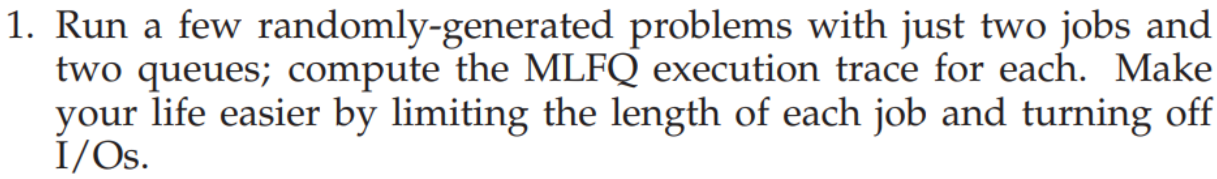
【OS】Day13(3)

【Ch8】Scheduling: Multi-Level-Feedback Queue Homework

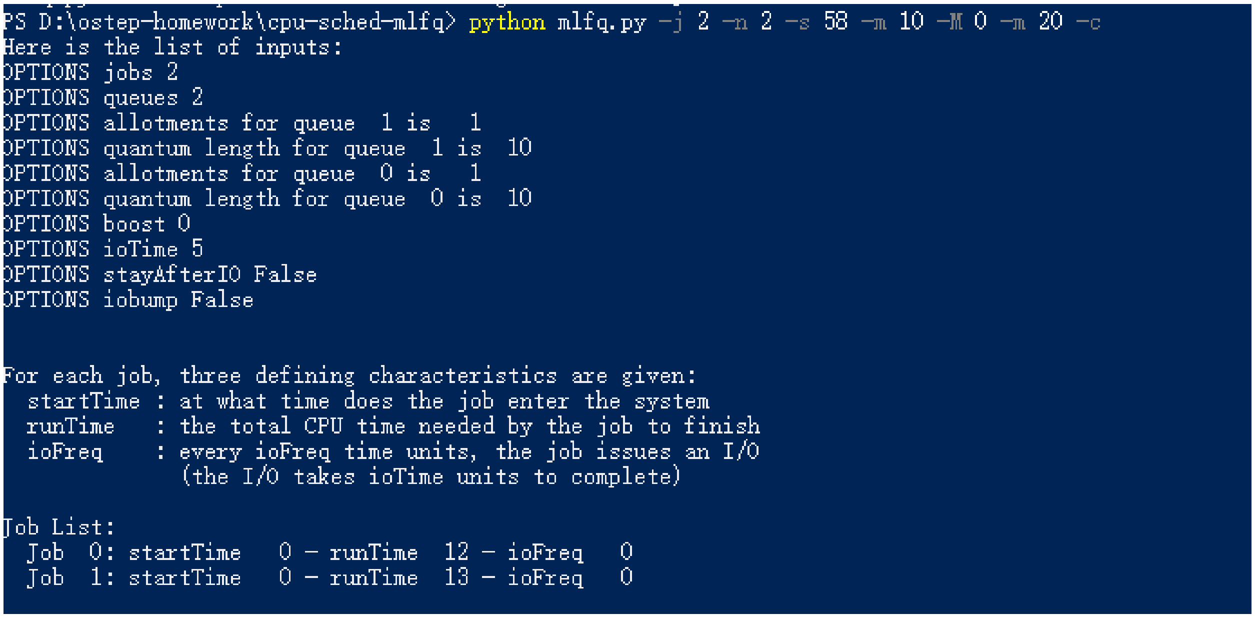
Question 1



1. 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/Os.

Genereate with the following code

```
#-j: Number of jobs
#-n: Number of queues
#-s: Random Seed
#-m: Maximum run-time
#-M: I/O Frequency
#-c: Show answer
Unix> python mlfq.py -j 2 -n 2 -s 58 -m 10 -M 0 -m 20 -c
```

