CSE31 : Lab #6 – C/MIPS

We will start look at the assembly code closely this week.

Overview

We will see how to write code and fix it in assembly. Also how a program in C gets translated by compiler into assembly and whether you can understand the code generated by that process.

(Exercise) Debugging MIPS

Debug the loop written in arrcp.s. The program is suppose to copy integers from memory address in $a0 to memory address in $a1, until it reads a zero value. The number of integers copied (up to but not including the zero value) should be returned so stored into $v0.

Q1. How many bugs are there?

Q2. How do you fix the bug(s)?

Q3. What is your strategy to finding the bug(s)?

Fix the code so it works in arrcp.s.

(Exercise) Compiled C → MIPS

This exercise contains a function that does the same array copy functionality. However, now we wrote the code in C and used a cross compiler to automatically generate the MIPS code. You will find the original C code in arrcopy.c and auto-generated assembly in arrcopy.s. Now look in arrcopy.s to answer the following:

Q4. Where is the `source` pointer stored originally?

Q5. Where is the `dest` pointer stored originally?

Q6. What instruction is used to load the address of `source` and `dest` pointers?

Q7. Where does the loop to copy values start? (give line # and the first instruction and/or label of where it is)

Q8. Explain what each line in the loop is trying to do in the following format:

Instruction : add $4, $0, $0 (as an example)
Purpose : to do nothing
`Corresponding C : x = 0;`

What to hand in

When you are done with this lab assignment, you are ready to submit your work. Make sure you have done the following ***before*** you press Submit:

- Answers **Q1-Q8**.
- Attach fixed arrcp.s
- List of collaborators
