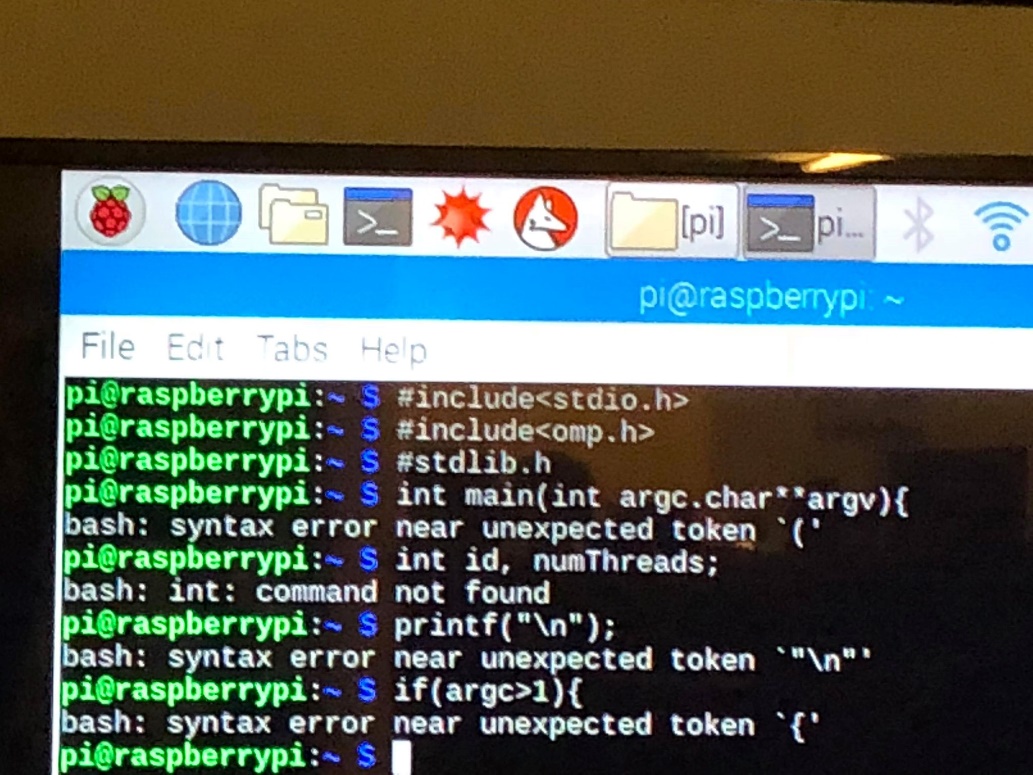
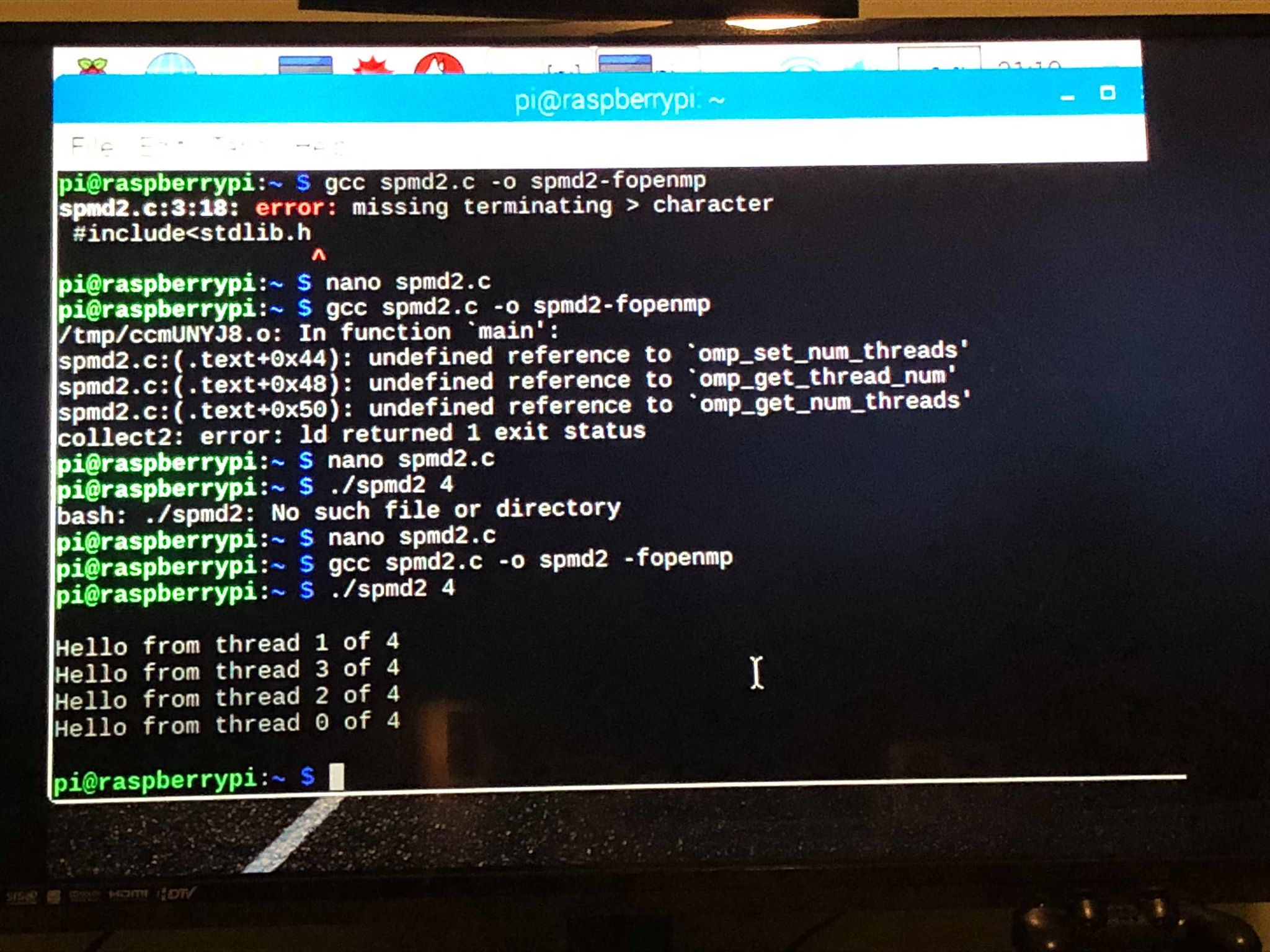
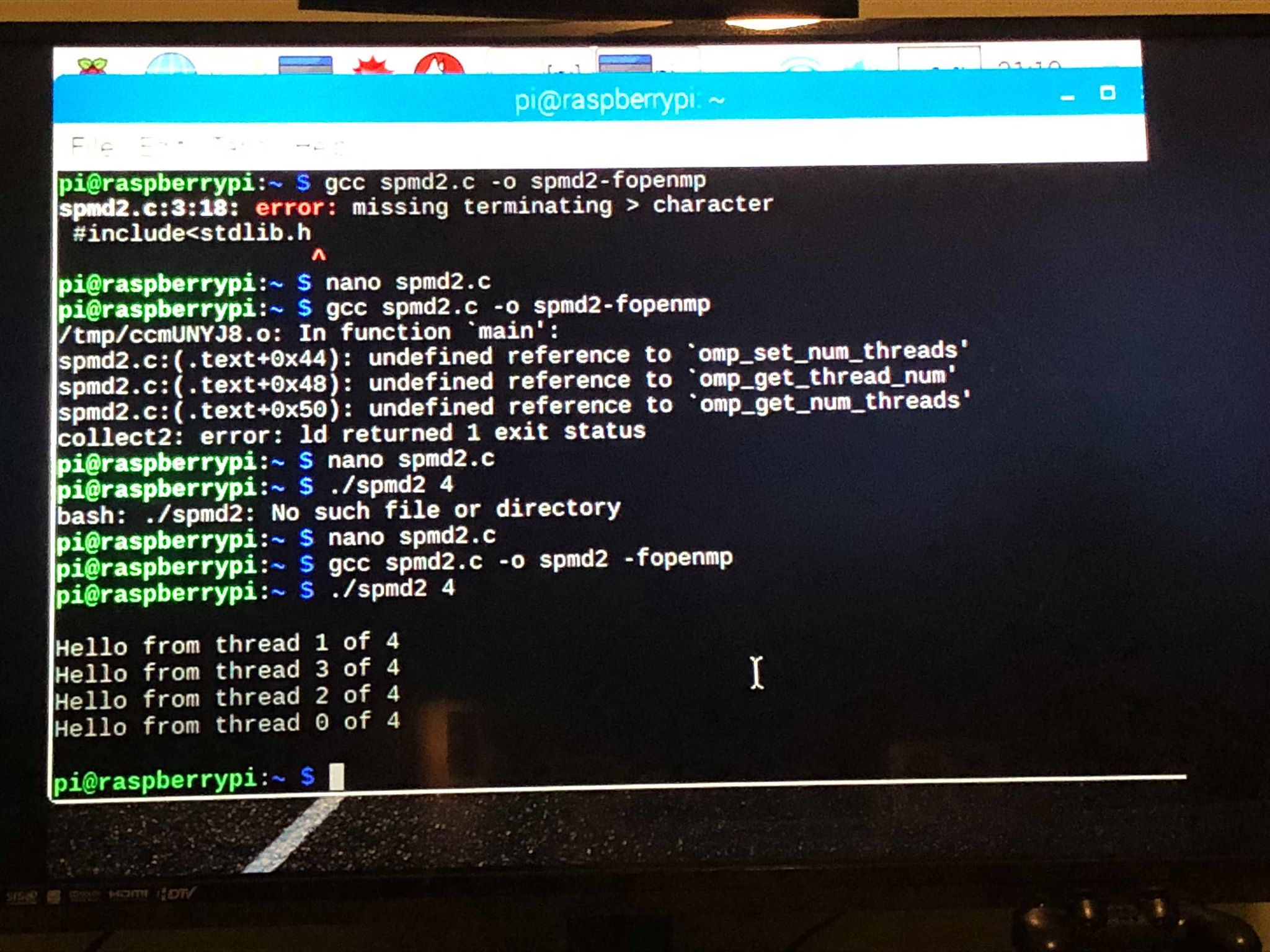
Dream Team Raspberry Pi Report  
  
What I have done was start understanding how the Raspberry PI works. Unlike some programming applications, you cannot run a program from the start, you would need to use the Editor known as “Nano.” Something I tested out during the task from the start was “What would Happen if I began typing in the code outside the Editor.”