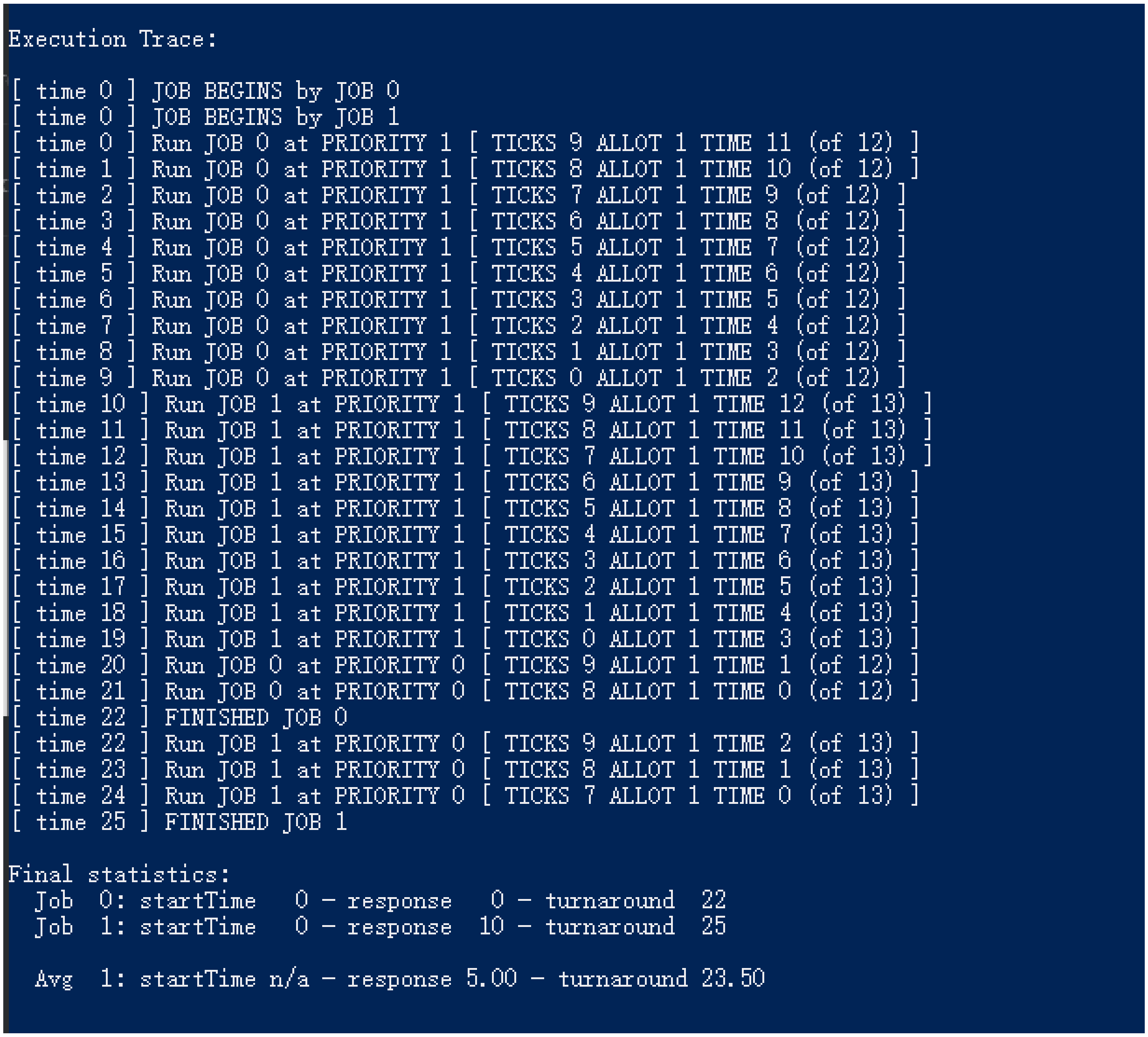


```
PS D:\ostep-homework\cpu-sched-mlfq> python mlfq.py -j 2 -n 2 -s 58 -m 10 -M 0 -m 20 -c
Here is the list of inputs:
OPTIONS jobs 2
OPTIONS queues 2
OPTIONS allotments for queue  1 is   1
OPTIONS quantum length for queue  1 is  10
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  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  12 - ioFreq   0
  Job  1: startTime   0 - runTime  13 - ioFreq   0
```

Execution Tree:



