Developing Soft and Parallel Programming Skills Using Project-Based Learning

Justin Choi, Adam Henrie, Jean-Pierre Sacha, Titilayo Shonuyi, Shanza Siddiqi

Spring 2019
Group Name: Qubits

Planning and Scheduling

Team name: qubits

Name	Email	Task	Duration	Dependency	Due Date	Note
Justin Choi	jchoi34@student.gsu.edu	Report	3 hours	Task 3 and Task 4	2/21/19	Report must be finished the day before the due date
Adam Henrie	ahenrie1@student.gsu.edu	Uploading video, Parallel programming	4 hours	Video	2/21/19	Please send the link to the video presentation task 3 information
JP Sacha	jsacha2@student.gsu.edu	Help with arm assembly programming and report	4 hours	None	2/21/19	Please assist in assembly programming
Titilayo Shonuyi	tshonuyi1@student.gsu.edu	Arm assembly programming	3 hours	None	2/21/19	Please send task 4 information for the report
Shanza Siddiqi	Ssiddiqi2@student.gsu.edu	Create To do/In Progress/Done columns and cards	3 hours	None	2/18/19	Please create new cards on github and provide screenshots of the tasks

Teamwork Basics

1. **What to do to get the task accomplished *and* the team members' satisfaction high?**

To keep the team member satisfaction high we can, set ground rules, use a facilitator, know the strengths of your group members, keep lines of communication open, and know how to avoid or solve common problems. We have set ground rules for tangents and overly talkative people. We have also acknowledged the need for a group facilitator. The position is on a volunteer basis.

2. **Answer all the questions in the Work Norms, Facilitator Norms, Communication Norms using your own words and your own context.**

- **Work Norms: How will work be distributed? Who will set deadlines? What happens if someone doesn't follow through on his/her commitment (for example, misses a deadline)? How will the work be reviewed? What happens if people have different opinions about the quality of the work? What happens if people have different work habits (e.g., some people like to get assignments done right away; others work better with the pressure of a deadline).**

  The group will volunteer for each task and will meet together to decide on the deadlines. If someone doesn't follow through on their commitment, the team will try to talk to that person about any problems they are facing and if they need any help with their task. All files can be shared through GitHub, so everyone is able to view the work that is being done and give input on the work. We all have similar work habits and will work consistently to finish the project. i.e. We have agreed to meet up at predetermined times and are aware that some work may be telecommuted from within the Slack platform.

- **Facilitator Norms: Will you use a facilitator? How will the facilitator be chosen? Will you rotate the position? What are the responsibilities of the facilitator? (see below)**

  JP has basically taken on the facilitator role right from the start and has kept the team progressing. The role of the facilitator is to focus the team on the task, get participation from all team members, keep the team to adhere to their deadlines, suggest alternative procedures when the team isn't progressing, help team members confront problems, and summarize and clarify the team's decisions. Facilitator will be a voluntary role.

- **Communication Norms: When should communication takes place and through what medium (e.g., do some people prefer to communicate through e-mail while others would rather talk on the phone)?**

  Communication should take place regularly in order to be up to date on everyone's progress. It should be done through Slack since it is the easiest way to contact every team member since we are all using it.

- **Meeting Norms: What is everyone's schedule? Should one person be responsible for coordinating meetings? Do people have a preference for when meetings are held? Where is a good place to hold meetings? What happens if people are late to a meeting? What happens if a group member misses a meeting? What if he/ she misses several meetings?**

  Since everyone has a different schedule, it is a collaborative effort to coordinate meetings. A good place to hold meetings is in the library. If a group member misses a meeting, it is his or her duty to inform the group and to make sure he or she gets all the information from the meeting. If a group member makes it a regular occurrence to miss meetings, then the rest of the group will confront the individual to ascertain what is going on.

- **Consideration Norms: Can people eat at meetings? smoke? What happens if someone is dominating the discussion? How can norms be changed if someone is not comfortable with what is going on in the team?**

  It is okay to eat in meetings, but it is definitely not okay to smoke. If someone is dominating a discussion, the facilitator could step in and allow other members to provide their input. If any group member is uncomfortable with anything that is going on in the team, then the group can alter current protocols to accommodate that individual.

3. **As a team, select two cases out of the four mentioned in Handling Difficult Behavior. (use your own words and your own context)**

Too quiet: The facilitator or any other group member could ask for the quiet individual's input during discussions if they are being noticeably quiet.

Overly talkative: If a certain member is being too talkative, someone can ask for additional input from the rest of the group to try and give someone else the spotlight.

4. **When making decisions, if the team is having trouble reaching a consensus, what should you do? (use your own words and your own context)**
If the group is having difficulties in reaching a consensus when making decisions, the group can compromise by voting on the most popular option. Any other dissention from within the group will be discussed from within Slack if a group member is still not cooperative.

5. **What should you do if person may reach a decision more quickly than others and pressure people to move on before it is a good idea to do so?**
If a person is pressuring people to move on before it's a good idea to do so, the group can agree to come back to the current decision at a later time in order to discuss it more. We will use Slack for this purpose. Which will allow us the opportunity to move forward with other tasks but still be able to address everyone's concerns eventually.

6. **What happens if most people on the team want to get an "A" on the assignment, but another person decides that a "B" will be acceptable?**
If the majority of the group wants to obtain an "A", but there is someone that is satisfied with only a "B", then the group can assign that individual somewhere between 0-100% on the note column of the report, based on their level of cooperation and contribution to the team's workload. Group consensus is that unless situations arise that prevent the entire team from getting a high grade, the grade goal is within the range of B to A.

Parallel Programming Skills

Foundation

1. **Identifying the components on the raspberry PI B+**
   Raspberry Pi B+ has 1gb of RAM, a quad core ARM based processor at 1.4ghz, a micro-usb power supply, 3.5mm audio out jack, HDMI out jack, Gigabit Ethernet, and 4 USB 2.0 ports, there is also a GPIO header and camera header on the motherboard.

2. **How many cores does the Raspberry Pi's B+ CPU have?**
   The B+ has a quadcore cpu at 1.4ghz clock speed.

3. **List three main differences between X86 (CISC) and ARM Raspberry PI (RISC). Justify you answer and use your own words (do not copy and past)**
   X86 (CISC) is considered to have a register memory structure whereas ARM(RISC) is a register-register implementation. CISC is considered more flexible because it allows for a more complex instruction set, however it has less registers than ARM. ARM uses Big endien notation but allows for switching between endiens. Finally ARM is considered faster in some respects because of less instructions needed to perform a task. However creative programming needs to be implemented in order to get similar overall results to that of x86.

4. **What is the difference between sequential and parallel computation and identify the practical significance of each?**
   Sequential computation is good for code that cannot be easily broken up into parallel chunks. Parallel is good for chunks of code that can be parallelized such as loops.

5. **Identify the basic form of data and task parallelism in computational problems.**
   Data parallelism is defined as parallel calculations that can performed to compute data in parallel. Task parallelism involves not just parallel data computations but the overall set of tasks that occur can be parallelized i.e. a whole set of equations. The main distinction here is that data parallelism happens to a strict set of data whereas Task is more broad.

6. **Explain the differences between processes and threads.**
   Processes are distinct instances of a program that perform a myriad of tasks, there can be sub processes in a process that allow for discrete processing however they should not be confused with threads which allow for multiple instances of code for a process to be split and be processed in parallel on multicore cpus before being joined after they output the desired result.

7. **What is OpenMP and what is OpenMP pragmas?**

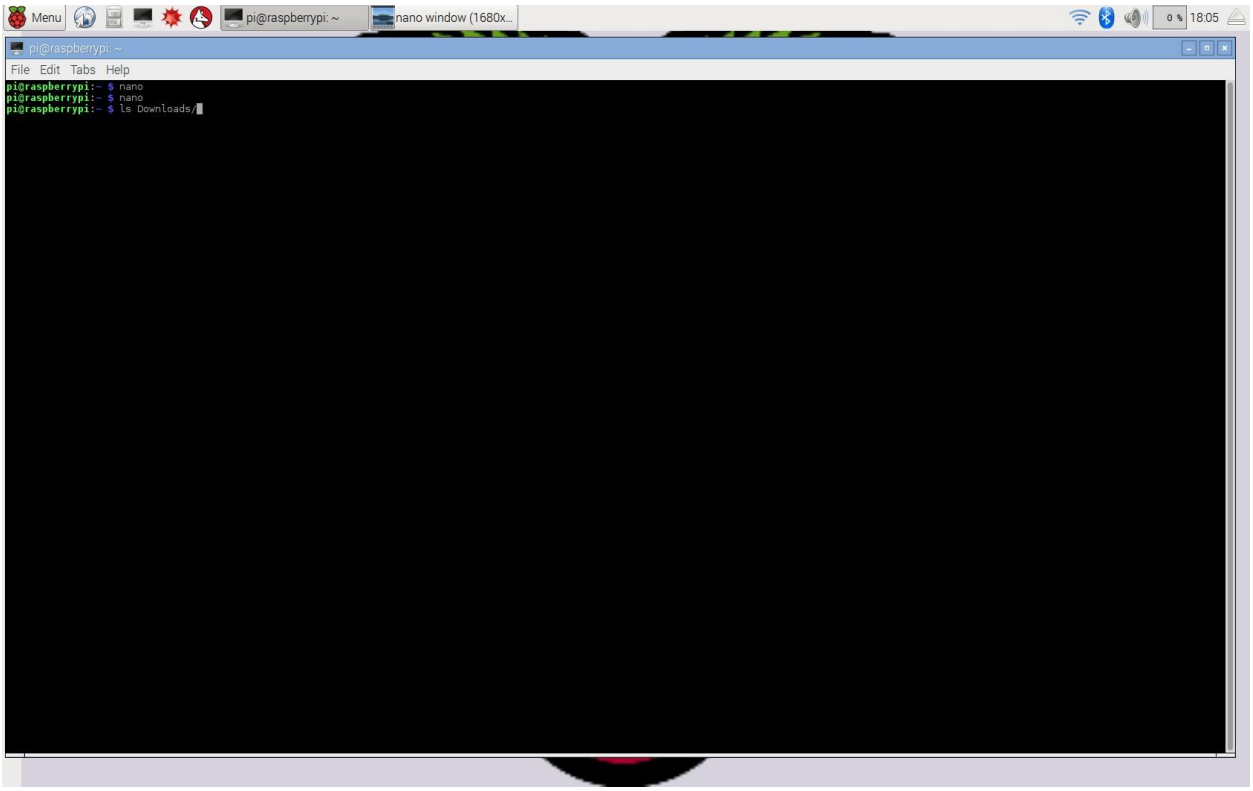
OpenMP is the framework designed for the C language that allows for implicit multi-threading and parallelism in processes. It is implicit because it allows for a pre-built framework to code and utilize multithreading through the OS itself. This function is achieved via a compiler directive called “pragmas” that signals the OS to treat the code segment as parallel. It is achieved via a single task multiple data implementation.

8. **What applications benefit from multi-core (list four)?**
   Applications that benefit heavily from multicore systems are Web browsers with multi-media plugins, web servers (where every new connection through a socket could be handled by a core on a multithreaded CPU, database servers which are also constantly working on multiple tasks at one time, and finally modern PC and console and mobile games that run on X86 and ARM architectures. Games such as Battlefield 3, and PUBG are designed from the ground up to utilize multi-core cpus with 4 and 6 cores respectively to process the games real time game engines.

