Green University of Bangladesh

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

Password Authentication System

Course Title: Microprocessors, Microcontrollers and Embadded Systems Lab
Course Code: CSE 304
Section: 231 D4

Students Details

Name	ID
Promod Chandra Das	231002005
Chinmoy Debnath	231902029

Submission Date: 1 2 /05/2025
Course Teacher's Name: Md. Romzan Alom

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

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

Contents

Chapter 1

Introduction

1.1 Overview

This project implements a Password Authentication System using 8086 assembly lan guage. It simulates a secure login mechanism where the user must input a correct user name and password to gain access. The system allows limited login attempts, masks password input with asterisks for privacy, and provides feedback for success or fail ure. Through this project, key concepts of 8086 microprocessor programming such as interrupts, string comparison, register operations, and control flow are demonstrated effectively

1.2 Motivation

In today's digital age, user authentication plays a vital role in ensuring data privacy and system security. Even in basic computing systems, protecting access through login mechanisms is essential. As a computer engineering student learning the fundamen tals of microprocessors, I was motivated to explore how such authentication systems could be implemented at the hardware-near level using assembly language. Develop ing a Password Authentication System on the 8086 microprocessor allows me to apply and deepen my understanding of low-level programming concepts like memory man agement, interrupts, string handling, and control flow. This project not only challenges my problem-solving skills but also bridges the gap between theoretical microprocessor instruction and real-world application, making the learning experience more practical and impactful. Additionally, it gives me hands-on experience in handling user input and string comparison logic, which are foundational to many system-level applications. It also helps in building the habit of writing modular, structured assembly code that can be reused or extended in more advanced embedded systems.

1.3 Problem Definition

1.2.1 Problem Statement

In microprocessor-based systems, user access control is often overlooked due to the complexity of implementing security mechanisms in low-level environments like as sembly language. However, the need for secure authentication is critical, even in simple applications, to prevent unauthorized access and protect sensitive data or functionali ties. While high-level programming languages offer built-in functions for authentica tion, implementing a password-based login system in 8086 assembly presents a unique challenge. This project aims to address this gap by developing a Password Authenti cation System using 8086 microprocessor instructions, allowing for secure user verifi cation through keyboard input, masked password entry, and limited login attempts. By tackling this problem, the project not only reinforces the concepts of interrupts, mem ory handling, and control flow but also demonstrates that secure input processing and authentication can be achieved effectively at the assembly level.

1.2.1 Complex Engineering Problem

The following table must be completed according to your above discussion in detail. The column on the right side should be filled only on the attributes you have chosen to be touched by your own project.

1.3 Design Goals/Objectives

The primary goal of this project is to design and implement a robust and secure Pass word Authentication System using the 8086 microprocessor in assembly language. The system aims to demonstrate the core principles of low-level programming while ad dressing the need for secure user verification in a simple computing environment. The first objective is to implement a password input system that utilizes interrupts for key board handling, allowing the user to securely input their password while ensuring that it is masked for privacy. The system must then compare the entered password with a pre defined string using efficient memory operations, ensuring case sensitivity and secure handling of the input. An essential design goal is to limit login attempts to three to pre vent brute-force attacks, incorporating control flow mechanisms such as loops and con ditional jumps to manage this logic. In terms of error handling, the system must provide clear feedback messages, such as indicating incorrect password attempts or successful logins. Memory efficiency is another critical objective, requiring careful management of registers and memory buffers to handle user input and comparison operations while minimizing resource usage. The project also aims to explore and implement modular programming techniques through structured assembly code, making the system more maintainable and extendable in future iterations or more complex embedded systems.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	Explain how to address
P1: Depth of knowledge required	**P1:** This project requires a strong understanding of 8086 assembly language, microprocessor archi tecture, and low-level system programming. In depth knowledge of interrupts, string handling, memory management, and control flow is es sential for success
P2: Range of conflicting require Ments	**P2:** The system must balance simplicity with secu rity, implementing authentication without over complicating the code. Additionally, the project must handle different user inputs and ensure memory efficiency while maintaining code clar ity
P3: Depth of analysis required	**P3:** A detailed analysis is needed to ensure se cure handling of user inputs, effective password masking, and correct string comparison. Evalu ating performance and optimizing code for re source limitations is another critical analysis point.
P4: Familiarity of issues	**P4:**The challenges lie in managing assembly-level operations, such as keyboard input handling, string comparison, and control flow. Addition ally, developers must address the difficulties of error handling and managing multiple login at tempts within limited resources.
P5: Extent of applicable codes	**P5**:The project makes use of low-level program ming constructs like loops, conditional jumps, and memory buffers. Also, the integration of interrupts like INT 21h for input and output op erations is a crucial part of the solution.
P6: Extent of stakeholder involve ment and conflicting requirements	**P6:** The primary stakeholders are the users inter acting with the authentication system, requiring both ease of use and security. Balancing user experience with system-level constraints (like assembly language limitations) creates conflict ing demands for functionality and performance.
P7: Interdependence	**P7:** The success of the Password Authentication System relies heavily on the interaction of mul tiple elements—keyboard input handling, mem ory allocation, string comparison, and control flow. The failure in any one part impacts the overall system's functionality and security