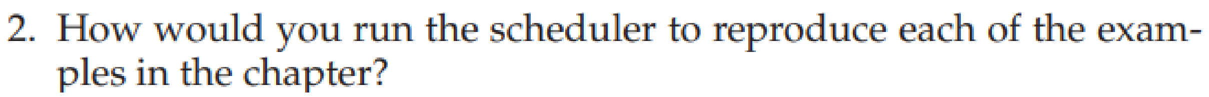
```
Execution Trace:

[ 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 11 (of 12) ]
[ time 1 ] Run JOB 0 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 10 (of 12) ]
[ time 2 ] Run JOB 0 at PRIORITY 1 [ TICKS 7 ALLOT 1 TIME 9 (of 12) ]
[ time 3 ] Run JOB 0 at PRIORITY 1 [ TICKS 6 ALLOT 1 TIME 8 (of 12) ]
[ time 4 ] Run JOB 0 at PRIORITY 1 [ TICKS 5 ALLOT 1 TIME 7 (of 12) ]
[ time 5 ] Run JOB 0 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 6 (of 12) ]
[ time 6 ] Run JOB 0 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 5 (of 12) ]
[ time 7 ] Run JOB 0 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 4 (of 12) ]
[ time 8 ] Run JOB 0 at PRIORITY 1 [ TICKS 1 ALLOT 1 TIME 3 (of 12) ]
[ time 9 ] Run JOB 0 at PRIORITY 1 [ TICKS 0 ALLOT 1 TIME 2 (of 12) ]
[ time 10 ] Run JOB 1 at PRIORITY 1 [ TICKS 9 ALLOT 1 TIME 12 (of 13) ]
[ time 11 ] Run JOB 1 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 11 (of 13) ]
[ time 12 ] Run JOB 1 at PRIORITY 1 [ TICKS 7 ALLOT 1 TIME 10 (of 13) ]
[ time 13 ] Run JOB 1 at PRIORITY 1 [ TICKS 6 ALLOT 1 TIME 9 (of 13) ]
[ time 14 ] Run JOB 1 at PRIORITY 1 [ TICKS 5 ALLOT 1 TIME 8 (of 13) ]
[ time 15 ] Run JOB 1 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 7 (of 13) ]
[ time 16 ] Run JOB 1 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 6 (of 13) ]
[ time 17 ] Run JOB 1 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 5 (of 13) ]
[ time 18 ] Run JOB 1 at PRIORITY 1 [ TICKS 1 ALLOT 1 TIME 4 (of 13) ]
[ time 19 ] Run JOB 1 at PRIORITY 1 [ TICKS 0 ALLOT 1 TIME 3 (of 13) ]
[ time 20 ] Run JOB 0 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 1 (of 12) ]
[ time 21 ] Run JOB 0 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 0 (of 12) ]
[ time 22 ] FINISHED JOB 0
[ time 22 ] Run JOB 1 at PRIORITY 0 [ TICKS 9 ALLOT 1 TIME 2 (of 13) ]
[ time 23 ] Run JOB 1 at PRIORITY 0 [ TICKS 8 ALLOT 1 TIME 1 (of 13) ]
[ time 24 ] Run JOB 1 at PRIORITY 0 [ TICKS 7 ALLOT 1 TIME 0 (of 13) ]
[ time 25 ] FINISHED JOB 1

Final statistics:
  Job  0: startTime   0 - response   0 - turnaround  22
  Job  1: startTime   0 - response  10 - turnaround  25

  Avg  1: startTime n/a - response 5.00 - turnaround 23.50
```

Question 2



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

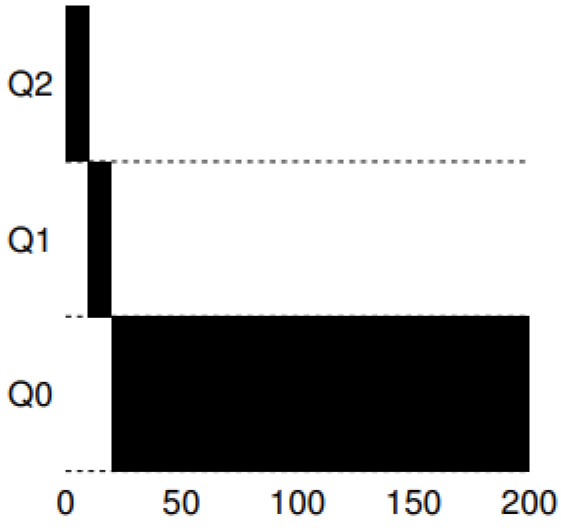


Figure 8.2: **Long-running Job Over Time**

```
#-l: Explicitly give the job list in the form of start-time,run-tim,I/O request frequency
#-q: Quantum Length
Unix> python mlfq.py -l 0,200,0 -n 3 -q 10 -c
```

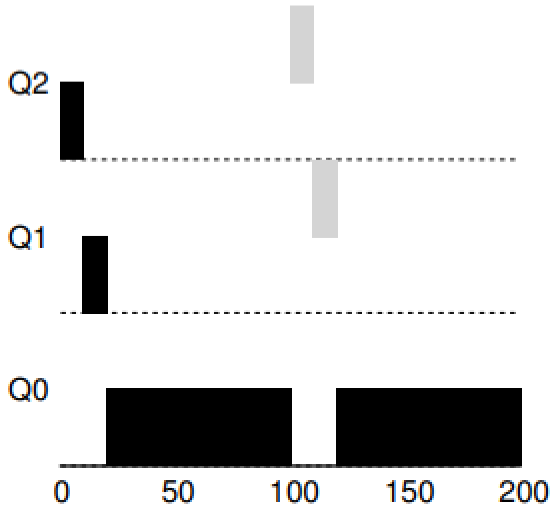
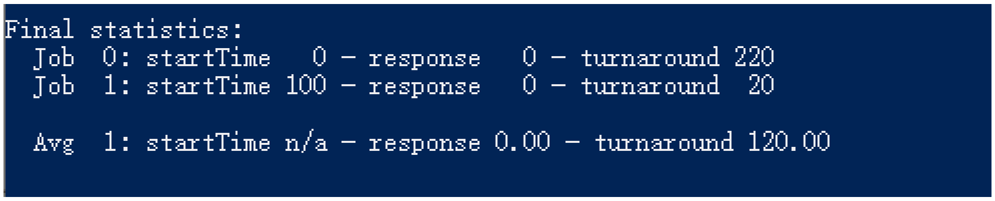


Figure 8.3: **Along Came An Interactive Job**

```
Unix> python mlfq.py -l 0,200,0:100,20,0 -n 3 -q 10 -c
```



```
Final statistics:
  Job  0: startTime   0 - response   0 - turnaround 220
  Job  1: startTime 100 - response   0 - turnaround  20

  Avg  1: startTime n/a - response 0.00 - turnaround 120.00
```

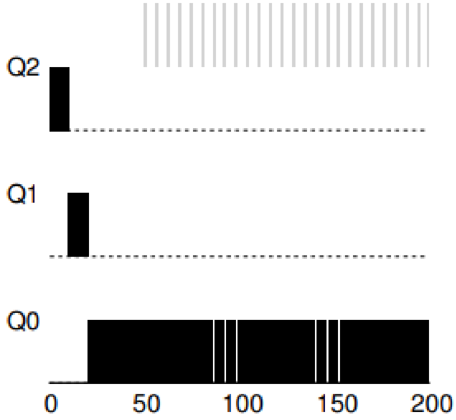
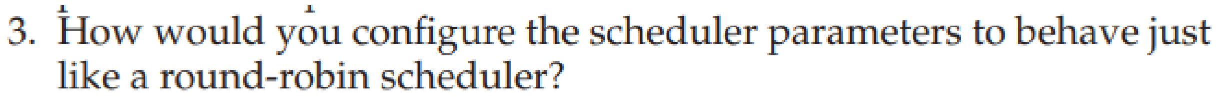


Figure 8.4: **A Mixed I/O-intensive and CPU-intensive Workload**

```
UNIX> python mlfq.py -l 0,200,0:50,25,1 -n 3 -q 10 -i 5
```

Question 3

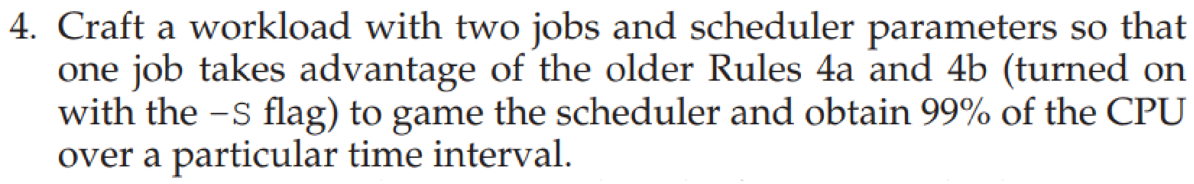


3. How would you configure the scheduler parameters to behave just like a round-robin scheduler?

```
UNIX> python mlfq.py -n 1
```

The scheduler would behave as a Round-Robin scheduler if we set the number of queues to be 1.

