

1. Overview:

The program is a simple number guessing game, with a little “Easter Egg” using MIPS

2. Description:

When the program starts, it will greet the player, provide instructions and generate a random number within the range of 0 to 100 for the user to guess. The user has 10 attempts to guess. After each attempt, the game will tell the player if the guessed number is lower/higher than the result. Otherwise, if the user correctly guesses the number, the program will congratulate the user, and count the scores based on the numbers of attempts left from the user. The game will ask the user if they want to continue or not, and will behave accordingly.

As an “Easter Egg” from the author, if the number generated is 23 and the user correctly guessed that, the program will print a special line.

3. Design decisions:

a. Instructions:

The program will have specific instructions on how to play, and how the game works (scoring, etc). Thus it expects the user to enter the correct input. Some of the input will crash the program, while the others can be caught and be reminded to the user. The details will be presented below.

b. Input format:

The program uses syscall 5 to read integers from the user input, so if the input is not an integer (i.e the input contains characters other than digits from 0 to 9), the program will crash (terminate with error). As a result, the user needs to enter an integer for the program to continue running.

c. Input validation:

The program supports integers in the range of 0 to 100. Thus it will notify the user if the guess is less than 0 or more than 100, and let the user make another guess.

d. Random Number Generator (RNG)

To generate a random number, the program uses syscall 42 (generate random number), with a1 set to 100 (the upperbound for syscall 42). The result will be stored in a0. As a side note, the MIPS assembly syscall 42 will generate a pseudo-random integer within the target range.

e. Easter egg:

The program is hard-coded to print a special line if the random number is 23, and the guess is correct. The chance to get this result is low (1% for the RNG to get the number, and the player only gets 10 attempts to correctly guess it).

4. Future features: The program is created with extensibility in mind, and thus it supports additional features to be implemented in the future. Some of the easy-to-do features include:
 - Global score: cumulative scores for consecutive games
 - Additional guidance not just higher or lower, we can add more range difference, e.g extremely high, abysmal, etc.
 - Range choice: The game can be made to support range chosen by the user
 - Attempt choice: The game can let the use choose the number of attempts they can have
 - Multiplayer support: the program can be edited so that one player will type a number, and the other will guess it.
5. Conclusion:

The program is my attempt to utilize MIPS assembly to make a project that I find interesting myself. I hope that you, and other users, will also enjoy it! Thank you so much for the course, and I hope to meet you in another course in the future.