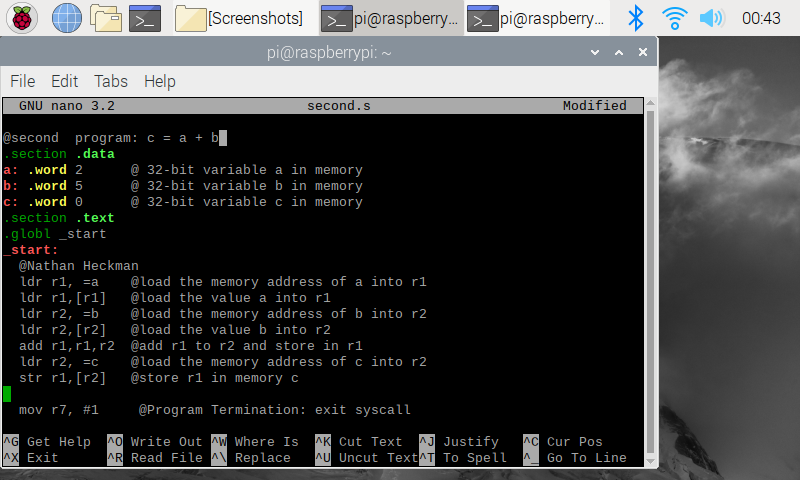
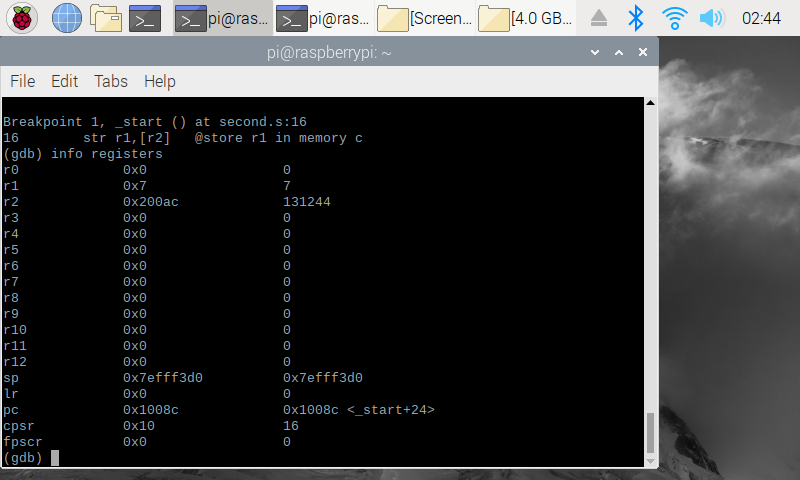
Nathan Heckman