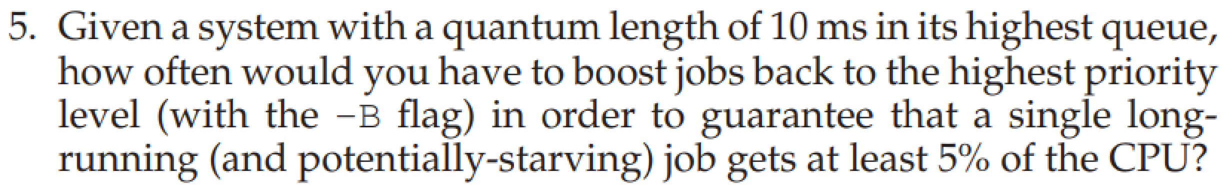
Question 4


4. 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.

```
UNIX> python mlfq.py -l 0,200,9:0,200,0 -q 10 -n 3 -S -c -i 0
```

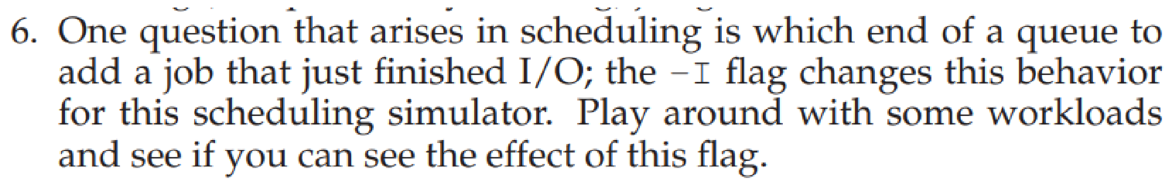
The process just needs to send an I/O request of 0 ms in every quantum length and thus it will monopolize the CPU.

