0 (of 36) ]
[ time 66 ] FINISHED JOB 0
[ time 66 ] Run JOB 1 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 4 (of 35) ]
[ time 67 ] Run JOB 1 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 3 (of 35) ]
[ time 68 ] Run JOB 1 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 2 (of 35) ]
[ time 69 ] Run JOB 1 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 1 (of 35) ]
[ time 70 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 0 (of 35) ]
[ time 71 ] FINISHED JOB 1

Final statistics:
  Job  0: startTime   0 - response   0 - turnaround  66
  Job  1: startTime   0 - response  10 - turnaround  71

  Avg  1: startTime n/a - response 5.00 - turnaround 68.50
```

Question No.02:

How would you run the scheduler to reproduce each of the examples in the chapter?

Answer:

For figure 8.2: **python mlfq.py -l 0,200,0** this will allot 1 job to 3 queues with pre-defined quantum 10.

For figure 8.3: **python mlfq.py -l 0,200,0:100,20,0** this will allot 2 jobs to 3 queues.

For figure 8.4: **python mlfq.py -l 0,200,0:0,20,1 -i 10 -S**

For figure 8.5: **python mlfq.py -l 0,200,0:20,10,0:30,10,0:40,10,0:50,10,0:60,10,0:70,10,0:80,10,0:90,10,0:100,10,0:110,10,0:120,10,0:130,10,0:140,10,0:150,10,0:160,10,0:170,10,0:180,10,0:190,10,0:200,10,0:210,10,0:220,10,0, -B 50**

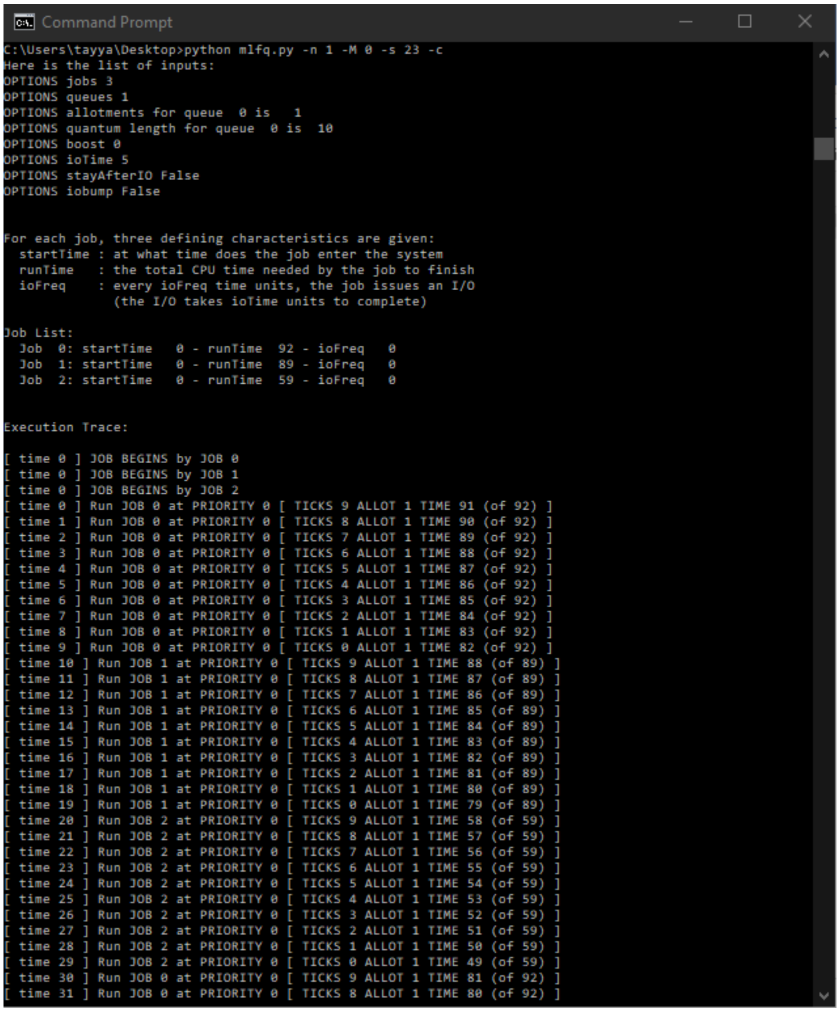
For figure 8.6: **python mlfq.py -l 0,120,0:20,100,9 -S -i 1**

For figure 8.7: **python mlfq.py -l 0,100,0:0,100,0 -Q 10,20,40**

Question No.03:

How would you configure the scheduler parameters to behave just like a round-robin scheduler?
Answer:

Jobs on the same queue are time-sliced, so the answer is to only have one queue.



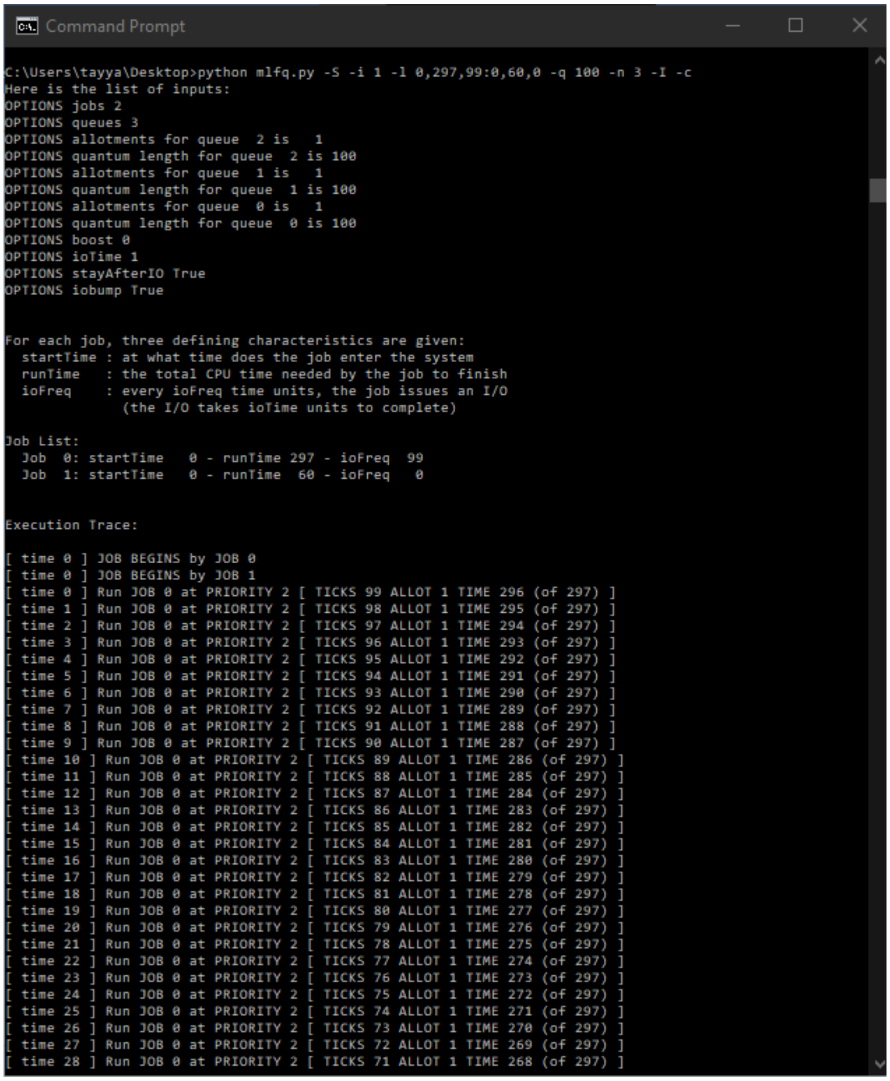
```
C:\Users\tayya\Desktop>python mlfq.py -n 1 -M 0 -s 23 -c
Here is the list of inputs:
OPTIONS jobs 3
OPTIONS queues 1
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  0 is  10
OPTIONS boost 0
OPTIONS ioTime 5
OPTIONS stayAfterIO False
OPTIONS iobump False


