

**CSE344 Systems
Programming
Midterm Project
Report**

**Tolga REİS
161044035**

There are some parts of the synchronization between processes. These are:

- Supplier and cook synchronization in the kitchen
- Cook and student synchronization in the counter
- Student and student synchronization in the table

First of all there are three types of processes and many samples of these processes except supplier. I started to solve problem with the easier one, supplier. Firstly, I created a structure to use as shared memory between the supplier and cook processes. This structure includes five counters and to provide synchronization includes six semaphores:

- `b_sem`, it can be imagined as a binary semaphore to control which one of the supplier and cook runs. To provide a critical section, this synchronization is vital.
- `full_sem` and `empty_sem`, there is a producer-consumer relation between the supplier and cooks, so `empty_sem` starts with the kitchen size and `full_sem` starts with the 0. When a cook does its job post `empty_sem` and when the supplier does its job post `full_sem`. To provide maximum number of plates according to size in kitchen, this synchronization is vital.
- `sem_P`, `sem_C`, `sem_D`; used for plates synchronization. When there is no such plates that is wanted by cook, it waits until the supplier post it. To provide always three different plates on the counter, this synchronization is vital.
- `total_plates`, is a counter and used for total plates number in the kitchen.
- `total_taken_plates`, is a counter and used for total taken plates from the kitchen by cooks.

As a second part for the synchronization between cooks-students, students-students; I created another structure that includes three semaphores and four counters:

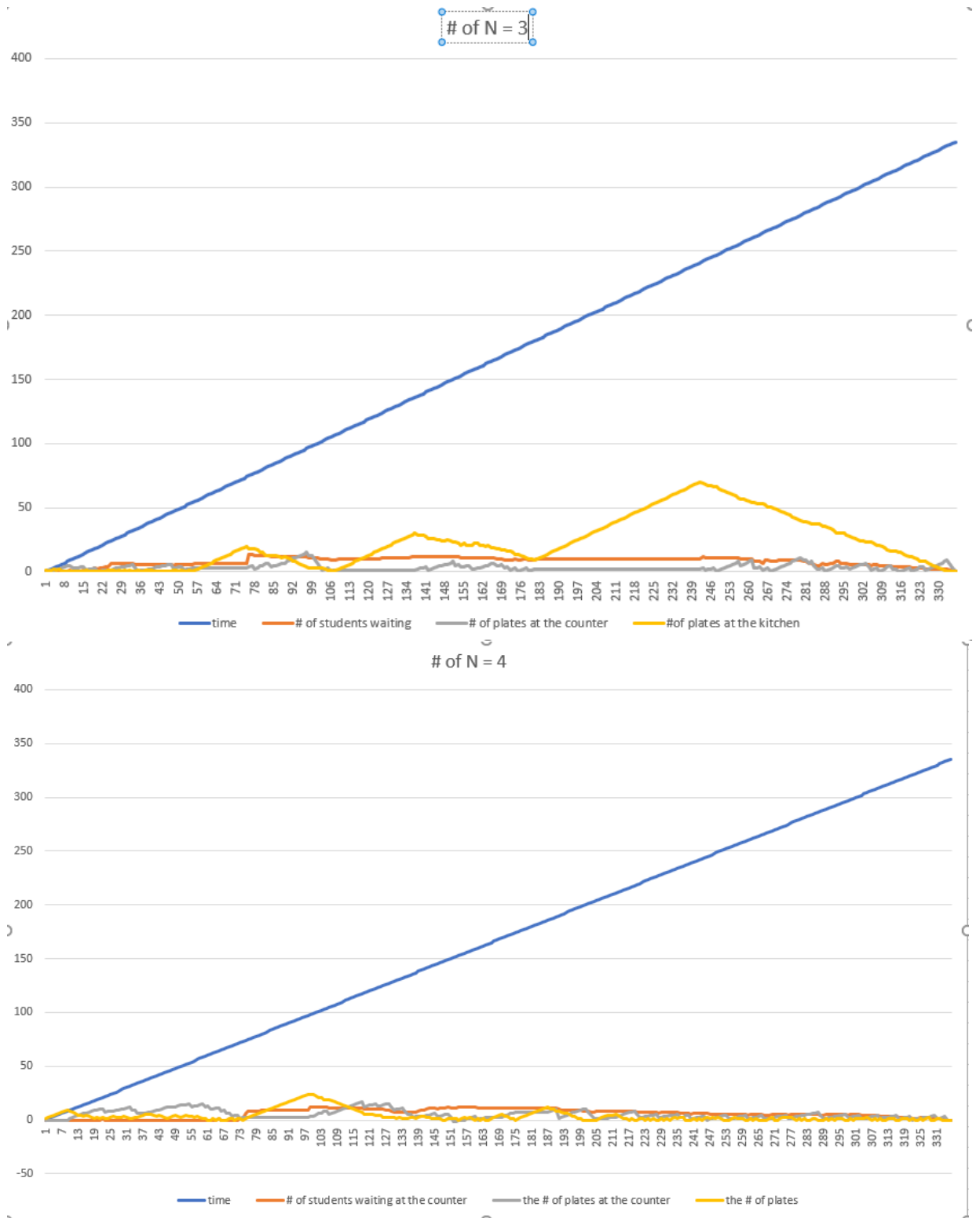
- `b_sem`, it can be also imagined as a binary semaphore to control which one of the cooks and students run. To provide critical section, adding something to and getting from counter; it is vital.
- `full_sem`, is used to provide to get three different plates by the student. When there are there different plates, cook posts it and student waits it.
- `table`, is also a semaphore to provide `T` students sit at the same time. Student process posts `table` when it sits, and waits it when lefts from the table.

