

# **OS Mini Project - Report**

*Name: Gayathri V.*

*Roll No: IMT2018026*

## **Overview**

This project aims to build a banking system using system calls and file locking. The system supports three types of users - normal, joint and admin. Users can use basic functionalities of the banking system, which include authentication, deposit and withdrawal. A concurrent server is used to facilitate multiple clients at once. In order to demonstrate file locking, we show that multiple users holding a joint account can query the server simultaneously.

## **Types of Commands**

1. User Commands (Normal and Joint)
  - a. Sign Up: Different options are provided to sign up as a normal and joint user. Username and password must be provided. One can't sign up if the username already exists. On sign up, your balance is initialized to 0.
  - b. Sign In: Username and password should be provided while signing in. One must specify what kind of user they are signing in as.
  - c. Deposit: Deposit money into your account.
  - d. Withdraw: Withdraw money from your account.
  - e. Check Balance: Used to check how much money your account holds.
  - f. Change Password: Used to change password to login.
  - g. View Details: Will return username, password, type of user and balance.
  - h. Exit: Used to terminate the program.
2. Admin Commands
  - a. Add User: Used to create a user of type normal or joint.
  - b. Delete User: Used to delete user given username
  - c. Modify User: Used to modify username and password given the old username.
  - d. Get User Details: Used to get password and balance given the username.

## **File Structure**

- client.c: This contains the code for the client (user and admin interface). It communicates with the server through the socket.
- server.c: This contains the code for the concurrent server. For every client, the server creates a *new thread* to ensure that multiple clients can query the server independently.
- menu.c: Contains the menu-driven part of the code.

## **Overview of Working**

When a user signs up for the first time, a file is created under the name 'username.txt'. This file stores all his relevant information - account type, balance etc. If 'username.txt' already exists, then that user cannot sign up. All queries are answered by opening the corresponding file, and reading from it. In order to delete a user, the corresponding file is deleted.

## **Execution**

Client: gcc client.c menu.c

./a.out

Server: gcc server.c -lpthread

./a.out