Project A3

Tek Acharya

900758663

GSU-Spring 2020

ARM\_Assembly\_Programming\_A3

**TASK 4a**

Edited the code into the compiler.

Found that the code has the error while trying to assemble it.

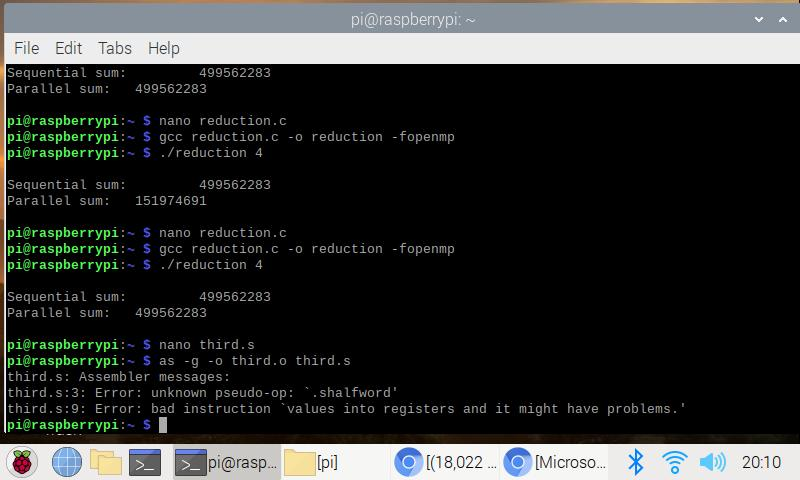
* What error message did you get and why? (Report that)

Answer: Error message: unknown pseudo-op: ‘.shalfword’

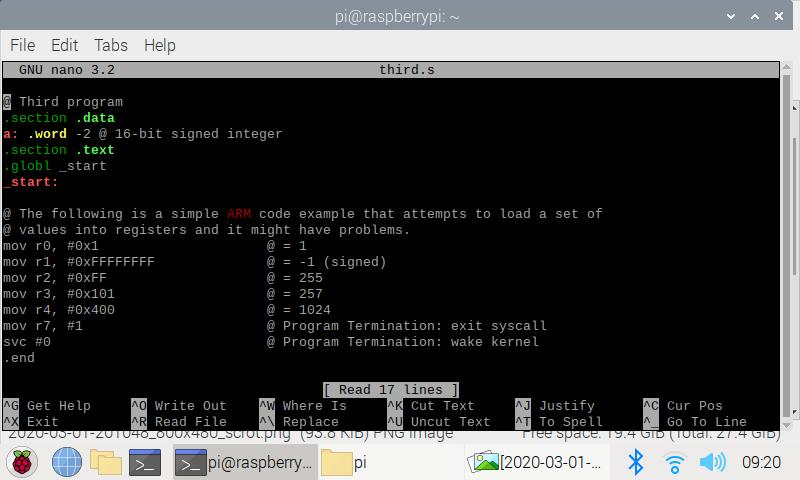
Bad instruction ‘values into the registers and it might have problems.’

Why?

.shalfword is not a valid data declaration. It only takes .word as a data declaration.

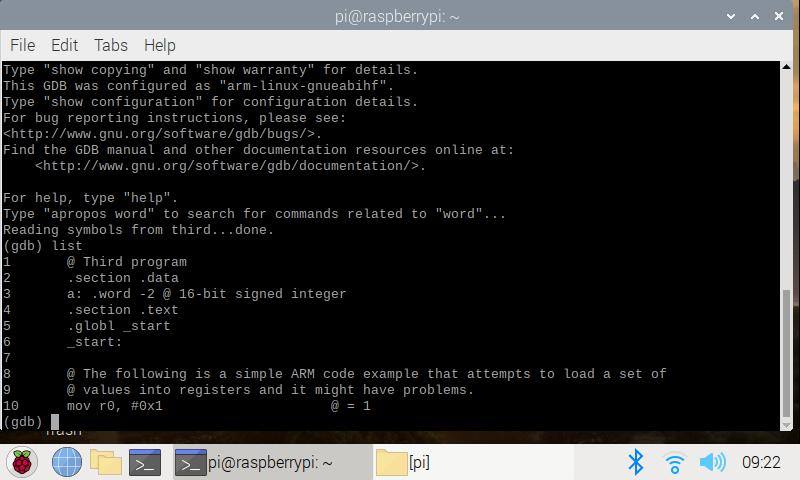


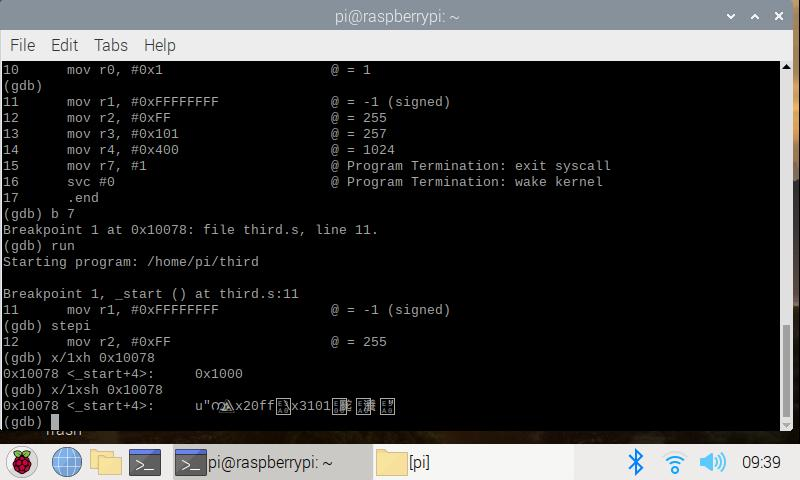
Error noticed.