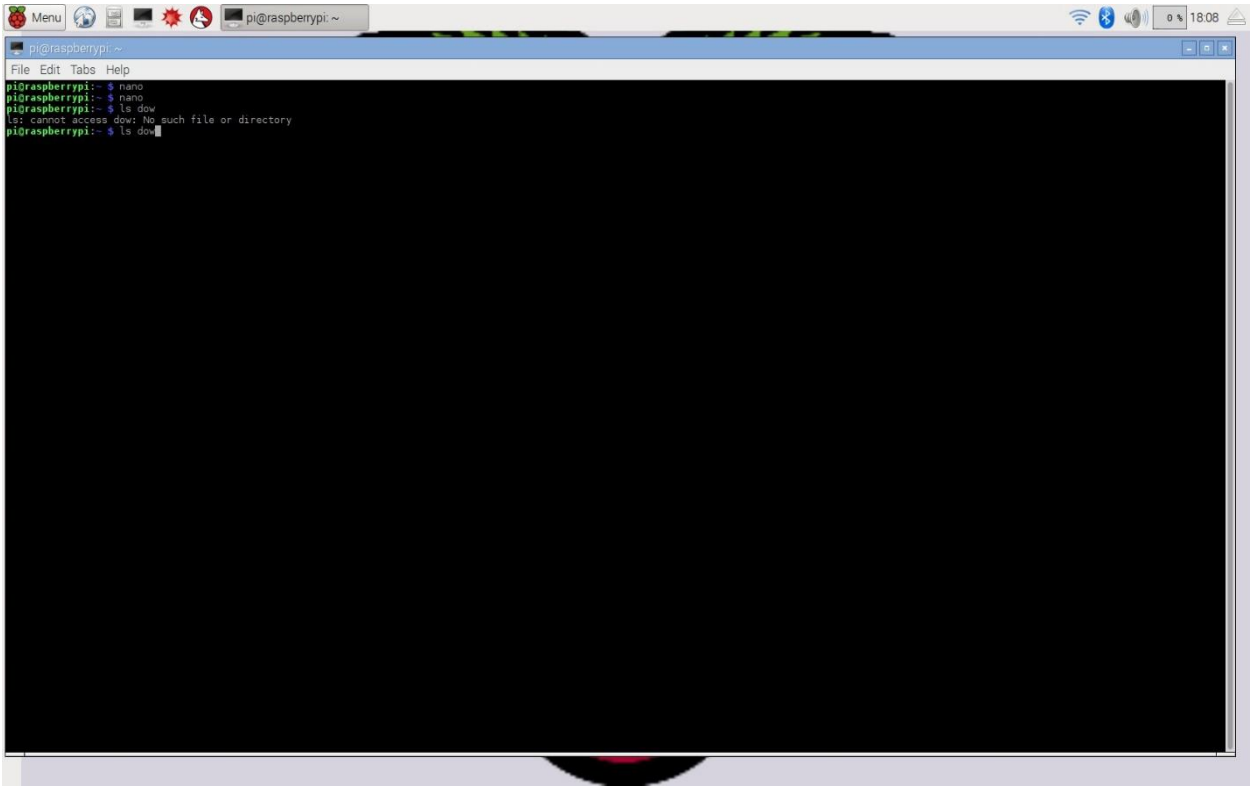
9. **Why Multicore? (why not single core, list four)**
   Multicore helps to solve the problem of increasingly difficult to achieve processor frequencies by splitting the workload. Decreases the need for dedicated server based computation by allowing for more speed on a personal workstation. Many applications that need more speed than can be supplied by a serialized single core processor benefit from multi-threading on multi-core CPUs. Multi-core also allows for the use of many separate serial computation only programs to happen on the same CPU die.

Parallel Programming Basics

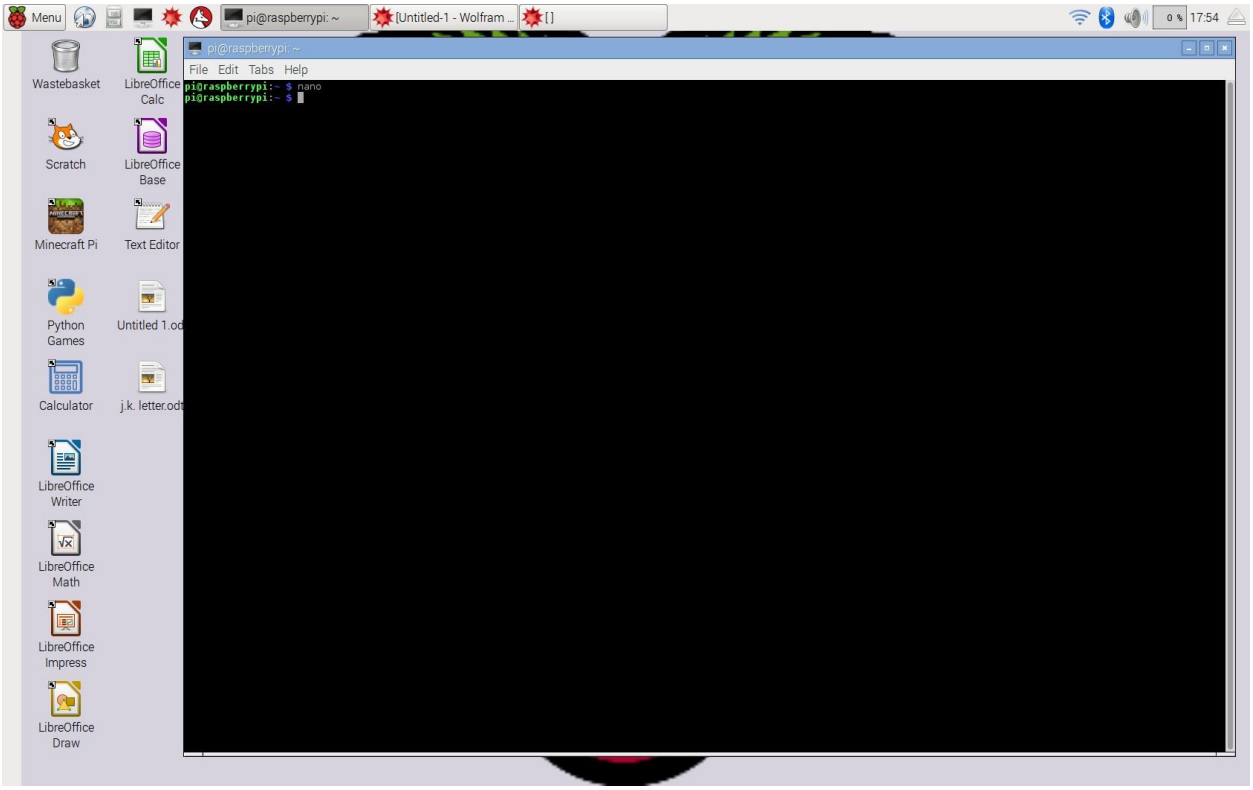
Here we can see that we can tab and auto-complete a command such as ls Downloads



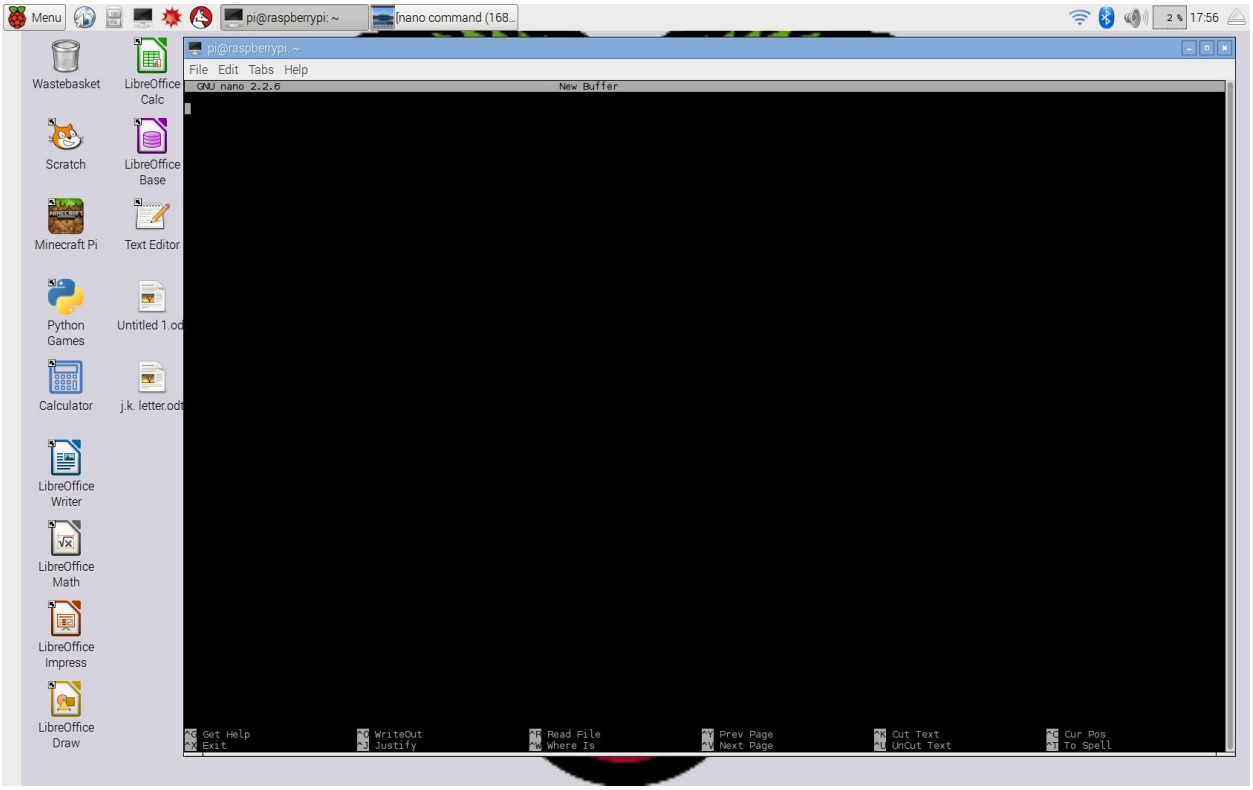
Here we can see that we can try a command and then hit the up arrow key in order to re-list the last command.



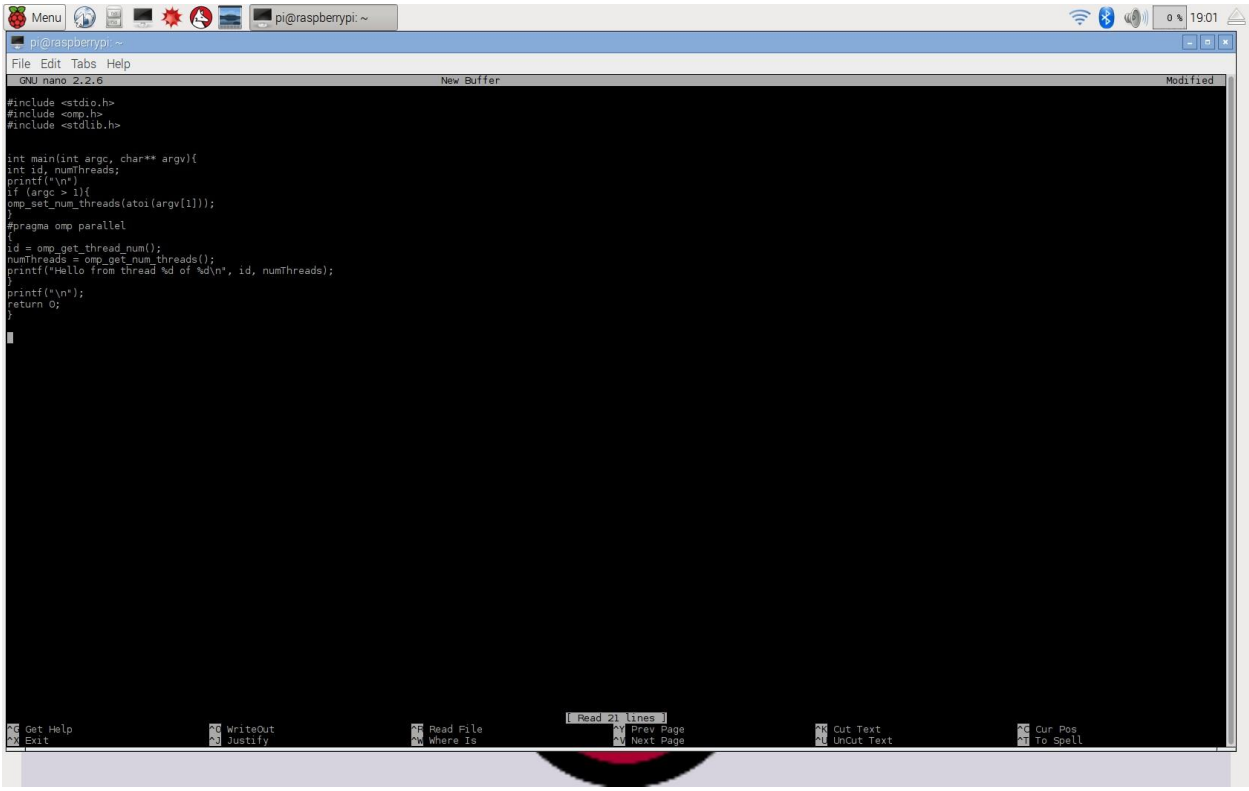
Here we can see the nano command brings up the nano editor



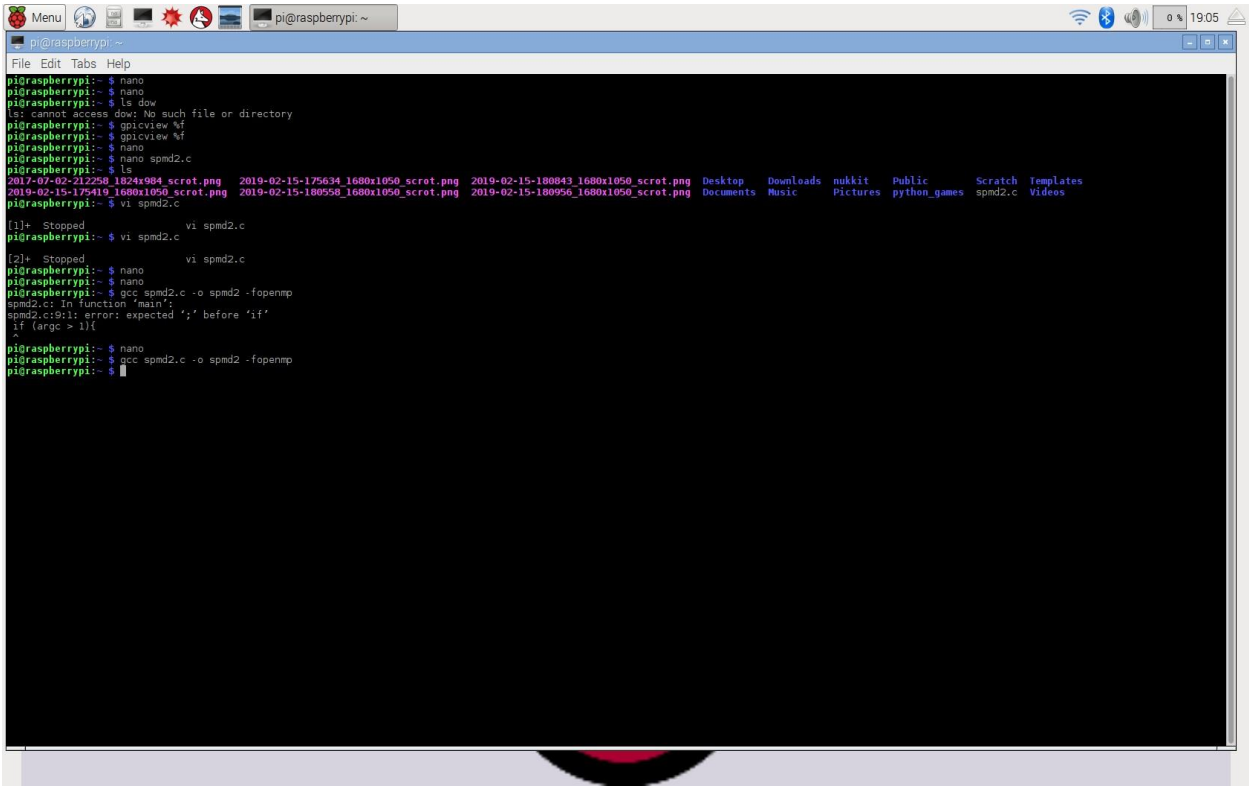
Here we can see the nano window



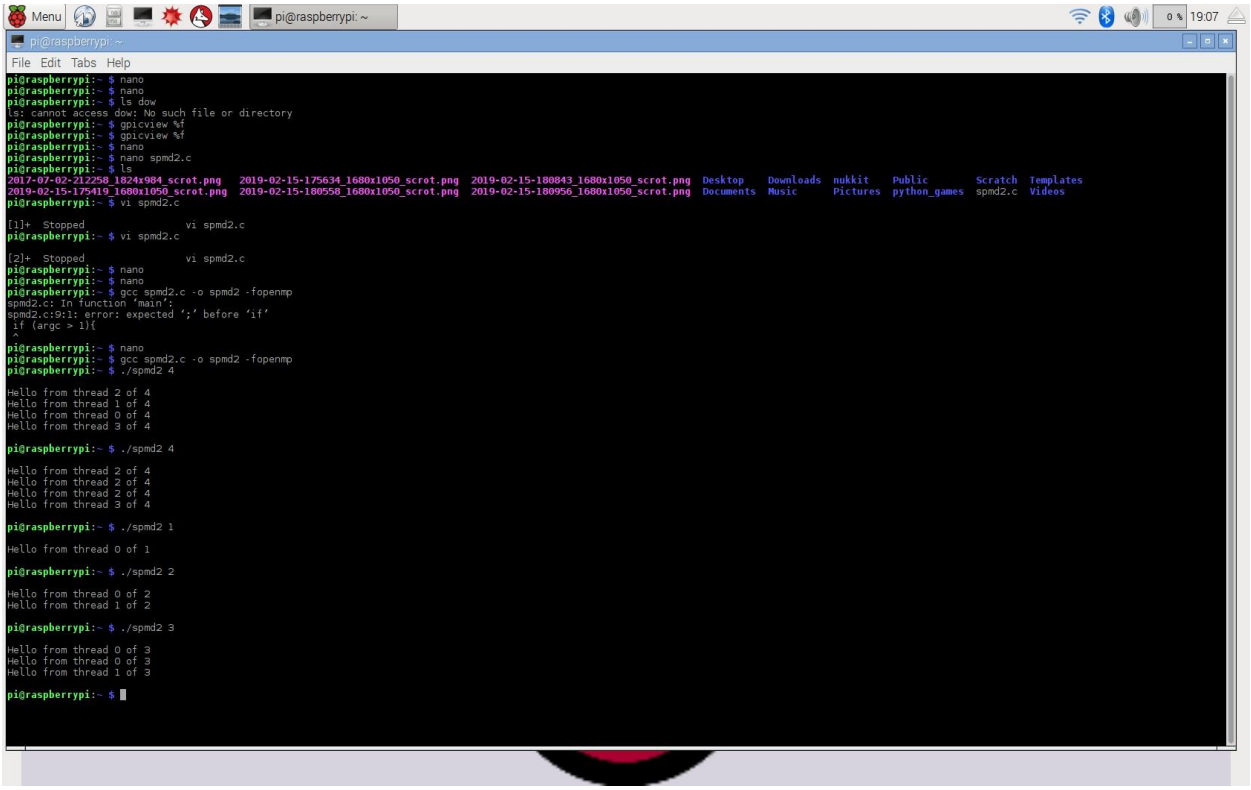
Here I input the command nano spmd2.c to create the file and then write this code.



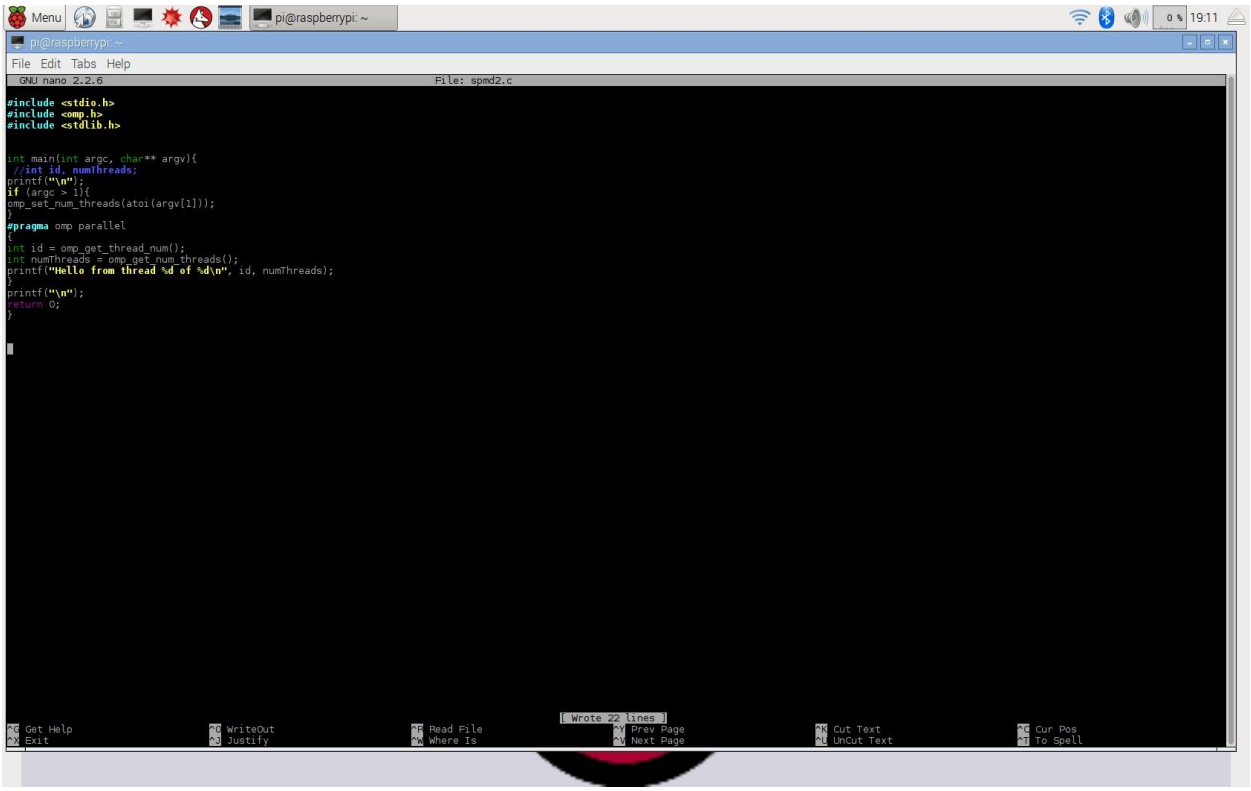
Next the code was compiled using this command: gcc spmd2.c -o spmd2 –fopenmpA compile error was encountered and then fixed.



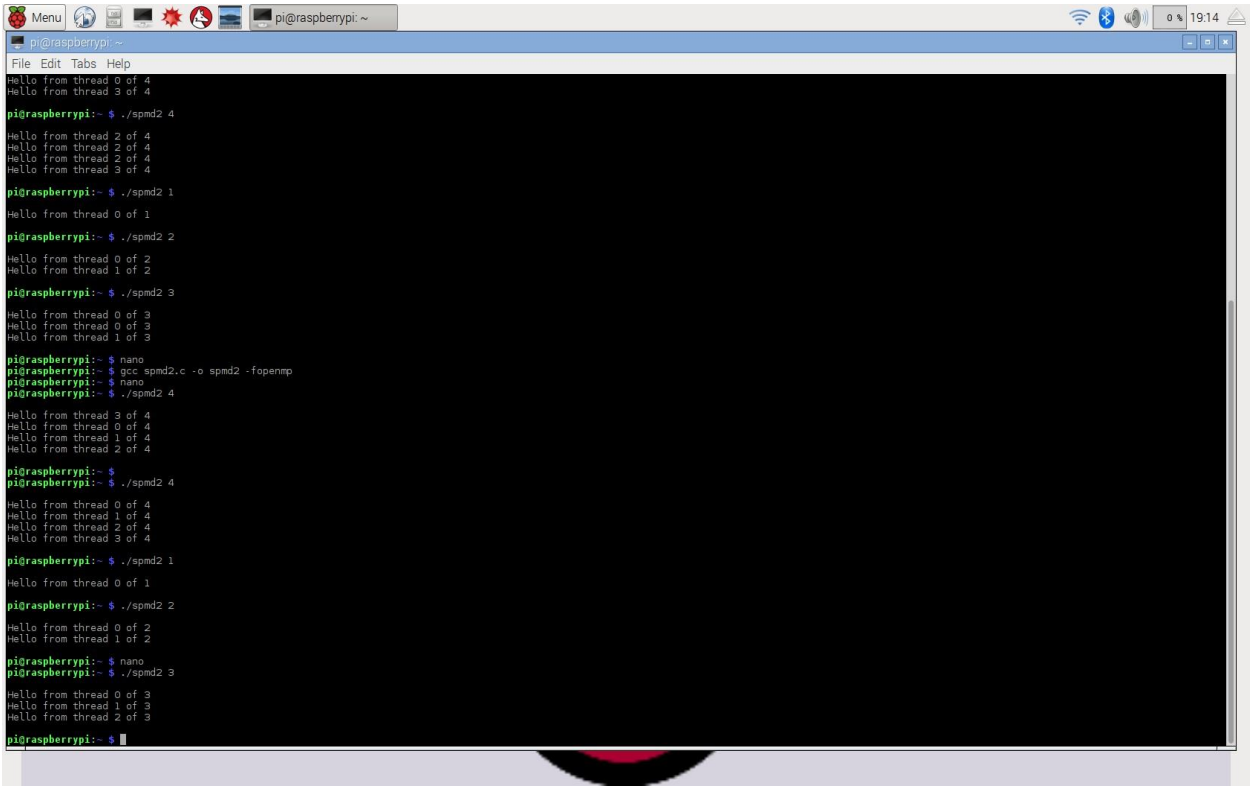
The program was run with an input of 4, 3, 2, 1. A critical section error occurred and the memory locations for each thread are shared and not mutually exclusive. Thus, the cores reported multiple duplicates of the same thread id.



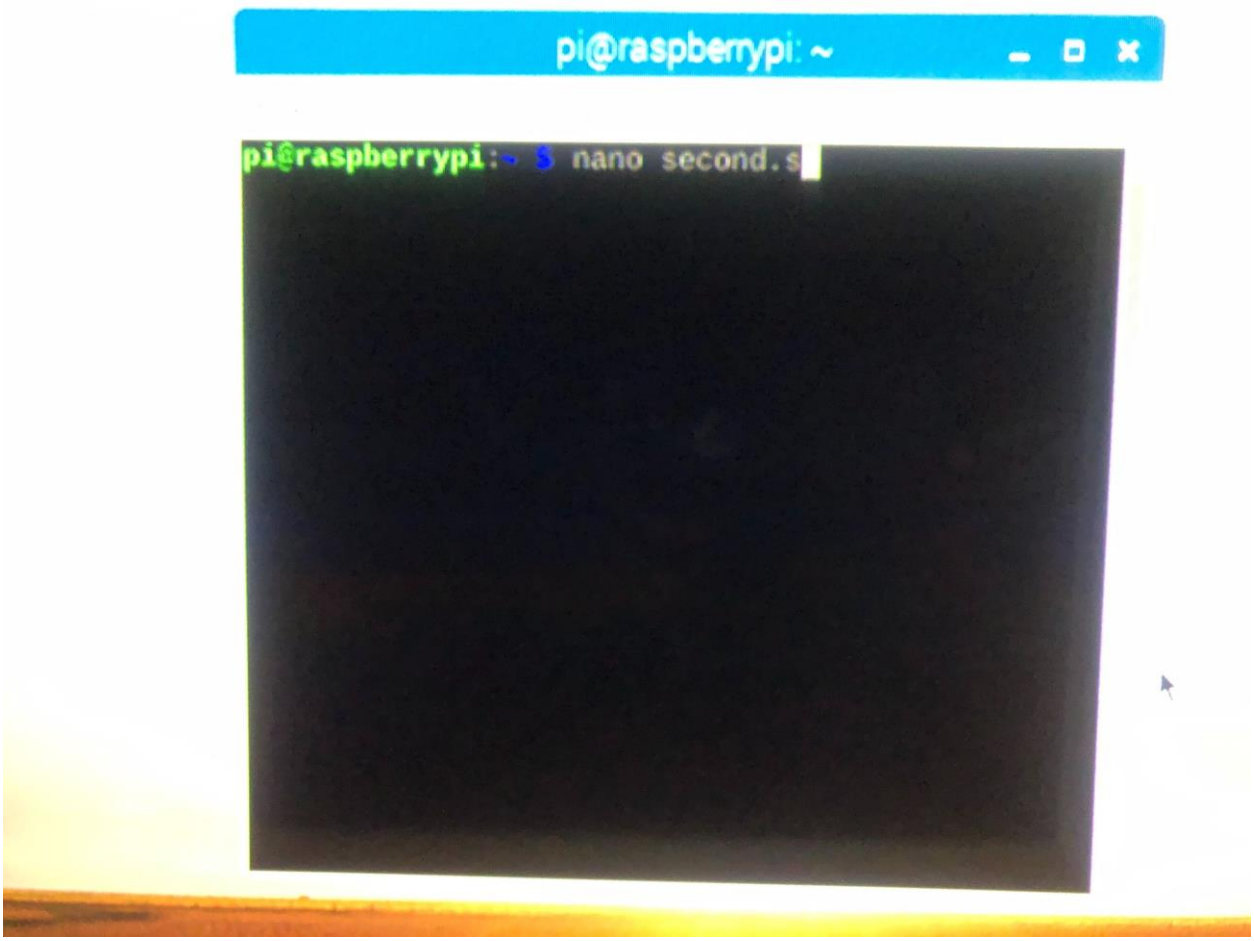
Here, two lines of code were added to allow for separate memory location for each core. Two lines were fully declared with int. And the original int variable declaration of id and numThreads were commented out.



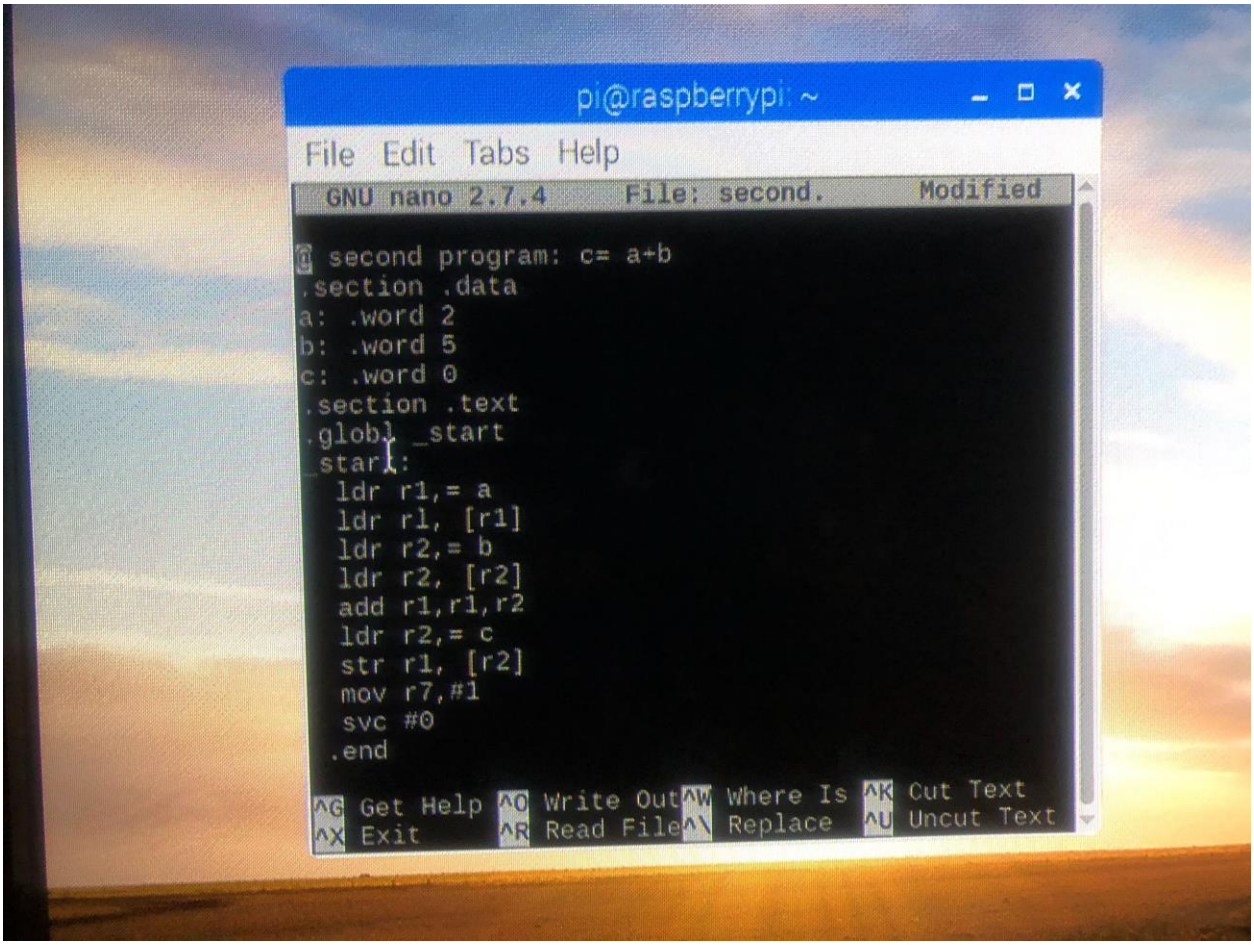
Here, below. The program has been recompiled and we have re-input 1-4 and each core is reporting in with its own unique thread id out of four.



