Introduction to Operating Systems

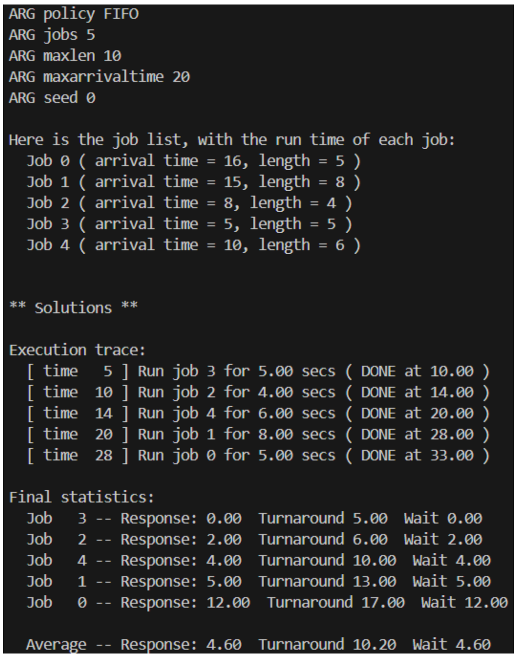
HW1

Name: 신종훈

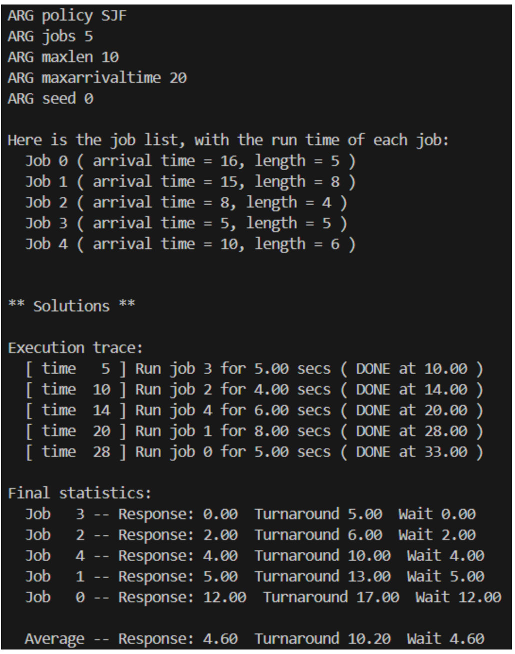
Student ID: 2020311573

Date: 2024.10.11

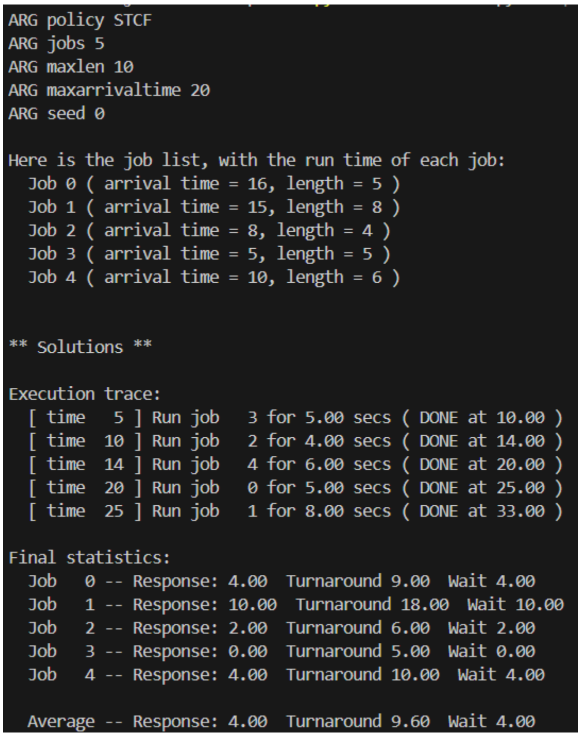
1. STCF Coding



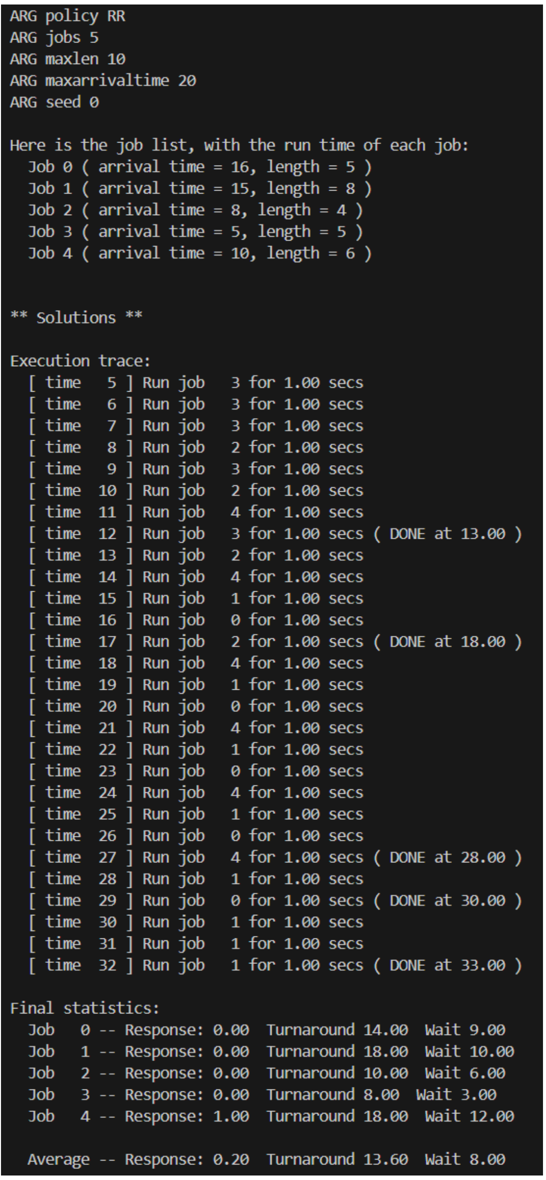
```
ARG policy FIFO
ARG jobs 5
ARG maxlen 10
ARG maxarrivaltime 20
ARG seed 0

Here is the job list, with the run time of each job:
  Job 0 ( arrival time = 16, length = 5 )
  Job 1 ( arrival time = 15, length = 8 )
  Job 2 ( arrival time = 8, length = 4 )
  Job 3 ( arrival time = 5, length = 5 )
  Job 4 ( arrival time = 10, length = 6 )


** Solutions **

Execution trace:
  [ time   5 ] Run job 3 for 5.00 secs ( DONE at 10.00 )
  [ time  10 ] Run job 2 for 4.00 secs ( DONE at 14.00 )
  [ time  14 ] Run job 4 for 6.00 secs ( DONE at 20.00 )
  [ time  20 ] Run job 1 for 8.00 secs ( DONE at 28.00 )
  [ time  28 ] Run job 0 for 5.00 secs ( DONE at 33.00 )

Final statistics:
  Job   3 -- Response: 0.00  Turnaround 5.00  Wait 0.00
  Job   2 -- Response: 2.00  Turnaround 6.00  Wait 2.00
  Job   4 -- Response: 4.00  Turnaround 10.00  Wait 4.00
  Job   1 -- Response: 5.00  Turnaround 13.00  Wait 5.00
  Job   0 -- Response: 12.00  Turnaround 17.00  Wait 12.00

  Average -- Response: 4.60  Turnaround 10.20  Wait 4.60
```



```
ARG policy SJF
ARG jobs 5
ARG maxlen 10
ARG maxarrivaltime 20
ARG seed 0

Here is the job list, with the run time of each job:
  Job 0 ( arrival time = 16, length = 5 )
  Job 1 ( arrival time = 15, length = 8 )
  Job 2 ( arrival time = 8, length = 4 )
  Job 3 ( arrival time = 5, length = 5 )
  Job 4 ( arrival time = 10, length = 6 )


** Solutions **

Execution trace:
  [ time   5 ] Run job 3 for 5.00 secs ( DONE at 10.00 )
  [ time  10 ] Run job 2 for 4.00 secs ( DONE at 14.00 )
