Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Summer, Year: 2025), B.Sc. in CSE (Day)

Banking System IPC

Course Title: Operating System Lab
Course Code: CSE 310
Section: D2 (223)

Students Details

Name	ID
ANISUR RAHAMAN MARUF	222902078
SAKIBU HASAN	222902083

Submission Date: August 26, 2025
Course Teacher's Name: Umme Habiba

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

0.1 Introduction

0.1.1 Overview

The Interactive Banking System is a terminal-based banking application developed using Bash scripting. This system implements Inter-Process Communication (IPC) through file-based data persistence, utilizing CSV files for account storage and transaction logging.

Key features include account management, money transfers, ATM operations, PIN-based security, daily withdrawal limits, and administrative controls with a color-coded terminal interface.

0.1.2 Motivation

This project was developed to understand core banking operations and IPC concepts through practical implementation. The motivation stems from:

- Learning file-based IPC mechanisms
- Understanding banking security principles
- Implementing real-world financial system operations
- Demonstrating concurrent data access handling

0.1.3 Problem Definition

Key challenges addressed during development:

- **Data Consistency:** Preventing corruption during concurrent file access
- **Security Issues:** Protecting PINs, preventing unauthorized access
- **File System Limitations:** Managing CSV operations safely
- **User Experience:** Creating intuitive terminal interface
- **Transaction Integrity:** Ensuring atomic financial operations

0.1.4 Design Goals/Objectives

Primary Objectives:

- Implement file-based IPC using CSV storage
- Develop multi-layered security with PIN authentication
- Create user-friendly terminal interface with color coding

- Ensure transaction integrity and data consistency
- Build modular, maintainable code architecture

Design Principles:

- Separation of concerns (UI, data, business logic)
- Comprehensive error handling and input validation
- Security by design with multiple authentication layers

0.1.5 Application

- **Educational Use:** Teaching system programming, IPC concepts, and bash scripting
- **Small-scale Banking:** Community banking, credit unions, personal finance tracking
- **Development Training:** Real-world application development experience

0.2 Design/Development/Implementation of the Project

0.2.1 Introduction

The system uses modular design with bash scripting, organizing functionality into data management, user interface, security, and transaction processing modules. IPC is achieved through shared CSV files enabling multiple user sessions.

0.2.2 Project Details

System Components:

1. **Data Layer:** `accounts.db` (CSV) and `transactions.log`
2. **Security Layer:** PIN authentication, attempt tracking, account locking
3. **UI Layer:** Color-coded interface with formatted output
4. **Business Logic:** Account operations and transaction processing

IPC Implementation:

- File-based communication through shared CSV files
- Atomic operations for data consistency
- State persistence across sessions

0.2.3 Implementation

Technologies Used:

- Bash scripting for core functionality
- CSV format for structured data storage
- AWK for text processing and data manipulation
- ANSI color codes for terminal styling

Key Functions: [language=bash] Data operations row$_g et()awk - F, -vA =$ ”1" ’1 == $Aprint; exit'$”ACCOUNTS$_F ILE$”; $update_f ield()awk - F, -vA =$ ”acc" -v I="2” − $vV =$ ”3" ’BEGINOFS="," 1 == AI=V1’ "$ACCOUNTS_F ILE$”;

Core banking operations create$_a ccount(), transfer_m oney(), withdraw_m oney(), check_b alance()$

0.2.4 Code Details

File Structure:

- Configuration and color definitions
- UI helper functions for formatting
- CSV data management functions
- Core banking operation functions
- Menu systems and user interaction

Security Features:

- PIN verification with attempt counting
- Account locking after 3 failed attempts
- Administrative password protection
- Complete transaction audit logging

0.3 Performance Evaluation

0.3.1 Simulation Environment

Testing Setup:

- Linux/Unix environment with Bash 4.0+
- Multiple test accounts with various scenarios
- Security and performance testing protocols

0.3.2 Results Analysis/Testing

Test Results:

- ✓ All CRUD operations functional
- ✓ Transaction integrity maintained
- ✓ Security features working properly
- ✓ UI components displaying correctly
- ✓ Data persistence across sessions

Performance Metrics:

- Account creation: $<$ 1 second
- Balance inquiry: $<$ 0.5 seconds
- Money transfer: $<$ 2 seconds
- Efficient for typical banking loads

0.3.3 Results Overall Discussion

The system successfully demonstrates IPC implementation with reliable data consistency, effective security measures, and intuitive user experience. Performance is excellent for small to medium-scale operations with room for optimization in larger deployments.

0.4 Conclusion

0.4.1 Discussion

The Interactive Banking System successfully implements core banking operations using IPC principles. Key achievements include:

- Successful file-based IPC implementation
- Comprehensive security model with authentication
- Robust CSV-based data management
- Complete banking functionality with user-friendly interface

0.4.2 Limitations

Current Constraints:

- Limited scalability with large datasets
- Basic concurrent access handling
- No data encryption for stored information
- Platform dependency on Unix/Linux systems
- No network or GUI capabilities

0.4.3 Scope of Future Work

Immediate Enhancements:

- Database integration (SQLite) for better performance
- Data encryption for security enhancement
- Improved concurrent access management
- Enhanced logging and audit trails

Advanced Features:

- Web interface development
- Mobile application integration
- Network-based operations
- Advanced security (2FA, biometrics)
- Real-time monitoring dashboard

References

1. Advanced Bash Scripting Guide - Mendel Cooper
2. Unix System Programming - Kay A. Robbins & Steven Robbins
3. Operating System Concepts - Abraham Silberschatz
4. Inter-Process Communication in Linux - John Shapley Gray