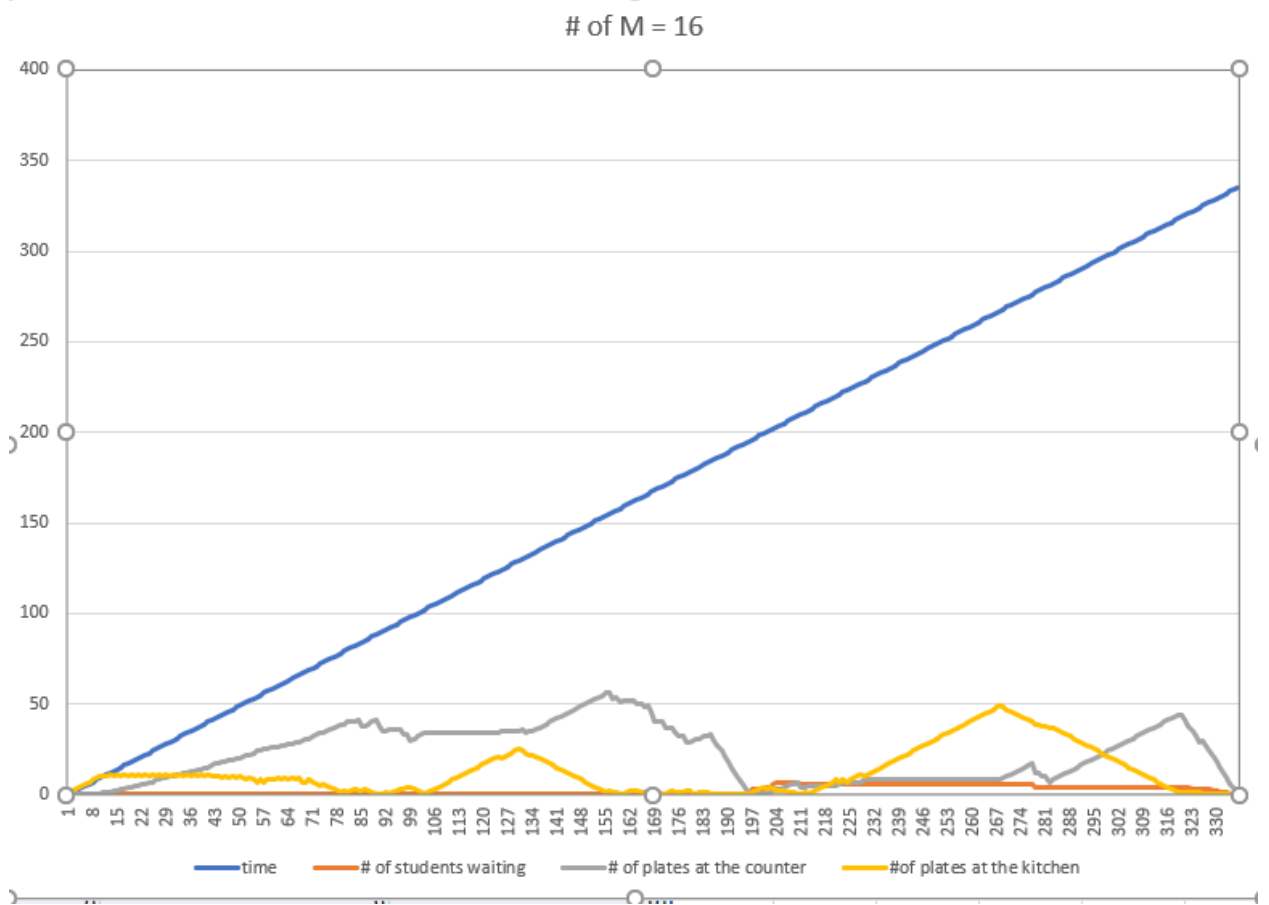
