Academia Portal - Course Registration System

Project Report
Submitted by – Kunal Mittal (IMT2023533)

1. Introduction

The Academia Portal is a terminal-based, multi-user course registration system developed to simplify academic management in educational institutions. It allows administrators, faculty members, and students to interact with a centralized system to perform various course-related operations. The system is built using low-level system programming in C and is operated through a network-based client-server model.

2. System Architecture

The system follows a client-server architecture using socket programming:

- Server: A concurrent server developed in C that handles multiple client connections, maintains data persistence, and enforces strict access controls.

- Client: Clients connect to the server using Telnet to access the system functionalities.

- Data Storage: System data is stored in plain text files with proper file locking mechanisms to ensure data consistency and prevent race conditions.

3. Implementation Details

3.1 Technologies Used

- Programming Language: C

- Networking: TCP/IP socket programming

- Concurrency: Process-based parallelism via fork()

- File I/O: System-level calls with locking using fcntl

- Build Tool: Make (GCC compiler)

3.2 Data Structure

Four colon-separated text files are used to store system data:

- students.txt: Student ID, name, password, and status (active/inactive)
- faculty.txt: Faculty ID, name, and password
- courses.txt: Course ID, name, faculty ID, maximum seats
- enrollments.txt: Course ID and enrolled student IDs

3.3 Key Functionalities

- Authentication & Authorization:
  - Role-based access (Admin, Faculty, Student)
  - Password protection
  - Status check for student accounts

- File Operations:
  - Atomic operations with proper locking
  - Efficient reading, updating, and writing of records
  - Support for concurrent access by multiple users

- User Management:
  - Add/update student and faculty accounts
  - Toggle student activation status
  - Change passwords

- Course Management:
  - Add or remove courses
  - Student enrollment and unenrollment

- Faculty view of course enrollments

4. User Roles and Functionalities

4.1 Administrator

- Add new students and faculty
- Update user information
- Toggle student active/inactive status

4.2 Faculty

- Add and remove courses
- View list of students enrolled in their courses
- Change password

4.3 Student

- Enroll and unenroll from courses
- View current course enrollments
- Change password

5. Security Features

- Role-based access control system
- Password-protected authentication
- Account status checks for students
- File locking to handle concurrency and avoid data corruption

6. Running the System

6.1 Compilation

make

gcc -o server server.c

6.2 Launching the Server

./server

- Initializes data directory and files (if missing)

- Listens for client connections on port 9000

6.3 Connecting as a Client

telnet localhost 9000 (install tellnet if not installed)

6.4 Default Admin Credentials

- Username: admin

- Password: admin123

7. Sample Interaction

Upon connecting, users are presented with:

=== Academia Portal ===

1)Admin 2)Faculty 3)Student 4)Exit

Admin Menu:

1)AddStu 2)AddFac 3)ToggleStu 4)UpdUser 5)Logout

Faculty Menu:

1)AddCourse 2)RemCourse 3)ViewEnroll 4)ChPwd 5)Logout

Student Menu:

1)Enroll 2)Unenroll 3)View 4)ChPwd 5)Logout

8. Limitations and Future Enhancements

Current Limitations

- Passwords are stored in plain text

- File-based storage may not scale efficiently

- Basic input validation and error handling

- No scheduling or prerequisite course constraints

Future Enhancements

- Encrypt password storage

- Migrate to a relational database (e.g., MySQL)

- Add a graphical or web-based interface

- Implement time-based registration windows

- Enhance reporting and analytics modules

9. Conclusion

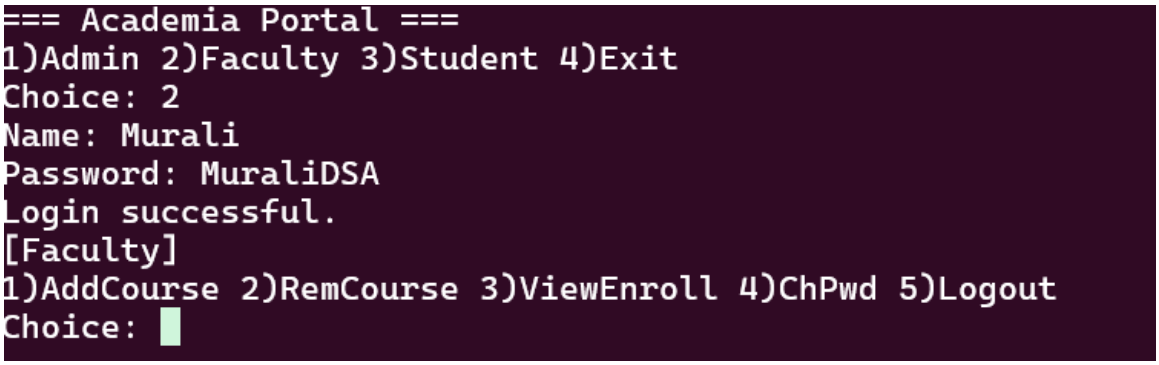
The Academia Portal showcases foundational concepts of system programming including process management, socket-based communication, and concurrent file operations. Despite its simplicity, it supports core academic operations with role-based access and secure, concurrent data handling. The system lays the groundwork for scaling into a more robust academic ERP solution in the future.

10. Screenshots

```
=== Academia Portal ===
1)Admin 2)Faculty 3)Student 4)Exit
Choice: 1
Name: admin
Password: admin123
Login successful.
[Admin]
1)AddStu 2)AddFac 3)ToggleStu 4)UpdUser 5)Logout
Choice:
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
=== Academia Portal ===
1)Admin 2)Faculty 3)Student 4)Exit
Choice:
```

```
=== Academia Portal ===
1)Admin 2)Faculty 3)Student 4)Exit
Choice: 3
Name: Kunal
Password: Kunal0025*
Login successful.
[Student]
1)Enroll 2)Unenroll 3)View 4)ChPwd 5)Logout
Choice:
```


=== Academia Portal ===
1)Admin 2)Faculty 3)Student 4)Exit
Choice: 2
Name: Murali
Password: MuraliDSA
Login successful.
[Faculty]
1)AddCourse 2)RemCourse 3)ViewEnroll 4)ChPwd 5)Logout
Choice: