Project Report

1. Program Description

This program is a binary conversion game where the player solves conversion problems between binary and decimal numbers. The game has 10 levels, and each level progresses with an additional problem until level 10 has 10 problems. The problems are generated randomly, where the user is given a decimal number and must input the binary representation, or vice versa. The user gets 10 points for each correct answer, and the game displays the final score at the end.

2. Challenges and Solutions

Challenge 1: Displaying the game board with ASCII characters
Creating a readable board with boxes was challenging because I needed to line everything up properly. I used the line and plus symbols to make borders, and printed each bit or space with a loop to fill the boxes correctly.

Challenge 2: Reading and validating binary string input
The hardest part was reading a string of 8 binary digits from the user and converting it to a number. I solved this by using syscall 8 to read the string, then looping through each character to check if it's a 0 or 1, and building the decimal value by shifting left and adding bits.

3. What I Learned

Through this project, I learned how to work with binary numbers at a low level using bit shifting and masking operations. I also learned how to structure a larger assembly program using multiple files and the .include directive to keep code organized. The most important skill I gained was understanding how to validate user input in assembly as well as how to use syscalls effectively to create a game with text input and output.

4. Algorithms and Techniques

How the Board is Displayed

The board is displayed using a loop that prints ASCII characters. First, I print the top border with plus and minus signs, then loop through bits 7 to 0, left to right. For each bit I print the actual bit value or spaces depending on the problem. I used bit shifting to extract each bit; I shift 1 left by the bit position and with the number, then shift right to get just 0 or 1. The decimal value is printed in the last box based on the current mode.

How Input is Validated

For decimal input I check if the number is between 0 and 255 using branch instructions. If it’s not then I print an error message and ask again. For binary input, I read the string and loop through each character checking if it's 0 or 1, and I count how many digits there are. If I find an invalid character or if there are not 8 digits, I return -1 which signals an error and prompts the user to try again.

How the Program Works

The main program starts at level 1 and score 0. For each level, it displays that corresponding number of questions. For each line, it generates a random mode (0 or 1) and a random 8 bit number. It displays the board based on the mode, gets the user's input, and compares it to the correct answer. If correct, it adds 10 points and says “Correct!”; if wrong, it shows the correct answer. After completing all 10 levels, it displays the final score and asks if the user wants to play again.