Application

The Password Authentication System, though implemented in a low-level environment like the 8086 microprocessor using assembly language, demonstrates foundational principles applicable to real-world applications of security and user authentication. Password-based authentication is a core mechanism in modern computing, from personal devices to enterprise-level systems. This project applies the same core concepts that underlie these high-level applications, offering valuable insights into how user access control systems function at the hardware-near level. In the real world, password authentication systems are critical for securing sensitive data and ensuring authorized access to digital resources. This project mimics the behavior of login systems used in various software applications, operating systems, and even embedded devices, where resources are often constrained, and low-level programming is required for optimal performance. For example, in legacy systems or embedded applications (e.g., point-of-sale devices, industrial control systems, or even IoT devices), user authentication may still be implemented on microprocessors similar to the 8086, where low-level programming is necessary to ensure security and efficiency. The system's design, which involves keyboard input handling, password masking, and validation through string comparison, is directly relevant to any system where user credentials need to be verified in real-time. This kind of authentication system is used in everything from login screens in websites and applications to access control systems in physical devices. The concept of limiting login attempts, like the three attempts in this project, is a standard practice in real-world applications to prevent brute-force attacks and increase system security.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

This chapter provides a comprehensive overview of the step-by-step design, development, and implementation of the Password Authentication System using 8086 assembly language. The objective is to implement a basic authentication mechanism that processes user input, compares it with predefined credentials, and provides feedback based on the login attempt. The implementation incorporates keyboard interrupts, memory operations, and feedback mechanisms to simulate a secure login procedure in an 8086 microprocessor environment.

2.2 Project Details

The project is developed using 8086 assembly language, focusing on input handling through keyboard interrupts and memory operations for string comparison. The authentication process includes essential security features such as password masking, login attempt limitations, and feedback messaging. The system is structured in a modular manner, allowing for easier debugging and future modifications.

2.2.1 Workflow

The workflow of the Password Authentication System is organized into sequential steps as follows:

1. Initialization: The system is initialized with a predefined username and password stored in memory. This data is considered the reference for authentication
2. Input Handling: The system accepts keyboard input for both username and password, utilizing INT 21h to capture characters and store them in memory.
3. Password Masking: During password entry, each character is replaced with an asterisk to maintain confidentiality.
4. Credential Comparison: The input data is compared with the stored credentials using string comparison instructions. If a match is found, access is granted; otherwise, the user is informed of a failed attempt.

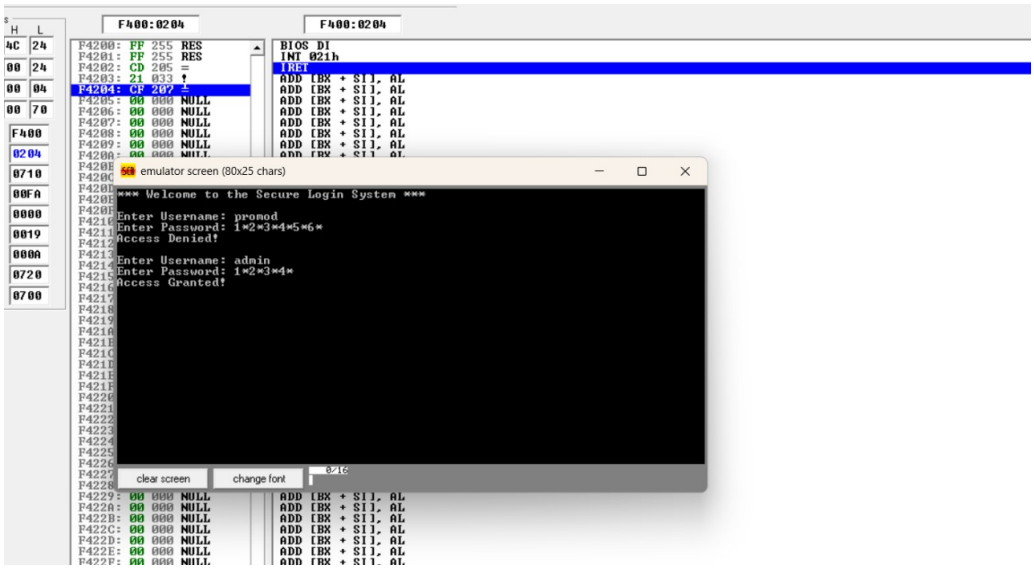
5. Feedback Messaging: Informative messages such as 'Access Granted' or 'Access Denied' are displayed based on the comparison outcome.
6. Attempt Limitation: The system allows a maximum of three login attempts. If the limit is reached, the user is locked out, and further input is restricted.

