

# **OS MINI PROJECT DOCUMENT**

*-Satyam Dewangan IMT2023545*

## **Implementation:**

- There are two c files-client.c and dbserver.c
- Client.c works as a client part of the project whereas dbserver.c acts as a server for the project that processes all the requests from the client.
- Client.c has 3 functions defined based on the role- adminclient() for admin, facultyClient() for the faculty and studentClient() for the student.
- Dbserver.c has one authentication() function for authorised access to the server. Also handle\_client() function is there for which serves other requests.
- For the database, I have defined 4 files - students.txt, courses.txt, faculty.txt and enrollment.txt.
- These are the common files for all the clients, therefore we have implemented locking mechanisms here.
- We do fork() everytime a new client enters so that that process runs independently without the interferences from other clients.

## **Basic Explanation:**

This code implements a client-server system for a course registration platform with three types of users: Admin, Faculty, and Student. Here's a brief overview of how it works:

### **Client-Side (client.c):**

1. Initialization: The client connects to the server on port 8080.
2. Authentication: The user selects their role (Admin, Faculty, or Student), enters their ID and password, and sends this information to the server for authentication.
3. Role-Based Menus:
  - Admin: Can add students/faculty, activate/deactivate students, and update details.
  - Faculty: Can add/remove courses, view enrollments, and change passwords.
  - Student: Can enroll/unenroll in courses, view enrolled courses, and change passwords.
4. Communication: The client sends commands to the server and receives responses, displaying them to the user.

### **Server-Side (dbserver.c)**

1. Initialization: The server starts listening on port 8080.
2. Authentication: Validates user credentials against stored data (admin has hardcoded credentials; others are checked from `students.txt` or `faculty.txt`).

3. Command Handling: Processes commands from the client:
  - Admin Commands: Add/update students/faculty, activate/deactivate students.
  - Faculty Commands: Add/remove courses, view enrollments, change passwords.
  - Student Commands: Enroll/unenroll in courses, view enrolled courses, change passwords.
4. File Operations: Uses system calls (`open`, `read`, `write`, etc.) to manage data files (`students.txt`, `faculty.txt`, `courses.txt`, `enrollment.txt`). File locking (`fcntl`) ensures data consistency during concurrent access.
5. Concurrency: Forks a new process for each client to handle multiple connections simultaneously.

### Key Features

- Text-Based Data Storage: Uses plain text files for persistence.
- Role-Based Access Control: Different menus and functionalities for each user type.
- Error Handling: Validates inputs and provides feedback for failures.
- File Locking: Prevents race conditions during file updates.

### Workflow

1. Client connects to the server and authenticates.
2. Server validates credentials and grants access.
3. Client sends role-specific commands.
4. Server processes commands, updates files, and sends responses.
5. Client displays results to the user.

This system efficiently manages course registrations and user interactions while ensuring data integrity through file locking and role-based access control.

## Important Screenshots:

1) Starting tab:

```
satyam@satyam-HP-Pavilion-x360-Convertible-14-dy1xxx:~/Documents/OS_lab/OS_project$ ./client
.....Welcome Back to Academia :: Course Registration.....
Login Type
Enter your choice{1.Admin , 2.Professor , 3.Student}
█
```

2)Professor's login(similar with other):

```
satyam@satyam-HP-Pavilion-x360-Convertible-14-dy1xxx:~/Documents/OS_lab/OS_project$ ./client
.....Welcome Back to Academia :: Course Registration.....
Login Type
Enter your choice{1.Admin , 2.Professor , 3.Student}
2
Enter ID: 1
Enter Password: thang
Login successful!
.....Welcome to Faculty menu.....
Do you want to:
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. Password Change
5. Exit
Enter your choice:
█
```

3)Adding a new student(same with adding any faculty):

```
satyam@satyam-HP-Pavilion-x360-Convertible-14-dy1xxx:~/Documents/OS_lab/OS_project$ ./client
.....Welcome Back to Academia :: Course Registration.....
Login Type
Enter your choice{1.Admin , 2.Professor , 3.Student}
1
Enter ID: 1234
Enter Password: 7565
Login successful!
.....Welcome to Admin menu.....
Do you want to:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 1
Enter roll number of student: 1
Enter name of student: Satyam
Enter password: satyam
Server response: SUCCESS: Student added
```

4)Activating a student(same with updating student's or faculty's name):

```
Do you want to:
1. Add Student
2. Add Faculty
3. Activate/Deactivate Student
4. Update Student/Faculty details
5. Exit
Enter your choice: 3
Enter Student id: 1
Enter 0 if you want to activate and 1 for deactivate: 0
Server response: SUCCESS: Student 1 activated
```

#### 5)Adding a new course:

```
.....Welcome Back to Academia :: Course Registration.....
Login Type
Enter your choice{1.Admin , 2.Professor , 3.Student}
2
Enter ID: 1
Enter Password: thang
Login successful!
.....Welcome to Faculty menu.....
Do you want to:
1. Add new Course
2. Remove offered Course
3. View enrollments in Courses
4. Password Change
5. Exit
Enter your choice:
1
Enter courseId: 1
Enter course name: Operating_system
Enter number of credits: 4
Server response: SUCCESS: Course added
```

#### 6)Changing password of a student(similar with faculty)

```
satyam@satyam-HP-Pavilion-x360-Convertible-14-dy1xxx:~/Documents/OS_lab/OS_project$ ./client
.....Welcome Back to Academia :: Course Registration.....
Login Type
Enter your choice{1.Admin , 2.Professor , 3.Student}
3
Enter ID: 1
Enter Password: satyam
Login successful!
.....Welcome to Student menu.....
Do you want to:
1. Enroll new Course
2. Unenroll from already enrolled Courses
3. View enrolled Courses
4. Password Change
5. Exit
Enter your choice:
4
Enter old password: satyam
Enter new password: Satyam12345
SUCCESS: Password updated
```

#### 7)When I add invalid credentials to login:

```
satyam@satyam-HP-Pavilion-x360-Convertible-14-dy1xxx:~/Documents/OS_lab/OS_project$ ./client
.....Welcome Back to Academia :: Course Registration.....
Login Type
Enter your choice{1.Admin , 2.Professor , 3.Student}
3
Enter ID: 1
Enter Password: Satyam
Authentication failed: AUTH FAILURE: Invalid credentials
```

#### 8)Enrolling to a new course

```
.....Welcome to Student menu.....  
Do you want to:  
1. Enroll new Course  
2. Unenroll from already enrolled Courses  
3. View enrolled Courses  
4. Password Change  
5. Exit  
Enter your choice:  
1  
Enter faculty Id1  
Enter course id: 1  
Server response: SUCCESS: Enrollment completed
```

#### 9)Server's side

```
satyam@satyam-HP-Pavilion-x360-Convertible-14-dy1xxx:~/Documents/OS_lab/OS_project$ ./server  
Server listening on port 8080...  
Received login : 3:1:satyam  
Received login : 3:1:Satyam  
Received login : 3:1:Satyam12345  
^C
```