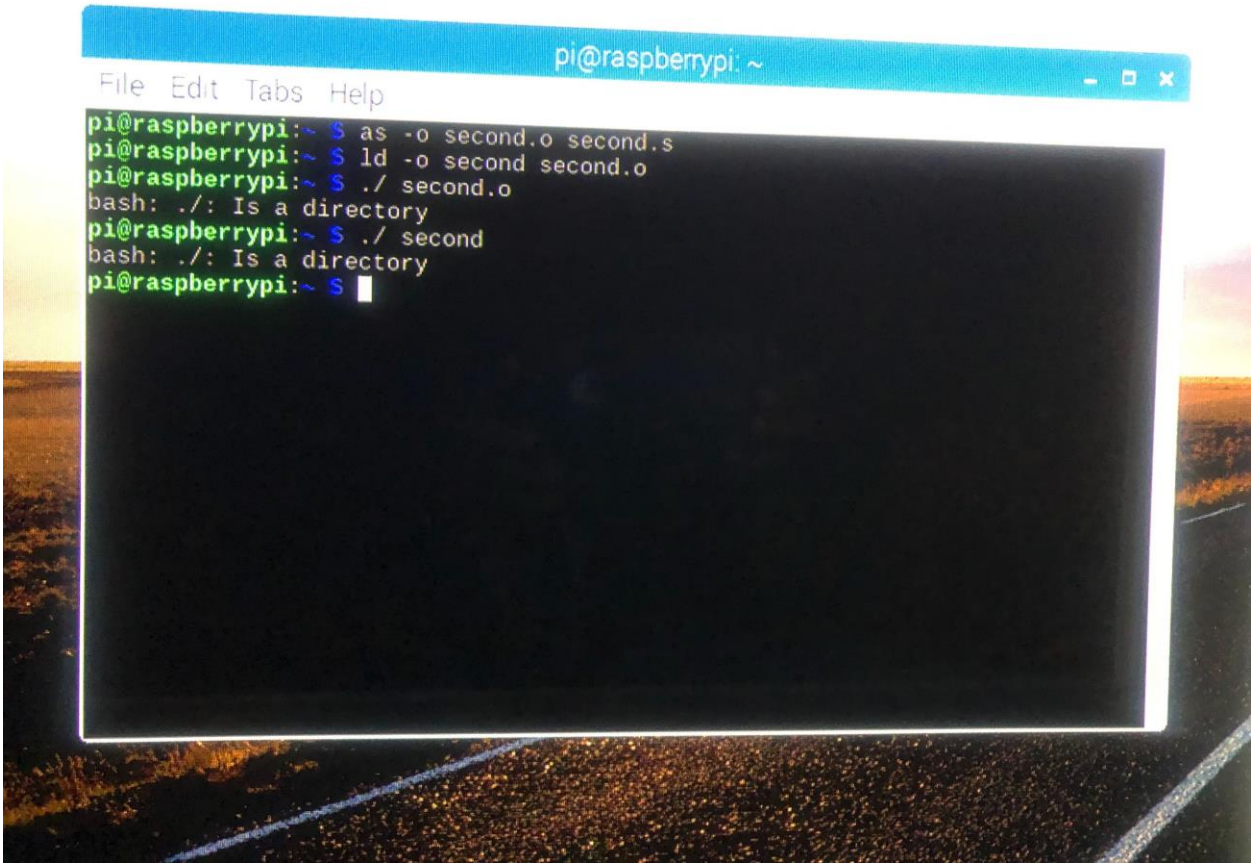
ARM Assembly Programming



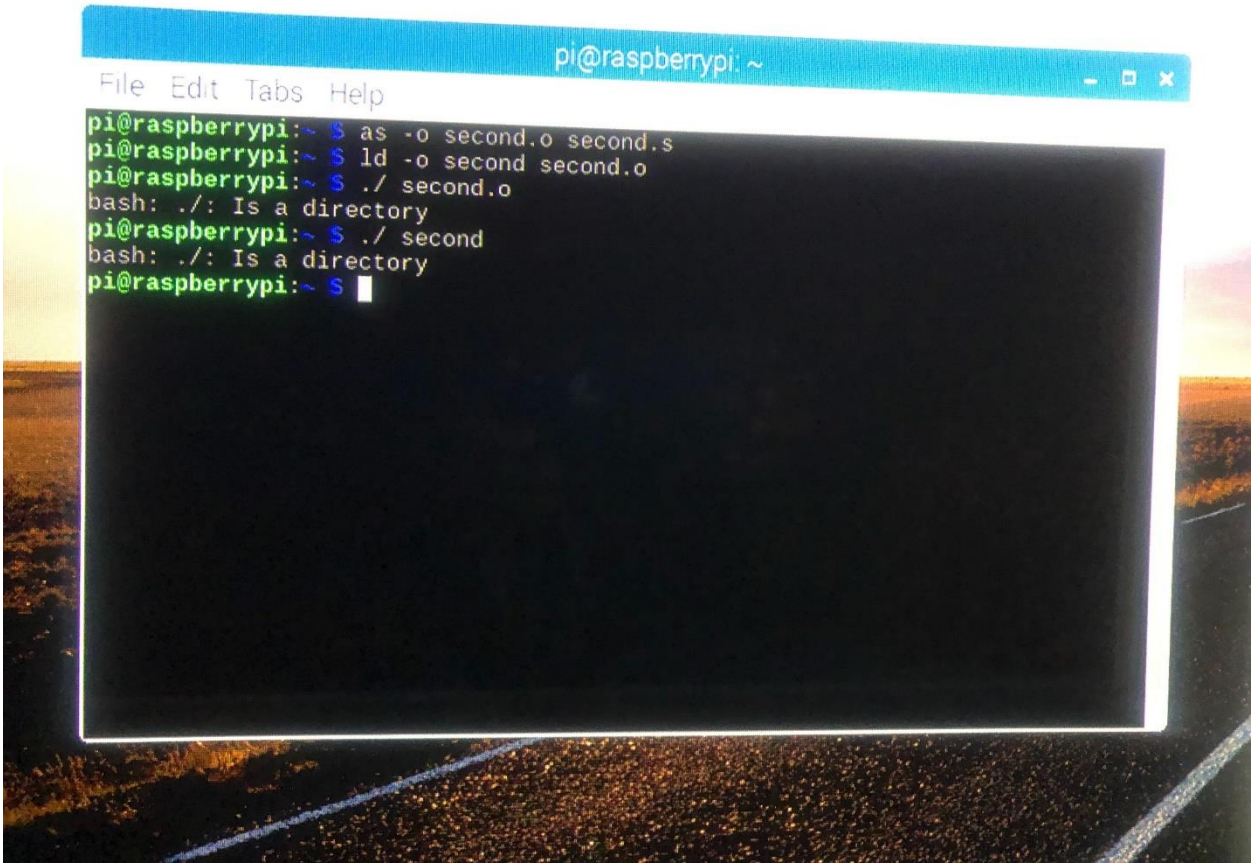
Writing the program.



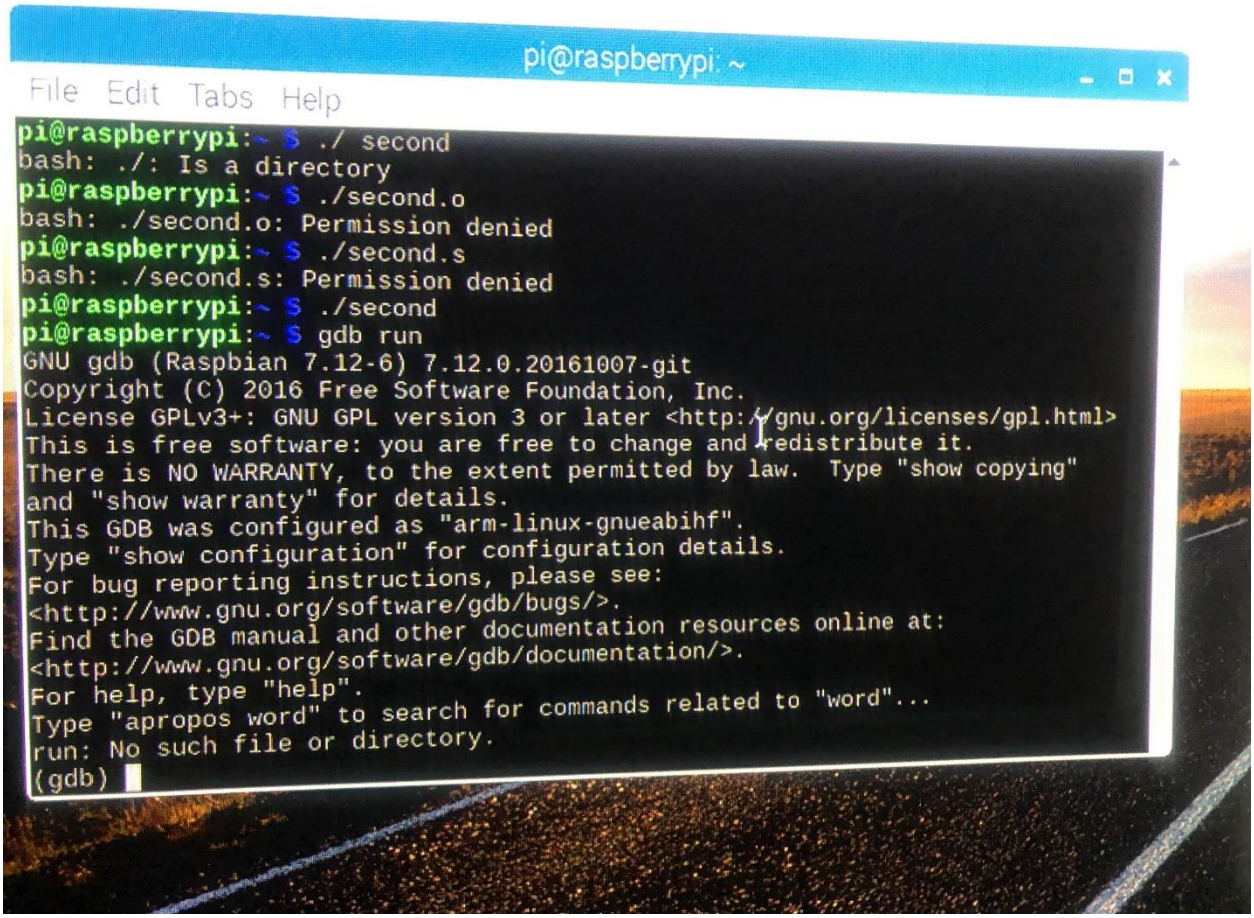
To Assembly the file.