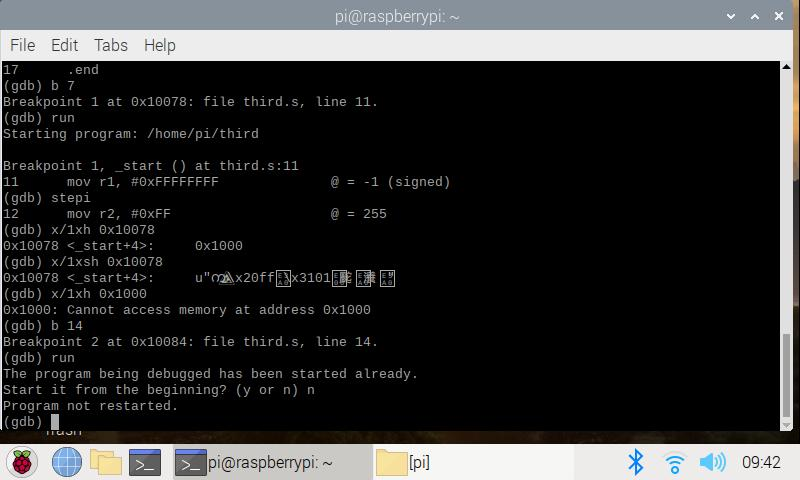
Error corrected with ‘.word’ data type declaration.

After correction, the code is reassembled, linked, and ran with a debugger as shown below.

Breakpoint created at line 7 but the it was automatically moved to line 10 by the gdb. Examined into the registers or memory using (gdb) x/1xh 0x10078



With sh(signed), instead of h(unsigned) got into the same upset, however, an unknown address is generated that I am unable to understand. screenshots of observation is shown above and below.

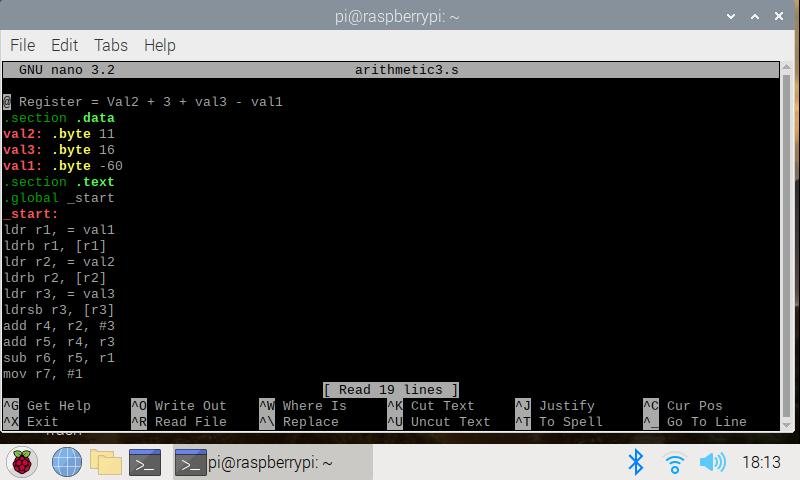
Tried sh instead of h and observed the changes inside the registers/memory. Also tried using different offset address but couldn’t understand what’s going on.

**Task 4b**

Referring to third.s as a base, and with the algorithm

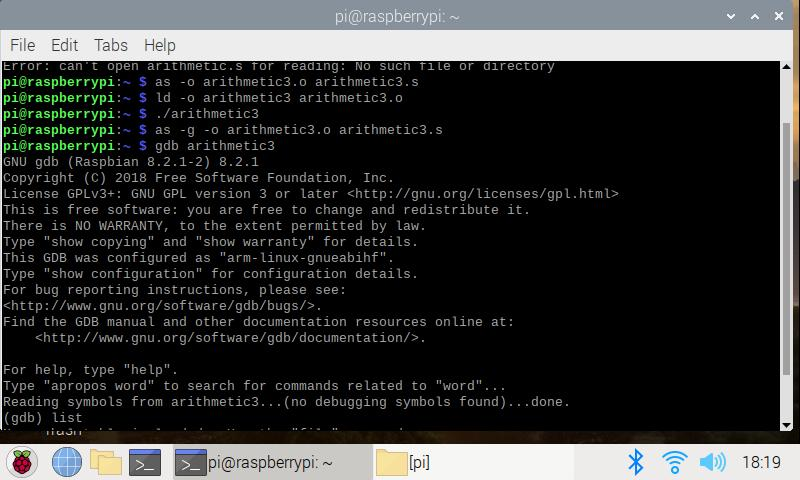
Register = val2 + 3 + val3 – val1; val1 = -60(signed), val2 = 11(unsigned), and val3 = 16(unsigned) and all these memories are 8-bit integer memory

we worked together and created the following ARM assembly code with the name arthmetic3.s

