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: 07/04/2025
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 language. It simulates a secure login mechanism where the user must input a correct username and password to gain access. The system allows limited login attempts, masks password input with asterisks for privacy, and provides feedback for success or failure. 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 fundamentals of microprocessors, I was motivated to explore how such authentication systems could be implemented at the hardware-near level using assembly language. Developing a Password Authentication System on the 8086 microprocessor allows me to apply and deepen my understanding of low-level programming concepts like memory management, 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].

1.3 Problem Definition

1.3.1 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 assembly language. However, the need for secure authentication is critical, even in simple applications, to prevent unauthorized access and protect sensitive data or functionalities. While high-level programming languages offer built-in functions for authentication, implementing a password-based login system in 8086 assembly presents a unique challenge. This project aims to address this gap by developing a Password Authentication System using 8086 microprocessor instructions, allowing for secure user verification through keyboard input, masked password entry, and limited login attempts. By tackling this problem, the project not only reinforces the concepts of interrupts, memory handling, and control flow but also demonstrates that secure input processing and authentication can be achieved effectively at the assembly level.

1.3.2 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.

1.4 Design Goals/Objectives

The primary goal of this project is to design and implement a robust and secure Password Authentication System using the 8086 microprocessor in assembly language. The system aims to demonstrate the core principles of low-level programming while addressing 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 keyboard 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 predefined 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 prevent brute-force attacks, incorporating control flow mechanisms such as loops and conditional 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	This project requires a strong understanding of 8086 assembly language, microprocessor architecture, and low-level system programming. In-depth knowledge of interrupts, string handling, memory management, and control flow is essential for success.
P2: Range of conflicting requirements	The system must balance simplicity with security, implementing authentication without overcomplicating the code. Additionally, the project must handle different user inputs and ensure memory efficiency while maintaining code clarity.
P3: Depth of analysis required	A detailed analysis is needed to ensure secure handling of user inputs, effective password masking, and correct string comparison. Evaluating performance and optimizing code for resource limitations is another critical analysis point.
P4: Familiarity of issues	The challenges lie in managing assembly-level operations, such as keyboard input handling, string comparison, and control flow. Additionally, developers must address the difficulties of error handling and managing multiple login attempts within limited resources.
P5: Extent of applicable codes	The project makes use of low-level programming constructs like loops, conditional jumps, and memory buffers. Also, the integration of interrupts like INT 21h for input and output operations is a crucial part of the solution.
P6: Extent of stakeholder involvement and conflicting requirements	The primary stakeholders are the users interacting with the authentication system, requiring both ease of use and security. Balancing user experience with system-level constraints (like assembly language limitations) creates conflicting demands for functionality and performance.
P7: Interdependence	The success of the Password Authentication System relies heavily on the interaction of multiple elements—keyboard input handling, memory allocation, string comparison, and control flow. The failure in any one part impacts the overall system's functionality and security.

1.5 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

Start the section with a general discussion of the project [2] [3] [4].

2.2 Project Details

In this section, you will elaborate on all the details of your project, using subsections if necessary.

2.2.1 Subsection_name

Figure 2.1: Figure name

You can fix the height, width, position, etc., of the figure accordingly.

2.3 Implementation

All the implementation details of your project should be included in this section, along with many subsections.

2.3.1 Subsection_name

This is just a sample subsection. Subsections should be written in detail. Subsections may include the following, in addition to others from your own project.

The workflow

Tools and libraries

Implementation details (with screenshots and programming codes)

Each subsection may also include subsubsections.

2.4 Algorithms

The algorithms and the programming codes in detail should be included .
Pseudo-codes are also encouraged very much to be included in this chapter for your project.

- Bullet points can also be included anywhere in this project report.

Algorithm 1: Sample Algorithm

Input: Your Input
Output: Your output
Data: Testing set x

```
1  ∑_{i=1}^{∞} := 0                                   // this is a comment
   /* Now this is an if...else conditional loop                    */
2  if Condition 1 then
3      Do something                              // this is another comment
4      if sub-Condition then
5          Do a lot
6  else if Condition 2 then
7      Do Otherwise
       /* Now this is a for loop                                   */
8      for sequence do
9          loop instructions
10 else
11     Do the rest
   /* Now this is a While loop                                     */
12 while Condition do
13     Do something
```

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

Discuss the experimental setup and environment installation needed for the simulation of your outcomes.

3.1.1 Subsection

3.1.2 Subsection

3.2 Results Analysis/Testing

Discussion about your various results should be included in this chapter in detail.

3.2.1 Result_portion_1

The results of any specific part of your project can be included using subsections.

3.2.2 Result_portion_2

Each result must include screenshots from your project. In addition to screenshots, graphs should be added accordingly to your project.

3.2.3 Result_portion_3

Each result must have a single paragraph describing your result screenshots or graphs or others. This is a simple discussion of that particular portion/part of your result.

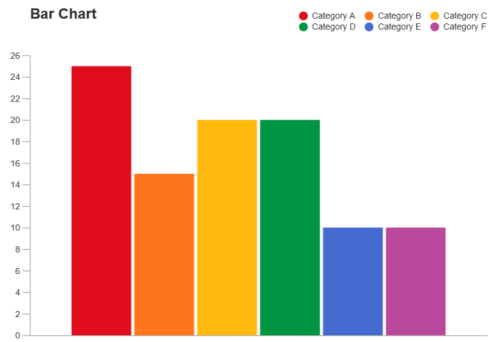

Figure 3.1: A graphical result of your project

3.3 Results Overall Discussion

A general discussion about how your result has arrived should be included in this chapter. Where the problems detected from your results should be included as well.

3.3.1 Complex Engineering Problem Discussion

[OPTIONAL] In this subsection, if you want, you can discuss in details the attributes that have been touched by your project problem in details. This has already been mentioned in the Table 1.1.

Chapter 4

Conclusion

4.1 Discussion

Discuss the contents of this chapter and summarized the description of the work and the results and observation. Generally, it should be in one paragraph.

4.2 Limitations

Discuss the limitations of the project. Limitations must be discussed, with the help of some critical analysis.

4.3 Scope of Future Work

Discuss the future work of the project, that is your plans for more work and extension of your project.

References

[1] Omid C Farokhzad and Robert Langer. Impact of nanotechnology on drug delivery. *ACS nano*, 3(1):16–20, 2009.

[2] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.

[3] Douglas Laney. 3d data management: controlling data volume, velocity and variety. gartner, 2001.

[4] MS Windows NT kernel description. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed Date: 2010-09-30.