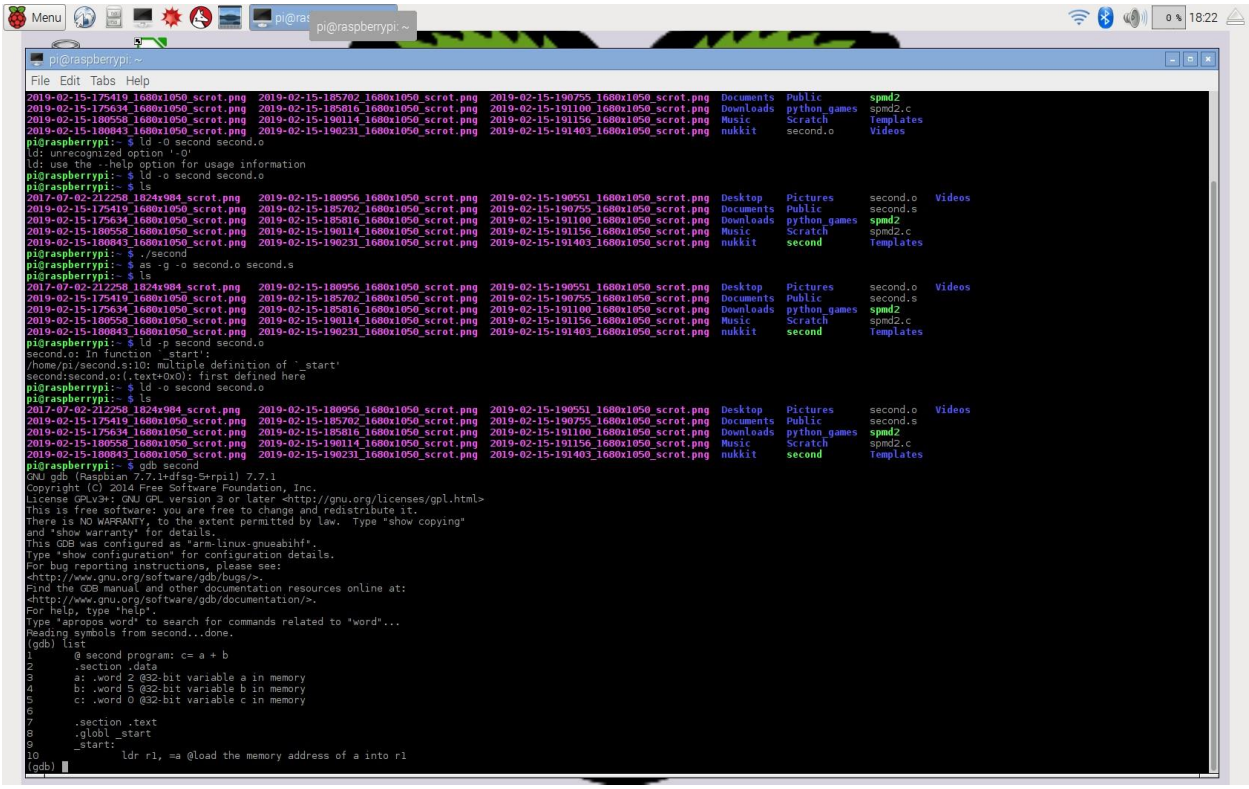
Linked the program. Then trying to run it. error message bash: ./: is a directory



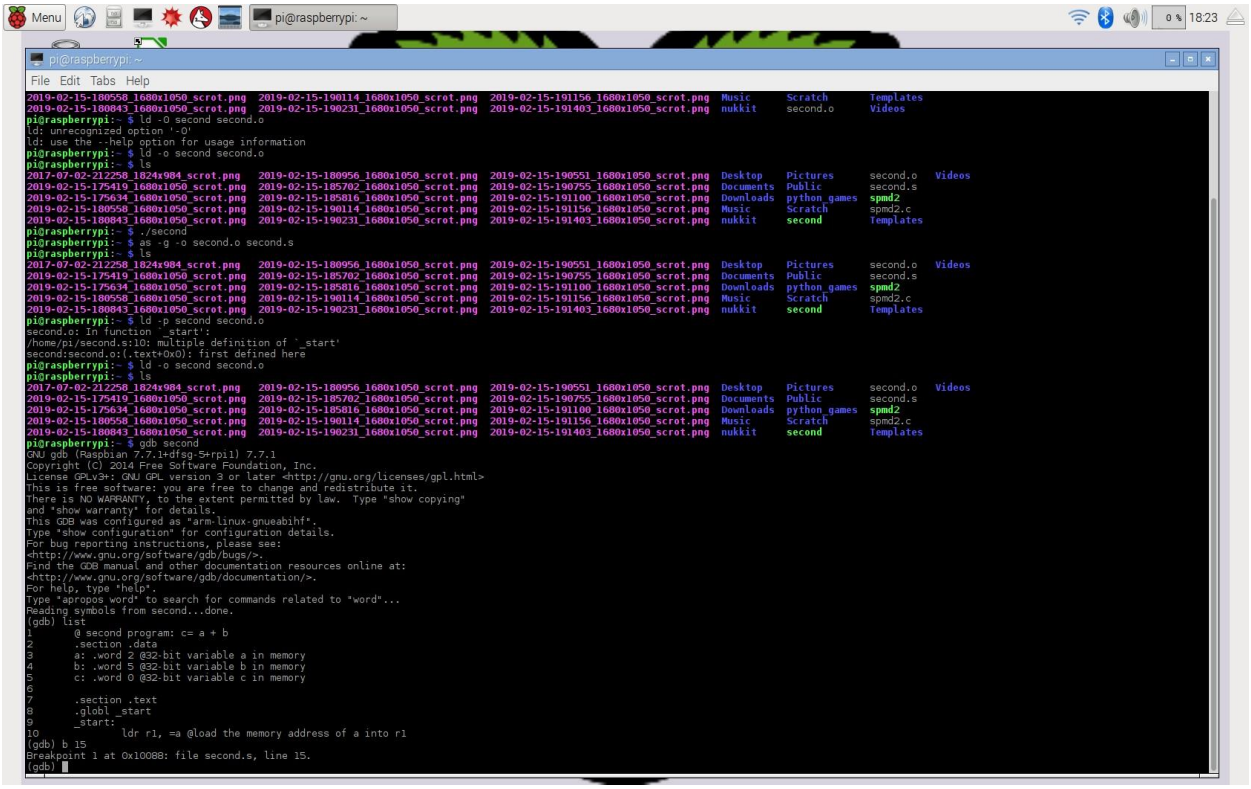
Fixed and starting debugging process.



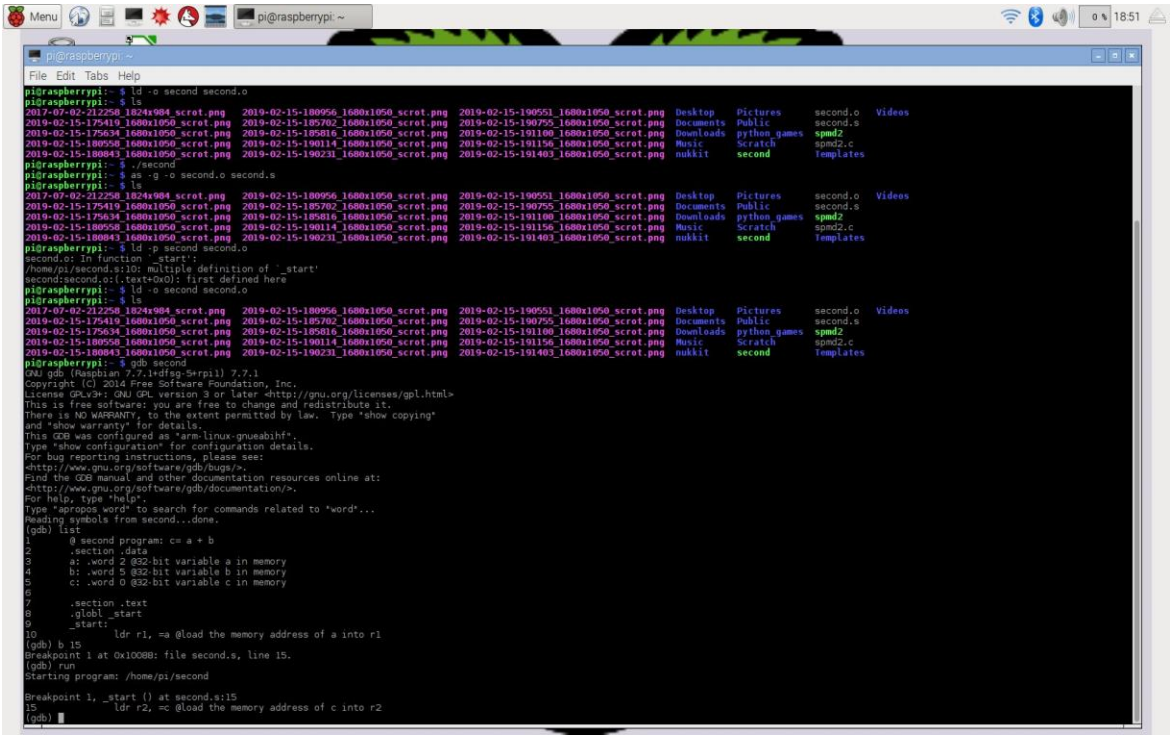
Here we have listed the symbols from the object file



