OS Mini Project

Design and Development of Course Registration Portal (Academia)

-Ankith Kini (IMT2023075)

Description: The project aims to develop a Academia Portal that is user-friendly and multifunctional.
The system comprises the following files:

1)Server Program

- **server.c:** Main server application that handles all client connections.

2)Client Program

- **client.c:** User interface for interacting with the server

3)Data Storage Files

- **students.dat:** Stores student records (ID, name, password, active status)
- **faculty.dat:** Stores faculty records (ID, name, password)
- **courses.dat:**Stores course information (Course ID, Course name, faculty details, seats, enrollments)

Roles

1) Admin
- Add new students and faculty.
- Activate/deactivate student accounts.
- Update student/faculty names.

2) Faculty
- Add new courses with seat limits.
- Remove existing courses.
- View enrollments for their courses.

3) Student
- Enroll in available courses.
- Unenroll from courses.
- View list of enrolled courses.

Source Code and Implementation details

1)Server Module (server.c): Acts as the backend, handling all logic and file operations.
Functions:

- *view_student_details():*Admin operation to view the details of a student with given ID.
- *add_faculty():* Admin operation to add a professor to the database.
- *add_student()*: Admin operation to add a student to the database.
- *view_faculty_details():* Admin operation to view the details of a professor with given ID.
- *activate_student():* Admin operation that activates a student.
- *block_student():* Admin operation that blocks a student.
- *modify_student_details():*Admin operation to modify the details of a particular student.
- *modify_faculty_details():* Admin operation that modifies the details of a particular professor.
- *view_offering_courses():* Faculty operation that displays the courses offered by that particular professor.
- *add_course()*: Faculty operation that adds a new course to the database with mentioned specifications.
- *remove_course():* Faculty operation that removes a specific course.
- *update_course_details():* Faculty operation that modifies the details of a particular course offered by the professor.
- *change_faculty_password()*: Faculty operation that changes the password.
- *view_all_courses():* Student operation that displays all the courses.
- *enroll_course():* Student operation that enrols the student to the mentioned course, if the maximum limit of the course is not reached.
- *drop_course():* Student operation that de-enrols the student from the mentioned course.
- *view_enrolled_courses():* Student operation that displays the list of courses enrolled by the student.
- *change_student_password():* student operation that changes the password of the student.
- load_data(): function to load the data from the .dta files.
- save_data(): function to save the modified data to the .dat files.
- *handle_client():* function that interacts with the client.

2) Client Module(client.c): It is the interface that users interact with.
 - It handles all server communication for student, faculty, and admin operations.

The server runs first on port 8080 to handle connections, and the client connects to it in a separate terminal to provide the user interface for course registration. Both programs work together, with the server managing data and logic while the client handles input/output.

- The application possesses a password-protected administrative access, thus preventing the whole management system from unauthorized access. If the login credentials are invalid, i.e. not matching with the ones stored in the .dat files, then the client connection is terminated.

The Academia database is initially populated with this data.(I manually populated this data by using the admin, faculty and student operations).

1	student1	stdpass1	1		1	prof1	pfpass1		1	course1	prof1	p1	30	s1,s3,s5	3
2	student2	stdpass2	1		2	prof2	pfpass2		2	course2	prof3	p3	40	s2,s3,s6	3
3	student3	stdpass3	1		3	prof3	pfpass3		3	course3	prof 2	p2	10	s1,s3,s6	3
4	student4	stdpass4	1		4	prof4	pfpass4		4	course4	prof 5	p5	15	s4,s8,s7,s9	4
5	student5	stdpass5	1		5	prof5	pfpass5		5	course5	prof4	p4	50	s1,s3,s6	3
6	student6	stdpass6	1		6	prof6	pfpass6		6	course6	prof6	p6	25	s9,s3,s7	3
7	student7	stdpass7	1						7	course7	prof 2	p2	30	s8,s3,s10	3
8	student8	stdpass8	1						8	course8	prof 5	p5	20	s4,s8,s9	3
9	student9	stdpass9	1						9	course9	prof1	p1	10	s10,s9	2
10	student10	stdpass10	1						10	course10	prof4	p4	40	s2,s8,s5	3

