Jiashuai Lu
JXL173630
Department of Computer Science

March 30, 2018

Project2 of Operating Systems Concepts

Project Design

1. Which Semaphores are Needed

1. Nursecall
    - Nurse call patient
    - One for each doctor
    - Init by 0
2. Nursetell
    - Nurse tell doctor
    - One for each doctor
    - Init by 0
3. Patients
    - Patients waiting for register
    - One semaphore
    - Init by 0
4. Symptoms
    - Patients telling doctor their symptoms
    - One for each patient
    - Init by 0
5. Finishadvising
    - Doctor finishing a patient's advising
    - One for each patient
    - Init by 0
6. Waitingfordoctor
    - Denoting there is patient waiting for doctor
    - One for each doctor

- Init by 0

7. Finishregistering

    - Receptionist finishing a patient's register
    - One for each patient
    - Init by 0

8. Leaves

    - Denote a patient leave status
    - One for each patient
    - Init by 0

9. Mutexes

    - Guarantee some operation to be safe
    - two
    - Init by 1

2. Pseudocodes

```
/* Clinic Simulator */
semaphore nursecall[3] = {0};
semaphore nursetell[3] = {0};
semaphore patients = 0;
semaphore symptoms[30] = {0};
semaphore finish_advising[30] = {0};
semaphore finish_registering[30] = {0};
semaphore leaves[30] = {0};
semaphore waitingfordoctor[3] = {0};
semaphore mutex1 = 1, mutex2 = 1;
void receptionist()
{
    while(true)
    {
        wait(patients);
        dequeue_patients(patient_id);
        assign_doctor_to_patient(patient_id);
        signal(finish_registering[patient_id]);
    }
}
void doctor_and_nurse()
{
    while(true)
    {
        wait(waitingfordoctor[doctor_id]);
        dequeue_patients_of_doctor_id(doctor_id, patient_id);
        signal(nursecall[doctor_id]);
        wait(nursetell[doctor_id]);
        wait(symptom[patient_id]);
        advising();
        signal(finish_advising[patient_id]);
        wait(leaves[patient_id]);
    }
}
void patient()
{
    enter_clinic();
    wait(mutex1);
    enqueue_patients(patient_id);
    signal(mutex1);
    wait(finish_registering[patient_id]);
    signal(waitingfordoctor[doctor_id]);
    wait(nursecall[doctor_id]);
    signal(nursetell[doctor_id]);
    walking_to_doctor_office(doctor_id);
    signal(symptom[patient_id]);
    wait(finish_advising[patient_id]);
    leave();
    signal(leaves[patient_id]);
}
```