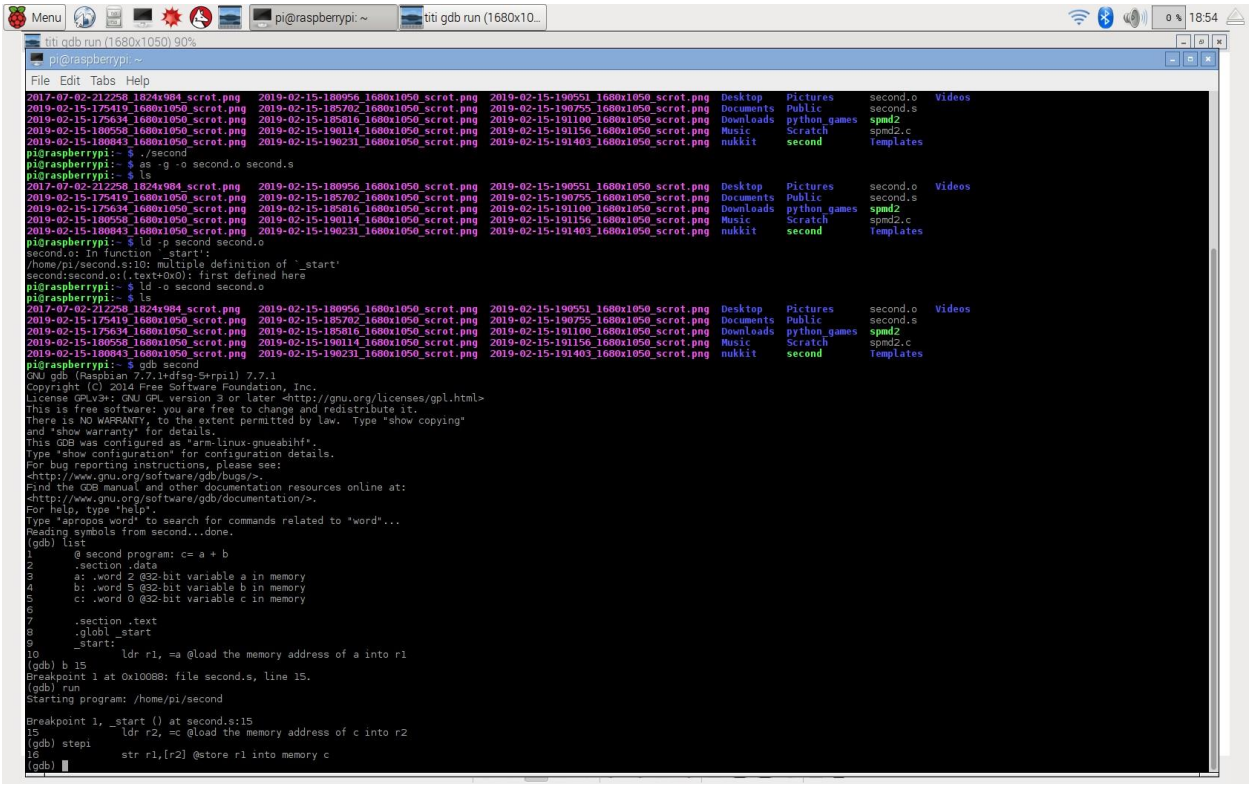
Here we are setting a break point for line 15 of the program.



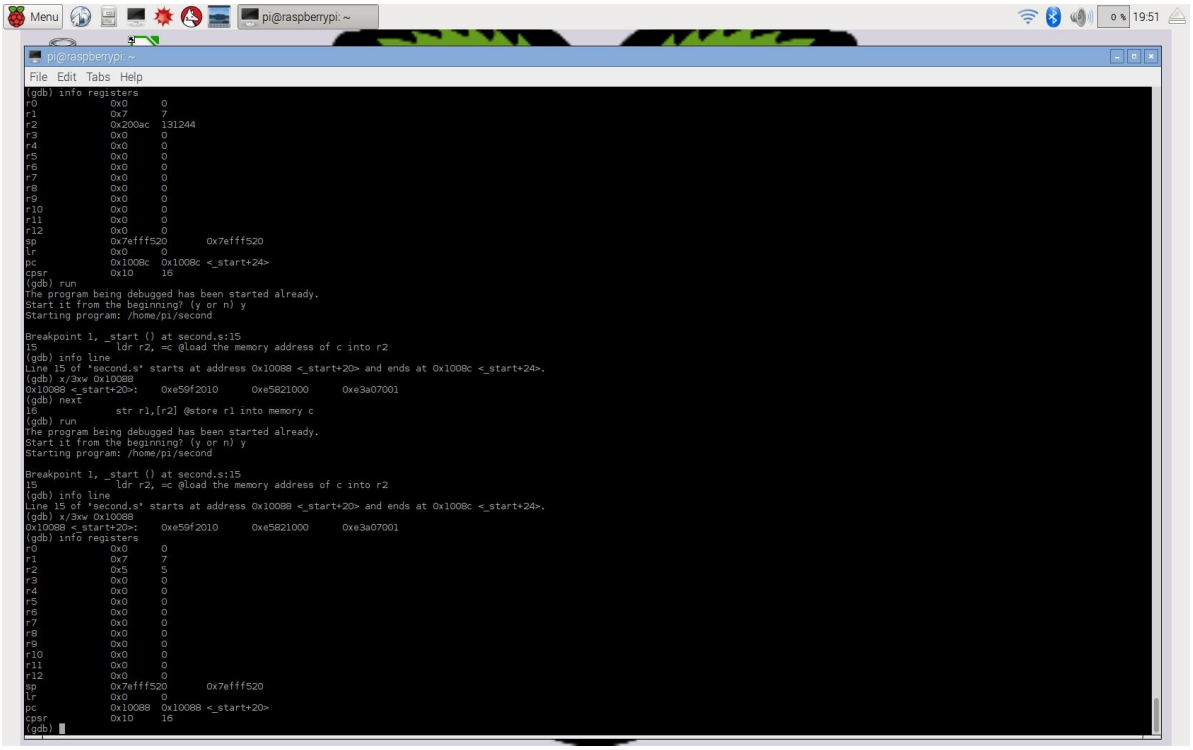
Here we run the code in gdb



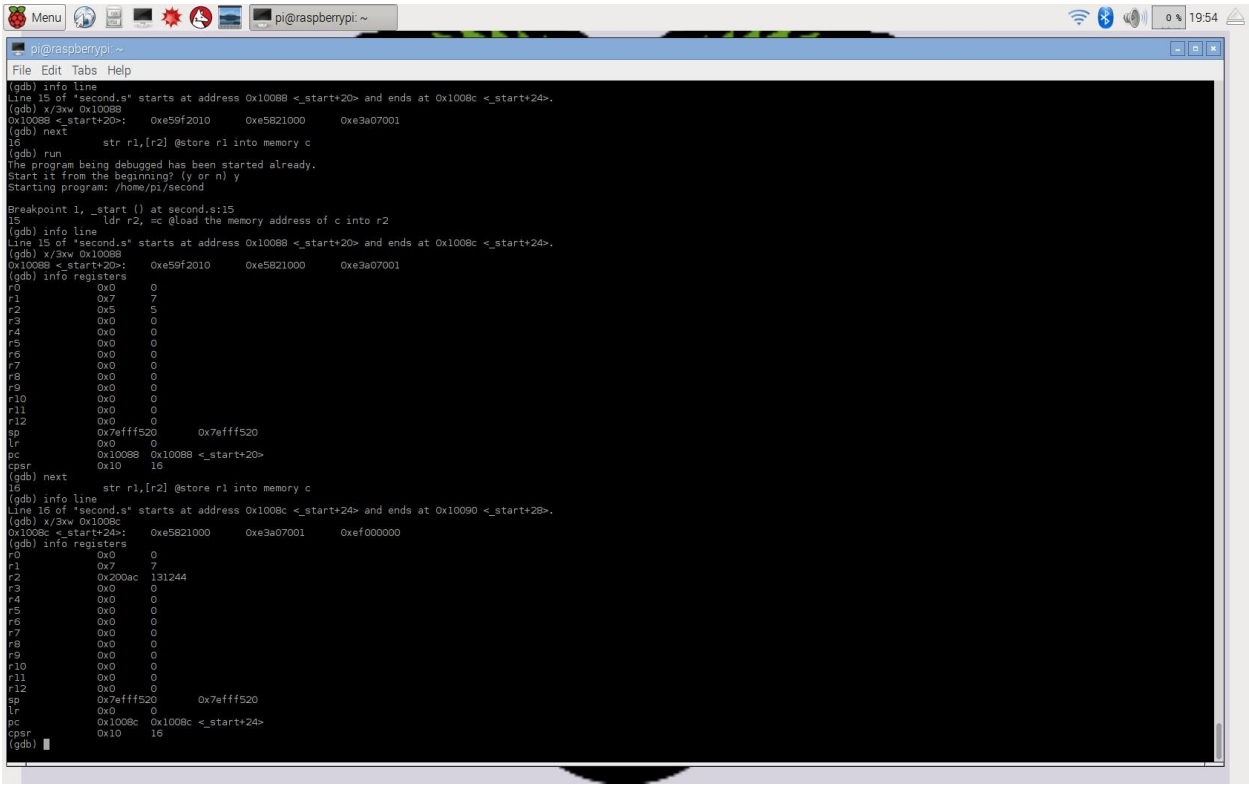
Here are stepping into instructions, the first instruction on line 15 has yet to be executed



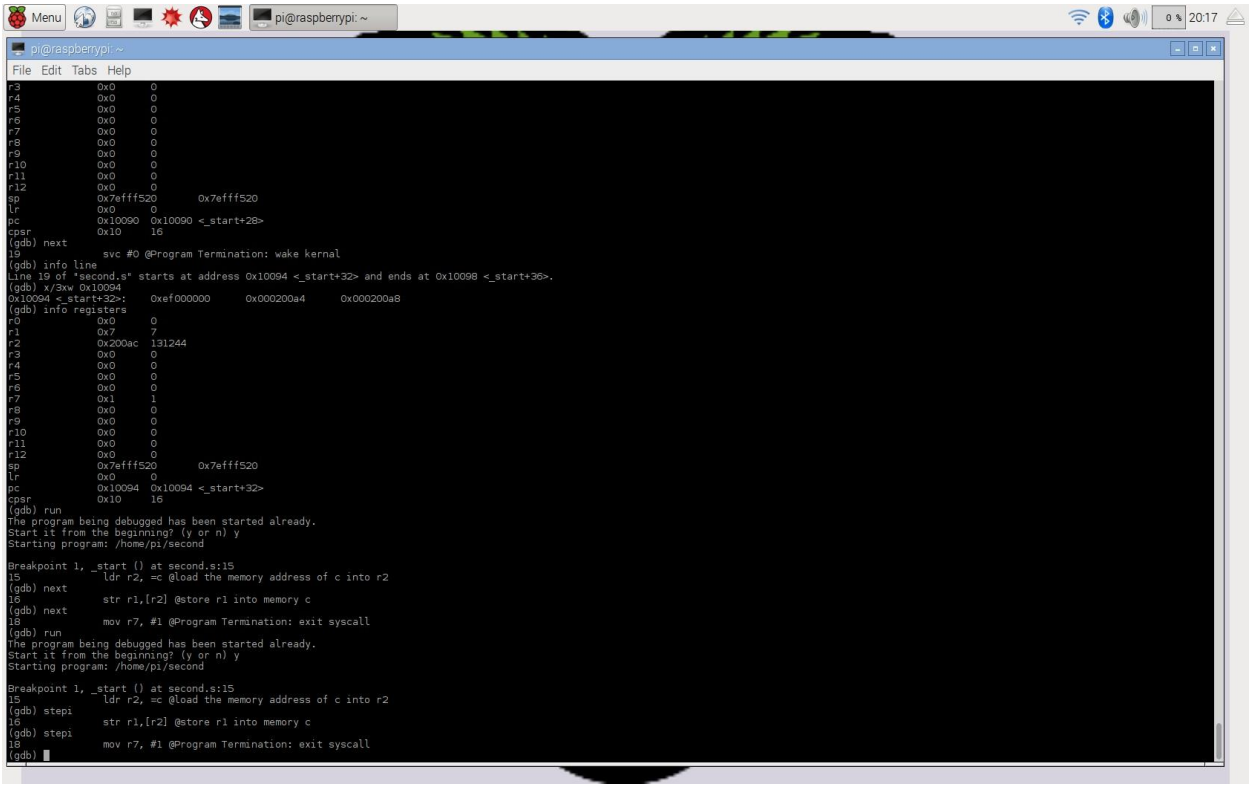
Here we can see info for line 15 the registers shows r1 = 7, r2 = 5