Question 5

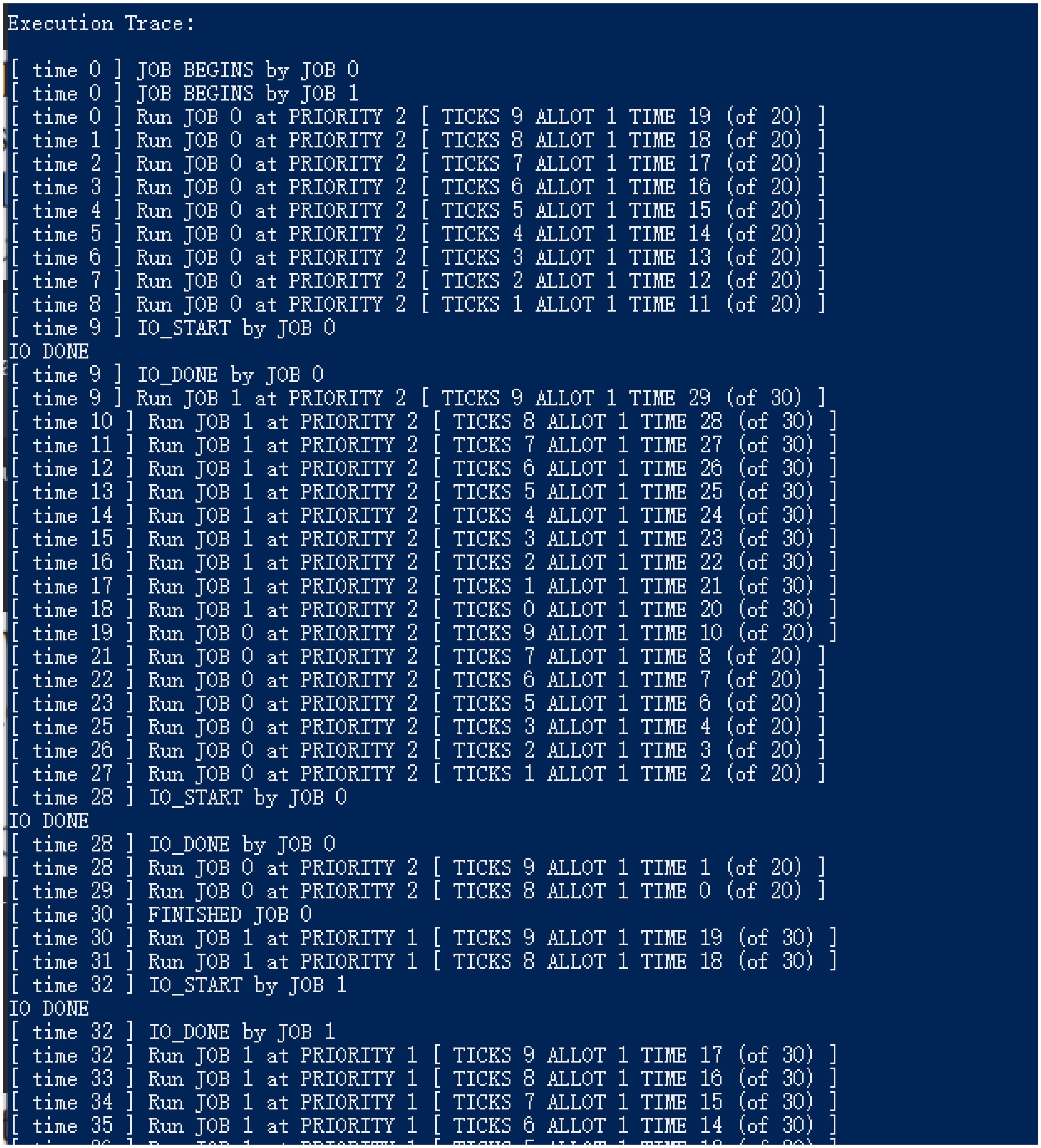

5. Given a system with a quantum length of 10 ms in its highest queue, how often would you have to boost jobs back to the highest priority level (with the -B flag) in order to guarantee that a single long-running (and potentially-starving) job gets at least 5% of the CPU?

Time quantum would have to be 200ms. The job will initially run for 10ms in Q1, then get demoted. After waiting for another 180ms, it can take the CPU again. $\frac{10}{200} = 5\%$

Question 6


6. One question that arises in scheduling is which end of a queue to add a job that just finished I/O; the -I flag changes this behavior for this scheduling simulator. Play around with some workloads and see if you can see the effect of this flag.