For each job, three defining characteristics are given:
  startTime : at what time does the job enter the system
  runTime   : the total CPU time needed by the job to finish
  ioFreq    : every ioFreq time units, the job issues an I/O
              (the I/O takes ioTime units to complete)

Job List:
  Job  0: startTime   0 - runTime  92 - ioFreq   0
  Job  1: startTime   0 - runTime  89 - ioFreq   0
  Job  2: startTime   0 - runTime  59 - ioFreq   0


Execution Trace:

[ time 0 ] JOB BEGINS by JOB 0
[ time 0 ] JOB BEGINS by JOB 1
[ time 0 ] JOB BEGINS by JOB 2
[ time 0 ] Run JOB 0 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 91 (of 92) ]
[ time 1 ] Run JOB 0 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 90 (of 92) ]
[ time 2 ] Run JOB 0 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 89 (of 92) ]
[ time 3 ] Run JOB 0 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 88 (of 92) ]
[ time 4 ] Run JOB 0 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 87 (of 92) ]
[ time 5 ] Run JOB 0 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 86 (of 92) ]
[ time 6 ] Run JOB 0 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 85 (of 92) ]
[ time 7 ] Run JOB 0 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 84 (of 92) ]
[ time 8 ] Run JOB 0 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 83 (of 92) ]
[ time 9 ] Run JOB 0 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 82 (of 92) ]
[ time 10 ] Run JOB 1 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 88 (of 89) ]
[ time 11 ] Run JOB 1 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 87 (of 89) ]
[ time 12 ] Run JOB 1 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 86 (of 89) ]
[ time 13 ] Run JOB 1 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 85 (of 89) ]
[ time 14 ] Run JOB 1 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 84 (of 89) ]
[ time 15 ] Run JOB 1 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 83 (of 89) ]
[ time 16 ] Run JOB 1 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 82 (of 89) ]
[ time 17 ] Run JOB 1 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 81 (of 89) ]
[ time 18 ] Run JOB 1 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 80 (of 89) ]
[ time 19 ] Run JOB 1 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 79 (of 89) ]
[ time 20 ] Run JOB 2 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 58 (of 59) ]
[ time 21 ] Run JOB 2 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 57 (of 59) ]
[ time 22 ] Run JOB 2 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 56 (of 59) ]
[ time 23 ] Run JOB 2 at PRIORITY 0 [ TICKS 6 ALLOT 1 TIME 55 (of 59) ]
[ time 24 ] Run JOB 2 at PRIORITY 0 [ TICKS 5 ALLOT 1 TIME 54 (of 59) ]
[ time 25 ] Run JOB 2 at PRIORITY 0 [ TICKS 4 ALLOT 1 TIME 53 (of 59) ]
[ time 26 ] Run JOB 2 at PRIORITY 0 [ TICKS 3 ALLOT 1 TIME 52 (of 59) ]
[ time 27 ] Run JOB 2 at PRIORITY 0 [ TICKS 2 ALLOT 1 TIME 51 (of 59) ]
[ time 28 ] Run JOB 2 at PRIORITY 0 [ TICKS 1 ALLOT 1 TIME 50 (of 59) ]
[ time 29 ] Run JOB 2 at PRIORITY 0 [ TICKS 0 ALLOT 1 TIME 49 (of 59) ]
[ time 30 ] Run JOB 0 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 81 (of 92) ]
[ time 31 ] Run JOB 0 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 80 (of 92) ]
```

Question No.04:

Craft a workload with two jobs and scheduler parameters so that one job takes advantage of the older Rules 4a and 4b (turned on with the -S flag) to game the scheduler and obtain 99% of the CPU over a particular time interval.

Answer:

First, set the I/O time to 1 with -i, then ensure that CPU hog calls an I/O operation on the 99th click of a 100 click quantum length. Also, include the -I flag to ensure the CPU gets back control when the I/O request finishes.



```
C:\Users\tayya\Desktop>python mlfq.py -S -i 1 -l 0,297,99:0,60,0 -q 100 -n 3 -I -c
Here is the list of inputs:
OPTIONS jobs 2
OPTIONS queues 3
OPTIONS allotments for queue  2 is   1
OPTIONS quantum length for queue  2 is 100
OPTIONS allotments for queue  1 is   1
OPTIONS quantum length for queue  1 is 100
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  0 is 100
OPTIONS boost 0
OPTIONS ioTime 1
OPTIONS stayAfterIO True
OPTIONS iobump True


For each job, three defining characteristics are given:
  startTime : at what time does the job enter the system
  runTime   : the total CPU time needed by the job to finish
  ioFreq    : every ioFreq time units, the job issues an I/O
              (the I/O takes ioTime units to complete)

Job List:
  Job  0: startTime   0 - runTime 297 - ioFreq  99
  Job  1: startTime   0 - runTime  60 - ioFreq   0


Execution Trace:

