

King Saud University
College of Computer and Information Sciences
Department of Information Technology

CSC227: Operating System

2nd Semester 1445 H

Project title

Multi-Level Queue

Group#2

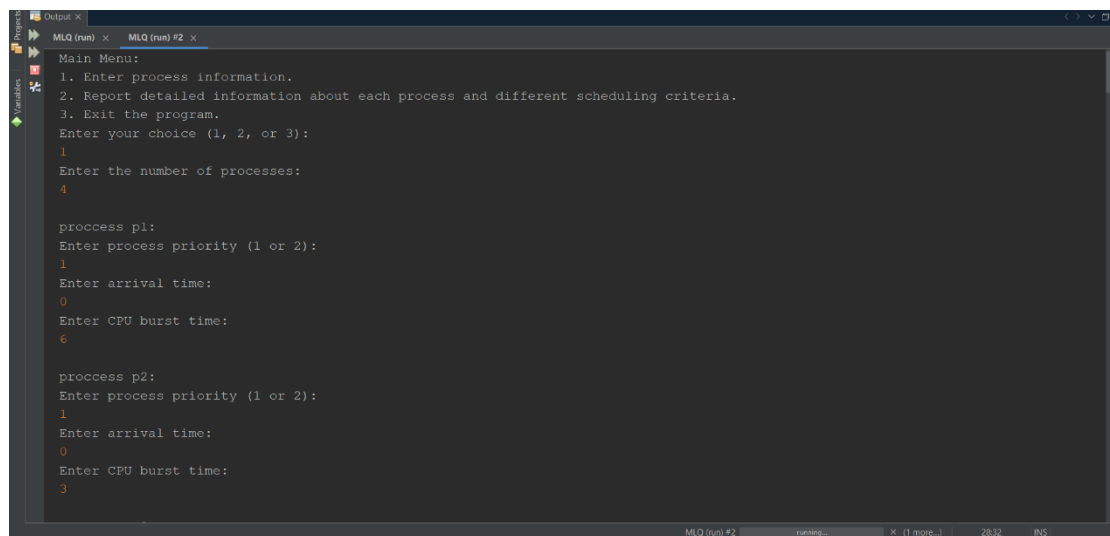
Section: 56304	
Name	ID
Wiam Baalahtar (Group Leader)	443200416
Raghad Alqabbani	443200508
Hind Alhijailan	443200971
Haya Alhawaimel	443200917
Hessa Alajalin	443200670

Supervised By: L.Malak Bagais

I. The RR condition is demonstrated with different scenarios:

We added four processes with different arrival times and CPU bursts (less/equal/more than time quantum) in queue 1 for showing the preemptive round robin with quantum 3. Each process will interrupt after three milliseconds and execute the next process based on the queue order until all processes have terminated.

