SYSTEM PROGRAMMING – CSE 344 – **MIDTERM**

REPORT

YAĞIZ DÖNER – 141044062

In this hw, parent process reads command line and parse them. It makes the control of command-line. If there is an error in the command line, it terminates the program. Main topic of this hw is the usage of semaphores. So, the parent process creates unnamed semaphores and shared memories. And then, it creates its children via fork. Parent process waits its children termination. In this purpose, I write a barrier semaphore. Every child which terminate is increase this semaphore value via sem_post. And so, parent can wait with sem_wait for the number of its children. The first sync. problem solved like this.

Supplier process reads from the file and then puts a plate to the kitchen. For every plate, it is increase the number of plates at kitchen semaphore(atKitchen) and decrease the total kitchen size semaphore(kitchenSize). If the kitchenSize is zero, sem_wait waits until one of the cooks takes a plate and posts a kitchenSize. For every plate it puts, it is increase the value of this plates semaphore(kitP, kitC or kitD). It terminates at the end of the file and post to barrier.

Cook waits atKitchen semaphore. When the supplier puts a plate, cooks can work. Cooks choose the plates intelligently.

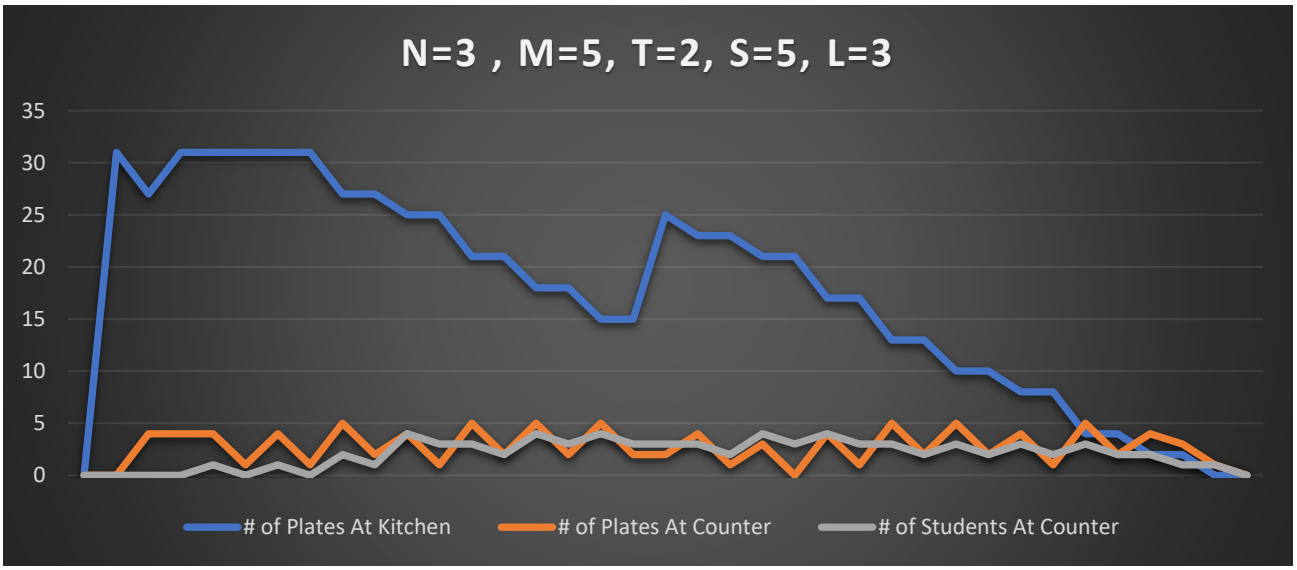
- If there is 2 more area for a plate at the Counter, they look other types of plates. If the others are equal to 0, cooks don't put this plate. Because, it causes a deadlock.
- If there is 1 more area for a plate at the Counter, they look other types of plates. If one of the other plates is equal to 0, cooks don't put this plate. Because, it causes a deadlock.
- Also, if everything is normal (it means that previous conditions is no problem and plate is in the kitchen), they choose plate which is less then the other.

