GEORGIA STATE UNIVERSITY

Department of Computer Science

CSC 3210 Computer Organization Programming

Lab Section:

Lab 3: Control Flow and Array Manipulation in Assembly

Linh H. Pham

Submitted: October 18, 2024

Contents

1 Introduction

This lab aims to explore the usage of array, heap, and stack in RISC-V assembly.

- The first task asks for the implementation of the Insertion Sort algorithm. This Task requires understanding of array manipulation.
- The second task asks for the implementation of a recursion for the ToH problem. To understand recursion, is to understand the working of stack and heap in RISC-V assembly.

2 Apparatus

- SNOWBALL server
- RISC-V on VS Code
- Code templates
- RISC-V reference card

3 Methods

3.1 Task 1

Run the insertion_sort.c file in the Snowball server. This is to show how the Insertion Sort algorithm works in the C language.

Convert the algorithm to RISC-V and run three sperate cases.

- A sorted array

- A reversed sorted array that also has negative integers
- An unsorted array with very large integer,

These cases show how RISC_V interacts with array and its elements. It also shows limitation of the memory and how to circumvent it.

3.2 Task 2

This task asks you to demonstrate your understanding of the ToH problem and solve it using recursion, in RISC-V assembly. Also, the task wants to implement the algorithm using different numbers of disks and to find out the largest number of disks the code can handle. Heap and stack are required for recursion. Stack is where the functions are stored and executed in a LIFO order. Heap is where variables are allocated.

4 Results and Discussion

In Task 1, three arrays are sorted successfully using insertion sort algorithm in RISC-V, but there are some variances in the results. Depending on the type of input array, the time complexity can go from O(n) to O(n^2).

In Task 2, the ToH recursive solution in RISC-V can solve the cases of 3,4 and 7 disks. If the number of disks is increased, the time to solve the problem increases exponentially.

5 Conclusions

This lab successfully demonstrates the application of array, heap, and stack in RISC-V assembly. Different input arrays will affect the time of the algorithm to produce outputs. A large input in a recursion can create problems for the machine. Therefore, it is important to understand the size of the data structure that you are working with to choose the best approach.