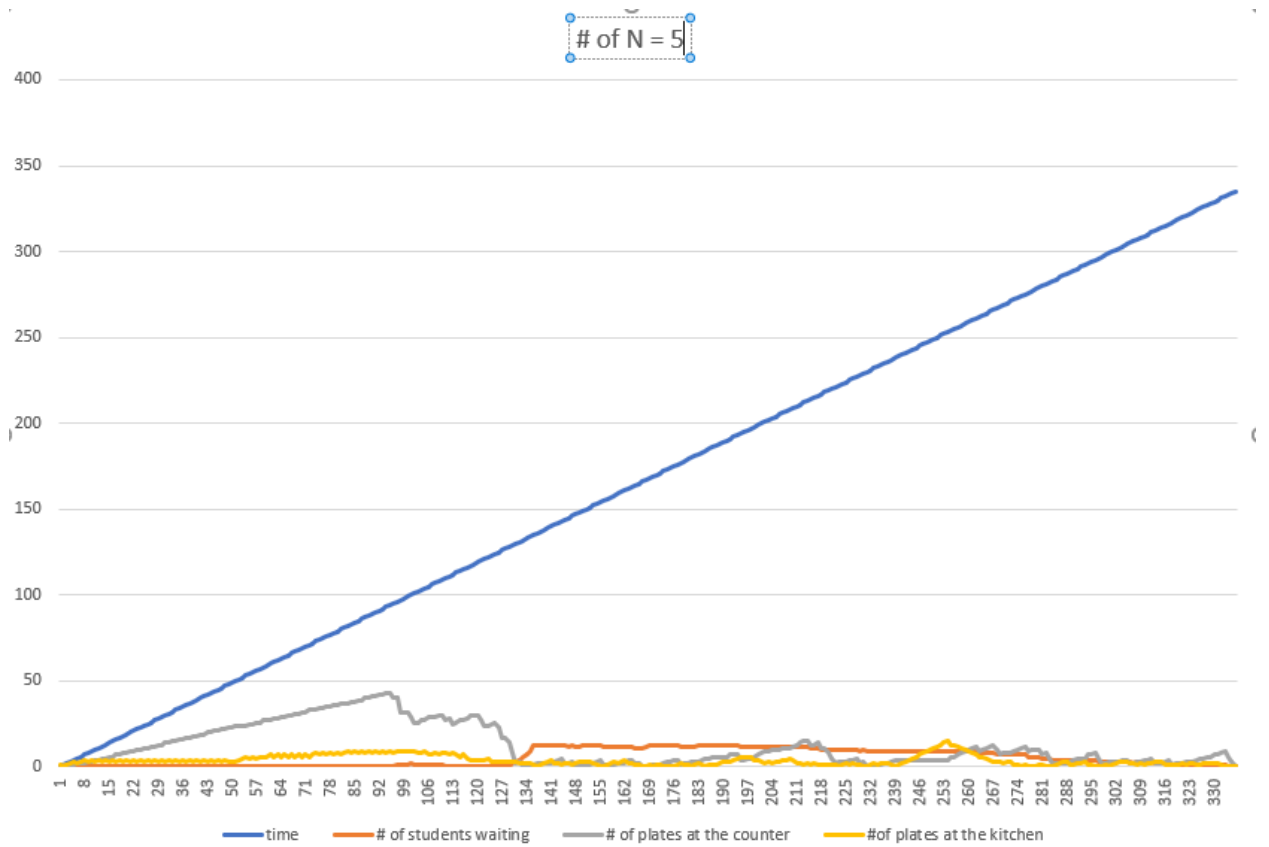
Plot cases:

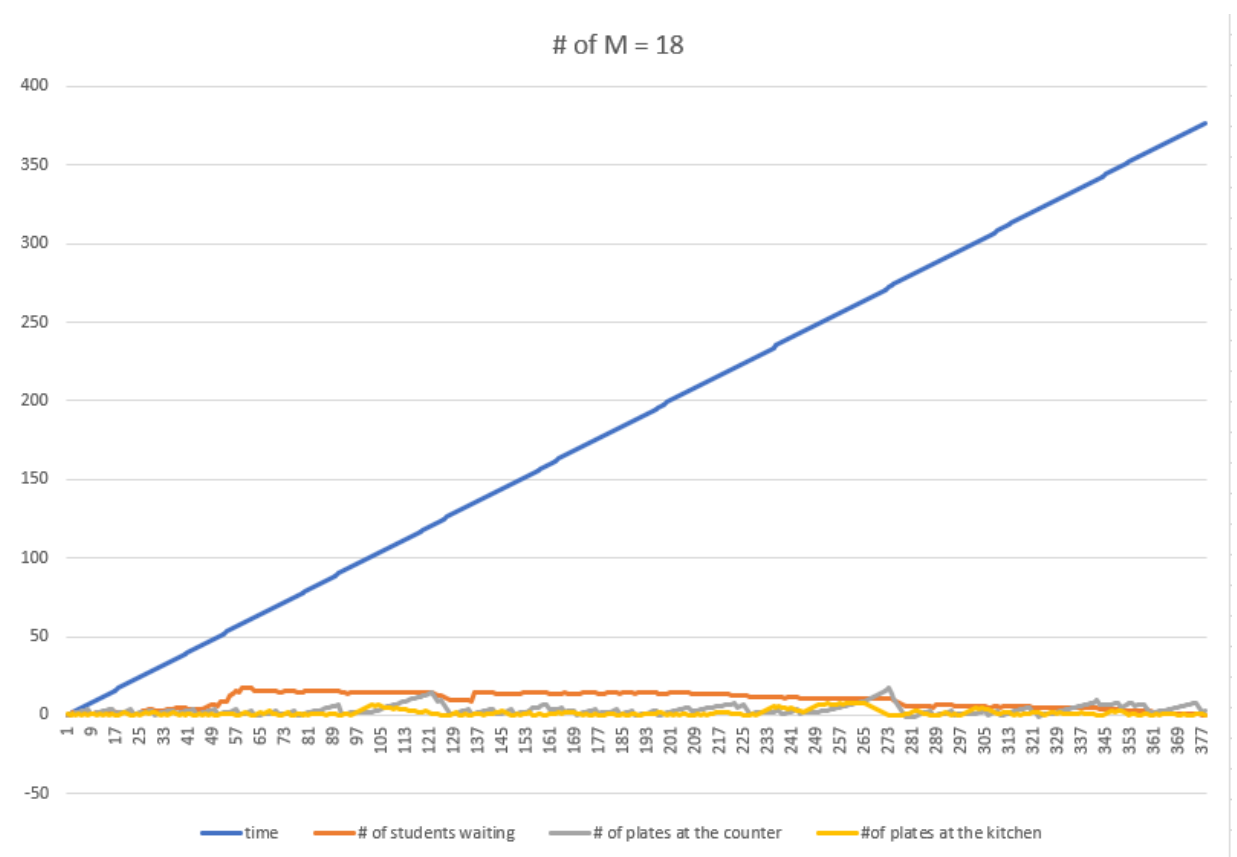
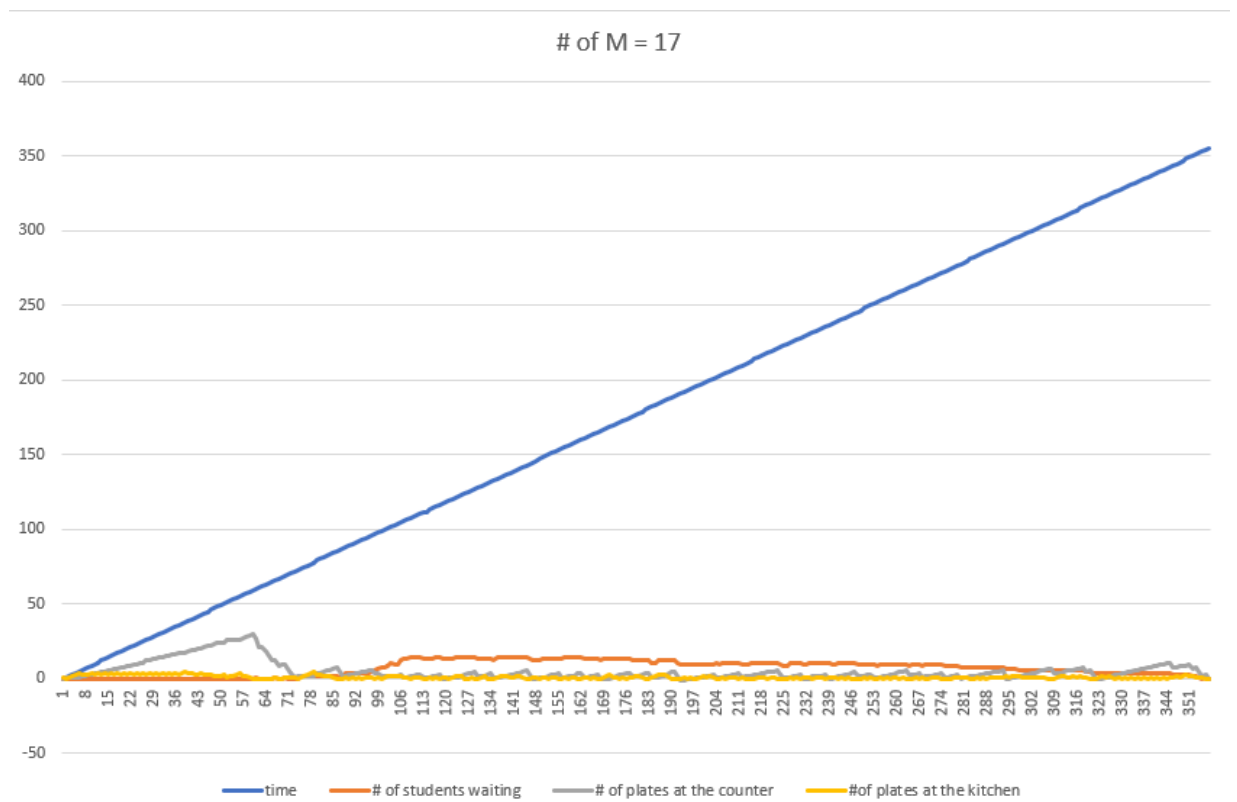
- `./program -N 3 -M 16 -T 5 -S 5 -L 3 -F fileName.txt`
- `./program -N 4 -M 16 -T 5 -S 5 -L 3 -F fileName.txt`
- `./program -N 5 -M 16 -T 5 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 16 -T 5 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 17 -T 5 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 18 -T 5 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 18 -T 6 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 18 -T 7 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 18 -T 8 -S 5 -L 3 -F fileName.txt`
- `./program -N 6 -M 18 -T 8 -S 6 -L 3 -F fileName.txt`

