\*Note to Report writer-this is not formatted so format accordingly\*

Assignment 4 Task 4

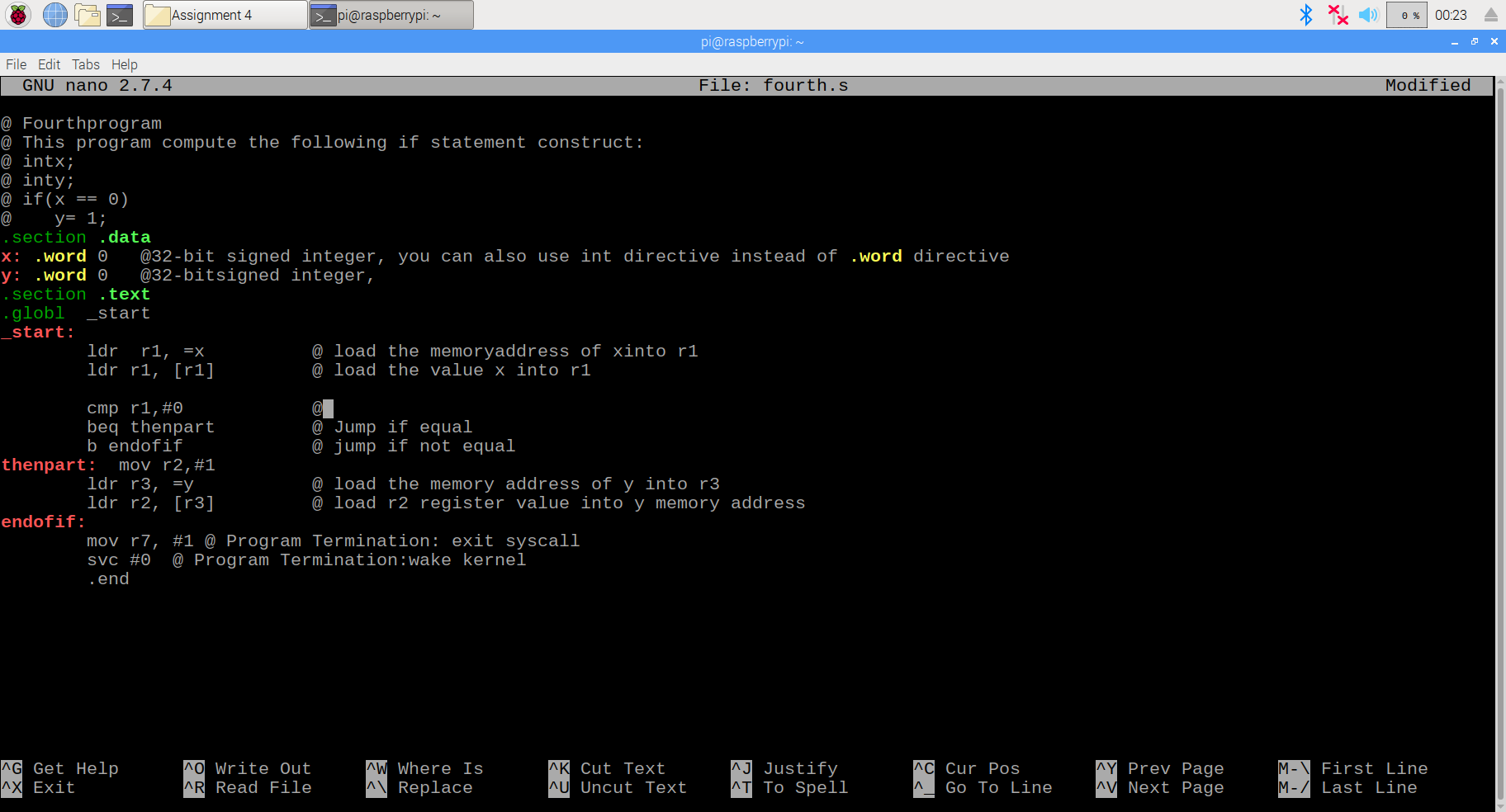
Johnathon Moore

My assignment was the Arm Assembly which I already had experience with when I handled parallel programing in a past assignment.