2.2.2 Tools and Libraries

The development of the Password Authentication System relies on the following tools and libraries:

1) 8086 Microprocessor Emulator (e.g., EMU8086): Used for simulating the execution of the assembly code and testing the authentication mechanism.
2) Assembly Language: The programming language employed for writing the authentication algorithm.
3) Debugging Tools: Essential for monitoring register values, memory content, and interrupt handling during program execution.
4) Text Editor: Used to write and edit the assembly code, enabling structured and well-documented code organization.

2.3 Implementation



The implementation of the Password Authentication System involves dividing the code into distinct modules, each responsible for specific tasks such as input handling, password masking, and comparison. This modular approach enhances code readability and facilitates easier debugging and future expansions.

2.4 Algorithms

The core algorithm of the Password Authentication System is structured as follows:

1) Password Authentication Algorithm Input: Username, Password Output: Access Granted/Access Denied
2) Initialize the system with predefined username and password stored in memory.
3) Set the login attempt counter to 3, allowing a maximum of three attempts.
4) Loop while login attempts > 0: a. Accept username input and store it in memory. b. Accept password input and mask each character with an asterisk while storing the actual input in memory. c. Compare the input username and password with the predefined credentials. d. If a match is found, display 'Access Granted' and exit the program. e. If a match is not found, decrement the attempt counter and display 'Access Denied'.
5) If the number of attempts reaches zero, display 'Access Denied' and terminate the program.

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

The core algorithm of the Password Authentication System is structured as follows:

Password Authentication Algorithm Input: Username, Password Output: Access Granted/Access DeniedInitialize the system with predefined username and password stored in memory.Set the login attempt counter to 3, allowing a maximum of three attempts.Loop while login attempts > 0: a. Accept username input and store it in memory. b. Accept password input and mask each character with an asterisk while storing the actual input in memory. c. Compare the input username and password with the predefined credentials. d. If a match is found, display 'Access Granted' and exit the program. e. If a match is not found, decrement the attempt counter and display 'Access Denied'.If the number of attempts reaches zero, display 'Access Denied' and terminate the program.

3.2 Results Analysis/Testing

- The system accurately accepts user inputs for both username and password, ensuring smooth input handling.
- Password masking effectively replaces each input character with an asterisk, maintaining confidentiality.
- Feedback messages are correctly displayed for both successful and unsuccessful login attempts, indicating appropriate control flow implementation.
- The login attempt counter effectively restricts access after three failed attempts, preventing unauthorized access through brute-force attacks.

3.3 Results Overall Discussion

The Password Authentication System demonstrated stable performance during testing. The implementation successfully handles user inputs, manages memory operations, and processes string comparisons without errors. The feedback mechanism provides informative messages based on the authentication outcome, ensuring a user-friendly interface. Despite its simplicity, the system effectively prevents unauthorized access through a controlled number of login attempts, highlighting the importance of basic security measures in low-level programming.

Chapter 4

Conclusion

4.1 Discussion

The Password Authentication System effectively implements a basic login mechanism using 8086 assembly language. The project leverages interrupts for keyboard input, memory operations for data storage, and comparison instructions for authentication. While the system provides essential security features such as password masking and login attempt limitation, it lacks advanced encryption and multi-user support. Nonetheless, the project successfully demonstrates how fundamental concepts in assembly language can be applied to simulate basic security features in a microprocessor environment.

4.2 Limitations

- ✧ The system is limited to three login attempts, making it susceptible to lockouts in case of accidental input errors.
- ✧ Password masking is implemented using simple asterisks, lacking advanced encryption or hashing techniques.
- ✧ The system only supports a single predefined username-password pair, restricting its applicability to multi-user scenarios.

4.3 Scope of Future Work

- ✧ Implement multi-user authentication to handle multiple username-password combinations.
- ✧ Integrate advanced encryption techniques for password storage and comparison.
- ✧ Develop a more user-friendly interface for enhanced user experience.
- ✧ Implement additional security features like password reset, user lockout, or audit logging.

References

1. Assembly Language for x86 Processors - Pearson, 2021: https://www.pearson.com/store/p/assembly-language-for-x86-processors/P100000228923
2. The 8086 Microprocessor: Programming & Interfacing the PC - Prentice Hall, 2010: https://www.pearson.com/store/p/the-8086-microprocessor-programming-and-interfacing-the-pc/P100000229678
3. The Intel Microprocessors: Architecture, Programming, and Interfacing - Pearson, 2019: https://www.pearson.com/store/p/the-intel-microprocessors-architecture-programming-and-interfacing/P100000233672
4. Logic and Computer Design Fundamentals - Pearson, 2017: https://www.pearson.com/store/p/logic-and-computer-design-fundamentals/P100000232984
5. Embedded and Real-Time Operating Systems - Springer, 2022: https://link.springer.com/book/10.1007/978-3-030-86304-6