Description of the Project

The project creates a system level simulation which demonstrates hospital operating room scheduling through fundamental operating system concepts. The system aims to develop a multithreaded scheduling platform which handles incoming surgical cases by allocating them to operational rooms and executing them in real time. The system enables users to input surgical data which then undergoes standard OS scheduling algorithm processing using First Come First Served (FCFS) and Shortest Job First (SJF) and Priority Scheduling to establish the sequence of medical procedures.

The simulation system operates through time increments of one second which allows it to place new surgical procedures into the ready queue when they become available and then allocate them to operating rooms based on resource availability. The operating system management of multiple processes which compete for restricted resources gets modeled through semaphores and mutex locks and thread synchronization mechanisms. The project shows how OS scheduling concepts affect healthcare system performance through their application to a healthcare environment which demonstrates their impact on system efficiency and patient wait times and system operational performance.

Significance of the Project

The project holds its value because it connects academic operating system principles to the actual problems which healthcare facilities encounter today. The process of scheduling operating rooms needs to operate at maximum efficiency because it helps hospitals decrease patient delays and enhance medical procedures and optimize their available resources. The simulation shows how algorithmic scheduling affects operational efficiency and patient care outcomes through its representation of surgeries as processes and operating rooms as restricted system resources. The project demonstrates why synchronization and real time coordination and prioritization need to happen immediately in situations which require immediate action because delayed responses would lead to critical outcomes. The system introduces new concepts through its implementation of OS scheduling algorithms which analyze healthcare delivery systems to demonstrate how computational methods enhance medical service distribution quality.

Code Structure

```
ORS/

├── include/
```

```
│   ├── surgery.h        (defines Surgery structure)
│   ├── queue.h          (linked list ready queue interface)
│   ├── algorithms.h     (algorithm selection and prototypes)
│   ├── scheduler.h      (scheduler thread and global state)
│   ├── or_thread.h      (operating room thread interface)
│   └── utils.h          (input/output utility functions)
│
└── src/
    ├── main.c           (program entry, initialization)
    ├── scheduler.c      (scheduling logic, global time management)
    ├── or_thread.c      (operating room thread behavior)
    ├── queue.c          (mutex protected ready queue implementation)
    ├── algorithms.c     (FCFS, SJF, and Priority scheduling logic)
    └── utils.c          (interactive input and summary reporting)
```

Description of Algorithms

The project includes three fundamental non preemptive scheduling methods which operating systems use for their design: First Come, First Served (FCFS), Shortest Job First (SJF) and Priority Scheduling. The FCFS scheduling system performs surgeries based on their arrival sequence in the ready queue which provides easy operation but leads to performance issues when longer tasks prevent shorter tasks from running. The system operates with SJF because it chooses the surgery that needs the shortest duration to minimize both patient wait times and enhance operational efficiency. The Priority Scheduling system chooses the surgery which needs immediate attention based on its lowest numerical priority value to start with the most critical cases. The shared ready queue becomes accessible through mutex protected operations which allow the algorithms to choose their next surgery assignment when an operating room becomes available. The project shows how different scheduling policies affect execution order and wait times and resource utilization in a healthcare environment through its implementation of these algorithms in a real time threaded simulation.

Verification of Algorithms

The scheduling algorithms were tested through small controlled toy examples which demonstrated their expected operational behavior. The FCFS scheduler chose surgeries based on their arrival times instead of their execution duration because it processed them in the order they arrived. The toy set under SJF scheduling received durations {5, 2, 7} at once which confirmed that the surgery with duration 2 would always receive first priority because the dequeue operation properly checked the queue for the shortest job. The Priority Scheduling algorithm received validation through testing with priority sets including {priority 1, priority 3, priority 2} which produced the expected result of choosing priority 1 surgery despite its delayed arrival compared to other tasks with lower priorities. The scheduler produced consistent results under each policy according to the controlled inputs which also showed that mutex protected queue operations and semaphore based coordination did not affect the algorithmic behavior. The scheduling tests demonstrate that the scheduling system follows the operating system principles which were presented in class.

Functionalities

The system provides multiple fundamental features which work together to create an authentic and interactive simulation for operating room scheduling operations. Users can enter any number of surgeries through the system which requires them to provide surgery names and both start and end times and priority levels for clinical needs. The system enables users to choose between three scheduling options which include FCFS and SJF and Priority for determining the operating room assignment sequence of surgeries. The multithreaded simulation runs in real time during execution because it adds new surgeries to the ready queue when they become available while using the selected scheduling policy to select operating rooms. The operating room functions as an independent thread which performs its designated surgical procedure before it returns a completion indicator to the scheduler. The system continues to update all timing data during the simulation by recording both the beginning and end points of each process and the duration of each waiting period. The program produces a complete summary which shows all surgery arrival times and execution sequences and waiting durations and priority levels to help users understand the scheduler operation and algorithm performance.

Execution Results and Analysis

The execution results show that the scheduler properly distributed incoming surgeries to available operating rooms while following the selected execution sequence. The screenshots demonstrate that surgical procedures start at their scheduled times and operate for their designated periods before finishing without any issues which proves that the threads operate correctly. The scheduling logic and synchronization mechanisms function properly because start times and end times and wait times align with the predicted results. The results demonstrate that the system operated precisely and reliably throughout all experimental conditions.

```
Choose scheduling algorithm:
1. FCFS
2. SJF (default)
3. Priority Scheduling
Selection: 3
Enter surgeries (type 'done' as name to finish):
```

```
Surgery name: Sam
Arrival time: 1
Duration: 3
Priority (1 = highest): 2

Surgery name: Chris
Arrival time: 2
Duration: 1
Priority (1 = highest): 2

Surgery name: Ishaq
Arrival time: 4
Duration: 4
Priority (1 = highest): 2

Surgery name: Timmy
Arrival time: 2
Duration: 1
Priority (1 = highest): 2
```

```
Surgery name: done
[time 1] Surgery Sam arrived
[time 1] Assigned Sam to OR 1
[time 1] OR 1 started surgery Sam
[time 2] Surgery Chris arrived
[time 2] Surgery Timmy arrived
[time 2] Assigned Chris to OR 2
[time 2] Assigned Timmy to OR 3
[time 2] OR 2 started surgery Chris
[time 2] OR 3 started surgery Timmy
[time 4] OR 2 completed surgery Chris
[time 4] OR 3 completed surgery Timmy
[time 6] OR 1 completed surgery Sam
[time 4] Surgery Ishaq arrived
[time 4] Assigned Ishaq to OR 1
[time 4] OR 1 started surgery Ishaq
[time 11] OR 1 completed surgery Ishaq

All surgeries completed.
```

Conclusions

The project shows how scheduling principles from operating systems work in a healthcare environment which faces restrictions on available operating room space. The system achieves real time resource competition modeling between surgeries through its implementation of multithreading and synchronization primitives and classical scheduling algorithms. The results demonstrate both the advantages and disadvantages of various scheduling methods together with the need for semaphores and mutex locks to achieve secure system operation. The simulator operates with reliability but developers should focus on adding three features to the system which include preemptive scheduling and emergency case handling and improved visualization capabilities. The project demonstrates how course principles apply to healthcare resource management problems which occur in actual healthcare facilities.

References

https://www.geeksforgeeks.org/operating-systems/cpu-scheduling-in-operating-systems/

https://man7.org/linux/man-pages/man7/pthreads.7.html