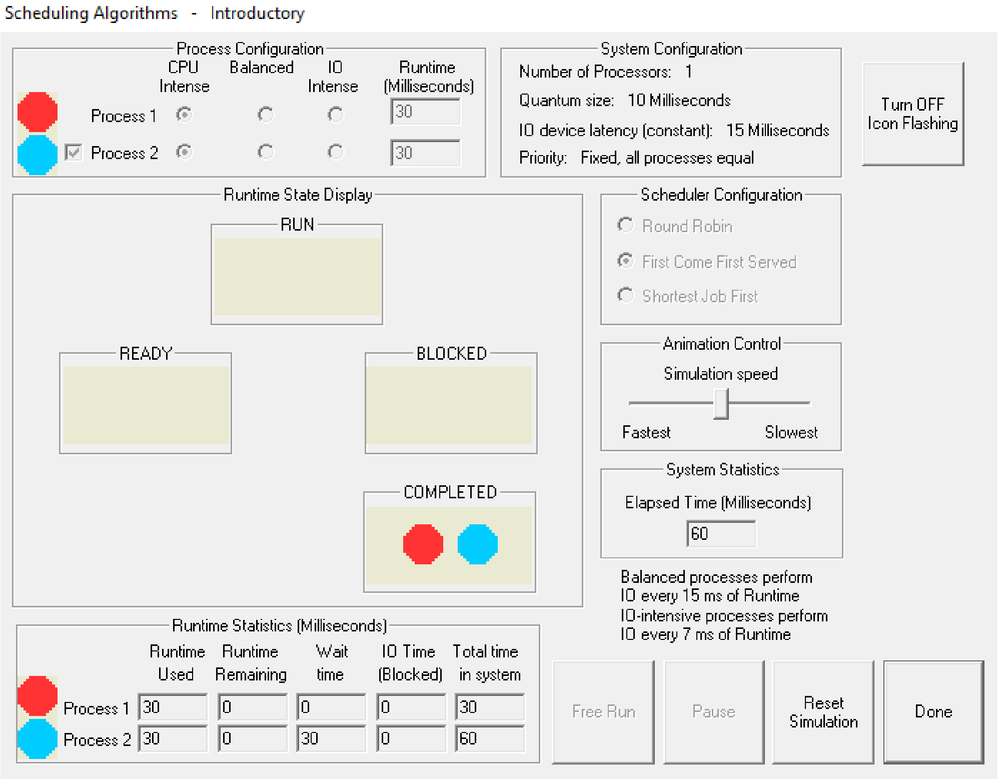
COMP1562 Logbook (Week 5)
Basic Information

1.1 Student name	**Trevor Kiggundu (001001720)**
1.2 Who did you work with? Name and/or id	**Maruf Hoque (001006731)**
1.3 Which lab topic does this document relate to?	Scheduling
1.4 How well do you feel you have done?	I have completed the exercise and am totally satisfied with my work.
1.5 Briefly explain your answer to question 1.4	My group and I were able to successfully follow and complete the tasks. Proof of that is shown below.

Annotated screenshots demonstrating what you have achieved:

EXERCISES:
Exercises a-h required us to run the various algorithm simulations in Workbench in order to monitor the results. Screenshots of the various simulations are shown below:

a. FCFS1



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	30
☑ Process 2	●	○	○	30

System Configuration
Number of Processors: 1
Quantum size: 10 Milliseconds
IO device latency (constant): 15 Milliseconds
Priority: Fixed, all processes equal

Scheduler Configuration
○ Round Robin
● First Come First Served
○ Shortest Job First

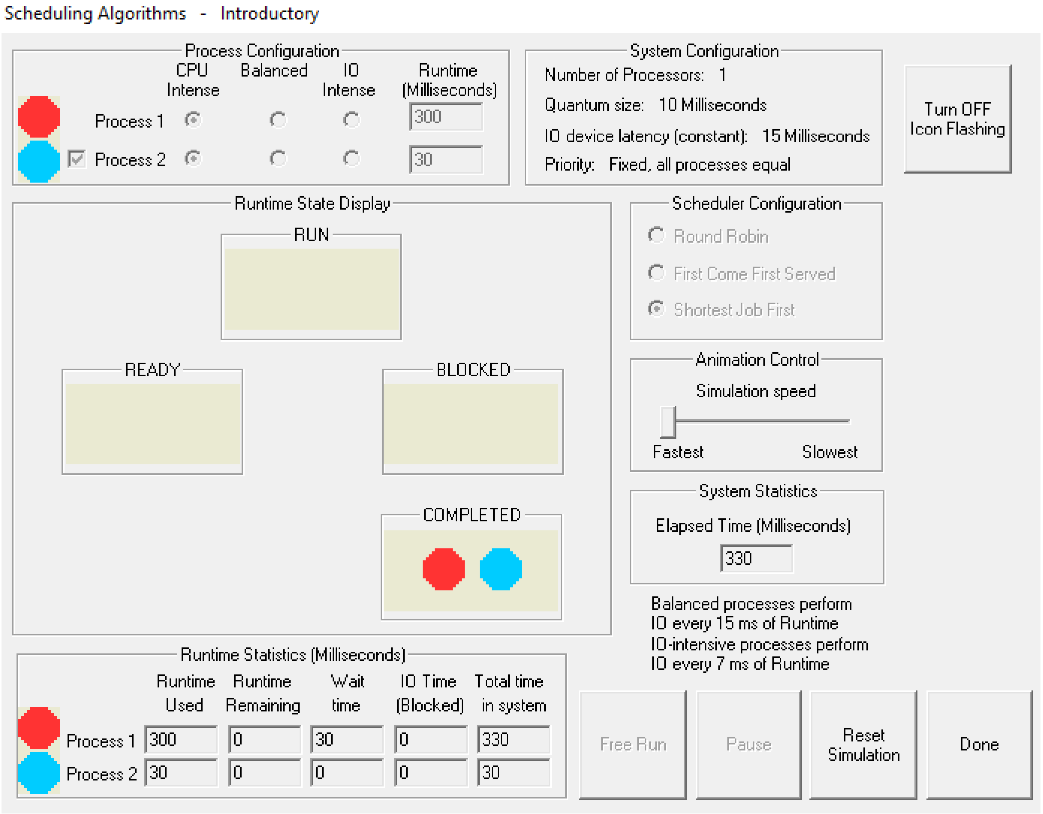
System Statistics
Elapsed Time (Milliseconds) 60