[ time 0 ] JOB BEGINS by JOB 0
[ time 0 ] JOB BEGINS by JOB 1
[ time 0 ] Run JOB 0 at PRIORITY 2 [ TICKS 99 ALLOT 1 TIME 296 (of 297) ]
[ time 1 ] Run JOB 0 at PRIORITY 2 [ TICKS 98 ALLOT 1 TIME 295 (of 297) ]
[ time 2 ] Run JOB 0 at PRIORITY 2 [ TICKS 97 ALLOT 1 TIME 294 (of 297) ]
[ time 3 ] Run JOB 0 at PRIORITY 2 [ TICKS 96 ALLOT 1 TIME 293 (of 297) ]
[ time 4 ] Run JOB 0 at PRIORITY 2 [ TICKS 95 ALLOT 1 TIME 292 (of 297) ]
[ time 5 ] Run JOB 0 at PRIORITY 2 [ TICKS 94 ALLOT 1 TIME 291 (of 297) ]
[ time 6 ] Run JOB 0 at PRIORITY 2 [ TICKS 93 ALLOT 1 TIME 290 (of 297) ]
[ time 7 ] Run JOB 0 at PRIORITY 2 [ TICKS 92 ALLOT 1 TIME 289 (of 297) ]
[ time 8 ] Run JOB 0 at PRIORITY 2 [ TICKS 91 ALLOT 1 TIME 288 (of 297) ]
[ time 9 ] Run JOB 0 at PRIORITY 2 [ TICKS 90 ALLOT 1 TIME 287 (of 297) ]
[ time 10 ] Run JOB 0 at PRIORITY 2 [ TICKS 89 ALLOT 1 TIME 286 (of 297) ]
[ time 11 ] Run JOB 0 at PRIORITY 2 [ TICKS 88 ALLOT 1 TIME 285 (of 297) ]
[ time 12 ] Run JOB 0 at PRIORITY 2 [ TICKS 87 ALLOT 1 TIME 284 (of 297) ]
[ time 13 ] Run JOB 0 at PRIORITY 2 [ TICKS 86 ALLOT 1 TIME 283 (of 297) ]
[ time 14 ] Run JOB 0 at PRIORITY 2 [ TICKS 85 ALLOT 1 TIME 282 (of 297) ]
[ time 15 ] Run JOB 0 at PRIORITY 2 [ TICKS 84 ALLOT 1 TIME 281 (of 297) ]
[ time 16 ] Run JOB 0 at PRIORITY 2 [ TICKS 83 ALLOT 1 TIME 280 (of 297) ]
[ time 17 ] Run JOB 0 at PRIORITY 2 [ TICKS 82 ALLOT 1 TIME 279 (of 297) ]
[ time 18 ] Run JOB 0 at PRIORITY 2 [ TICKS 81 ALLOT 1 TIME 278 (of 297) ]
[ time 19 ] Run JOB 0 at PRIORITY 2 [ TICKS 80 ALLOT 1 TIME 277 (of 297) ]
[ time 20 ] Run JOB 0 at PRIORITY 2 [ TICKS 79 ALLOT 1 TIME 276 (of 297) ]
[ time 21 ] Run JOB 0 at PRIORITY 2 [ TICKS 78 ALLOT 1 TIME 275 (of 297) ]
[ time 22 ] Run JOB 0 at PRIORITY 2 [ TICKS 77 ALLOT 1 TIME 274 (of 297) ]
[ time 23 ] Run JOB 0 at PRIORITY 2 [ TICKS 76 ALLOT 1 TIME 273 (of 297) ]
[ time 24 ] Run JOB 0 at PRIORITY 2 [ TICKS 75 ALLOT 1 TIME 272 (of 297) ]
[ time 25 ] Run JOB 0 at PRIORITY 2 [ TICKS 74 ALLOT 1 TIME 271 (of 297) ]
[ time 26 ] Run JOB 0 at PRIORITY 2 [ TICKS 73 ALLOT 1 TIME 270 (of 297) ]
[ time 27 ] Run JOB 0 at PRIORITY 2 [ TICKS 72 ALLOT 1 TIME 269 (of 297) ]
[ time 28 ] Run JOB 0 at PRIORITY 2 [ TICKS 71 ALLOT 1 TIME 268 (of 297) ]
```

Question No.05:

Given a system with a quantum length of 10 ms in its highest queue, how often would you have to boost jobs back to the highest priority level (with the -B flag) in order to guarantee that a single long   running (and potentially-starving) job gets at least 5% of the CPU?

Answer:

If a new 10ms-long job arrives every 10ms - like pictured in Figure 8.5 (left), boosting every 200 ticks ensures the long process will run 10/200 CPU ticks = 5%.