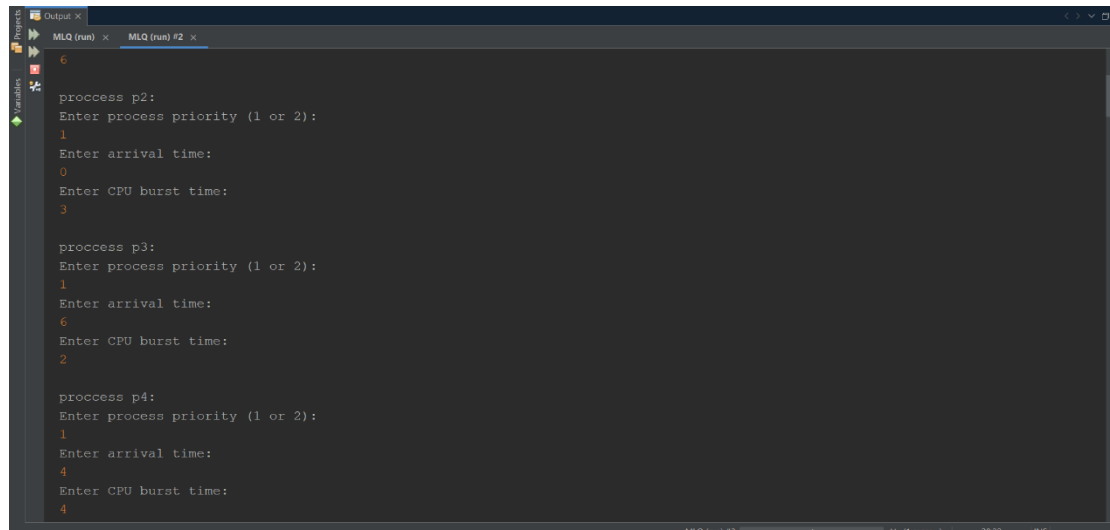
Input:



```
Output X
MLQ (run) x MLQ (run) #2 x
Main Menu:
1. Enter process information.
2. Report detailed information about each process and different scheduling criteria.
3. Exit the program.
Enter your choice (1, 2, or 3):
1
Enter the number of processes:
4

process p1:
Enter process priority (1 or 2):
1
Enter arrival time:
0
Enter CPU burst time:
6

process p2:
Enter process priority (1 or 2):
1
Enter arrival time:
0
Enter CPU burst time:
3
```



```
6
process p2:
Enter process priority (1 or 2):
1
Enter arrival time:
0
Enter CPU burst time:
3

process p3:
Enter process priority (1 or 2):
1
Enter arrival time:
6
Enter CPU burst time:
2

process p4:
Enter process priority (1 or 2):
1
Enter arrival time:
4
Enter CPU burst time:
4
```

Output:

```
MLQ (run) x MLQ (run) #2 x
3. Exit the program.
Enter your choice (1, 2, or 3):
2
Scheduling order of the processes: [P1 | P2 | P1 | P4 | P3 | P4]

Process ID: P1
Priority: 1
Arrival time: 0
CPU burst: 6
Start time: 0
Termination time: 9
Turnaround time: 9
Waiting time: 3
Response time: 0

Process ID: P2
Priority: 1
Arrival time: 0
CPU burst: 3
Start time: 3
Termination time: 6
Turnaround time: 6
Waiting time: 3
Response time: 3
```

```
Output x
MLQ (run) x MLQ (run) #2 x
Response time: 3

Process ID: P4
Priority: 1
Arrival time: 4
CPU burst: 4
Start time: 9
Termination time: 15
Turnaround time: 11
Waiting time: 7
Response time: 5

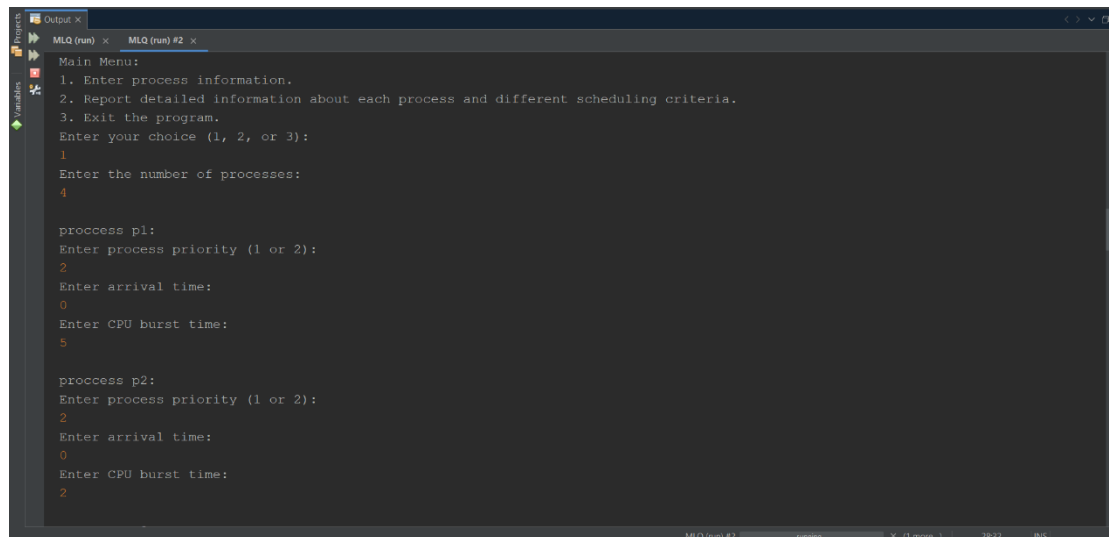
Process ID: P3
Priority: 1
Arrival time: 6
CPU burst: 2
Start time: 12
Termination time: 14
Turnaround time: 8
Waiting time: 6
Response time: 6

Average Turnaround Time: 8.5
Average Waiting Time: 4.75
Average Response Time: 3.5
```