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	30	0	0	0	30
Process 2	30	0	30	0	60

b. SJF 1



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	300
☑ Process 2	●	○	○	30

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

○ Round Robin

○ First Come First Served

● Shortest Job First

Animation Control

Simulation speed

Fastest Slowest

System Statistics

Elapsed Time (Milliseconds) 330

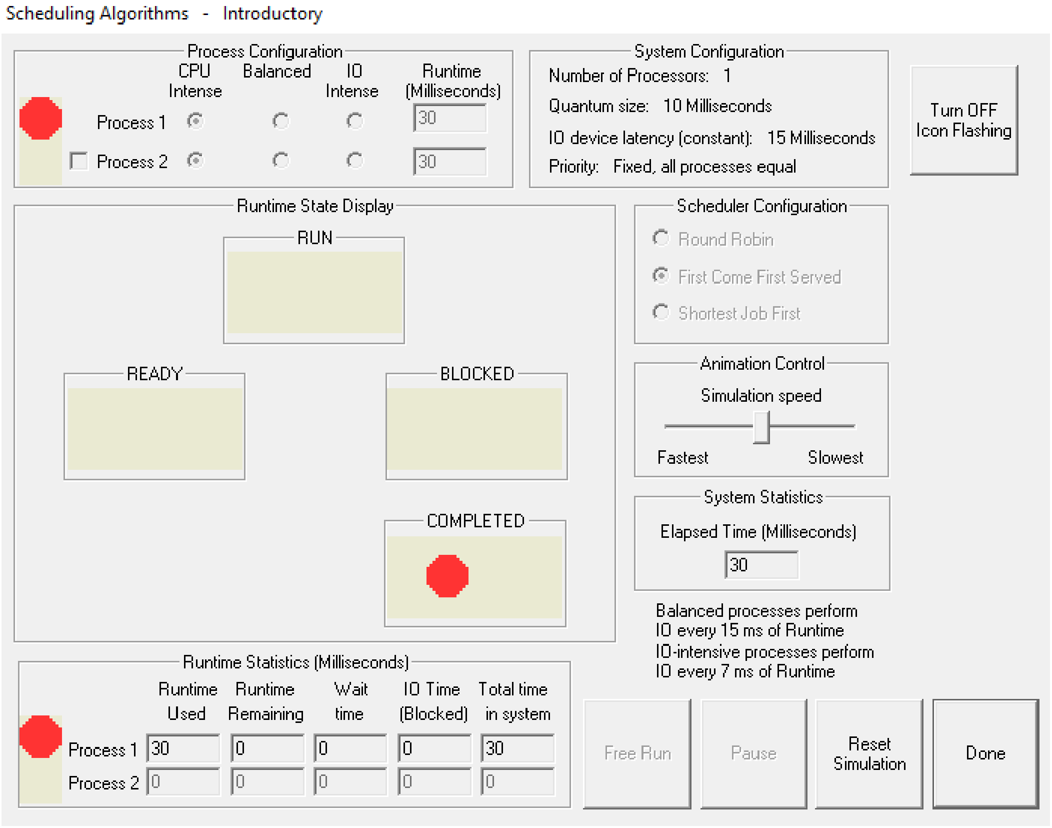
Balanced processes perform IO every 15 ms of Runtime

IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	300	0	30	0	330
Process 2	30	0	0	0	30

c. FCFS2



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	30
□ Process 2	●	○	○	30

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

○ Round Robin

● First Come First Served

○ Shortest Job First

Animation Control

Simulation speed

Fastest Slowest

System Statistics

Elapsed Time (Milliseconds) 30

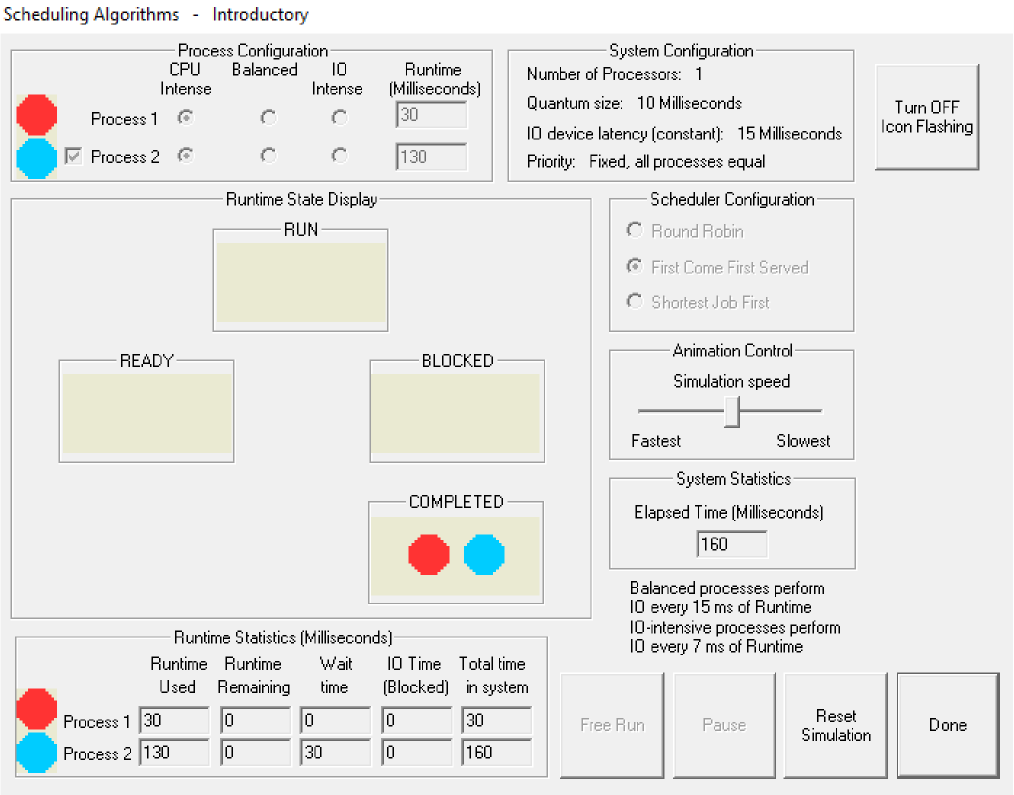
Balanced processes perform IO every 15 ms of Runtime

IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	30	0	0	0	30
Process 2	0	0	0	0	0

d. FCFS3



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
● Process 1	●	○	○	30
● ☑ Process 2	●	○	○	130

