CS250 - Computer Organization and Architecture

A0 - Simulation of MIPS Assembly in SPIM

Points to Note

- Write the MIPS-I assembly code for all the questions below. Simulate your code on `spim` (text based simulator) or `xspim` or `qtspim` (GUI based simulators).
- In the comments section of each MIPS assembly code, include a short description of the program, your name, roll number, date of writing the program and other information you deem relevant.
- Your submission is an archive containing the following: One README file + One directory per assignment question. This assignment (A0) has N questions (1,. . .,N) - your submission will be a zip file containing N directories and a README file. For each question (in each directory) include the following: MIPS code, snapshots of the simulator at key execution stages, and other relevant information.
- This is an individual assignment. One submission per student.
- (a) Submission is due on Jan, 24, end-of-day. Pack your report, code, screenshots and other files in an archive.

Write the MIPS assembly code for the following.

1. Load two 32b constants on to $t0 and $t1. Add them and store the result in $t2.
2. Load a 32b constant in the memory location pointed to by the global pointer. Load another 32b constant in the first memory location of the data segment. Add them both and save the sum in the 100^{th} word of the data segment.
3. **Hello World Program**. Print "Hello World" on the output screen.
4. Initialize a contiguous chunk of memory to contain 10 two digit decimal numbers. Your program should calculate the sum of these numbers and put the result into $v0.
5. Write a program to reverse a string (choose your favorite string as a global parameter).
6. Implement a function that calculates the sum of the elements of the array. The function accepts the length of the array and the address of the first element of the array. The function returns the sum to the caller. The main procedure calculates the average of the elements of the array. The main procedure uses the mentioned the function. Use the MIPS compiler subroutine conventions for this code.
7. Along the same lines as the previous question, implement a matrix multiplication program using functions.
8. **Factorial Program**. Load a random integer in $t0. Calculate its factorial using (a) loops, (b) recursion.
9. Write a program in MIPS assembly language to convert an ASCII number string containing positive and negative integer decimal strings, to an integer. Your program should expect register $a0 to hold the address of a null-terminated string containing some combination of the digits 0 through 9. Your program should compute the integer value equivalent to this string of digits, then place the number in register $v0 . If a non-digit character appears anywhere in the string, your program should stop with the value -1 in register $v0 . For example, if register $a0 points to a sequence of three bytes 50_{ten}, 52_{ten}, 0_{ten} (the null-terminated string "24"), then when the program stops, register $v0 should contain the value 24_{ten}.
10. **Fibonacci Series**. Input: user provided command line argument, a random positive integer, say n. Calculate the n^{th}, and the next number in the Fibonacci sequence using (a) loops, (b) recursion. The two outputs should be in registers $v0, and $v1.