ii. The non-preemptive Shortest-Job-First (SJF) scheduling is demonstrated:

As shown, we add four processes to queue 2 so we can show the non-preemptive short job. The process that arrived and has the shorter burst from than other processes that arrived too, will execute first until process termination, and we repeat this cycle with rest of processes until all processes terminate.

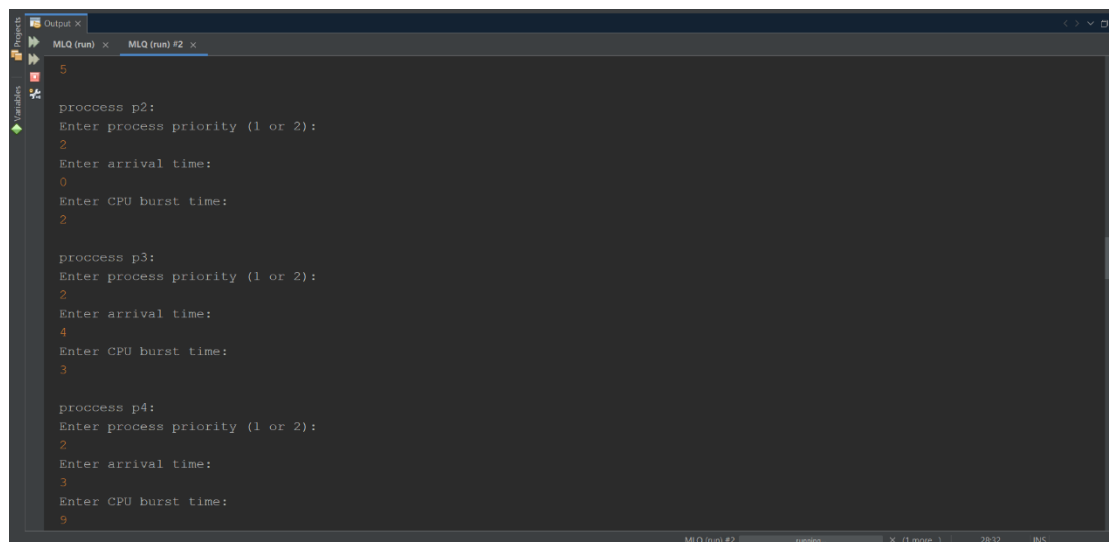
Input:



```
Output x
MLQ (run) x MLQ (run) #2 x
Main Menu:
1. Enter process information.
2. Report detailed information about each process and different scheduling criteria.
3. Exit the program.
Enter your choice (1, 2, or 3):
1
Enter the number of processes:
4

process p1:
Enter process priority (1 or 2):
2
Enter arrival time:
0
Enter CPU burst time:
5

process p2:
Enter process priority (1 or 2):
2
Enter arrival time:
0
Enter CPU burst time:
2
```



```
5
process p2:
Enter process priority (1 or 2):
2
Enter arrival time:
0
Enter CPU burst time:
2

process p3:
Enter process priority (1 or 2):
2
Enter arrival time:
4
Enter CPU burst time:
3

process p4:
Enter process priority (1 or 2):
2
Enter arrival time:
3
Enter CPU burst time:
9
```