3/13 Received the PI

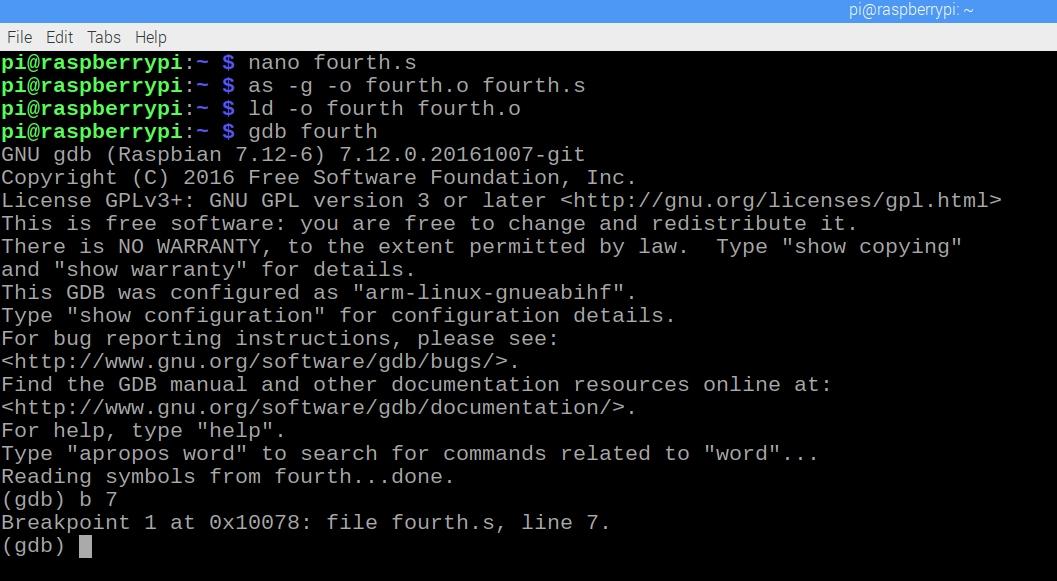
3/23 Because of the extended time of the break I had extra time till my assignment was due But I finally got around to it. Programming begins.

Question 1) I began by copying the given code into a program file named fourth.s



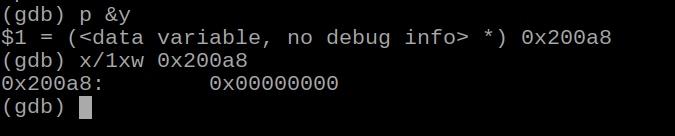
I noticed some issues which would not allow my program to be assembled correctly so I corrected them in the code like adding a “#” before 1 in the line that begins with “thenpart:”

After correcting the mistakes I assembled the program and began the debugging section

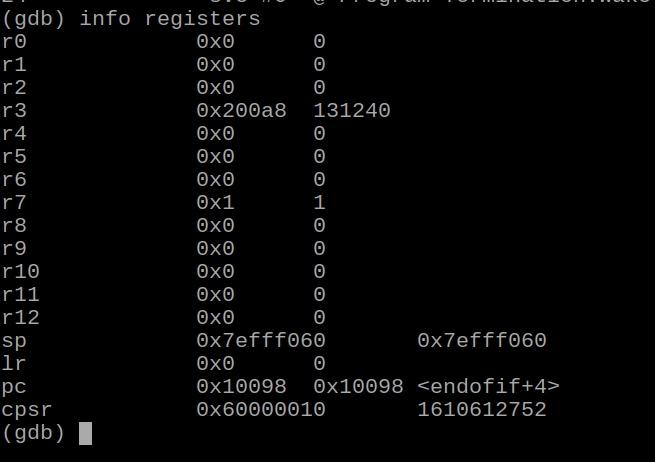


After seeing up the breakpoint and running the program, I stepped through the program far enough to see the affected values.

First the Y (using &y to find the address and then x/1xw address to retrieve the value)



Then I locate the value of the Z-flag by using “info registers” and determining it through the cpsr register