  [ time  14 ] Run job 4 for 6.00 secs ( DONE at 20.00 )
  [ time  20 ] Run job 1 for 8.00 secs ( DONE at 28.00 )
  [ time  28 ] Run job 0 for 5.00 secs ( DONE at 33.00 )

Final statistics:
  Job   3 -- Response: 0.00  Turnaround 5.00  Wait 0.00
  Job   2 -- Response: 2.00  Turnaround 6.00  Wait 2.00
  Job   4 -- Response: 4.00  Turnaround 10.00  Wait 4.00
  Job   1 -- Response: 5.00  Turnaround 13.00  Wait 5.00
  Job   0 -- Response: 12.00  Turnaround 17.00  Wait 12.00

  Average -- Response: 4.60  Turnaround 10.20  Wait 4.60
```



```
ARG policy STCF
ARG jobs 5
ARG maxlen 10
ARG maxarrivaltime 20
ARG seed 0

Here is the job list, with the run time of each job:
  Job 0 ( arrival time = 16, length = 5 )
  Job 1 ( arrival time = 15, length = 8 )
  Job 2 ( arrival time = 8, length = 4 )
  Job 3 ( arrival time = 5, length = 5 )
  Job 4 ( arrival time = 10, length = 6 )


** Solutions **

Execution trace:
  [ time   5 ] Run job   3 for 5.00 secs ( DONE at 10.00 )
  [ time  10 ] Run job   2 for 4.00 secs ( DONE at 14.00 )
  [ time  14 ] Run job   4 for 6.00 secs ( DONE at 20.00 )
  [ time  20 ] Run job   0 for 5.00 secs ( DONE at 25.00 )
  [ time  25 ] Run job   1 for 8.00 secs ( DONE at 33.00 )

Final statistics:
  Job   0 -- Response: 4.00  Turnaround 9.00  Wait 4.00
  Job   1 -- Response: 10.00  Turnaround 18.00  Wait 10.00
  Job   2 -- Response: 2.00  Turnaround 6.00  Wait 2.00
  Job   3 -- Response: 0.00  Turnaround 5.00  Wait 0.00
  Job   4 -- Response: 4.00  Turnaround 10.00  Wait 4.00

  Average -- Response: 4.00  Turnaround 9.60  Wait 4.00
```



```
ARG policy RR
ARG jobs 5
ARG maxlen 10
ARG maxarrivaltime 20
ARG seed 0

Here is the job list, with the run time of each job:
  Job 0 ( arrival time = 16, length = 5 )
  Job 1 ( arrival time = 15, length = 8 )
