Name: Yahya Alkendi

Date: 06/10/2021

CSCI 4350: Computer Architecture

Project 1: Guess the Number Game in MIPS Assembly

Both the MIPS assembly program and the C program have the same structure and approach in implementing Guess the Number game.

The program is structured in two functions excluding the main function which are:

* generate\_random
  + Arguments: int n - The upper limit of the random number generator range
  + Return: The random integer value generated.
  + It includes seeding the generator using the syscall 30 to get the time and syscall 40 for seeding.
  + Uses stack pointer for allocating space for the used variables in the register.
* Compare
  + Arguments: int x, int y
    - X represent the guessed number.
    - Y represent the random number.
  + Return: returns -1 if (x < y), 0 if (x == y), and 1 if (x > y).

The MIPS program follows the rules below:

* Uses stack to store and load values of register in function calls if needed.
* A pair of JAL and JR instructions for function calls
* Uses branches for the loops and if statements.
* Use $a0 and $a1 for the function arguments and $v0 for the return value.

The MIPS program works in Mars 4.5 simulator. Some notes for running the program:

* The program does not implement inputs validation solution therefore please enter the correct inputs as the program prompts.
* Please do not enter an empty input. This will cause an error.
* The program shows the random number for testing and grading purposes.
* The program asks the user the following “You found the number! Play again? (Y/N)” Please note that the input character is case sensitive, it must be in capital case. Again, the program does not validate the inputs.