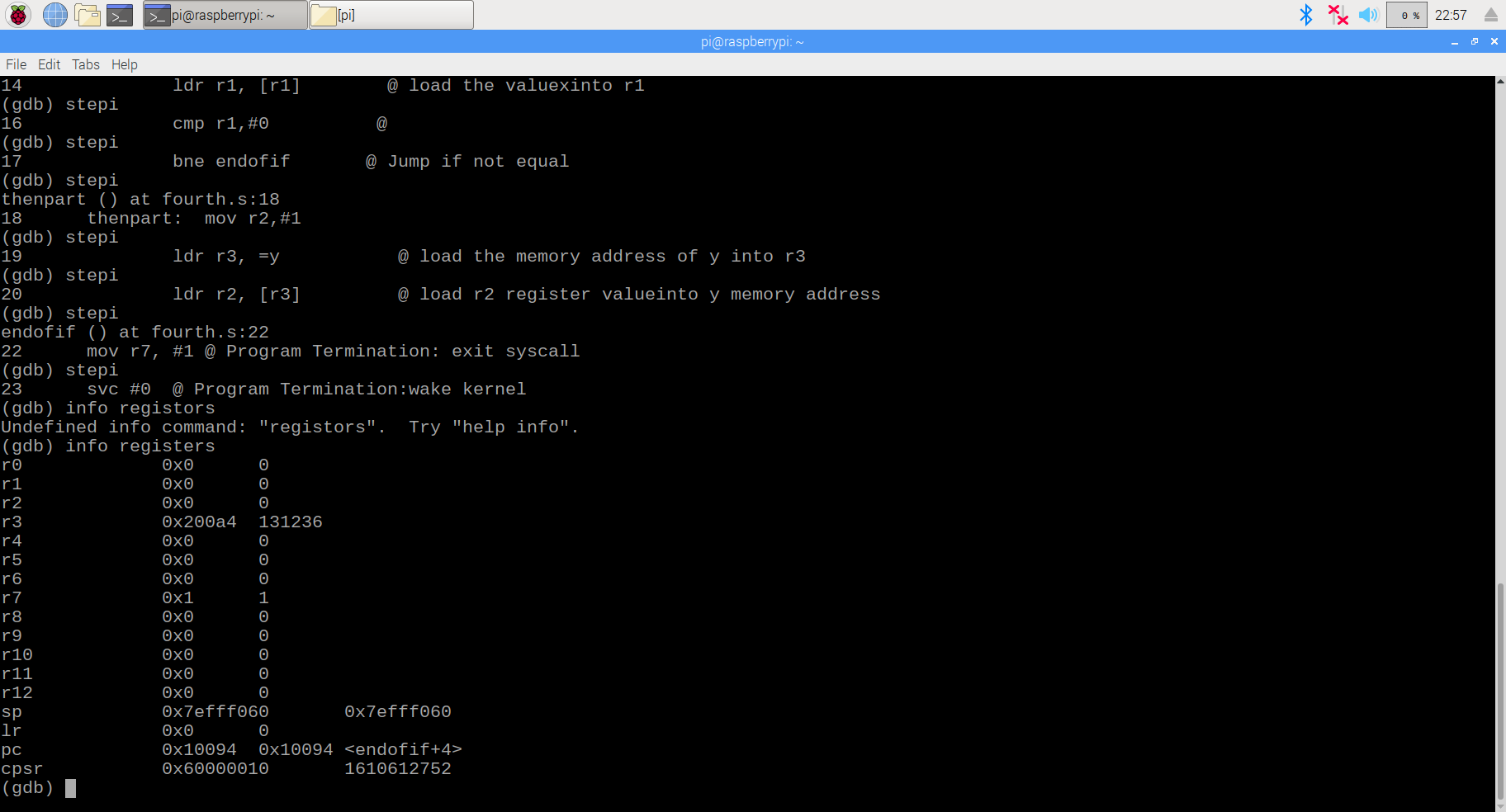


This process was a bit confusing but I learned that cpsr is the register that holds flag values. To find the Z-flag we convert the “60000010” from hex to binary getting 0110 0000 0000… (The rest does not matter for this assignment). We can use prior knowledge to tell that the second binary number from the right represents the status of the Z-Flag. The Z-flag is 1 which is what we are looking for.

2) Question 2 - Again I am assuming there was an error in the instructions as it tells us to replace beq with “bnq” however it appears to be referencing “bne” so I used that instead.

Other that that I just removed the the line b endofit and replace the thenpart to endpart.

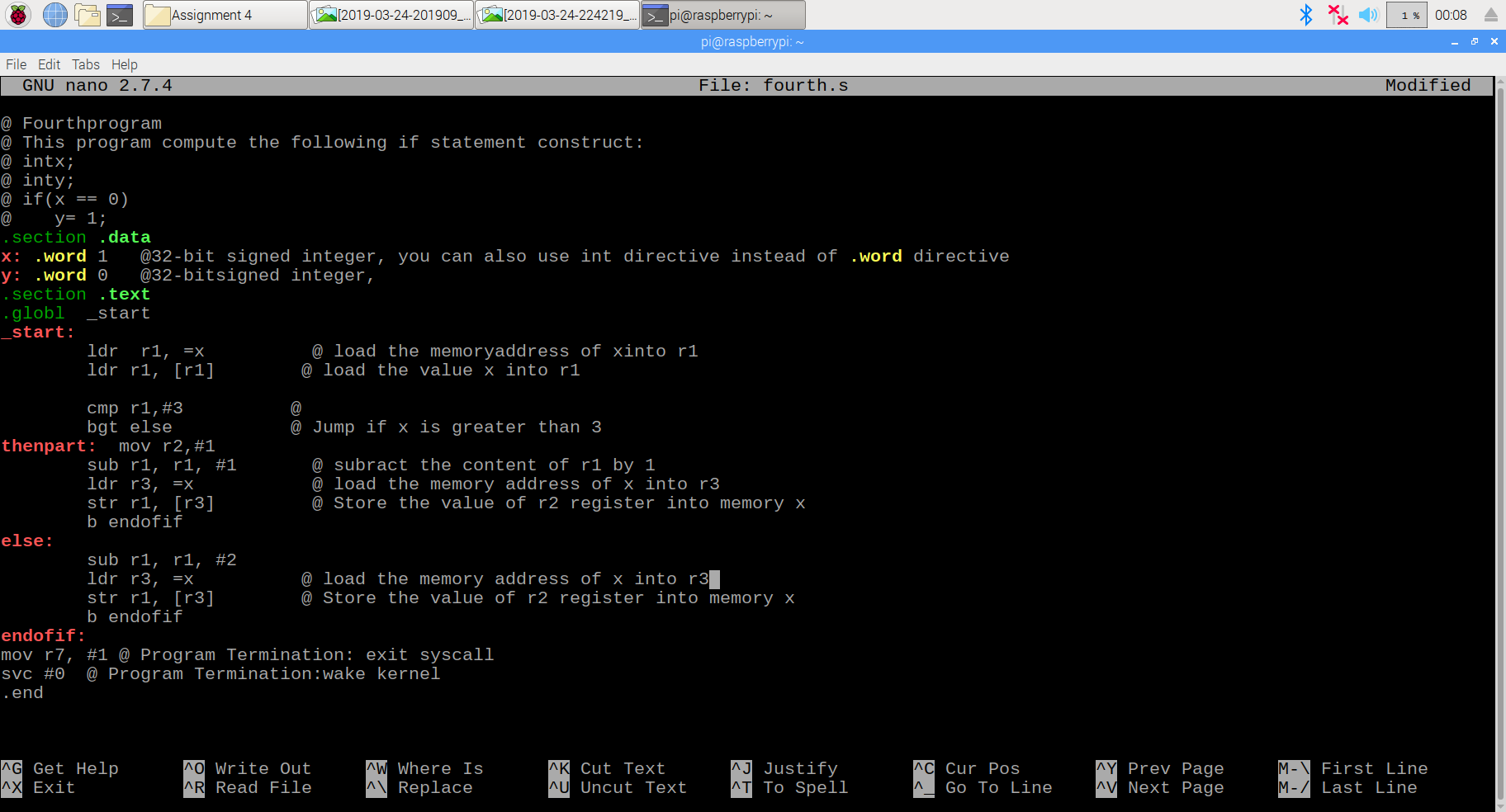
I then assembled the code, ran it through debugging and found the Z-flag through the cpsr just like before.



Just as before, I found cpsr and translated the value given (60...) to binary (01100000…)

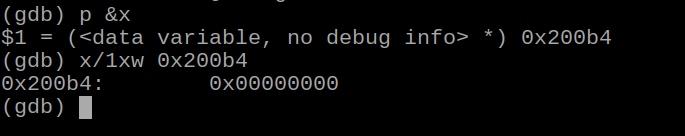
Finding that the Z-flag is set just as expected.

