Tony Ngo – CSC 3210 Project A1 Task 4

Task A: Connecting to GitHub via SSH

My personal experience while trying to install GitHub onto the Raspberry Pi was seamless except that my Pi would not connect to SSH at first, but you can change that in the settings.

Task B: ARM Assembly Programming

Part 1: First Program

A close up of text on a black background

Description automatically generated Picture 1: Assembly, Link, Run, & Beginning of debugging

A close up of text on a black background

Description automatically generated Picture 2: Debugging of First program

Part 2: Arithmetic Program

A close up of a sign

Description automatically generated Picture 1: Assembly, Link, Run, & Beginning of debugging

A close up of text on a black background

Description automatically generatedPicture 2: Debugging of Arithmetic1

Part 1: First Program Explanation

The first part of this assignment was program was made simply just to add and subtract ARM code as an example. The command “as” assembles the code that you wrote, while the command “ld” is the linker and creates the executable of the file. Adding “-g” in the code assembles the code and also allows for the GDB debugger to be used. The GDB Debugger serves as a way to check where specifically your code has bugs, but in our case we want to see what values were added to each register. We used the command “b <number>” to see specifically when we wanted to see the state of each register and “info registers” to see the registers themselves. In our example, we used “b 11” to see what our registers would look like post-operations and “info registers” to execute, which in our example register r2 has 8 because 5 + 4 – 1 = 8.

I did not have difficulties running the original “first” program. I believe that running “./first” doesn’t have any output because we are only adding values to each register, and the output itself does not show the register. I was able to run the debugger seamlessly without any error and register r1 has the correct number in the entry (5 - 1 + 4 = 8). When I did “list”, I did not press enter so it only shows 10 lines.

Part 2: Arithmetic Program Explanation

The second part of this assignment we were supposed to write a program that had the arithmetic sequence A = (A + B) – (C \* D) where A = 10, B = 11, C = 7, and D = 2. I assembled the code using “as”, linked the code and created the .exe using “ld” and re-assembled the code for debugging in GDB uses using “as -g” vs “as”. I ran the debugger using “gdb arithmetic1” and set the breakpoint to be at “b 15” because that was after I had completed my operations. I was able to see that my registers were set correct because the arithmetic was right.

While writing the code, I did experience some difficulties but that was due to user error. When I originally compiled this code, I had sub “r1, r4” but that would be incorrect since the value of (C \* D) is stored in C (r3 in this case), so the correct line would be “sub r1, r3”, which is seen in the screenshot.

That error caused me to run into issues while I was trying to debug it, because if you just reassemble the line “as -o Assembly1.o Assembly1.s”, it will not affect the line in the debugger, so I had to reassemble “as -g -o Assembly1.o Assembly1.s” to get the correct output in r1, which is 7.