

Project 2 Report

Omar Al Asaad 0075155

Ahmed Jareer 0074982

Versions and Dependencies:

- GCC: 12.2.0
- Argparse: 2.9

Compiling & Running:

- To compile the code:

```
cd src/  
make
```
- To Run the code

```
./main -p NORMAL_VOTER_PROB -t SIMUL_TIME -c STATION_COUNT \  
-n LOG_INIT_WAIT -f FAILURE_PROB
```

Project Separation:

We worked together on all the parts.

Implementation:

Functions:

- Main
 - Parsing logic
 - Initializes barrier which is used for syncing threads
 - Initializes threads
 - Joins threads
- Create_voters (Thread Func)
 - Creates the first 2 voters and waits at the barrier
 - Creates the rest of the voters
 - Each voter is a thread
 - A voter is created each second
 - Checks the least crowded station using the `get_least_crowded_station()`
- Voter_thread_func (Thread Func)
 - Waits until it gets a signal from the assigned station
 - Voters vote and sleep 2 seconds to simulate voting

- It calls log_voter_data to add the voter to the log file.
- Log (Thread Func)
 - Prints the log from the nth sec every second.
- Start_station (Thread Func)
 - Parses arguments and waits at the barrier
 - Implements the mechanic logic here
 - Checks for voters
 - Pops queues according to priority and given conditions
 - Starvation free
- Rest
 - Helpers for Thread Funcs

Classes:

- Station
 - has a map of two queues, Ordinary and Special (elderly/Pregnant).
 - Has Mutexes for multiples uses
 - Thread-safe
 - Has setters and getters
 - Has a constructor and a destructor where mutexes are initialized and then destroyed, setting the station number
- Voter
 - Holds the thread of the voter
 - Has Mutexes for multiples uses
 - Thread-safe
 - Has setters and getters
 - Has a constructor and a destructor where mutexes are initialized and then destroyed