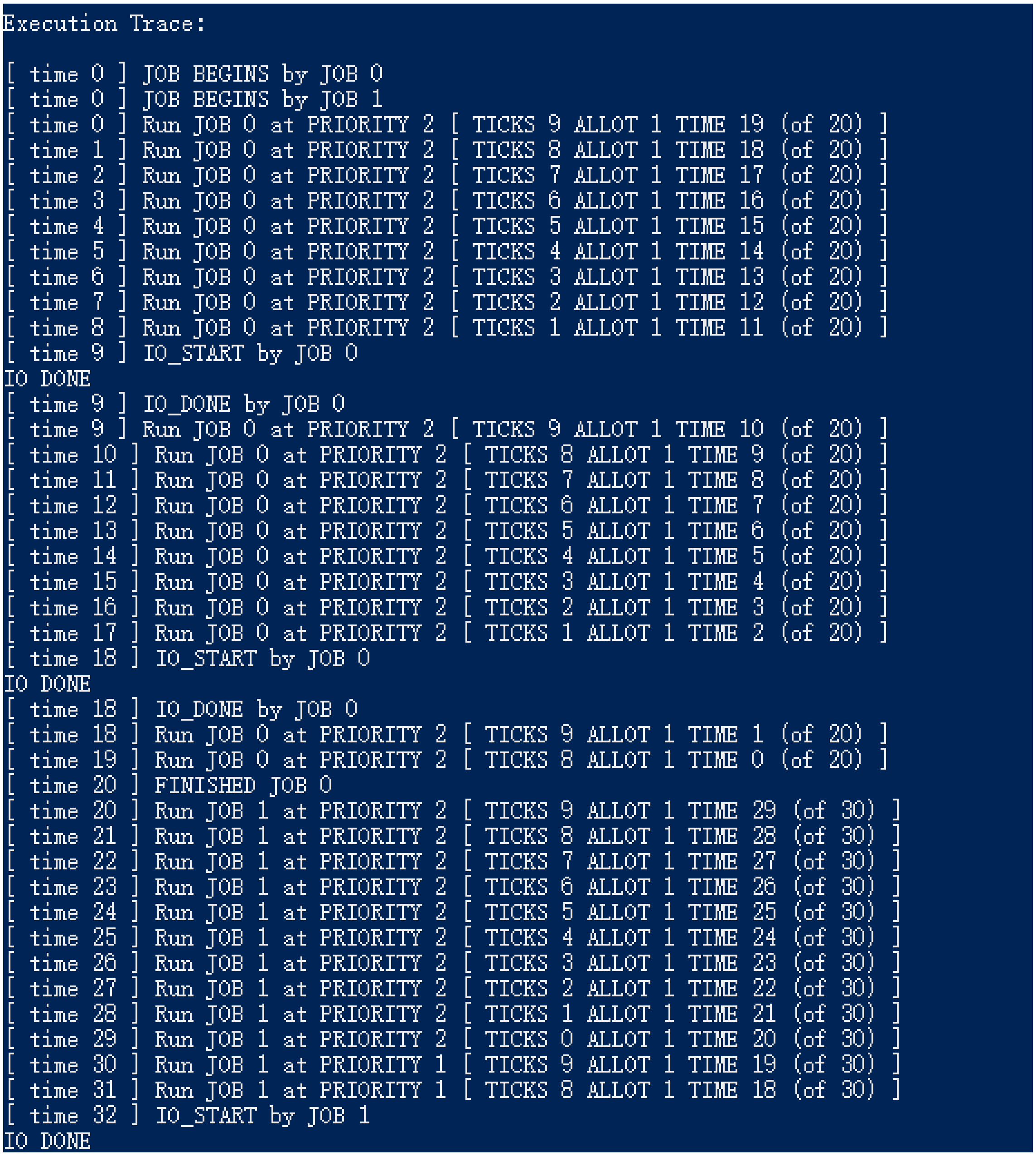
```
UNIX> python ./mlfq.py -l 0,20,9:0,30,12 -s 1 -c -i 0 -S
```



```
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 9 ALLOT 1 TIME 19 (of 20) ]
[ time 1 ] Run JOB 0 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 18 (of 20) ]
[ time 2 ] Run JOB 0 at PRIORITY 2 [ TICKS 7 ALLOT 1 TIME 17 (of 20) ]
[ time 3 ] Run JOB 0 at PRIORITY 2 [ TICKS 6 ALLOT 1 TIME 16 (of 20) ]
[ time 4 ] Run JOB 0 at PRIORITY 2 [ TICKS 5 ALLOT 1 TIME 15 (of 20) ]
[ time 5 ] Run JOB 0 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 14 (of 20) ]
[ time 6 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 13 (of 20) ]
[ time 7 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 12 (of 20) ]
[ time 8 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 11 (of 20) ]
[ time 9 ] IO_START by JOB 0
IO DONE
[ time 9 ] IO_DONE by JOB 0
[ time 9 ] Run JOB 1 at PRIORITY 2 [ TICKS 9 ALLOT 1 TIME 29 (of 30) ]
[ time 10 ] Run JOB 1 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 28 (of 30) ]
[ time 11 ] Run JOB 1 at PRIORITY 2 [ TICKS 7 ALLOT 1 TIME 27 (of 30) ]
[ time 12 ] Run JOB 1 at PRIORITY 2 [ TICKS 6 ALLOT 1 TIME 26 (of 30) ]
[ time 13 ] Run JOB 1 at PRIORITY 2 [ TICKS 5 ALLOT 1 TIME 25 (of 30) ]
[ time 14 ] Run JOB 1 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 24 (of 30) ]
[ time 15 ] Run JOB 1 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 23 (of 30) ]
[ time 16 ] Run JOB 1 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 22 (of 30) ]
[ time 17 ] Run JOB 1 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 21 (of 30) ]
[ time 18 ] Run JOB 1 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 20 (of 30) ]
[ time 19 ] Run JOB 0 at PRIORITY 2 [ TICKS 9 ALLOT 1 TIME 10 (of 20) ]
[ time 21 ] Run JOB 0 at PRIORITY 2 [ TICKS 7 ALLOT 1 TIME 8 (of 20) ]
[ time 22 ] Run JOB 0 at PRIORITY 2 [ TICKS 6 ALLOT 1 TIME 7 (of 20) ]
[ time 23 ] Run JOB 0 at PRIORITY 2 [ TICKS 5 ALLOT 1 TIME 6 (of 20) ]
[ time 25 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 4 (of 20) ]
[ time 26 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 3 (of 20) ]
[ time 27 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 2 (of 20) ]
[ time 28 ] IO_START by JOB 0
IO DONE
[ time 28 ] IO_DONE by JOB 0
[ time 28 ] Run JOB 0 at PRIORITY 2 [ TICKS 9 ALLOT 1 TIME 1 (of 20) ]
[ time 29 ] Run JOB 0 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 0 (of 20) ]
[ time 30 ] FINISHED JOB 0
[ time 30 ] Run JOB 1 at PRIORITY 1 [ TICKS 9 ALLOT 1 TIME 19 (of 30) ]
[ time 31 ] Run JOB 1 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 18 (of 30) ]
[ time 32 ] IO_START by JOB 1
IO DONE
[ time 32 ] IO_DONE by JOB 1
[ time 32 ] Run JOB 1 at PRIORITY 1 [ TICKS 9 ALLOT 1 TIME 17 (of 30) ]
[ time 33 ] Run JOB 1 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 16 (of 30) ]
[ time 34 ] Run JOB 1 at PRIORITY 1 [ TICKS 7 ALLOT 1 TIME 15 (of 30) ]
[ time 35 ] Run JOB 1 at PRIORITY 1 [ TICKS 6 ALLOT 1 TIME 14 (of 30) ]
