CPEG-323: Introduction to Computer Systems Engineering

Machine Problem #2: C to MIPS translation (100 points)

You may work alone, or in groups of up to two.

Due on Monday October 15 (11:59:59pm)

This MP will give you an introduction to MIPS programming, using code that you have already written and/or used. You will learn to use bit-wise operators in MIPS, arrays in MIPS, and MIPS control flow. It will also give you a chance to familiarize yourself with the MIPS calling conventions.

Please go through the provided SPIM Tutorial carefully to get you acquainted with the SPIM simulator.

Problems

In this MP you will be:

- Translating the ***singleton*** and ***get_singleton*** functions from MP1 To MIPS (50 points). The ***singleton*** method only requires "if" control flow, the ***get_singleton*** method adds loop.
- Translating the ***board_done*** and ***print_board*** functions from MP1 to MIPS (50 points). These functions require: 1) implementing doubly nested loops, 2) indexing into a two-dimensional array, and 3) saving/restoring registers and calling other functions. You will note that the two functions are very similar, so we recommend writing one and completely debugging it and then using that as a template for the second one.
  Some hints/suggestions/information:
  - You should hard code GRIDSIZE and GRID_SQUARED as 3 and 9, respectively.
  - Two dimensional arrays in C are laid out contiguously in memory like a singledimension array; for an M*N array (int A[M][N]) the locations are laid out in the following order: A[0][0], A[0][1], A[0][2], …, A[0][N-1], A[1][0],A[1][1],…, A[1][N-1],…,A[M-1][N-1]. The address of element A[i][j] is:
    the address of A[0][0] + (((i*N)+j)*sizeof(element))
    The two dimensional arrays that you are dealing with are fixed size (9*9) matrices (i.e., int board[9][9]) and of integers, so N is the constant 9 and sizeof(element) is 4.
  - Use the $s[0-7] registers! If you store some of the $s registers on the stack at the beginning of a function call you can use these registers without worrying about any callee corrupting the values.
  - The not instruction does a bit-wise not (which is the C operator "~"), not a logical not (which is the C operator "!").

sudoku.s contains the main program for testing your functions. It contains two invocations of each of ***singleton***, ***get_singleton***, and ***board_done*** and a test case for ***print_board***. We encourage you to come up with other tests to test your code thoroughly.

Submission

Please email the following files to cpeg323.udel@gmail.com.

- Modified code: sudoku.s.
- Partners.txt containing the names of everyone in your group.