Output:

```
Output X
MLQ (run) x MLQ (run) #2 x
Enter your choice (1, 2, or 3):
2
Scheduling order of the processes: [P2 | P1 | P3 | P4]

Process ID: P2
Priority: 2
Arrival time: 0
CPU burst: 2
Start time: 0
Termination time: 2
Turnaround time: 2
Waiting time: 0
Response time: 0

Process ID: P1
Priority: 2
Arrival time: 0
CPU burst: 5
Start time: 2
Termination time: 7
Turnaround time: 7
Waiting time: 2
Response time: 2
```

```
Output X
MLQ (run) x MLQ (run) #2 x
Response time: 2

Process ID: P3
Priority: 2
Arrival time: 4
CPU burst: 3
Start time: 7
Termination time: 10
Turnaround time: 6
Waiting time: 3
Response time: 3

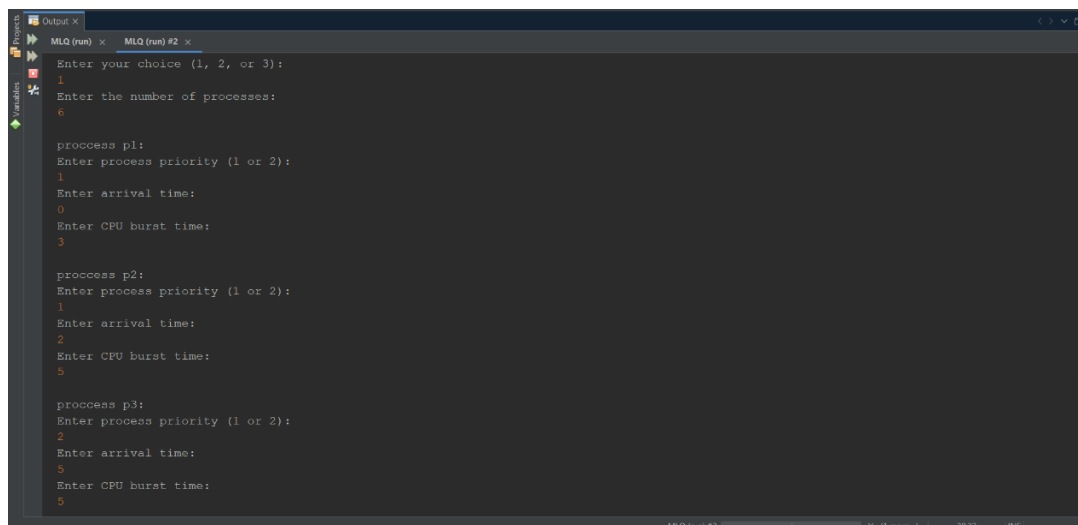
Process ID: P4
Priority: 2
Arrival time: 3
CPU burst: 9
Start time: 10
Termination time: 19
Turnaround time: 16
Waiting time: 7
Response time: 7

Average Turnaround Time: 7.75
Average Waiting Time: 3.0
Average Response Time: 3.0
```

iii. Preemption is demonstrated (Round Robin and Short Job First):

In order to demonstrate working with two queues together, as shown, we add six processes (RR=4 and SJF=2), because round robin (queue1) has higher priority than short job first (queue2). The arrival processes in queue1 will be execution first, followed by the execution arrival processes in queue2. After each process that executes from queue2, we check to see if any processes have arrived in queue1; if so, we stop executing processes in queue2 and begin executing in queue1.

Input:

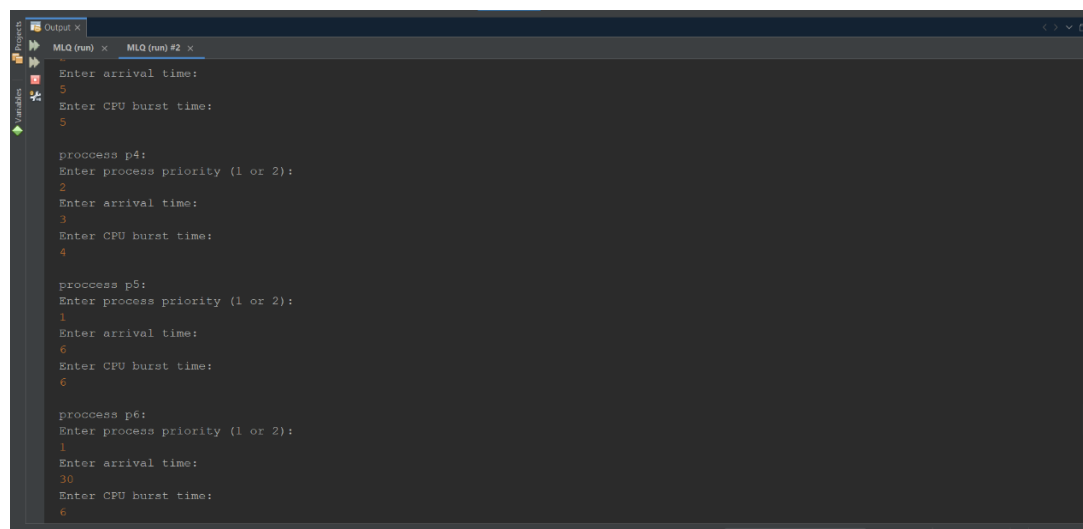


```
Output x
MLQ (run) x MLQ (run) #2 x
Enter your choice (1, 2, or 3):
1
Enter the number of processes:
6

process p1:
Enter process priority (1 or 2):
1
Enter arrival time:
0
Enter CPU burst time:
3

process p2:
Enter process priority (1 or 2):
1
Enter arrival time:
2
Enter CPU burst time:
5

process p3:
Enter process priority (1 or 2):
2
Enter arrival time:
5
Enter CPU burst time:
5
```



```
Output x
MLQ (run) x MLQ (run) #2 x
Enter arrival time:
5
Enter CPU burst time:
5

process p4:
Enter process priority (1 or 2):
2
Enter arrival time:
3
Enter CPU burst time:
4

process p5:
Enter process priority (1 or 2):
1
Enter arrival time:
6
Enter CPU burst time:
6

process p6:
Enter process priority (1 or 2):
1
Enter arrival time:
30
Enter CPU burst time:
6
```

Output:

```
Output x
MLQ (run) x MLQ (run) #2 x
Scheduling order of the processes: [P1 | P2 | P5 | P2 | P5 | P4 | P3 | P6 | P6]

Process ID: P1
Priority: 1
Arrival time: 0
CPU burst: 3
Start time: 0
Termination time: 3
Turnaround time: 3
Waiting time: 0
Response time: 0

Process ID: P2
Priority: 1
Arrival time: 2
CPU burst: 5
Start time: 3
Termination time: 11
Turnaround time: 9
Waiting time: 4
Response time: 1

Process ID: P5
Priority: 1
Arrival time: 6
CPU burst: 6
Start time: 6
Termination time: 14
```

```
Output x
MLQ (run) x MLQ (run) #2 x
Priority: 1
Arrival time: 2
CPU burst: 5
Start time: 3
Termination time: 11
Turnaround time: 9
Waiting time: 4
Response time: 1

Process ID: P5
Priority: 1
Arrival time: 6
CPU burst: 6
Start time: 6
Termination time: 14
Turnaround time: 8
Waiting time: 2
Response time: 0

Process ID: P4
Priority: 2
Arrival time: 3
CPU burst: 4
Start time: 14
Termination time: 18
Turnaround time: 15
Waiting time: 11
Response time: 11
```