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

○ Round Robin

● First Come First Served

○ Shortest Job First

Animation Control

Simulation speed

Fastest Slowest

System Statistics

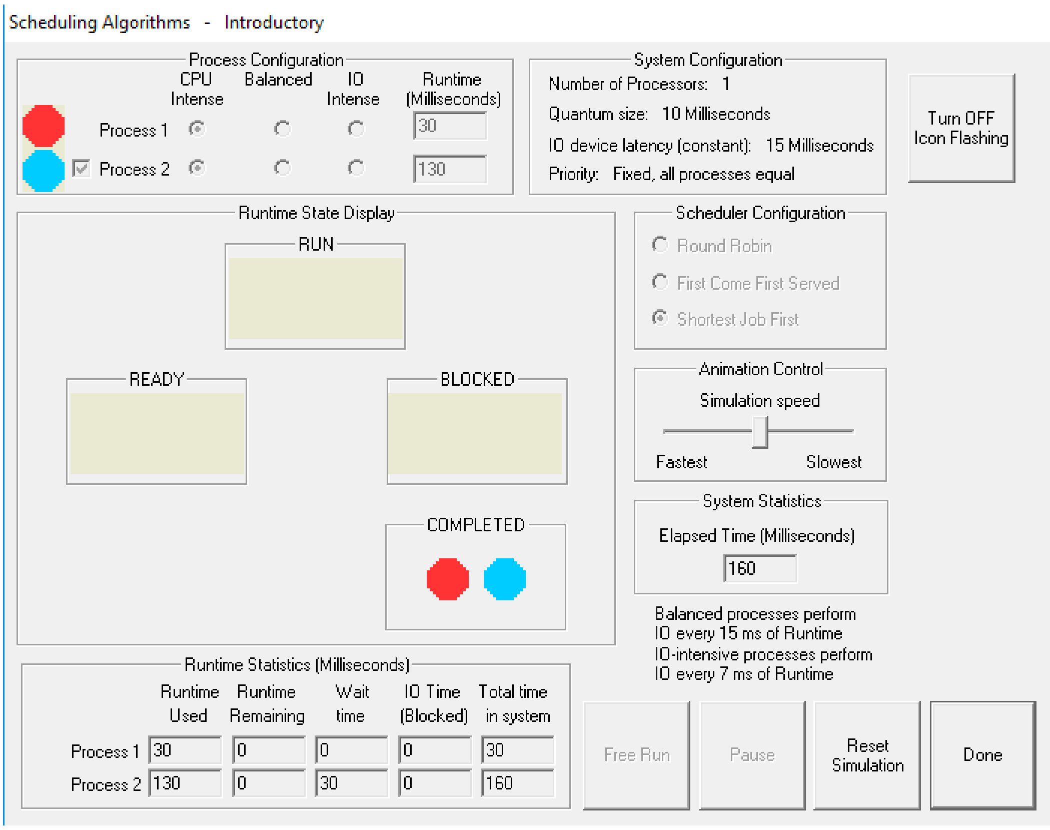
Elapsed Time (Milliseconds) 160

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
● Process 1	30	0	0	0	30
● Process 2	130	0	30	0	160

e. SJF2



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
● Process 1	●	○	○	30
● ☑ Process 2	●	○	○	130

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

○ Round Robin

○ First Come First Served

● Shortest Job First

Animation Control

Simulation speed

Fastest Slowest

System Statistics

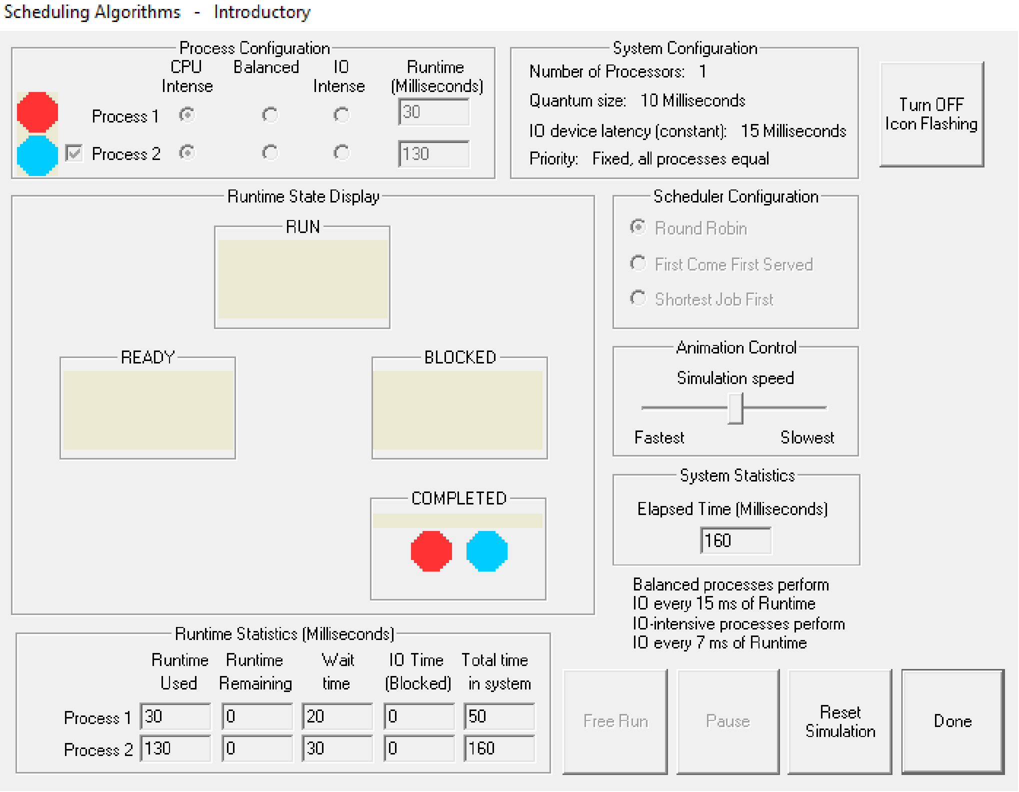
Elapsed Time (Milliseconds) 160

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	30	0	0	0	30
Process 2	130	0	30	0	160