  Job 2 ( arrival time = 8, length = 4 )
  Job 3 ( arrival time = 5, length = 5 )
  Job 4 ( arrival time = 10, length = 6 )


** Solutions **

Execution trace:
  [ time   5 ] Run job   3 for 1.00 secs
  [ time   6 ] Run job   3 for 1.00 secs
  [ time   7 ] Run job   3 for 1.00 secs
  [ time   8 ] Run job   2 for 1.00 secs
  [ time   9 ] Run job   3 for 1.00 secs
  [ time  10 ] Run job   2 for 1.00 secs
  [ time  11 ] Run job   4 for 1.00 secs
  [ time  12 ] Run job   3 for 1.00 secs ( DONE at 13.00 )
  [ time  13 ] Run job   2 for 1.00 secs
  [ time  14 ] Run job   4 for 1.00 secs
  [ time  15 ] Run job   1 for 1.00 secs
  [ time  16 ] Run job   0 for 1.00 secs
  [ time  17 ] Run job   2 for 1.00 secs ( DONE at 18.00 )
  [ time  18 ] Run job   4 for 1.00 secs
  [ time  19 ] Run job   1 for 1.00 secs
  [ time  20 ] Run job   0 for 1.00 secs
  [ time  21 ] Run job   4 for 1.00 secs
  [ time  22 ] Run job   1 for 1.00 secs
  [ time  23 ] Run job   0 for 1.00 secs
  [ time  24 ] Run job   4 for 1.00 secs
  [ time  25 ] Run job   1 for 1.00 secs
  [ time  26 ] Run job   0 for 1.00 secs
  [ time  27 ] Run job   4 for 1.00 secs ( DONE at 28.00 )
  [ time  28 ] Run job   1 for 1.00 secs
  [ time  29 ] Run job   0 for 1.00 secs ( DONE at 30.00 )
  [ time  30 ] Run job   1 for 1.00 secs
  [ time  31 ] Run job   1 for 1.00 secs
  [ time  32 ] Run job   1 for 1.00 secs ( DONE at 33.00 )

Final statistics:
  Job   0 -- Response: 0.00  Turnaround 14.00  Wait 9.00
  Job   1 -- Response: 0.00  Turnaround 18.00  Wait 10.00
  Job   2 -- Response: 0.00  Turnaround 10.00  Wait 6.00
  Job   3 -- Response: 0.00  Turnaround 8.00  Wait 3.00
  Job   4 -- Response: 1.00  Turnaround 18.00  Wait 12.00

  Average -- Response: 0.20  Turnaround 13.60  Wait 8.00
```

