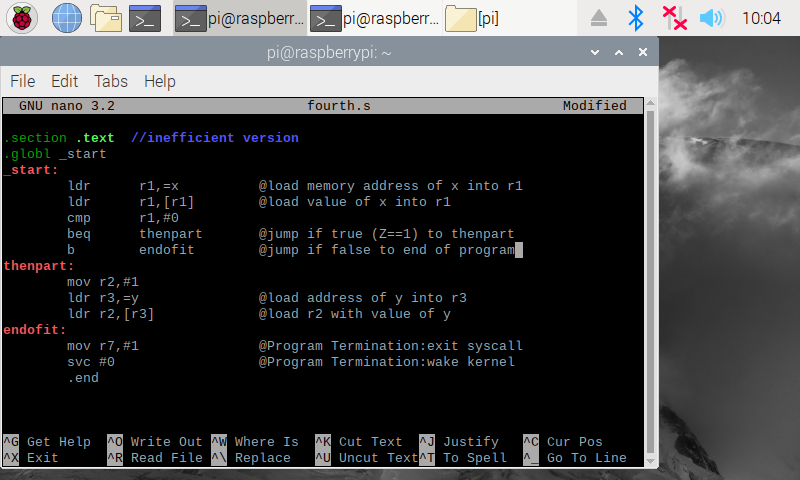
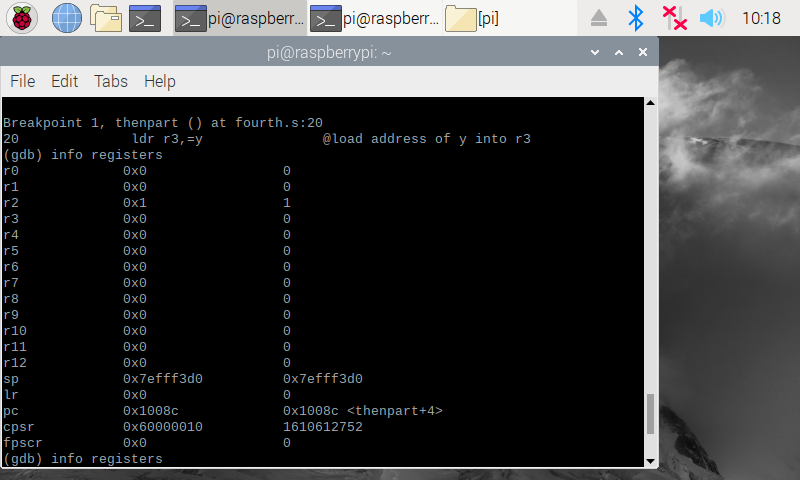
Nathan Heckman