ADMIN

Once the Admin connects to the server, He/She should get a login and password prompt.
After successful login, he/she gets the following menu with these operations.

```
......Welcome Back to Academia :: Course Registration......
Login Type
Enter Your Choice { 1.Admin , 2.Professor, 3.Student } : 1
Enter username: admin
Enter password: admin123

Welcome to Admin Menu
1. Add Student
2. View Student Details
3. Add Faculty
4. View Faculty Details
5. Activate Student
6. Block Student
7. Modify Student Details
8. Modify Faculty Details
9. Logout and Exit
Enter Your Choice:

_____
1
Enter student name: student11
Enter student password: stdpass11
Student added successfully!
```

The admin then chooses the **add student** option to add a new student with name
student11 and password **stdpass11**. The confirmation message is also printed
indicating the new student is added.(student1 to student10 are already present)

The admin then chooses the **View Student Details** option and enters the Student ID
of the student he/she wants to view, in this case it is 11 (student11).
The Student ID, Student Name and Status of the student is printed.

If the Student ID is not present, an error is thrown.

```
Welcome to Admin Menu
1. Add Student
2. View Student Details
3. Add Faculty
4. View Faculty Details
5. Activate Student
6. Block Student
7. Modify Student Details
8. Modify Faculty Details
9. Logout and Exit
Enter Your Choice:

_____
2
Enter student ID: 11
ID: 11
Name: student11
Status: Active

Welcome to Admin Menu
1. Add Student
```

```
_____
2
Enter student ID: 109
Student not found!
```

Fig: error handling for faulty Student ID

The admin now chooses the **Add Faculty** option to add a new Professor with the
name **Prof7** and password **pfpass7**. The confirmation message is also printed
indicating the new professor is added.(prof1 to prof6 are already present)

The admin then chooses the **View Faculty Details** option and enters the Faculty ID
of the professor he/she wants to view, in this case it is 7(prof7).
The Faculty ID and Faculty Name is printed.

If the Faculty ID is not present, an error is thrown.

```
 Welcome to Admin Menu
 1. Add Student
 2. View Student Details
 3. Add Faculty
 4. View Faculty Details
 5. Activate Student
 6. Block Student
 7. Modify Student Details
 8. Modify Faculty Details
 9. Logout and Exit
 Enter Your Choice:

 _____
 3
 Enter faculty name: prof7
 Enter faculty password: pfpass7
 Faculty added successfully!
 Welcome to Admin Menu
```

```
Faculty added successfully!
Welcome to Admin Menu
1. Add Student
2. View Student Details
3. Add Faculty
4. View Faculty Details
5. Activate Student
6. Block Student
7. Modify Student Details
8. Modify Faculty Details
9. Logout and Exit
Enter Your Choice:

_____
4
Enter faculty ID: 7
ID: 7
Name: prof7

Welcome to Admin Menu
```

```
------------------------------------------------------------
4
Enter faculty ID: 189
Faculty not found!

Welcome to Admin Menu
```

Fig: error handling for faulty Faculty ID

The admin then chooses the **Block Student** operation to block a student with the particular Student ID. The confirmation message indicated that the student is blocked, which can be further verified by **viewing the student details (operation 2)**.

```
------------------------------------------------------------
6
Enter student ID to block: 11
Student blocked successfully!

Welcome to Admin Menu
```

```
------------------------------------------------------------
2
Enter student ID: 11
ID: 11
Name: student11
Status: Blocked
```

Fig: Cross Verification

The admin then chooses the **Activate Student** operation to activate a student with the particular Student ID. The confirmation message indicated that the student is blocked, which can be further verified by **viewing the student details (operation 2)**.

```
------------------------------------------------------------
5
Enter student ID to activate:11
Student activated successfully!
Welcome to Admin Menu
```

```
------------------------------------------------------------
2
Enter student ID: 11
ID: 11
Name: student11
Status: Active

Welcome to Admin Menu
```
Fig: Cross Verification

The admin might choose the **Modify Student Details** operation which modifies the **Name** or **Password** of the student specified by his/her Student ID.
An incorrect choice will lead to an error as shown.

```
----------------------------------------------------
7
Enter student ID to modify: 11
What do you want to modify?
1. Name
2. Password
Enter your choice: AnkithKini
Invalid choice!
```

Here, the admin chooses to modify the name of **Student11 to AnkithKini**, which is successfully completed as verified by the **View Student Details** operation that Student.

```
----------------------------------------------------
7
Enter student ID to modify: 11
What do you want to modify?
1. Name
2. Password
Enter your choice: 1
Enter new name: AnkithKini
Student details updated successfully!
Welcome to Admin Menu
```

```
----------------------------------------------------
2
Enter student ID: 11
ID: 11
Name: AnkithKini
Status: Active

Welcome to Admin Menu
```