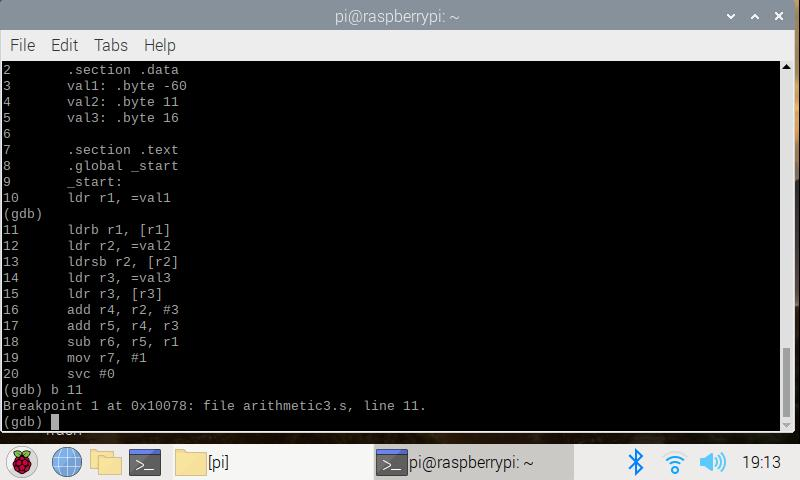


The code compiled

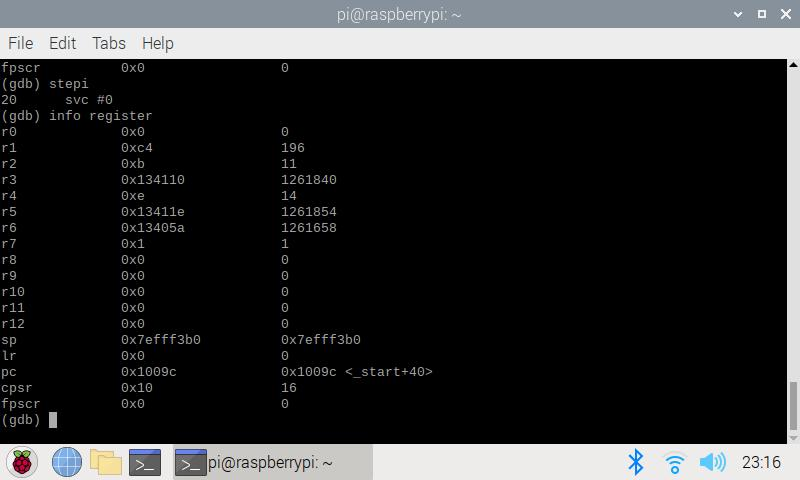
Then we assembled, linked and ran the code as following



The code assembled, linked, debugged, created a breakpoint at line 10, and ran



Debugged with breakpoint at 10



The assigned arithmetic to be computed with their assigned value is: 11+3+16-(-60) = 90.

Our

r1 = C4h which corresponds to -60 (11000100) that is our Val1

r2 = 0Bh which corresponds to 11 that is our val2

r3 = 10h which corresponds to 16 that is our val3

similarly

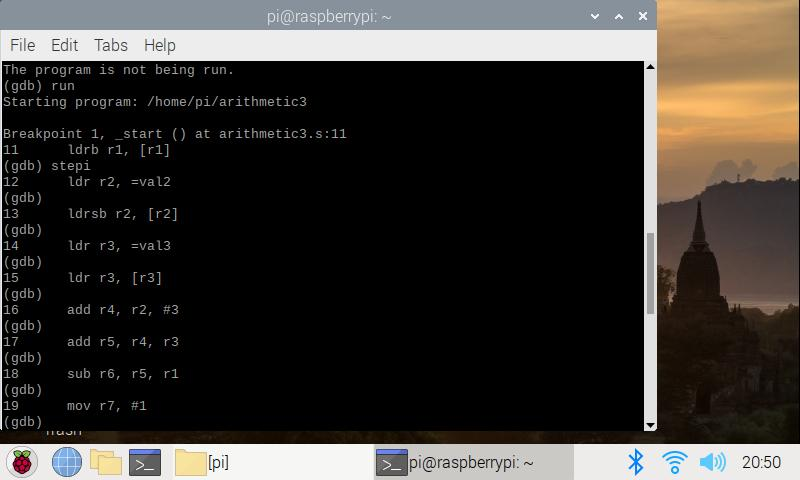
r4 = 14

r5 = 30

r6 = 90 (5a) which is our final answer of the “Register”

all these final and intermediate arithmetic values are as expected.

Upon checking the ‘cpsr’ value is 0x10 which tells us that the negative flag is set as we worked with signed value in ARM.



We went further to learn more with the stepi and check some details of the values in memory and registers of our program.

Stepped in into each register and studied the value of registers