- ./program -N 6 -M 18 -T 8 -S 7 -L 3 -F fileName.txt
- ./program -N 6 -M 18 -T 8 -S 8 -L 3 -F fileName.txt
- ./program -N 4 -M 5 -T 3 -S 8 -L 6 -F fileName.txt
- ./program -N 4 -M 5 -T 3 -S 8 -L 7 -F fileName.txt
- ./program -N 4 -M 5 -T 3 -S 8 -L 8 -F fileName.txt

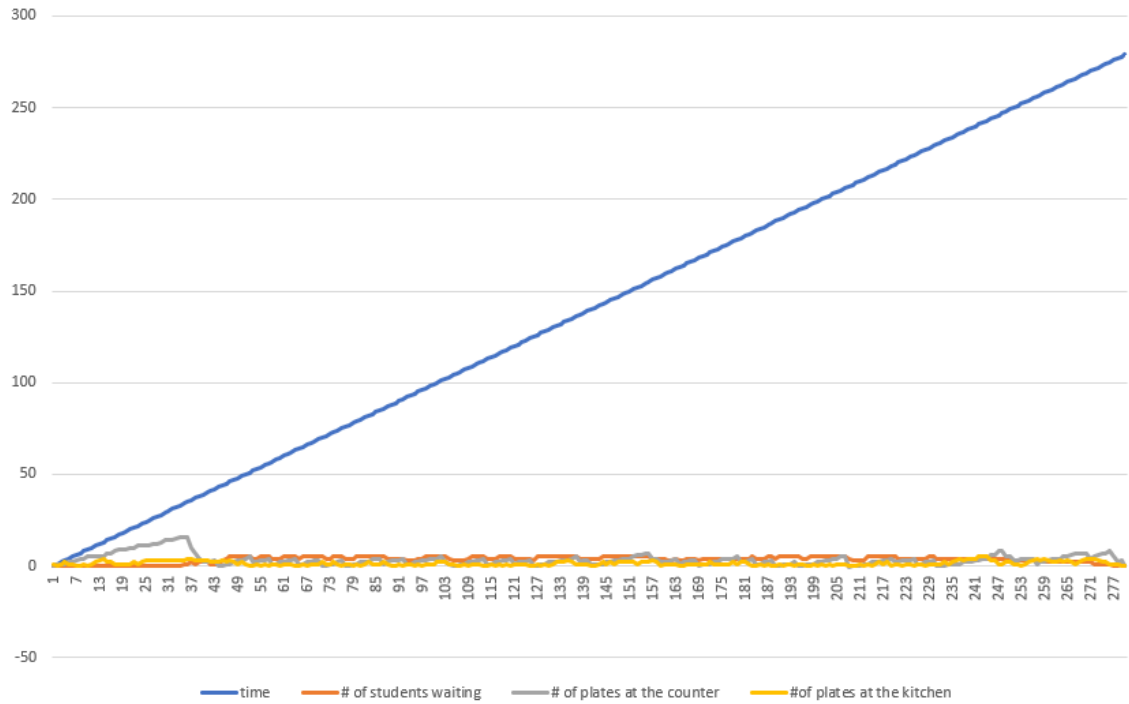
Plots



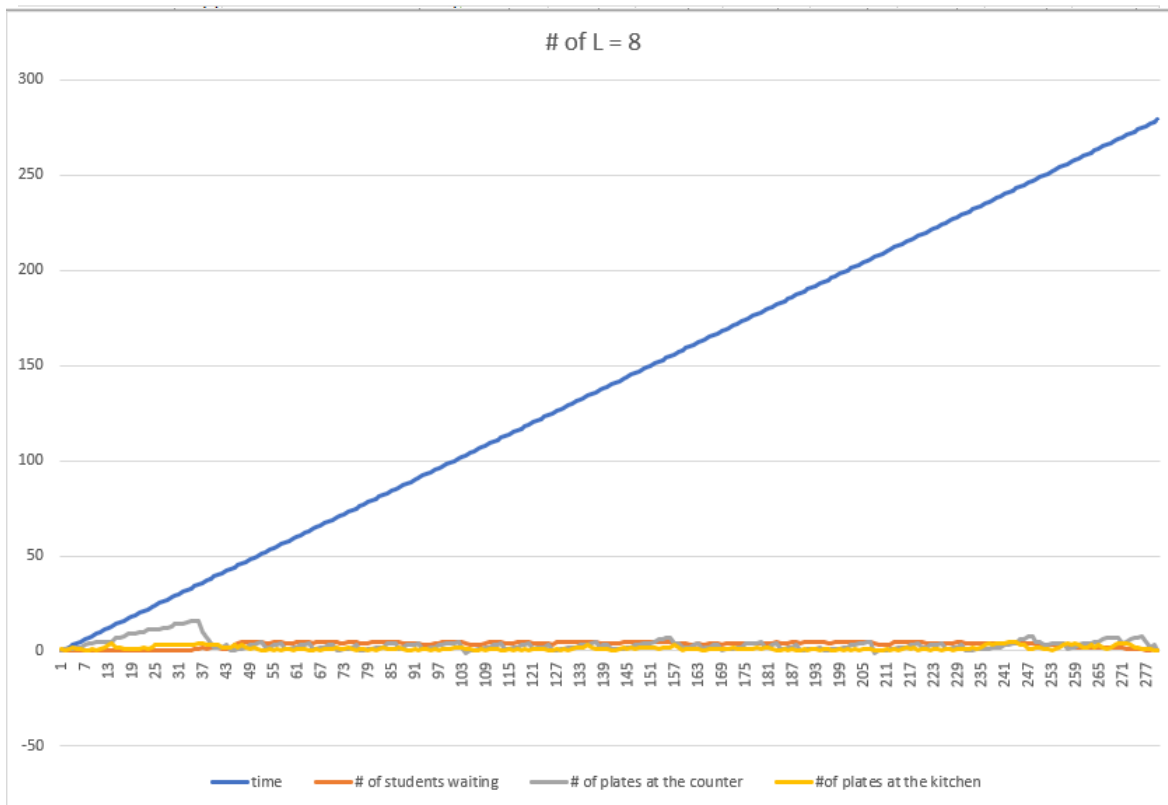


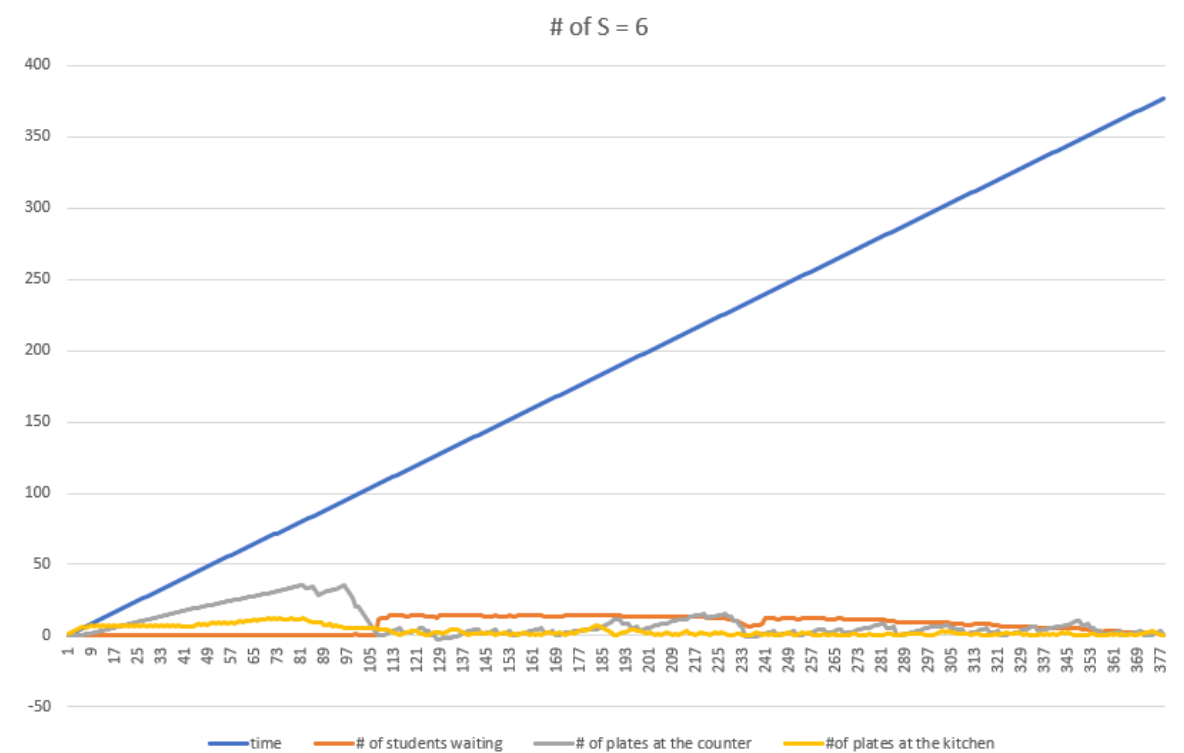
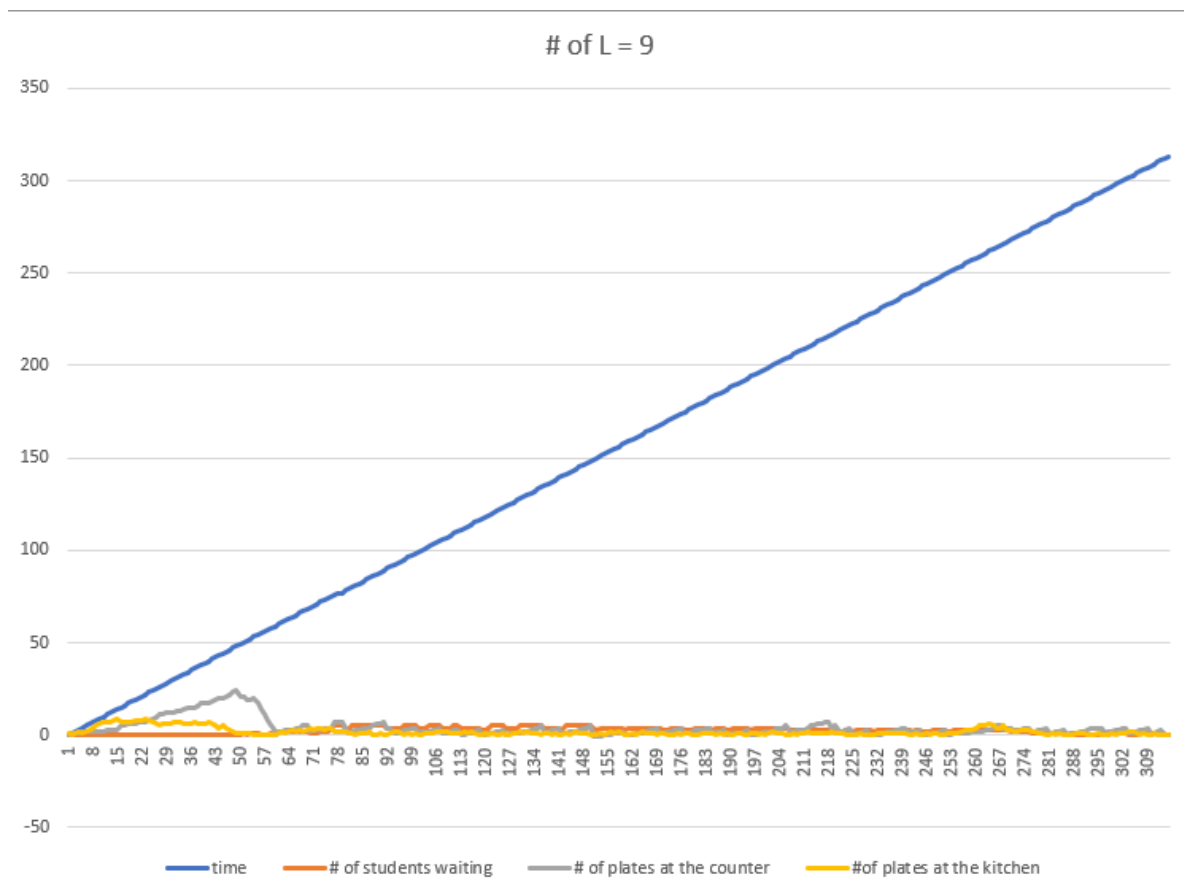