The admin might choose the **Modify Faculty Details** operation which modifies the **Name** or **Password** of the Faculty specified by his/her Faculty ID.
As shown below, entering the old value of the password will lead to Authentication Failure and hence the new password works.

```
--------------------------------------------------------------
8
Enter faculty ID to modify: 7
What do you want to modify?
1. Name
2. Password
Enter your choice: 2
Enter new password: newpass
Faculty details updated successfully!
Welcome to Admin Menu
```

```
● ankith@LAPTOP-33SKEA6M:~/OS/lab/project$ ./a.out
 ......Welcome Back to Academia :: Course Registration......
 Login Type
 Enter Your Choice { 1.Admin , 2.Professor, 3.Student } : 2
 Enter username: prof7
 Enter password: pfpass7
 EXIT: Authentication failed.
```

Once the admin is done with all the operations, he/she chooses Logout and Exit operation to end the session.

```
 Welcome to Admin Menu
 1. Add Student
 2. View Student Details
 3. Add Faculty
 4. View Faculty Details
 5. Activate Student
 6. Block Student
 7. Modify Student Details
 8. Modify Faculty Details
 9. Logout and Exit
 Enter Your Choice:

 --------------------------------------------------------------
 9
○ ankith@LAPTOP-33SKEA6M:~/OS/lab/project$ ▌
```

STUDENT

Once the student logins successfully, he/she is met with a list of operations that can be performed. The admin chooses the **View All Courses** which displays all the courses that can be enrolled in, by the student (here it is course1 to course10).
The 10 courses along with their details are printed.
Some of the available seats are already occupied by the Students student1 to student 10.

Once the student is done, he/she can choose the LOgout and Exit option to edn the session as shown.

```
---------------------------------------------
1
Course ID: 1
Name: course1
Faculty: prof1
Available Seats: 27/30

Course ID: 2
Name: course2
Faculty: prof3
Available Seats: 37/40

Course ID: 3
Name: course3
Faculty: prof2
Available Seats: 7/10

Course ID: 4
Name: course4
Faculty: prof5
Available Seats: 11/15

Course ID: 5
Name: course5
Faculty: prof4
Available Seats: 47/50

Course ID: 6
Name: course6
Faculty: prof6
Available Seats: 22/25

Course ID: 7
Name: course7
Faculty: prof2
Available Seats: 27/30

Course ID: 8
Name: course8
Faculty: prof5
Available Seats: 17/20
```

```
 Course ID: 9
 Name: course9
 Faculty: prof1
 Available Seats: 8/10

 Course ID: 10
 Name: course10
 Faculty: prof4
 Available Seats: 37/40


 Welcome to Student Menu
 1. View All Courses
 2. Enroll (pick) New Course
 3. Drop Course
 4. View Enrolled Course Details
 5. Change Password
 6. Logout and Exit
 Enter Your Choice:

 _____
```

Notice that **AnkithKini**(formerly student11) has logged in and since he is a newly created student, he has not enrolled in any of the courses. Therefore, the **View Enrolled Course Details** operation states that the student has not enrolled in any course.

```
 Enter Your Choice { 1.Admin , 2.Professor, 3.Student } : 3
 Enter username: AnkithKini
 Enter password: stdpass11

 Welcome to Student Menu
 1. View All Courses
 2. Enroll (pick) New Course
 3. Drop Course
 4. View Enrolled Course Details
 5. Change Password
 6. Logout and Exit
 Enter Your Choice:

 _____
 4
 You are not enrolled in any courses!
```

```
Enter Your Choice:

_____

2
Enter course ID to enroll: 10
Enrolled in course successfully!
```

The Student then chooses to Enroll in a New Course with CourseID=10, since he hasn't already enrolled in the course, he successfully enrolls himself in that course.

```
Welcome to Student Menu
1. View All Courses
2. Enroll (pick) New Course
3. Drop Course
4. View Enrolled Course Details
5. Change Password
6. Logout and Exit
Enter Your Choice:

_____

2
Enter course ID to enroll: 1
You are already enrolled in this course!
```

Fig: For student1, since he/she has already enrolled in course1, he is not allowed to enroll again, and thus the error message pops up.