Uninformed about my current situation, I found it interesting that I would constantly get an error after every inputted line. This was until I thoroughly read through and understood the editor, “nano spmd2.c”. Upon entering the code using the editor I was met with 2 errors at first, one being a missing terminating character,

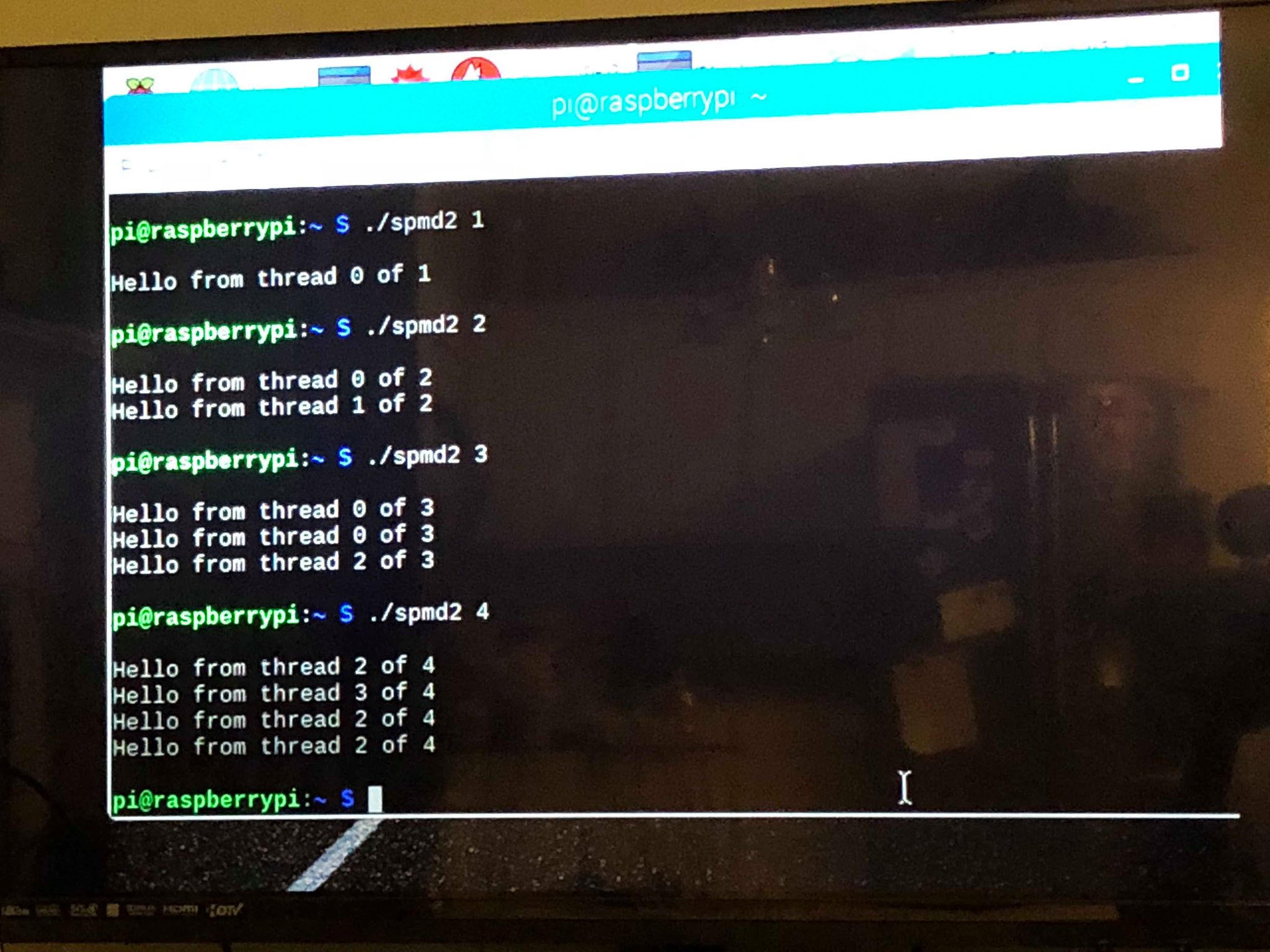


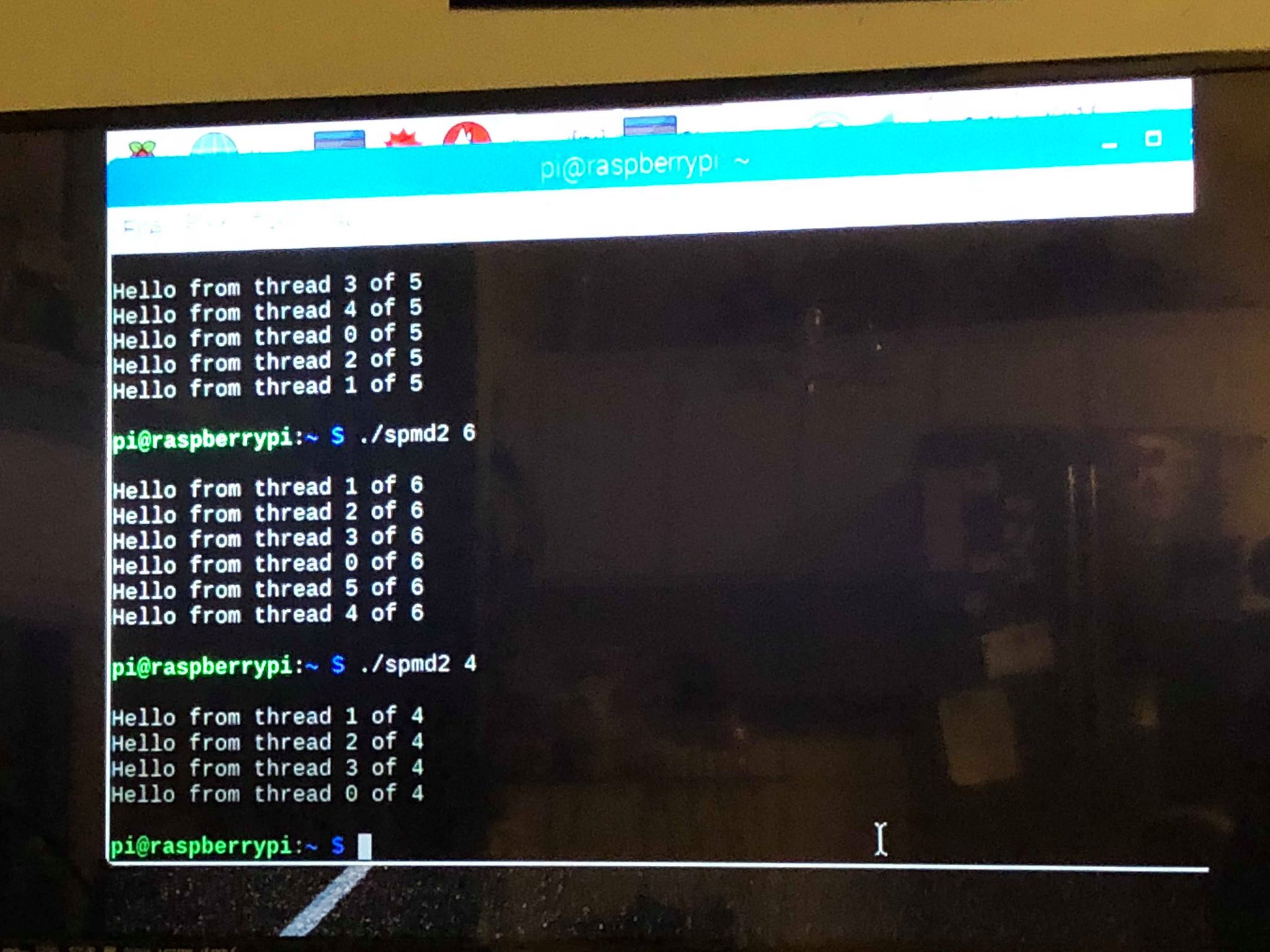
And undefined references.



While the solution to the terminating character was as simple as adding the “>” character, I had to progress further through the task to figure out the undefined references error. What I particularly found interesting was after applying the steps in section 2.4.1, “ //int id, NumThreads;” running the program, and then removing the “//,” the error ceased and the program continued running regularly. Moving onto running the program itself, I observed the code segment in charge of making the executable program, “ gcc spmd2.c -o spmd2 -fopenmp” as well as the code segment in charge of running the program, “ ./spmd2 4.” If the code segment has any lingering errors, the executable program will not be created, and therefore cannot run.

After running the code several times with 4 threads as instructed, as well as running it using a different number of threads. I noticed that while order was not only the issue, but a thread number appears more than once, rather than having its own number.



This was due to the variable being declared outside the block, causing all the threads to share the same memory and therefore ID. The tasked then showed me how to fix this issue, by adding a “//” in front of “id” and “numThreads” as well as adding an “int” in front of the id and numThreads variables. This was the result. I also used multiple numbers 4 and up for the threads. 

I thought it was interesting how it still wasn’t in order, however how C is similar to Java, I assumed there would need to be a loop to check the order.