f. RR1



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	30
☑ Process 2	●	○	○	130

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

● Round Robin

○ First Come First Served

○ Shortest Job First

System Statistics

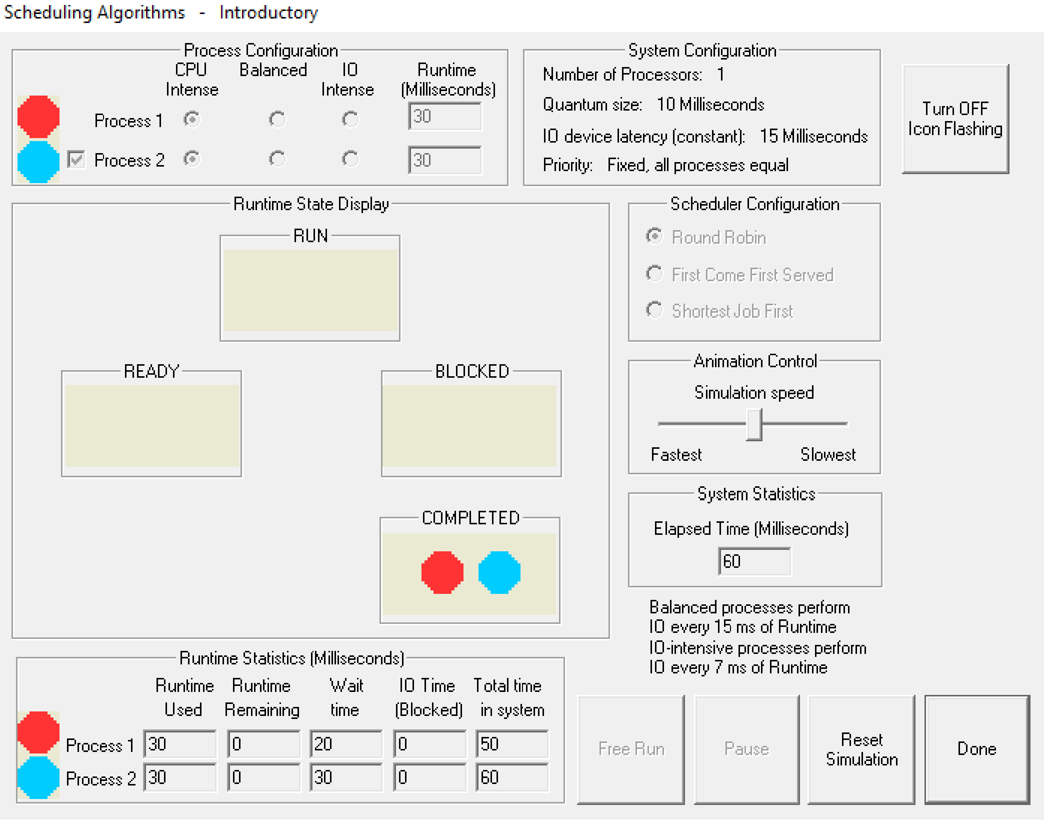
Elapsed Time (Milliseconds): 160

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	30	0	20	0	50
Process 2	130	0	30	0	160

g. RR2



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	30
☑ Process 2	●	○	○	30

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

● Round Robin

○ First Come First Served

○ Shortest Job First

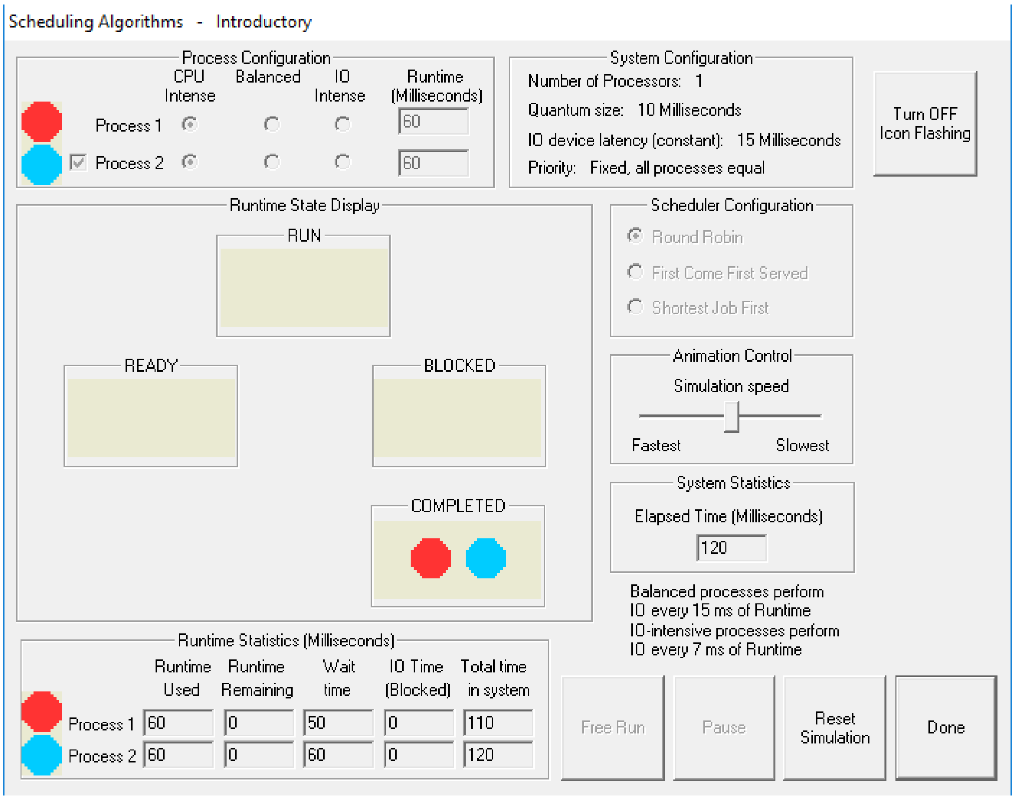
System Statistics

Elapsed Time (Milliseconds): 60

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	30	0	20	0	50
Process 2	30	0	30	0	60