Question 3) - Next I had to edit fourth program to make a specific conditional statement that was specified, the issue here was deciding what conditional branch jump commands that I needed for the particular problem. Here is the code:

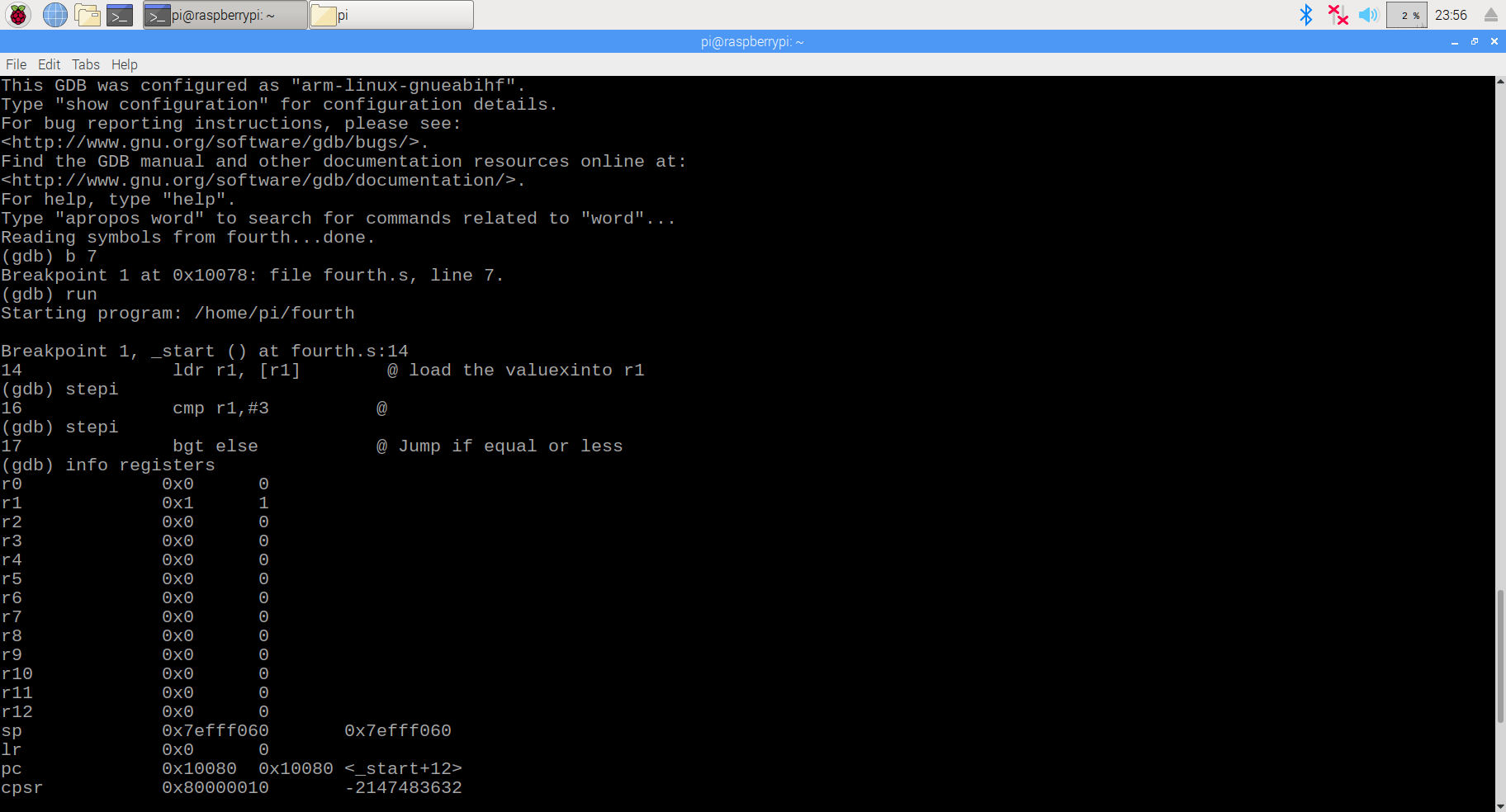


In this code I replace bne with bgt (which jumps to \_\_\_\_ when X is greater than 3) and had it jump to else, thus skipping thenpart. We were asked to find “if x<=3” but it was simpler to use the inverse statement (x>3) to skip to the else statement. I also added a line “b endofif” which skips the else statement and goes straight to the end of the code thus making it impossible to have both if and else happen at the same time.

I found the X value by:



And the Z-flag, the same as before by:



These values matched with what we were looking for using the given x=1. The Z-flag is set to 0 which means that 1-3!=0 which is true. The x value is 0 as 1-3=-2 which is Less than 3 so we would subtract x-1 which is 1-1=0 which is what we got!