CS 4348.004 Operating Systems Concepts
Spring 2024
Design Document
Due: 4/6/24

Max Lichter

Semaphores List:

Semaphore	**Purpose**	**Initial Value**
receptionist	To control access to the receptionist by patients	0
reception	To ensure exclusive access to the reception buffer	1
register	To indicate when a patient is registered	0
patientRdy	To signal when a patient is ready to enter the doctor’s office	0
nurse[] (semaphore array)	To signal nurses when patients are ready to be taken to the doctor's office.	0 for each semaphore in the array
enterOff[] (semaphore array)	To signal when a patient enters a doctor's office.	0 for each semaphore in the array
doctor[] (semaphore array)	To signal doctors when patients are ready to be seen.	0 for each semaphore in the array
advice[] (semaphore array)	To signal when a doctor finishes giving advice to a patient.	0 for each semaphore in the array
adviceRec[] (semaphore array)	To signal when a patient receives advice from a doctor.	0 for each semaphore in the array
yesNo[] (semaphore array)	To ensure mutual exclusion when accessing doctor's buffers.	1 for each semaphore in the array

CS 4348.004 Operating Systems Concepts Max Lichter
Spring 2024
Design Document
Due: 4/6/24

Pseudocode:

```
class Receptionist {
	loop forever {
		wait(receptionist)	// Wait for a signal from recep. semaphore indicating a patient is ready
		patientID = getPatientFromBuffer()	// Get patient ID from reception buffer
		doctorID = assignPatientToDoctor()	// Assign patient to a doctor
		addPatientToDoctorBuffer(patientID, doctorID)	// Add patient to doctor's buffer
		assignDoctorInformation(patientID, doctorID)	// Assign doctor information to patient
		signal(register)	// Signal to register the patient
		wait(patientRdy)	// Wait for patient to be ready
		signalNurseThatPatientIsReady(doctorID)	// Signal nurse that patient is ready
	}
}
```

```
class Patient {
	run() {
		enterWaitingRoom()	// Enter the waiting room
		registerWithReceptionist()	// Register with the receptionist
		sitInWaitingRoom()	// Sit in the waiting room
		enterDoctorOffice()	// Enter the doctor's office
		receiveAdvice()	// Receive advice from the doctor
		leave()	// Leave the doctor's office
		signalPatientDone()	// Signal that the patient is done
	}
}
```

```
class Nurse {
	run():
		loop forever {
			wait(nurse[threadID])	// Wait for signal from nurse semaphore indicating patient needs attention
			handlePatient()	// Handle the patient
```

```
                takePatientToOffice()    // Take patient to doctor's office
                signalPatientEnteredOffice()    // Signal that patient has entered the office
                signalDoctorToCome()    // Signal the doctor to come and attend the patient
            }
        }
}
```

```
class Doctor{
    run():
        loop forever{
            wait(doctor[threadID])  // Wait for  signal from doctor semaphore indicating patient needs attention
            patientID = retrievePatientInformation()    // Retrieve patient information
            listenToSymptoms(patientID)    // Listen to symptoms reported by the patient
            giveAdviceToPatient()    // Give advice to patient
            wait(adviceRec[threadID])    // Wait for patient to leave
            releaseResources()    // Release resources
        }
    }
}
```
