

CISC324 W2024

Instructor: Dr. Anwar Hossain

Group 15: Ethan Fighel, George Salib, Will Wu

Lab 2

**Compilation:** (run below code to compile the project, ensure you're in the project directory)

```
clang++ -std=c++11 main.cpp schedulingalgorithms/FirstComeFirstServedScheduler.cpp  
schedulingalgorithms/RoundRobinScheduler.cpp -o schedulerApp
```

(\*NOTE: if using windows, replace clang++ with g++)

After compilation, a schedulerApp.exe file will be made which shall be run to get results.

**Results:**

```
#####  
##### TEST CASE #1: First Come First Served Scheduler #####  
Average Wait Time: 7.8  
Average Turnaround Time: 14.2  
#####  
##### TEST CASE #1: Round Robin Scheduler + time quantum = 2 #####  
Average Wait Time: 21.4  
Average Turnaround Time: 22.4  
#####  
##### TEST CASE #2: First Come First Served Scheduler #####  
Average Wait Time: 24.3  
Average Turnaround Time: 30  
#####  
##### TEST CASE #2: Round Robin Scheduler + time quantum = 2 #####  
Average Wait Time: 39.2  
Average Turnaround Time: 40.2  
#####  
##### TEST CASE #3: First Come First Served Scheduler #####  
Average Wait Time: 7.8  
Average Turnaround Time: 14.2  
#####  
##### TEST CASE #3: Round Robin Scheduler + time quantum = 4 #####  
Average Wait Time: 23  
Average Turnaround Time: 24  
#####
```

##### TEST CASE #4: First Come First Served Scheduler #####

Average Wait Time: 24.3

Average Turnaround Time: 30

#####

##### TEST CASE #4: Round Robin Scheduler + time quantum = 4 #####

Average Wait Time: 41.6

Average Turnaround Time: 42.6

#####

### **Discussion:**

For different Round Robin (RR) time quantum, First Come First Serve (FCFS) outperformed RR on all counts. It seems the cost for context switches outweighed the cost of waiting times under FCFS. The downside of FCFS starving processes is not really seen so strongly when the number of processes is so small (5 in the first two tests, 10 in the last two).

### **Contributions:**

Ethan	33.33%
George	33.33%
Will	33.33%