	FIFO	SJF	STCF	RR
RESPONSE TIME [MS]	4.60	4.60	4.00	0.20
TURNAROUND TIME [MS]	10.20	10.20	9.60	13.60

Execution trace를 통해서 FIFO, SJF, STCF, RR 모두 의도한 대로 동작하고 있음을 확인할 수 있다.

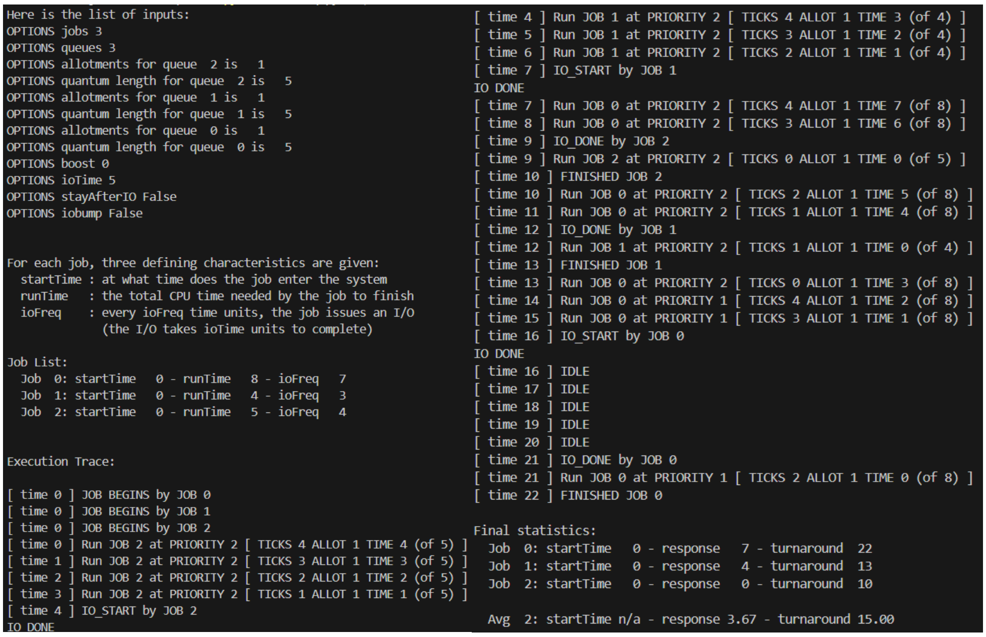
모든 job이 서로 다른 arrival time을 지니고 있어서 FIFO와 SJF는 동일한 response time과 turnaround time을 지닌다. SJF에 preemption을 추가한 STCF는 SJF보다 적은 response time과 turnaround time을 지닌다. RR에서 time slice를 매우 작은 1로 설정하여 가장 작은 response time을 지닌 반면에 가장 큰 turnaround time을 지닌다.

2. MLFQ Analysis

   1) 해당 코드에서는 처리 중인 job이 time slice를 모두 채우기 전에는 해당 queue에서 pop되지 않습니다. 따라서 어떤 job이 time slice를 다 못 채우고 preempt되는 경우에 해당 job은 기존 queue에서 기존의 위치를 유지합니다.
   2) 해당 코드에서 priority boost가 발생할 경우에 boost되는 모든 job들은 highest queue에 배치된다. 기존에 상위 큐에 있던 job들이 highest queue의 앞에 배치되고 하위 큐에 있던 job들은 뒤에 배치된다. 또한 동일한 큐에 있던 job들은 기존 큐에서의 순서를 유지하며 highest queue에 배치된다.

