**Design Document**

|  |  |  |
| --- | --- | --- |
| Semaphore | Purpose | Initial Value |
| capacitySem | Controls access to the waiting room, ensuring it doesn't exceed capacity. | 15 (program defines waiting room size) |
| printSem | Protects threads when printing to the console to prevent overlapping output. | 1 (ensures mutual exclusion for printing) |
| pwrSem | Signals the receptionist that a patient is waiting. | 0 (initially no patients waiting) |
|  |  |  |
| recepSem | Indicates if the receptionist is available to register patients. | 1 (receptionist starts available) |
| registerSem | Indicates if a patient has been registered with an assigned office. | 0 (no patients registered initially) |
| wrSem | Signals that a patient has entered the waiting room. | 0 (no patients in waiting room initially) |
| enterSem | Indicates when a patient enters their assigned doctor's office. | 0 (no patients in offices initially) |
| docSems[i]  (i = 0 to docNum-1) | Controls access to individual doctor's offices. | 1 (all doctors available initially) |
| nurseSems[i]  (i = 0 to docNum-1) | Indicates when a nurse is complete escorting patient | 0 (no patient has been escorted) |
| officeSems[i]  (i = 0 to docNum-1) | Controls access to individual doctor's offices | 0 (all offices initially unoccupied) |
| patientSems[i]  (i = 0 to docNum-1) | Indicates how many patients are waiting for a specific doctor's office (used by nurses). | 0 (no patients waiting for any office initially) |
| assesmentSems[i]  (i = 0 to docNum-1) | Indicates when a doctor's assessment of a patient is complete. | 0 (no assessments done initially) |

|  |  |
| --- | --- |
| /\* **program doctorsOffice** \*/ **semaphore capacity = 15;**  **semaphore printSem = 1;**  **semaphore recepSem = 1;**  **semaphore pwrSem = 0;**  **semaphore registerSem = 0;**  **semaphore wrSem = 0;**  **semaphore enterSem = 0;**  **semaphore docSems[i..docNum-1] = {1};**  **semaphore nurseSems[i..docNum-1] = {0};**  **semaphore officeSems[i..docNum-1] = {0};**  **semaphore patientSems[i..docNum-1] = {0};**  **semaphore assesmentSems[i..docNum-1] = {0};**  **queue<int> recepQueue;**  **queue<int> officeQueues[docNum];**  **int assignedOffice[patientNum];** | |
| void patient(int patientId){  // Register  sem\_wait(capacitySem);  sem\_wait(recepSem);  enqueue(recepQueue, patientId);  sem\_post(pwrSem);  sem\_wait(registerSem);  sem\_wait(print);  leaveReception();  sem\_post(print);  sem\_post(wrSem);  sem\_post(recepSem);    // Wait for nurse  sem\_wait(nurseSems[assignedOffice]);  sem\_wait(print);  enterOffice();  sem\_post(print)  sem\_post(enterSem);    // Wait for assessment  sem\_wait(assesmentSems[assignedOffice]);  sem\_wait(print);  recieveAdvice();  leaveOffice();  sem\_post(print);    // Exit  sem\_post(officeSems[assignedOffice]);  sem\_post(capacitySem);  } | void receptionist(){  int patientId;  int office;  while(true){  // Register patient  sem\_wait(pwrSem);  patientId =dequeue(recepQueue);  office = randomOffice();  assignedOffice[patientId] = office; enqueue(officeQueues[office], patientId);  sem\_post(registerSem);  // Signal nurse that a patient is waiting  sem\_wait(wrSem);  sem\_post(patientSems[ranOffice]);  }  } |
| void nurse(int nurseId){  int patient id;  while(true){  // Take patient to doctor's office  semWait(patientSems[nurseId]);  patientId = officeQueues[nurseId].front();  sem\_wait(print);  escortPatient();  sem\_post(print);  sem\_post(nurseSems[nurseId]);  sem\_wait(enterSem);  // Notify doctor  sem\_post(officeSems[nurseId]);  // Wait for doctor to complete  sem\_wait(docSems[nurseId]);  }  } | void doctor(int docId){  int patientId;  while(true){  // Assess patient  sem\_wait(officeSems[doctorId]);  int patientId = dequeue(officeQueues[doctorId]);    sem\_wait(print);  listenPatient()  sem\_post(print);  sem\_post(assesmentSems[doctorId]);  // Wait for patient to leave office  sem\_wait(officeSems[doctorId]);  // Signal doctor is available again  sem\_post(docSems[doctorId]);  }  } |
| **void** main**(){**  int docNum = getUserInput();  int patientNum = getUserInput();;  **parbegin** (receptionist, patient, . . . patientNum times, . . ., doctor, . . . docNum times, . . .,   nurse, . . . docNum times,. . .);  } | |