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	60
☑ Process 2	●	○	○	60

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

● Round Robin

○ First Come First Served

○ Shortest Job First

System Statistics

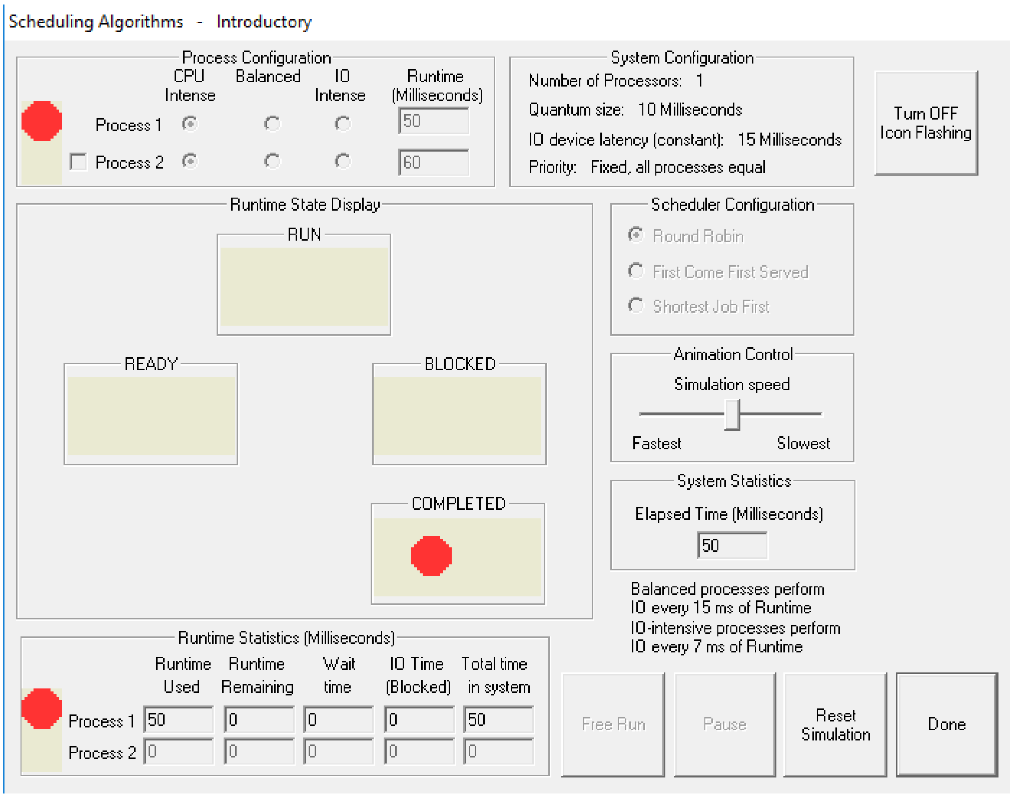
Elapsed Time (Milliseconds) 120

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	60	0	50	0	110
Process 2	60	0	60	0	120

h. RR3



Scheduling Algorithms - Introductory

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	50
□ Process 2	●	○	○	60

System Configuration

Number of Processors: 1

Quantum size: 10 Milliseconds

IO device latency (constant): 15 Milliseconds

Priority: Fixed, all processes equal

Scheduler Configuration

● Round Robin

○ First Come First Served

○ Shortest Job First

System Statistics

Elapsed Time (Milliseconds) 50

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	50	0	0	0	50
Process 2	0	0	0	0	0

TASKS:

For the following scenario with four processes please draw Gannt's charts and calculate *average waiting time* t_{AWT} and *average time a process remains in the system* t_{ATT} for the following scheduling algorithms:

SCENARIO:

Process	Arrival Time (When it arrives in the system) [ms]	Service Time (From arrival to finish) [ms]
P1	0	11
P2	2	8
P3	4	4
P4	6	6

TASK 5.1:

First Comes First Served:

The FCFS algorithm is shown below. The processes in the algorithm ran completely in order (P1, P2, P3, P4), with the total time in the system increasing as each process was carried out. The calculated average wait time for each process was 10.25 ms, and the average time a process remained in the system was 17.5 ms. The scenario table was unchanged, and the detailed t_{AWT} and t_{ATT} table, code, gantt chart and screenshots are shown below:

t_{AWT} and t_{ATT} table:

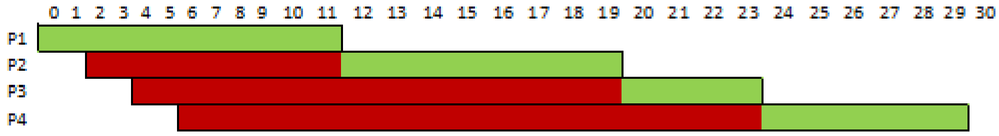
Process	Wait Time (t_{AWT}) [ms]	Average time a process remains in system (t_{ATT}) [ms]
P1	0	11
P2	9	17
P3	15	19
P4	17	23
Average:	10.25	17.5

Scriptcheck Code:

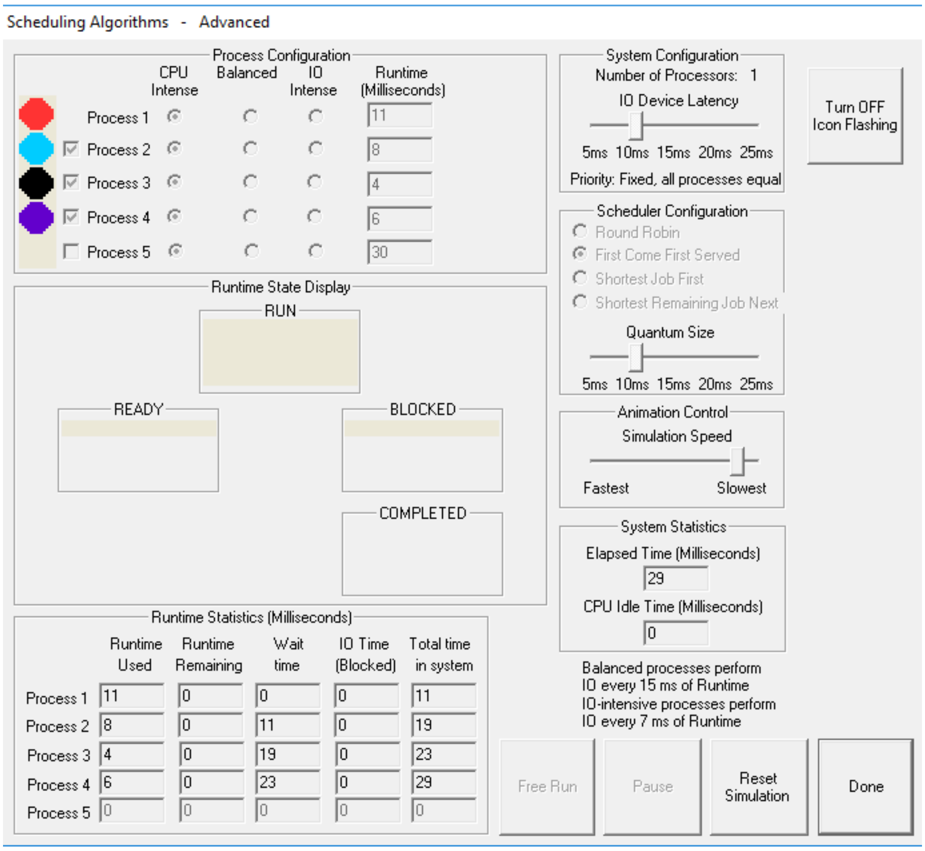
rrrrrrrrrr

--wwwwwwwwwrrrrrrrr
----wwwwwwwwwwwwwwwrrrr
------wwwwwwwwwwwwwwwwwrrrrrr

Gantt Chart: (x-axis: process, y-axis: time[ms]):



Workbench Screenshot:


Scheduling Algorithms - Advanced

Process Configuration

		CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
	Process 1	●	○	○	11
☑	Process 2	●	○	○	8
☑	Process 3	●	○	○	4
☑	Process 4	●	○	○	6
□	Process 5	●	○	○	30

System Configuration
Number of Processors: 1
IO Device Latency
5ms 10ms 15ms 20ms 25ms
Priority: Fixed, all processes equal

Scheduler Configuration
○ Round Robin
● First Come First Served
○ Shortest Job First
○ Shortest Remaining Job Next
Quantum Size
5ms 10ms 15ms 20ms 25ms

Animation Control
Simulation Speed
Fastest Slowest

System Statistics
Elapsed Time (Milliseconds)
29
CPU Idle Time (Milliseconds)
0

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime State Display
RUN
READY
BLOCKED
COMPLETED

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	11	0	0	0	11
Process 2	8	0	11	0	19
Process 3	4	0	19	0	23
Process 4	6	0	23	0	29
Process 5	0	0	0	0	0

TASK 5.2:

Shortest Job Next:

The SJN algorithm is shown below. The processes in the algorithm ran in a different order than FCFS (P1, P3, P4, P2), with the total time in the system fluctuating as each process was carried out. The calculated average wait time for each process was 8.75 ms, and the average time a process remained in the system was 16 ms. The updated scenario table, detailed t_{AWT} and t_{ATT} table, code, gantt chart and screenshots are shown below:

SCENARIO:

Process	Arrival Time (When it arrives in the system) [ms]	Service Time (From arrival to finish) [ms]
P1	0	11
P3	4	4
P4	6	6
P2	2	8

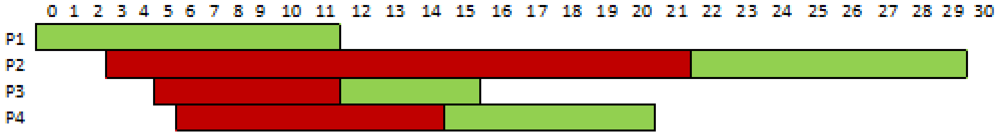
t_{AWT} and t_{ATT} table:

Process	Wait Time (t_{AWT}) [ms]	Average time a process remains in system (t_{ATT}) [ms]
P1	0	11
P2	9	15
P3	19	27
P4	7	11
Average:	8.75	16

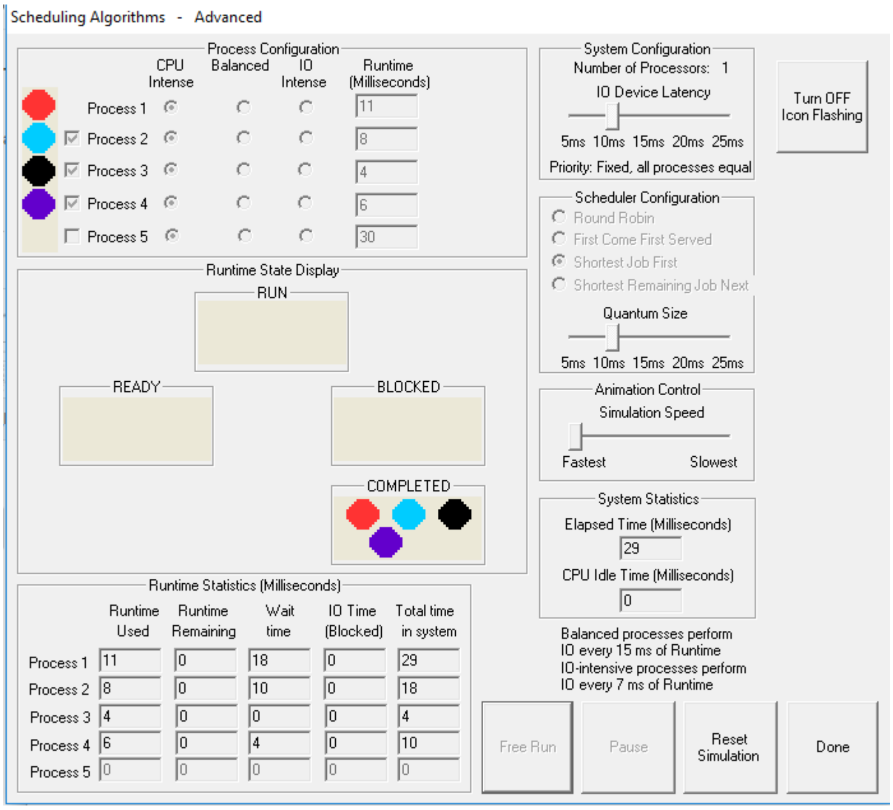
Scriptcheck Code:

```
rrrrrrrrrrr
----wwwwwwwrrrr
------wwwwwwwwwrrrrrr
--wwwwwwwwwwwwwwwwwwwrrrrrrrr
```

Gantt Chart: (x-axis: process, y-axis: time[ms]):



Workbench Screenshot:



Scheduling Algorithms - Advanced

Process Configuration

	CPU Intense	Balanced	IO Intense	Runtime (Milliseconds)
Process 1	●	○	○	11
☑ Process 2	●	○	○	8
☑ Process 3	●	○	○	4
☑ Process 4	●	○	○	6
□ Process 5	●	○	○	30

System Configuration
Number of Processors: 1
IO Device Latency
5ms 10ms 15ms 20ms 25ms
Priority: Fixed, all processes equal

Scheduler Configuration
○ Round Robin
○ First Come First Served
● Shortest Job First
○ Shortest Remaining Job Next
Quantum Size
5ms 10ms 15ms 20ms 25ms

Animation Control
Simulation Speed
Fastest Slowest

System Statistics
Elapsed Time (Milliseconds) 29
CPU Idle Time (Milliseconds) 0

Balanced processes perform IO every 15 ms of Runtime
IO-intensive processes perform IO every 7 ms of Runtime

Runtime State Display

Runtime Statistics (Milliseconds)

	Runtime Used	Runtime Remaining	Wait time	IO Time (Blocked)	Total time in system
Process 1	11	0	18	0	29
Process 2	8	0	10	0	18
Process 3	4	0	0	0	4
Process 4	6	0	4	0	10
Process 5	0	0	0	0	0

Personal Reflection:

This weeks' task was surprisingly easy compared to last weeks'. We were required to familiarize ourselves with the WorkBench program, one which I had never encountered before. However, after running through the pre-task exercises, I was able to confidently use the program. I found it so cool that I could pause and simulate multiple processes and algorithms, mimicking how they would actually run in real time situations. Unlike the previous weeks' we had no problem using scriptcheck to upload out answers, as the input system was fairly straight forward. The tasks required us to learn what we had used in the exercises and apply it to different scenarios, which was easy enough. We had also made gantt charts in previous modules so that was an easy task as well. Calculating the wait and average time in the system was a matter of simple math, and before we knew it, we were finished with week 5 tasks. I thoroughly enjoyed the task and hope the future ones are as easy to understand as this one. I do believe that I have furthered my understanding of operating systems and look forward to continuing so next week.