3. MLFQ Coding

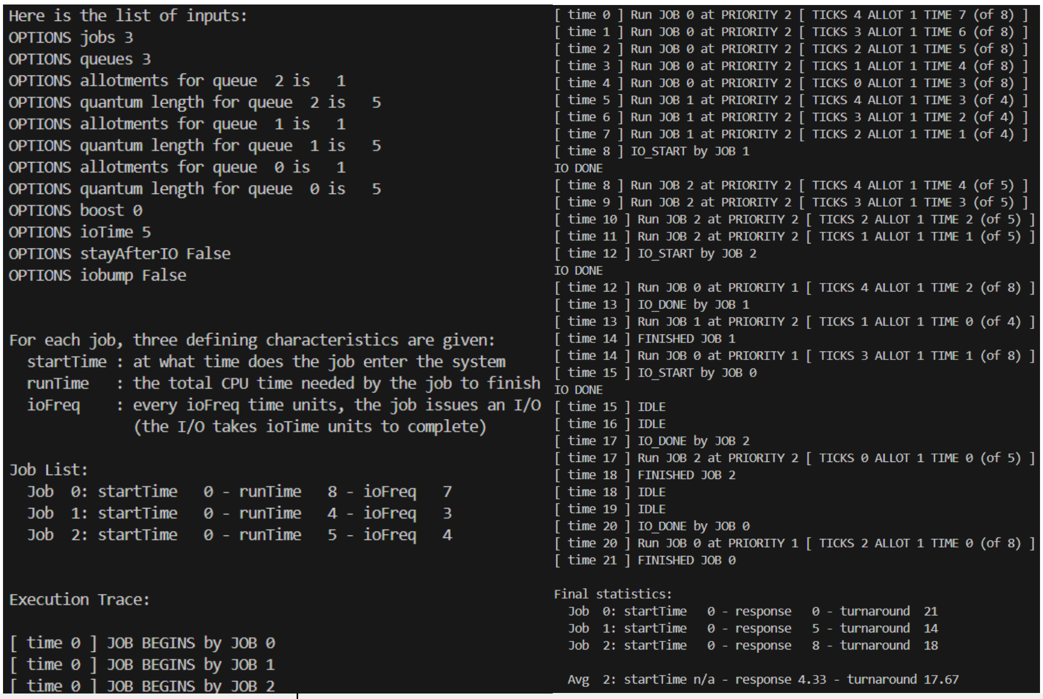
arrjobFront == False:



```
Here is the list of inputs:
OPTIONS jobs 3
OPTIONS queues 3
OPTIONS allotments for queue  2 is   1
OPTIONS quantum length for queue  2 is   5
OPTIONS allotments for queue  1 is   1
OPTIONS quantum length for queue  1 is   5
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  0 is   5
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   8 - ioFreq   7
  Job  1: startTime   0 - runTime   4 - ioFreq   3
  Job  2: startTime   0 - runTime   5 - ioFreq   4


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 2 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 4 (of 5) ]
[ time 1 ] Run JOB 2 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 3 (of 5) ]
[ time 2 ] Run JOB 2 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 2 (of 5) ]
[ time 3 ] Run JOB 2 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 1 (of 5) ]
[ time 4 ] IO_START by JOB 2
IO DONE
```

```
[ time 4 ] Run JOB 1 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 3 (of 4) ]
[ time 5 ] Run JOB 1 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 2 (of 4) ]
[ time 6 ] Run JOB 1 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 1 (of 4) ]
[ time 7 ] IO_START by JOB 1
IO DONE
[ time 7 ] Run JOB 0 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 7 (of 8) ]
[ time 8 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 6 (of 8) ]
[ time 9 ] IO_DONE by JOB 2
[ time 9 ] Run JOB 2 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 0 (of 5) ]
[ time 10 ] FINISHED JOB 2
[ time 10 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 5 (of 8) ]
[ time 11 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 4 (of 8) ]
[ time 12 ] IO_DONE by JOB 1
[ time 12 ] Run JOB 1 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 0 (of 4) ]
[ time 13 ] FINISHED JOB 1
[ time 13 ] Run JOB 0 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 3 (of 8) ]
[ time 14 ] Run JOB 0 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 2 (of 8) ]
[ time 15 ] Run JOB 0 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 1 (of 8) ]
[ time 16 ] IO_START by JOB 0
IO DONE
[ time 16 ] IDLE
[ time 17 ] IDLE
[ time 18 ] IDLE
[ time 19 ] IDLE
[ time 20 ] IDLE
[ time 21 ] IO_DONE by JOB 0
[ time 21 ] Run JOB 0 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 0 (of 8) ]
[ time 22 ] FINISHED JOB 0

Final statistics:
  Job  0: startTime   0 - response   7 - turnaround  22
  Job  1: startTime   0 - response   4 - turnaround  13
  Job  2: startTime   0 - response   0 - turnaround  10

  Avg  2: startTime n/a - response 3.67 - turnaround 15.00
```