This is verified by **Viewing Enrolled Course Details**, which displays the enrolled course (course10).

```
Enter Your Choice:

_____

4
Course ID: 10
Name: course10
Faculty: prof4
```
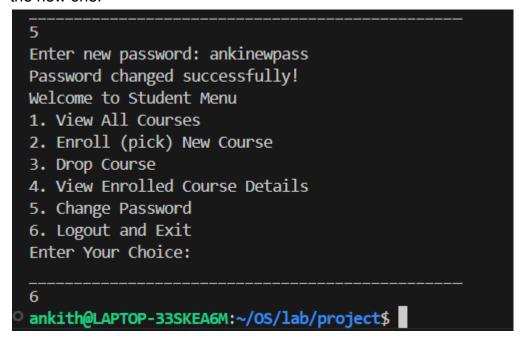
The student can **Drop a course** which he/she has enrolled in by specifying the course ID. Here, the student drops the Course10 and this is **verified by View Enrolled Courses Details.**

```
_____

3
Enter course ID to drop: 10
Course dropped successfully!
Welcome to Student Menu
```

```
———————————————————————————————————————
4
You are not enrolled in any courses!
Welcome to Student Menu
```

This states that the student is not enrolled in any course indicating he/she has successfully dropped Course10.

The student can also **Change Password** to anything he/she desires and similar to shown above, it shows **Authentication Error** on entering any password other than the new one.

```
———————————————————————————————————————
5
Enter new password: ankinewpass
Password changed successfully!
Welcome to Student Menu
1. View All Courses
2. Enroll (pick) New Course
3. Drop Course
4. View Enrolled Course Details
5. Change Password
6. Logout and Exit
Enter Your Choice:

———————————————————————————————————————
6
ankith@LAPTOP-33SKEA6M:~/OS/lab/project$ 
```

PROFESSOR

```
Login Type
Enter Your Choice { 1.Admin , 2.Professor, 3.Student } : 2
Enter username: prof1
Enter password: pfpass1

Welcome to Faculty Menu
1. View Offering Courses
2. Add New Course
3. Remove Course from Catalog
4. Update Course Details
5. Change Password
6. Logout and Exit
Enter Your Choice:

_____
1
Course ID: 1
Name: course1
Faculty: prof1
Seats: 30
Enrolled: 3

Course ID: 9
Name: course9
Faculty: prof1
Seats: 10
Enrolled: 2


Welcome to Faculty Menu
```

Upon logging in successfully with the right credentials, the professor chooses to **View Course Offered** by him/her.

```
Enter Your Choice:

_____
2
Enter course name: democourse
Enter number of seats: 2
Course added successfully!
```

The professor can then choose to **Add a New Course** which he/she plans to offer. This operation, on success, can be verified using the **View Offering Courses operation.**

```
_____
1
Course ID: 1
Name: course1
Faculty: prof1
Seats: 30
Enrolled: 3

Course ID: 9
Name: course9
Faculty: prof1
Seats: 10
Enrolled: 2

Course ID: 11
Name: democourse
Faculty: prof1
Seats: 2
Enrolled: 0
```

The professor can then choose to **Remove a Course** offered under him, by specifying the courseID. The operation, on success, can be verified using the V**iew Offered Course operations.**

```
_____
3
Enter course ID to remove: 11
Course removed successfully!
```

```
------------------------------------------------------------
1
Course ID: 1
Name: course1
Faculty: prof1
Seats: 30
Enrolled: 3

Course ID: 9
Name: course9
Faculty: prof1
Seats: 10
Enrolled: 2
```

The professor can then choose to **Update Course Details**, either the **Course Name** or **Number of seats**. Upon completion, the Professor chooses the **Logout and Exit** option to terminate the session.

```
------------------------------------------------------------
4
Enter course ID to update: 1
What do you want to update?
1. Course name
2. Number of seats
Enter your choice: 2
Enter new number of seats: 3
Course updated successfully!

Welcome to Faculty Menu
1. View Offering Courses
2. Add New Course
3. Remove Course from Catalog
4. Update Course Details
5. Change Password
6. Logout and Exit
Enter Your Choice:

------------------------------------------------------------
6
ankith@LAPTOP-33SKEA6M:~/OS/lab/project$
```

```
Enter username: student2
Enter password: stdpass2

Welcome to Student Menu
1. View All Courses
2. Enroll (pick) New Course
3. Drop Course
4. View Enrolled Course Details
5. Change Password
6. Logout and Exit
Enter Your Choice:

_____
2
Enter course ID to enroll: 1
No seats available in this course!
```

A student cannot enroll in a course that has no spots left, i.e. the maximum occupancy has been reached.

_____