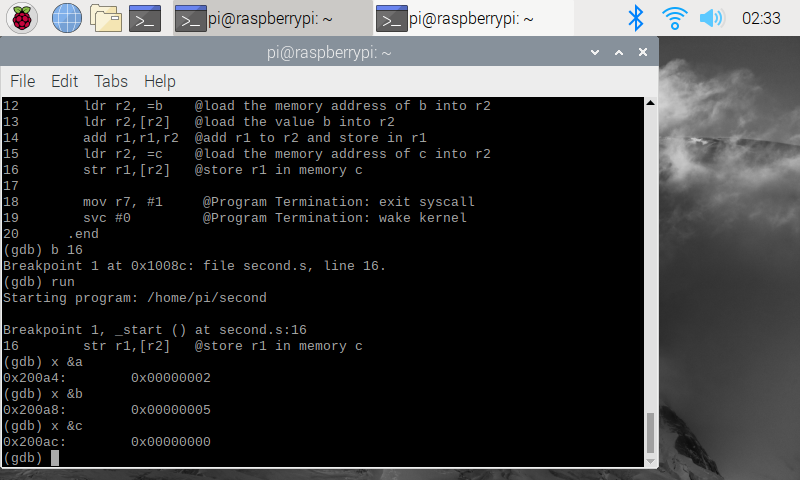
A2 ARM Assembly Report

**Part 1 - Question 3 Program**

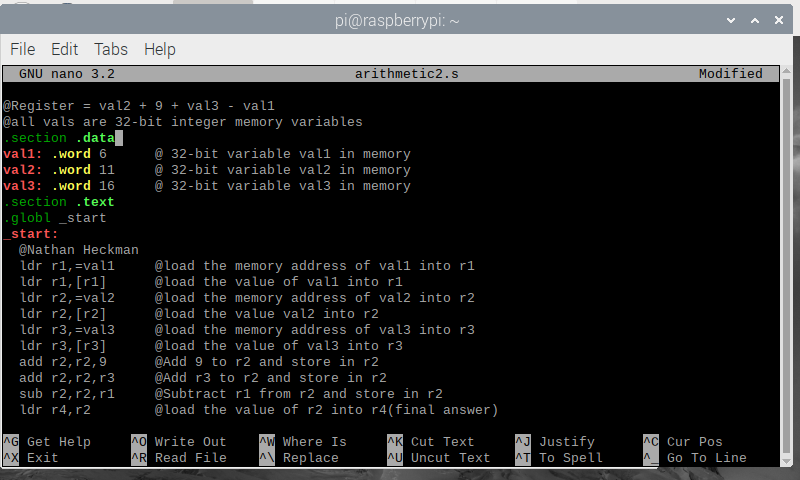
The program is saved in second.s and then taken to the command line in order to complete the debug process and view info registers. While coding this program I also made myself more familiar with the ARM syntax and how to access values inside of the memory address.

**Part 1 – Question 3 Info Registers **

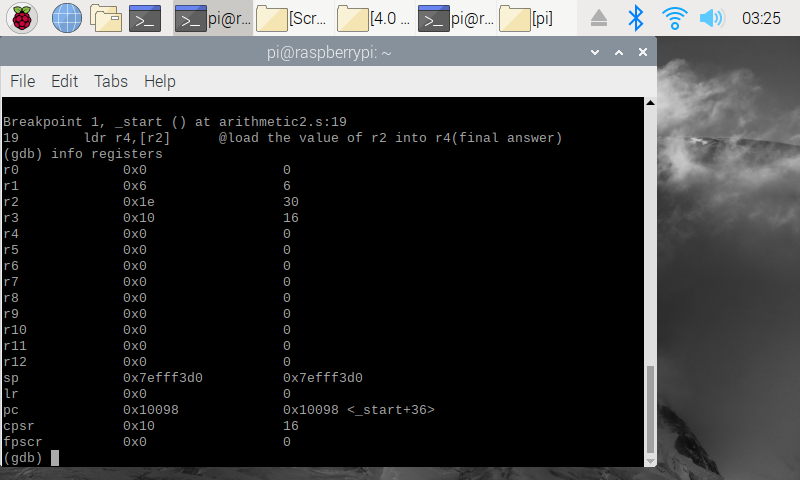
Looking at the various registers, r1 has been updated with the correct value of 7 and r2 has stored the memory address of variable c.

**Part 1 – Question 3 Variable Memory**

Shows setting the breakpoint and then running the program. While figuring out how to access the variable memory I figured out that you can search by the variable name in the same way that you can in x86 assembly language. Each variable has correctly been assigned its 32-bit integer value initialized in the .data.

**Part 2 – Program Code**

The program is structured similarly to the previous program. Each val variable is assigned as a word so they fit the 32-bit size requirement. After I took this screenshot, I noticed that the integer immediate 9 was causing the program to not assemble. Adding a # to the front of it fixed that issue. The final ldr command was also an issue and I had to add square brackets to r2.

**Part 2 – Program Info Registers**

After making the fixes, the program produces the desired output of 30 in the r2 register. I decided to remove the code that stored the final answer in r4 and instead simply leave it in r2, since all of the operations were already completed.