arrjobFront == True:



```
Here is the list of inputs:
OPTIONS jobs 3
OPTIONS queues 3
OPTIONS allotments for queue  2 is   1
OPTIONS quantum length for queue  2 is   5
OPTIONS allotments for queue  1 is   1
OPTIONS quantum length for queue  1 is   5
OPTIONS allotments for queue  0 is   1
OPTIONS quantum length for queue  0 is   5
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   8 - ioFreq   7
  Job  1: startTime   0 - runTime   4 - ioFreq   3
  Job  2: startTime   0 - runTime   5 - ioFreq   4


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 2 [ TICKS 4 ALLOT 1 TIME 7 (of 8) ]
[ time 1 ] Run JOB 0 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 6 (of 8) ]
[ time 2 ] Run JOB 0 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 5 (of 8) ]
[ time 3 ] Run JOB 0 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 4 (of 8) ]
[ time 4 ] Run JOB 0 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 3 (of 8) ]
[ time 5 ] Run JOB 1 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 3 (of 4) ]
[ time 6 ] Run JOB 1 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 2 (of 4) ]
[ time 7 ] Run JOB 1 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 1 (of 4) ]
[ time 8 ] IO_START by JOB 1
IO DONE
[ time 8 ] Run JOB 2 at PRIORITY 2 [ TICKS 4 ALLOT 1 TIME 4 (of 5) ]
[ time 9 ] Run JOB 2 at PRIORITY 2 [ TICKS 3 ALLOT 1 TIME 3 (of 5) ]
[ time 10 ] Run JOB 2 at PRIORITY 2 [ TICKS 2 ALLOT 1 TIME 2 (of 5) ]
[ time 11 ] Run JOB 2 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 1 (of 5) ]
[ time 12 ] IO_START by JOB 2
IO DONE
[ time 12 ] Run JOB 0 at PRIORITY 1 [ TICKS 4 ALLOT 1 TIME 2 (of 8) ]
[ time 13 ] IO_DONE by JOB 1
[ time 13 ] Run JOB 1 at PRIORITY 2 [ TICKS 1 ALLOT 1 TIME 0 (of 4) ]
[ time 14 ] FINISHED JOB 1
[ time 14 ] Run JOB 0 at PRIORITY 1 [ TICKS 3 ALLOT 1 TIME 1 (of 8) ]
[ time 15 ] IO_START by JOB 0
IO DONE
[ time 15 ] IDLE
[ time 16 ] IDLE
[ time 17 ] IO_DONE by JOB 2
[ time 17 ] Run JOB 2 at PRIORITY 2 [ TICKS 0 ALLOT 1 TIME 0 (of 5) ]
[ time 18 ] FINISHED JOB 2
[ time 18 ] IDLE
[ time 19 ] IDLE
[ time 20 ] IO_DONE by JOB 0
[ time 20 ] Run JOB 0 at PRIORITY 1 [ TICKS 2 ALLOT 1 TIME 0 (of 8) ]
[ time 21 ] FINISHED JOB 0

Final statistics:
  Job  0: startTime   0 - response   0 - turnaround  21
  Job  1: startTime   0 - response   5 - turnaround  14
  Job  2: startTime   0 - response   8 - turnaround  18

  Avg  2: startTime n/a - response 4.33 - turnaround 17.67
```

arrjobFront	False	True
Response time [ms]	3.67	15.00
Turnaround time [ms]	4.33	17.67

Execution Trace를 통해서 arrjobFront가 False인 경우와 True인 경우 모두 의도한 바와 같이 작동함을 확인할 수 있다. arrjobFront의 값을 False로 하여 처음 arrive 한 job이 queue의 뒤에 추가되도록 하는 것이 response time과 turnaround time 모두를 감소시킴을 확인할 수 있다.