lication Design

Number System Conversion Project Report

**Submitted To :**

#### **Name** : Saima Siddique Tashfia

#### **Designation :** Lecturer of uits

#### **Date of Submission :** 09/12/2024

#### **Cours code** : CSE 412

**Course title :** Operating Systems Lab

ion :Date of Submission :

et of Things Lab

ion :Date of Submission :

**Submitted By :**

**Name :** Md Shariful islam sajib sarker (2125051016)

**Name :** Kamrun nahar mitu (2125051019)

**Name :** Kamrul Huda Sattari Saad (2125051041)

**Name :** Md Tajbeer Ahamed Rimon (2125051007)

**Batch** : 50

**Submitted By :**

**Name :** Md Shariful islam sajib sarker (2125051016)

**Name :** Kamrun nahar mitu (2125051019)

**Name :** Kamrul Huda Sattari Saad (2125051041)

**Name :** Md Tajbeer Ahamed Rimon (2125051007)

**Batch** : 50

**Abstract**

This project focuses on the implementation of a Bash script designed to facilitate number system conversions. The script offers a menu-driven approach where users can select their desired conversion, input the number, and receive the converted result. The program supports all fundamental conversions between binary, decimal, octal, and hexadecimal number systems. Additionally, the user can exit the application through an explicit option in the menu, making the interaction intuitive and efficient.

**Introduction**

Number system conversions are essential in computer science and programming, especially when working with data at different abstraction levels. This project demonstrates how Bash scripting can simplify this task. The script serves as a lightweight solution, capable of performing conversions without the need for external libraries or software. The primary objectives of this project are:

1. To create a user-friendly interface for number system conversion.
2. To understand the functionality of loops, conditionals, and user input in Bash scripting.
3. To demonstrate the utility of command-line scripting in solving practical problems.

**Methodology**

The project script was developed using the Bash shell scripting language. The methodology involved:

1. Menu Design:
   * A menu was created to display conversion options to the user.
   * The menu includes options for conversions like:
     + Binary to Decimal
     + Decimal to Binary
     + Decimal to Octal
     + Decimal to Hexadecimal
     + Exit
2. Input Handling:
   * The script prompts the user to choose a conversion type.
   * Based on the selection, the script asks for a number input in the respective format.
3. Conversion Logic:
   * Conversion logic is implemented using built-in Bash utilities such as bc (arbitrary-precision calculator language).
   * Conditional statements (case) were used to handle various conversions.
4. Looping Mechanism:
   * The script runs in a loop to allow continuous conversions until the user explicitly chooses to exit.

**Implementation**

Below is the code implemented for this project:

|  |
| --- |
| #!/bin/bash  while true; do  echo "----------------------------------"  echo "Number System Conversion Menu"  echo "----------------------------------"  echo "1. Binary to Decimal"  echo "2. Decimal to Binary"  echo "3. Decimal to Octal"  echo "4. Decimal to Hexadecimal"  echo "5. Exit"  echo "----------------------------------"  read -p "Choose an option (1-5): " choice  case $choice in  1)  read -p "Enter a binary number: " binary  decimal=$((2#$binary))  echo "Decimal: $decimal"  ;;  2)  read -p "Enter a decimal number: " decimal  binary=$(echo "obase=2; $decimal" | bc)  echo "Binary: $binary"  ;;  3)  read -p "Enter a decimal number: " decimal  octal=$(echo "obase=8; $decimal" | bc)  echo "Octal: $octal"  ;;  4)  read -p "Enter a decimal number: " decimal  hex=$(echo "obase=16; $decimal" | bc)  echo "Hexadecimal: $hex"  ;;  5)  echo "Exiting the program. Goodbye!"  break  ;;  \*)  echo "Invalid option! Please try again."  ;;  esac  done |

**Results**

The script was tested with various inputs, and the results were verified to be accurate for all supported conversions. Below are some test cases:

| Input | Conversion Type | Output |
| --- | --- | --- |
| 1010 (Binary) | Binary to Decimal | 10 |
| 15 (Decimal) | Decimal to Binary | 1111 |
| 20 (Decimal) | Decimal to Octal | 24 |
| 255 (Decimal) | Decimal to Hexadecimal | FF |

The exit functionality also worked as intended, gracefully terminating the program.

**Conclusion**

This project highlights the versatility of Bash scripting in performing number system conversions effectively. The script provides a seamless and interactive user experience with its menu-driven design. Through this project, concepts like loops, conditionals, and arithmetic operations in Bash were reinforced. The system is simple yet robust, making it a handy tool for quick number system conversions.

**Future Enhancements**

1. Adding more number systems, such as base-3 or base-7.
2. Validating inputs more rigorously to handle invalid numbers.
3. Improving the UI by integrating color-coded messages for clarity.
4. Extending the script to handle batch conversions using files.

**References**

1. GNU Bash Manual
2. Linux bc Command Documentation
3. Classroom Lectures and Course Materials