[ time 36 ] Run JOB 1 at PRIORITY 1 [ TICKS 5 ALLOT 1 TIME 13 (of 30) ]
```

```
UNIX> python ./mlfq.py -l 0,20,9:0,30,12 -s 1 -c -i 0 -S -I
```



```
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 9 ALLOT 1 TIME 19 (of 20) ]
[ time 1 ] Run JOB 0 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 18 (of 20) ]
[ time 2 ] Run JOB 0 at PRIORITY 2 [ TICKS 7 ALLOT 1 TIME 17 (of 20) ]
[ time 3 ] Run JOB 0 at PRIORITY 2 [ TICKS 6 ALLOT 1 TIME 16 (of 20) ]
[ time 4 ] Run JOB 0 at PRIORITY 2 [ TICKS 5 ALLOT 1 TIME 15 (of 20) ]
[ time 5 ] Run JOB 0 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 14 (of 20) ]
[ time 6 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 13 (of 20) ]
[ time 7 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 12 (of 20) ]
[ time 8 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 11 (of 20) ]
[ time 9 ] IO_START by JOB 0
IO DONE
[ time 9 ] IO_DONE by JOB 0
[ time 9 ] Run JOB 0 at PRIORITY 2 [ TICKS 9 ALLOT 1 TIME 10 (of 20) ]
[ time 10 ] Run JOB 0 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 9 (of 20) ]
[ time 11 ] Run JOB 0 at PRIORITY 2 [ TICKS 7 ALLOT 1 TIME 8 (of 20) ]
[ time 12 ] Run JOB 0 at PRIORITY 2 [ TICKS 6 ALLOT 1 TIME 7 (of 20) ]
[ time 13 ] Run JOB 0 at PRIORITY 2 [ TICKS 5 ALLOT 1 TIME 6 (of 20) ]
[ time 14 ] Run JOB 0 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 5 (of 20) ]
[ time 15 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 4 (of 20) ]
[ time 16 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 3 (of 20) ]
[ time 17 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 2 (of 20) ]
[ time 18 ] IO_START by JOB 0
IO DONE
[ time 18 ] IO_DONE by JOB 0
[ time 18 ] Run JOB 0 at PRIORITY 2 [ TICKS 9 ALLOT 1 TIME 1 (of 20) ]
[ time 19 ] Run JOB 0 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 0 (of 20) ]
[ time 20 ] FINISHED JOB 0
[ time 20 ] Run JOB 1 at PRIORITY 2 [ TICKS 9 ALLOT 1 TIME 29 (of 30) ]
[ time 21 ] Run JOB 1 at PRIORITY 2 [ TICKS 8 ALLOT 1 TIME 28 (of 30) ]
[ time 22 ] Run JOB 1 at PRIORITY 2 [ TICKS 7 ALLOT 1 TIME 27 (of 30) ]
[ time 23 ] Run JOB 1 at PRIORITY 2 [ TICKS 6 ALLOT 1 TIME 26 (of 30) ]
[ time 24 ] Run JOB 1 at PRIORITY 2 [ TICKS 5 ALLOT 1 TIME 25 (of 30) ]
[ time 25 ] Run JOB 1 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 24 (of 30) ]
[ time 26 ] Run JOB 1 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 23 (of 30) ]
[ time 27 ] Run JOB 1 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 22 (of 30) ]
[ time 28 ] Run JOB 1 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 21 (of 30) ]
[ time 29 ] Run JOB 1 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 20 (of 30) ]
[ time 30 ] Run JOB 1 at PRIORITY 1 [ TICKS 9 ALLOT 1 TIME 19 (of 30) ]
[ time 31 ] Run JOB 1 at PRIORITY 1 [ TICKS 8 ALLOT 1 TIME 18 (of 30) ]
[ time 32 ] IO_START by JOB 1
IO DONE
```

With the `-S` and `-I` option, we can have a process fully monopolize the CPU once issued an I/O request.