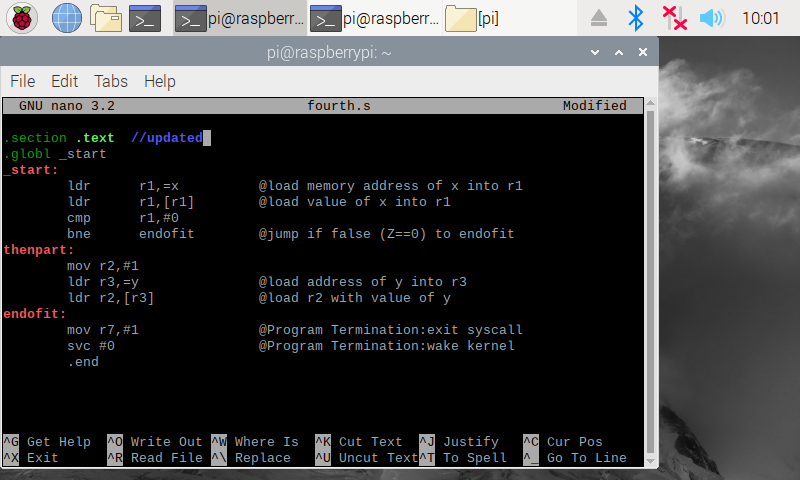
A3 ARM Assembly Report

**Part 1 – Inefficient fourth.s Code**

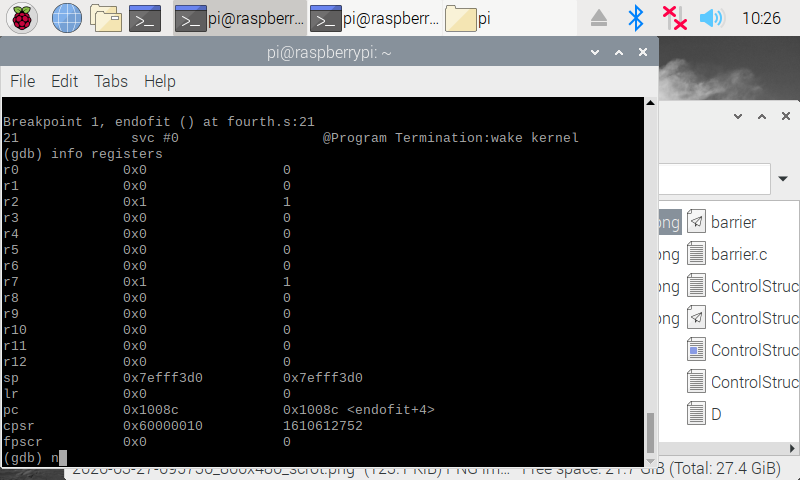
This is the first version of the fourth.s code, with two loops right after one another. It’s not efficient to do this because branches may cause a delay slot. The rest of the code is the same as the corrected version, detailed below.

**Part 1 – Inefficient fourth.s Code Registers**

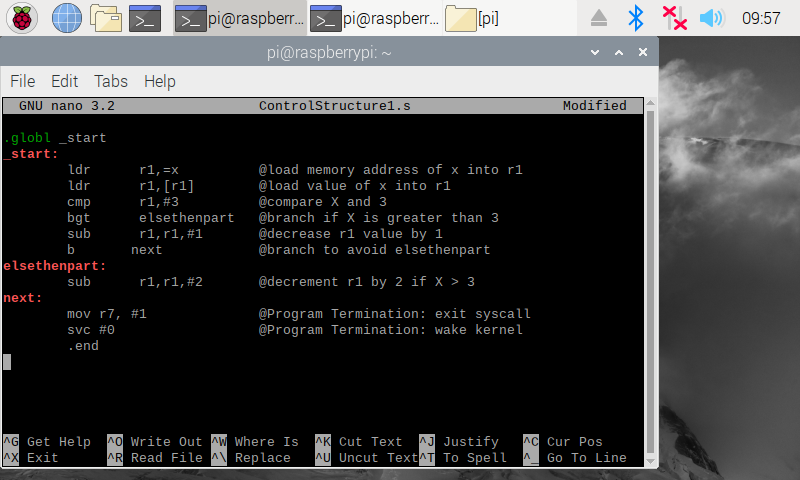
The inefficient code produces the desired r2 value of 1, but could be better. The red box highlights the zero flag in the Current Program Status Register(cpsr). This shows the zero flag as 1 since r1 was set to zero at the beginning of the program and is never changed. We can also see that the desired output of 1 in the r2 register was given.

**Part 2 – Updated fourth.s Code**

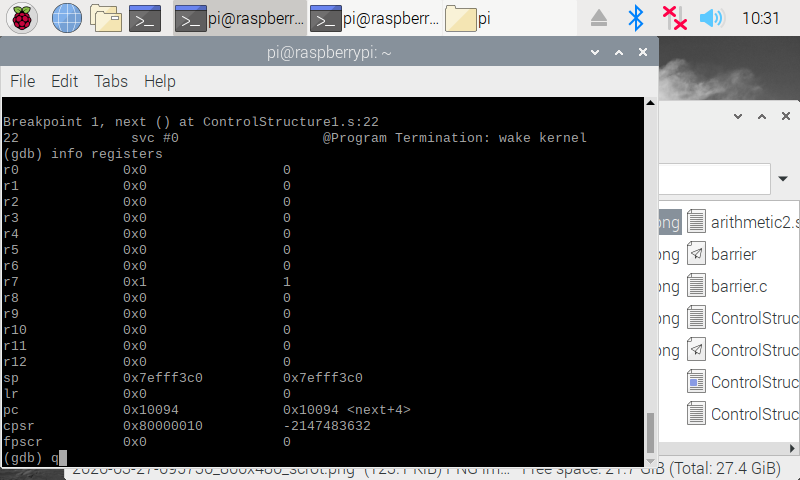
The new fourth.s code only requires one loop to execute the program properly since we are now branching if the statement is false rather than true. The rest of the program is the same as the inefficient one, so now we can view the registers and output of the program.

**Part 2 – Updated fourth.s Registers**

Like before, the zero flag is still 1 and the value of r2 is the same since we only changed how the program branches, not its actual operation or output values.

**Part 3 – ControlStructure1.s Code**

Since this program has two separate conditions that could be executed (one in the if portion and one in the else portion), two branches are required. One branch avoids elsethenpart when X is greater than 3, and the other jumps to the end of the program when X is less than or equal to 3 after executing the subtraction of 1 from r1.

**Part 3 – ControlStructure1.s Registers**

All of the registers show zero because X started as 1 and was less than or equal to 3. This caused the program to subtract one from r1, which held the X value. The program then skipped over the elsethenpart and terminated. The zero flag is again set, as the operation on r1 caused its value to equal 0.