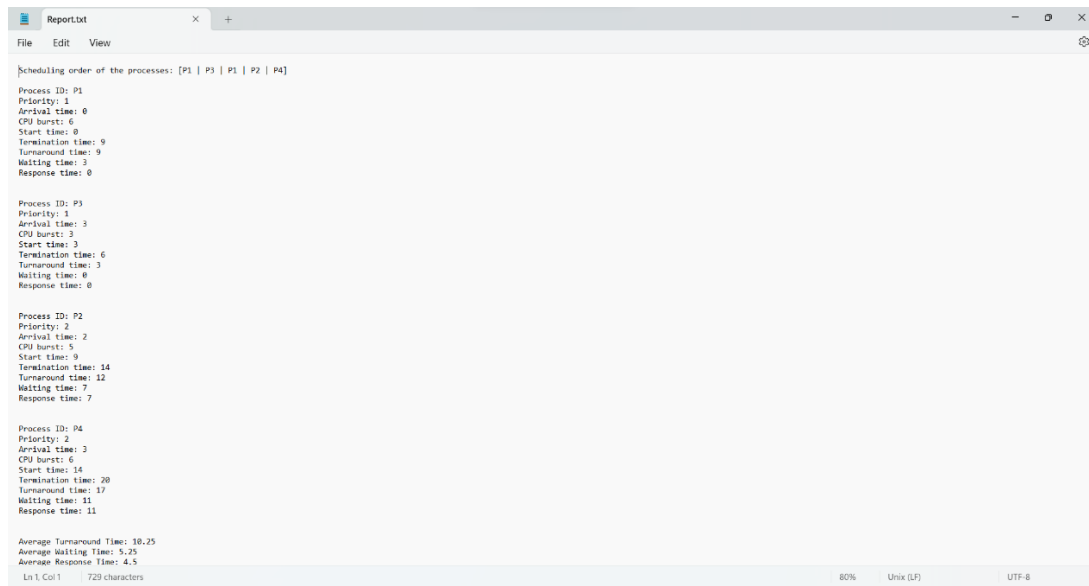
```
Output x
MLQ (run) x MLQ (run) #2 x
Termination time: 18
Turnaround time: 15
Waiting time: 11
Response time: 11

Process ID: P3
Priority: 2
Arrival time: 5
CPU burst: 5
Start time: 18
Termination time: 23
Turnaround time: 18
Waiting time: 13
Response time: 13

Process ID: P6
Priority: 1
Arrival time: 30
CPU burst: 6
Start time: 30
Termination time: 36
Turnaround time: 6
Waiting time: 0
Response time: 0

Average Turnaround Time: 9.833333333333334
Average Waiting Time: 5.0
Average Response Time: 4.166666666666667
```

iv. Screenshot of the report (Report.txt):



```
Report.txt
File Edit View

Scheduling order of the processes: [P1 | P3 | P1 | P2 | P4]

Process ID: P1
Priority: 1
Arrival time: 0
CPU burst: 6
Start time: 0
Termination time: 9
Turnaround time: 9
Waiting time: 3
Response time: 0

Process ID: P3
Priority: 1
Arrival time: 3
CPU burst: 3
Start time: 3
Termination time: 6
Turnaround time: 3
Waiting time: 0
Response time: 0

Process ID: P2
Priority: 2
Arrival time: 2
CPU burst: 5
Start time: 9
Termination time: 14
Turnaround time: 12
Waiting time: 7
Response time: 7

Process ID: P4
Priority: 2
Arrival time: 3
CPU burst: 6
Start time: 14
Termination time: 20
Turnaround time: 17
Waiting time: 11
Response time: 11

Average Turnaround Time: 10.25
Average Waiting Time: 5.25
Average Response Time: 4.5
Ln 1, Col 1 | 729 characters | 80% | Unix (LF) | UTF-8
```


Teamwork Evaluation:

Criteria	Hessa Alajalin	Hind Alhijaila	Raghad Alqabbani	Haya Alhawaimel	Wiam Baalahtar
Work division: Contributed equally to the work	1	1	1	1	1
Peer evaluation: Level of commitments (Interactivity with other team members), and professional behavior towards team & TA	1	1	1	1	1
Project Discussion: Accurate answers, understanding of the presented work, good listeners to questions	1	1	1	1	1
Time management: Attending on time, being ready to start the demo, good time management in discussion and demo.	1	1	1	1	1
Total/4	4	4	4	4	4

Table of task distribution:

Name:	Task:
Wiam Baalahtar	RR Logic, Logic of scheduling in multi-level queue, logic for computing the average (turn around, waiting, and response), Read me file.
Haya Alhawaimel	PCP and driver class, enterProcessInformation method, Read Me file.
Raghad Alqabbani	RR logic, generateReport method, enterProcessInformation method, Solve cod's errors.
Hind Alhijailan	SJF logic Logic for computing the average turn around, waiting, and response.
Hessa Alajalin	Report, Menu main, enterProcessInformation method, generateReport method.