Minjae Chae

mxc210045

CS 4348.001

Project 2 – Summary

My simulation goes like this. First, the patient comes in and counts the patient count. The patient let reception know that he is waiting for registration. Once reception notices a patient, he registers the patient and tells the nurse there is a patient in the waiting room and gives the patient information. Once the nurse is ready and gets a call that there is a patient in the waiting room, he gets patient information and once he gets signal that patient is sitting in waiting room and the office is ready, nurse directs patient to doctor office. Then the nurse lets the patient know which doctor's office he is going to, and lets the doctor know patient is waiting in the office and gives patient information. Once the doctor gets the notice, he gets patient information and once he gets the signal that patient is sitting in the office, he enters the office and listens to the symptoms of the patient and gives advice. Once the patient gets advice, he exits. Whenever a patient sits in a waiting room or office, they let the nurse or doctor know that they are ready to go.

As I encountered many difficulties and learned a lot from project 1, this project was very tough and helpful. The first difficulty was writing pseudocode. I referred to a lot from barbershop example from our lecture slide, and concepts of mutex and array of semaphore were hard to understand. Mutex was to communicate or deal with information one by one, and array of semaphore was to have every patient their own semaphore that we could make sure which patient is doing what. Next difficulty was figuring out which semaphores we need. For example, when I didn't have 'patient_office' semaphore, which means if patient is sitting in the office, message of doctor listening to the patient came out earlier than patient entering doctor's office. Another difficulty was communication between nurse and doctor. I used a single queue between nurse and doctor to give patient information, but then there was a problem that another doctor taking care of wrong patient. This problem happened because patient information goes in queue, which means whoever went into queue gets out of it first, that when another doctor tries to dequeue before the right doctor (the same id as nurse id) tries to dequeue the patient information. I figured out this problem by making array of queues as same amount of doctor. The last difficulty was that dequeuing when the queue is empty error message kept coming out. I didn't know why it happened and don't even understand why my work fixed it, but I fixed it. My array semaphores were all set to 'MAX_CAPACITY' which value is 15, but I should change some of them to 'MAX_DOCTOR' which value is 3. I fixed the dequeue problem by fixing initializing for loop of semaphores which were array of 3. These are difficulties that come up to me, but there were a lot more small difficulties especially regard to deadlock that I should move the order of semaphores a lot. Through these difficulties and getting those fixed, I could be able to think about how I should place semaphores and structure the code better than before that I feel very good about that.

My program now runs well, and logic looks straightforward. result matches the example from the document, and this program taught me a lot.