Amogh Yatnatti

Professor Ozbirn

CS 4348.001

10 April 2021

**Project 2 Design**

**Semaphores**:  
Semaphore mutexReception = new Semaphore(1);

* This semaphore is used to make sure that only one patient at a time (mutual exclusion) is able to add to the receptionBuffer, meet the receptionist, and get registered (assigned a doctor).

Semaphore receptionist = new Semaphore(0);

* This semaphore is used by the patient to signal the receptionist that the patient is ready. The receptionist waits for this semaphore before doing anything.

Semaphore register = new Semaphore(0);

* This semaphore is used by receptionist to signal the patient that they have been registered. This signal is done after the receptionist assigns the patient to a random doctor and adds the patient information to that doctor’s buffer. The patient waits for this semaphore and then gets their assigned doctor information. They then leave the receptionist.

Semaphore patientRdy = new Semaphore(0);

* This semaphore is used by the patient to signal that they have left the room. The receptionist is waiting for this semaphore so he/she/they can call the nurse to go see the patient.

**For the Semaphore arrays, the index corresponds to the nurse/doctor’s ID**:

Semaphore nurse[]; // Each semaphore is initialized to a value of 0

* This semaphore is used by the receptionist to signal the corresponding nurse of that their patient is ready.

Semaphore mutex[]; // Each semaphore is initialized to a value of 1

* This semaphore is used to ensure that each nurse and doctor is only dealing with one patient at a time. The semaphore waits after the first patient enters. It is only signaled after the doctor is finished with the patient and the patient has left. The doctor is able to dequeue (mutually exclusive)

Semaphore enterOff[]; // Each semaphore is initialized to a value of 0

* This semaphore is used by the patient to signal that they have entered the doctor’s office. The nurse waits for this semaphore and then notifies the doctor that the patient is ready.

Semaphore doctor[]; // Each semaphore is initialized to a value of 0

* This semaphore is used by the nurse to signify the patient is ready and the doctor can come in.

Semaphore advice[]; // Each semaphore is initialized to a value of 0

* This semaphore is used by the doctor to give advice to the patient. The doctor signals this semaphore and the patient will be waiting for it. After that, the patient will say that he received the advice.

Semaphore adviceRec[]; // Each semaphore is initialized to a value of 0

* This semaphore is used by the patient to signal the doctor that he has received the advice and will be leaving. The doctor will be waiting for this so nurse and him can move on to the next patient.

**Pseudocode**:

class Patient

{

int threadID;

run()

{

enter\_waiting\_room();

wait(mutexReception);

receptionBuffer.add(threadID);

signal(receptionist);

wait(register);

docID = get\_docID();

signal(patientRdy);

signal(mutexReception);

wait(enterOff[docID]);

enter\_doc\_office();

wait(advice[docID]);

receive\_advice();

leave();

signal(adviceRec[docID]);

class Receptionist

{

run()

{

while (true)

{

wait(receptionist);

int copy = receptionBufer.poll();

int docID = assignRandomDoc();

docBuffer[docID].add(copy);

assignedDoc[patID] = docID;

signal(register);

wait(patientRdy);

signal(nurse[docID]);

}

}  
}

class Nurse

{

int threadID;

run()

{

while (true)

{

wait(nurse[threadID]);

wait(mutex[threadID]);

int copy = docBuffer[threadID].peekFirst();

take\_patient\_to\_office();

signal(enterOff[threadID]);

signal(doctor[threadID]);

}

}  
}

class Doctor

{

int threadID;

run()

{

while (true)

{

wait(doctor[threadID]);

int copy = docBuffer[threadID].poll();

listen\_to\_patient();

signal(advice[threadID]);

wait(adviceRec[threadID]);

signal(mutex[threadID]);

}

}  
}