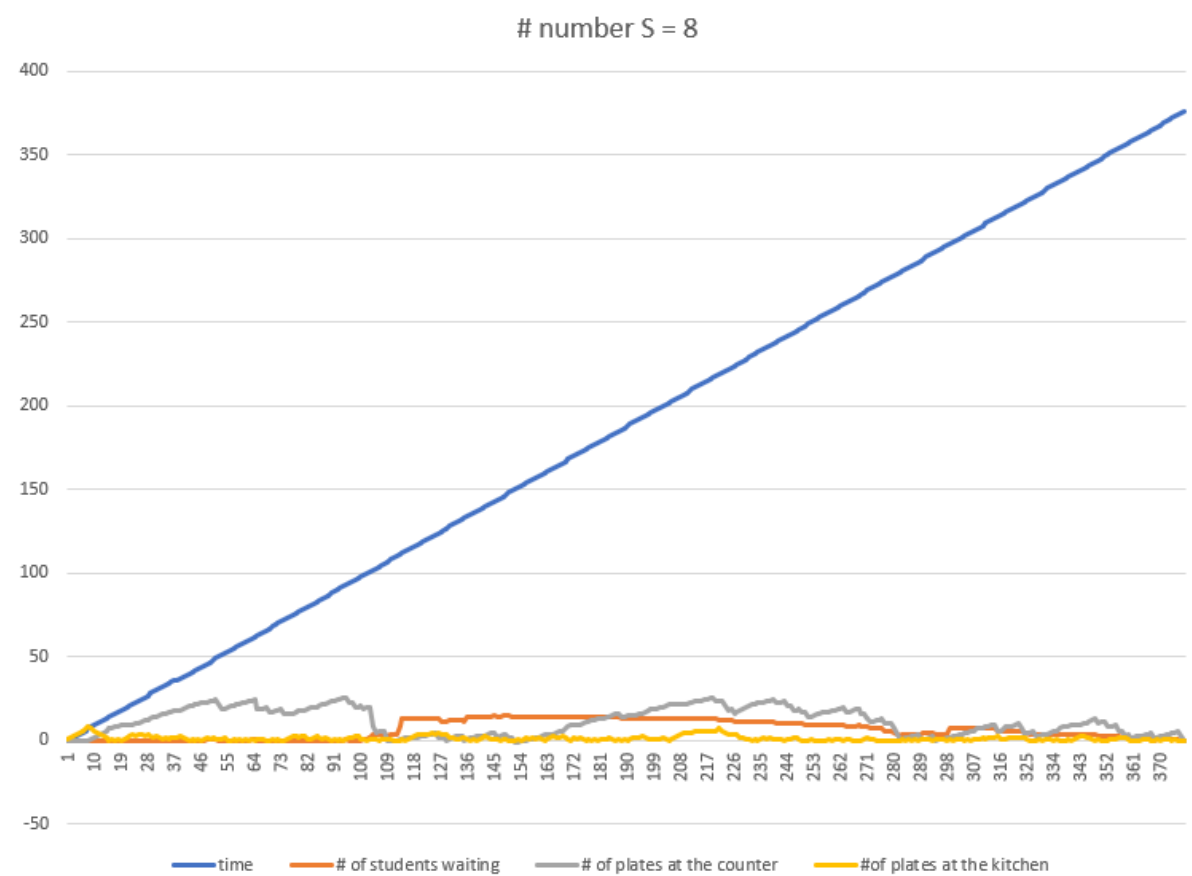
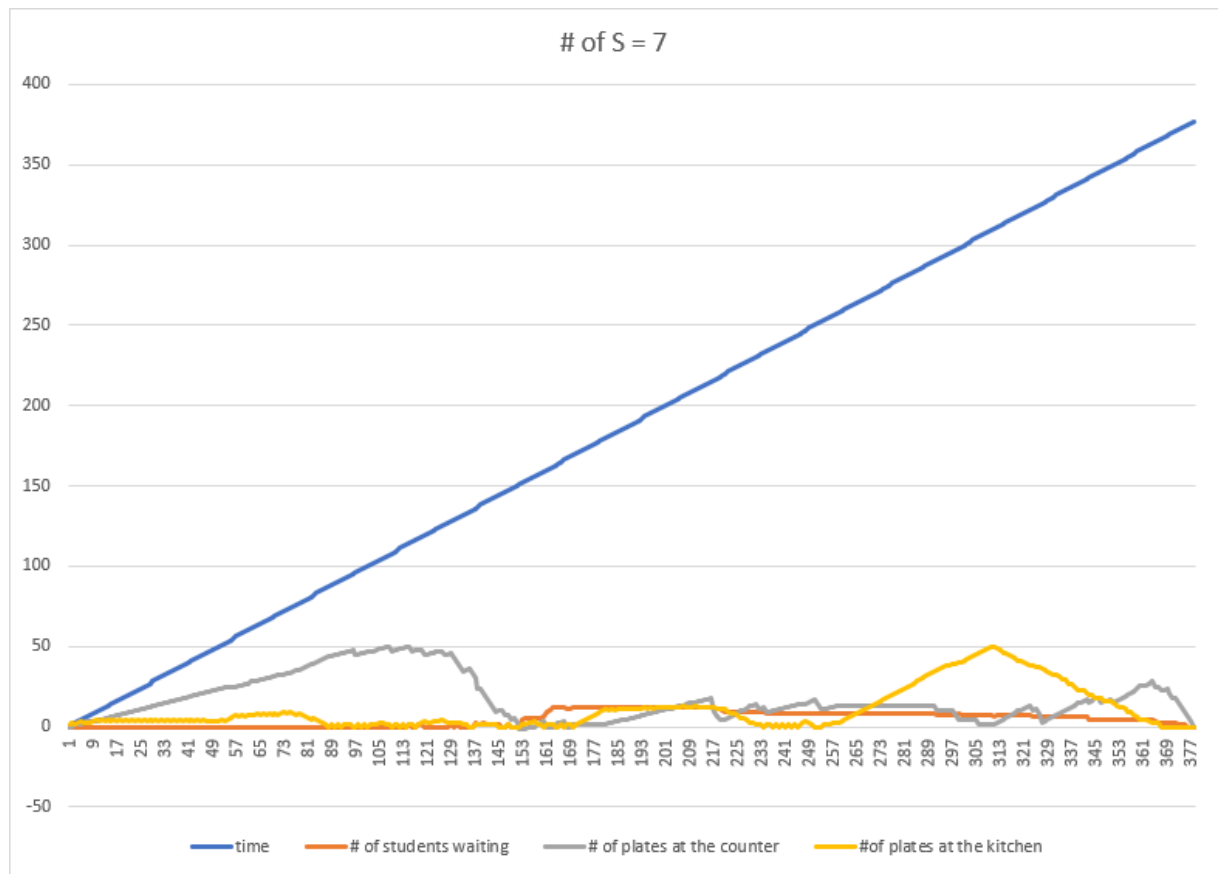
of L = 7



