



EAST WEST UNIVERSITY

Course Title : Operating System

Course Code : ICE275/CSE325

Section : 02

Group : 01

Project Title : Library Reservation System

Submitted by-

1. MD. Fahad Mia
ID: 2021-1-50-002
2. Tahrima Billal Biva
ID: 2020-3-60-049
3. Miftha Ahona Majid
ID: 2022-1-50-015

Submitted To-

Dr. Md. Nawab Yousuf Ali

Professor, Department of Computer Science and Engineering

Library Reservation System

Project Description

The library reservation system is a software application designed to streamline the process of reserving and managing library resources. It allows library patrons to easily browse, search, and reserve books, ebooks, journals, DVDs, and other materials available in the library's collection. Here are the key aspects of this project:

The primary goal is to create an efficient system that allows users to manage their library accounts, check book availability, and track borrowing limits. The system replaces traditional paper-based processes with an intuitive online interface.

Features:

User Authentication: Users can log in to their accounts securely.

Book Availability: Users can check whether a book is available for borrowing.

Reservation: Users can reserve study rooms, resources, or specific books.

Notifications: Users receive notifications about due dates, reservations, and other relevant information.

Admin Dashboard: Administrators can manage user accounts, book inventory, and reservations.

Benefits:

Efficiency: The system automates routine tasks, reducing manual effort for both users and library staff.

Accuracy: Real-time data ensures accurate information about book availability and reservations.

Convenience: Users can access the system from anywhere, anytime.

Resource Optimization: Efficient room/resource allocation prevents conflicts and maximizes utilization.

Overview of Library Reservation System

In a bustling city, the local library has become a popular destination for students and book enthusiasts alike. However, with the increasing demand for study rooms and resources, the library faces challenges in managing reservations efficiently. To address this, a multi-threaded C program with semaphores, processes, threads, and mutex locks can be implemented.

The problem to be addressed involves:

1. Coordinating the reservation of study rooms and resources such as computers and study materials to avoid conflicts and ensure fair access.
2. Managing the shared digital reservation system to update and maintain accurate records of reservations, including time slots and allocated resources.
3. Preventing double bookings and ensuring that no two users are assigned the same study room or resource simultaneously.

By implementing this system, the library can streamline the reservation process, providing a conducive environment for learning and study sessions.

Code :

```
#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

#define NUM_ROOMS 5

#define NUM_COMPUTERS 3

pthread_mutex_t room_mutex[NUM_ROOMS];

pthread_mutex_t computer_mutex[NUM_COMPUTERS];
```

```

sem_t room_sem;

sem_t computer_sem;

void* reserve_study_room(void* arg) {

    int id = (int)arg;

    int room_id;

    sem_wait(&room_sem);

    for (room_id = 0; room_id < NUM_ROOMS; room_id++) {

        if(pthread_mutex_trylock(&room_mutex[room_id]) == 0) {

            printf("User %d reserved study room %d.\n", id, room_id);

            sleep(rand() % 3 + 1);

            pthread_mutex_unlock(&room_mutex[room_id]);

            printf("User %d released study room %d.\n", id, room_id);

            break;

        }

    }

    sem_post(&room_sem);

    return NULL;

}

void* reserve_computer(void* arg) {

    int id = (int)arg;

    int computer_id;

    sem_wait(&computer_sem);

    for (computer_id = 0; computer_id < NUM_COMPUTERS; computer_id++) {

```

```

        if (pthread_mutex_trylock(&computer_mutex[computer_id]) == 0) {
            printf("User %d reserved computer %d.\n", id, computer_id);
            sleep(rand() % 3 + 1);
            pthread_mutex_unlock(&computer_mutex[computer_id]);
            printf("User %d released computer %d.\n", id, computer_id);
            break;
        }
    }
    sem_post(&computer_sem);
    return NULL;
}

int main() {
    pthread_t threads[10];
    int user_ids[10];
    for (int i = 0; i < NUM_ROOMS; i++) {
        pthread_mutex_init(&room_mutex[i], NULL);
    }
    for (int i = 0; i < NUM_COMPUTERS; i++) {
        pthread_mutex_init(&computer_mutex[i], NULL);
    }
    sem_init(&room_sem, 0, NUM_ROOMS);
    sem_init(&computer_sem, 0, NUM_COMPUTERS);
    for (int i = 0; i < 10; i++) {

```

```

    user_ids[i] = i + 1;

    if (rand() % 2) {

        pthread_create(&threads[i], NULL, reserve_study_room, &user_ids[i]);

    } else {

        pthread_create(&threads[i], NULL, reserve_computer, &user_ids[i]);

    }

}

for (int i = 0; i < 10; i++) {

    pthread_join(threads[i], NULL);

}

for (int i = 0; i < NUM_ROOMS; i++) {

    pthread_mutex_destroy(&room_mutex[i]);

}

for (int i = 0; i < NUM_COMPUTERS; i++) {

    pthread_mutex_destroy(&computer_mutex[i]);

}

sem_destroy(&room_sem);

sem_destroy(&computer_sem);

return 0;

}

```

GCC code compiler compilation :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SERIAL MONITOR
leopard@leopard-workstation-2:~/Downloads/Classroom/NYA_Sir_05/project-1$ sudo gcc ./Library_Reservation.c -o LibRes
```

Running compiled executable file :

```
leopard@leopard-workstation-2:~/Downloads/Classroom/NYA_Sir_05/project-1$ ./LibRes
```

Output of the compiled program :

```
leopard@leopard-workstation-2:~/Downloads/Classroom/NYA_Sir_05/project-1$ ./LibRes
User 4 reserved study room 1.
User 2 reserved computer 0.
User 3 reserved study room 0.
User 5 reserved study room 2.
User 6 reserved study room 4.
User 7 reserved computer 1.
User 8 reserved computer 2.
User 1 reserved study room 3.
User 2 released computer 0.
User 3 released study room 0.
```

```
User 3 reserved study room 0.
User 5 reserved study room 2.
User 6 reserved study room 4.
User 7 reserved computer 1.
User 8 reserved computer 2.
User 1 reserved study room 3.
User 2 released computer 0.
User 3 released study room 0.
User 10 reserved computer 0.
User 9 reserved study room 0.
User 1 released study room 3.
```

```
User 2 released computer 0.  
User 3 released study room 0.  
User 10 reserved computer 0.  
User 9 reserved study room 0.  
User 1 released study room 3.  
User 4 released study room 1.  
User 5 released study room 2.  
User 6 released study room 4.  
User 7 released computer 1.  
User 8 released computer 2.  
User 9 released study room 0.  
User 10 released computer 0.
```

Conclusion :

The Library Reservation System provides an efficient solution to manage study room reservations in a bustling city library. By implementing a multi-threaded C program with semaphores, processes, threads, and mutex locks, the system achieves the following objectives:

Coordinated Reservations: Students can reserve study rooms and resources without conflicts. The system ensures fair access and prevents double bookings.

Accurate Records: The shared digital reservation system maintains precise records of reservations, including time slots and allocated resources.

Streamlined Process: Through effective coordination and synchronization, the library can streamline the reservation process, creating a conducive environment for learning and study sessions.

In summary, the Library Reservation System optimizes resource utilization, enhances user experience, and contributes to a well-organized library environment.