For every plate they takes, they are decrease kitchen value of this plate(kitP,C or D) and increase counter value of this plate(couP,C or D). If there is a deadlock situation in the counter, cooks wait supplier to get more plates. When the supplier terminates and no more plates at kitchen, cooks can also terminate and increase the value of barrier.

Other actors are students. Students go to counter and wait the three types of plates. When the plates available, students are decrease their counter values(couP,C or D) and increase the counter size semaphore(atCounter). When they take the plates, it wait an empty table and eat their plates. After the eat, they go to the counter L times. For table sync, I use also a semaphore(tableSize). After L times, they terminate. they are increase the value of the barrier at the end.

General topics like this. There is also mainSem semaphore. This semaphore used in the sync. of the child processes. So, processes cannot diveded by the kernel. One of the processes is running, the other one cannot work. And also, all of the processes can catch SIGINT (Ctrl+C) signal. I only use unnamed semaphore and shared memory.

I create graph of the sample usage of my program. I use small input as much as possible for the graphs. Because, larger inputs have too many records and i cannot follow them.

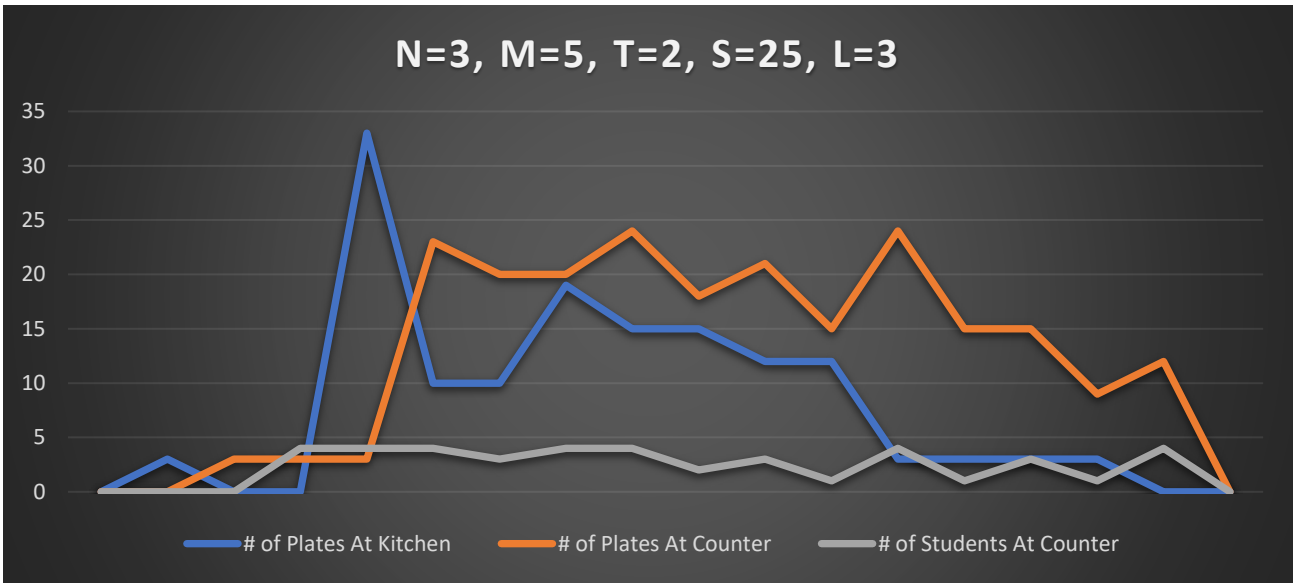


These two graphs show that smaller counter size (S) cause more valotile situations. When the counter size is bigger, operation goes more stable.

The effect of the table size(T) is not too much. So i don't create its graph.

M and L are effect the number of plates. It has only larger value graphs.

In here, the most important part of the values is S and i show that.



Yağız Döner - 141044062