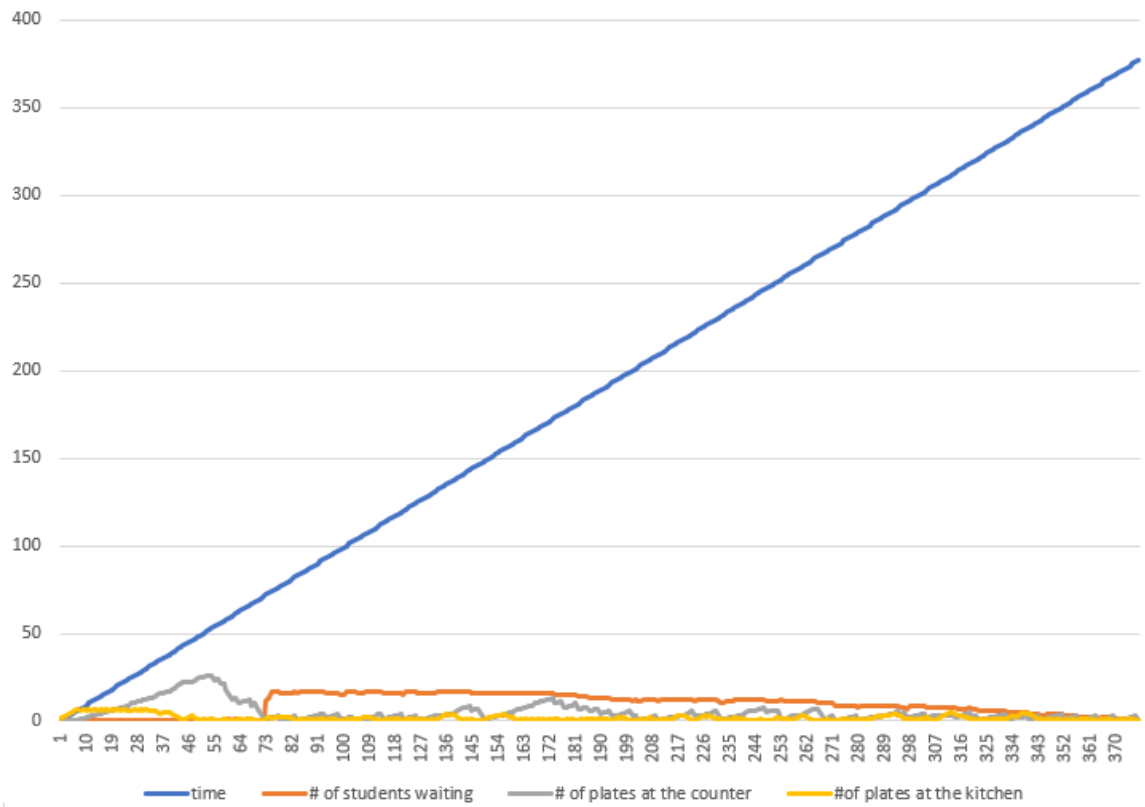
of L = 8







of T = 6



of T = 7