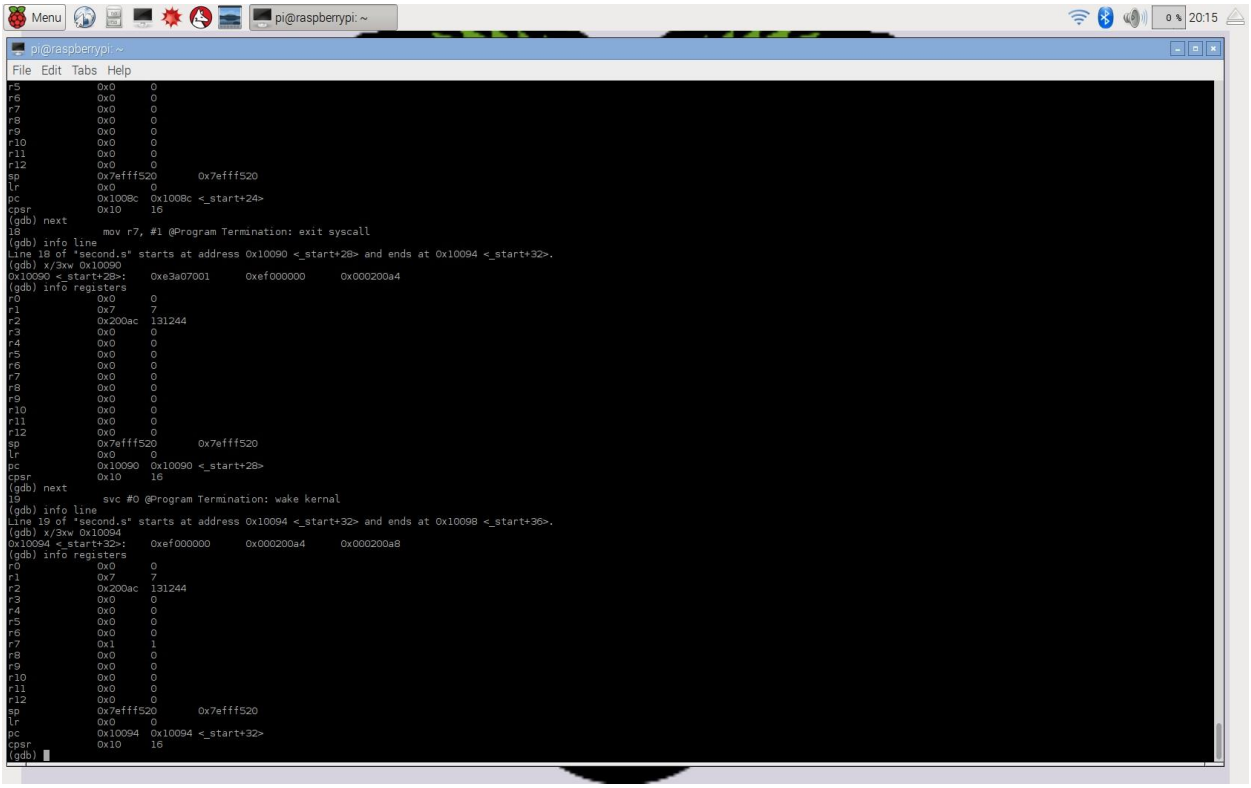
Next we see line 16. Here we can see that r3 is holding the memory location for c.



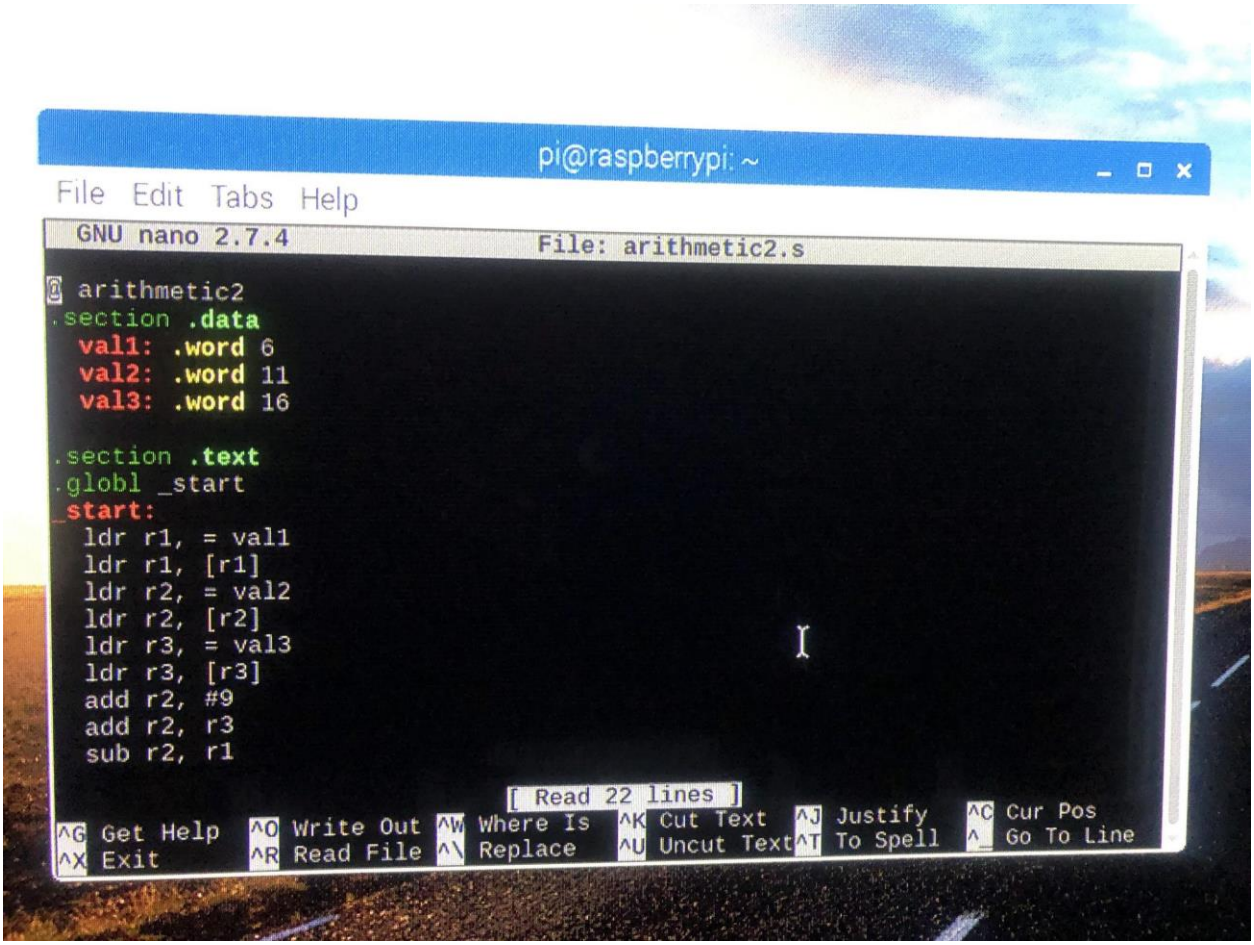
Below we can see that line 17 was not shown in the gdb debugger.



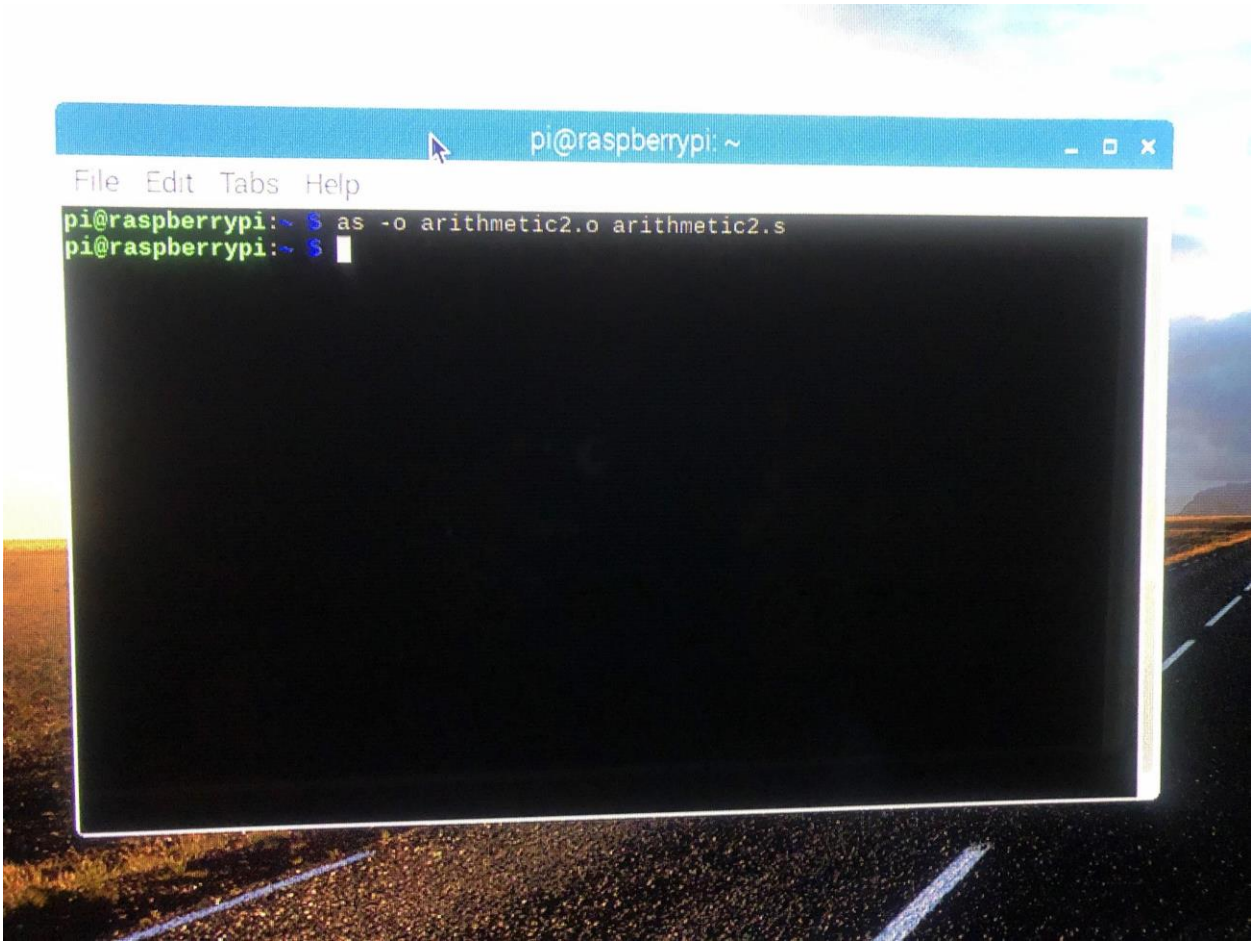
Here below we can see that a 1 was put into r7 and the program exited to the kernal.



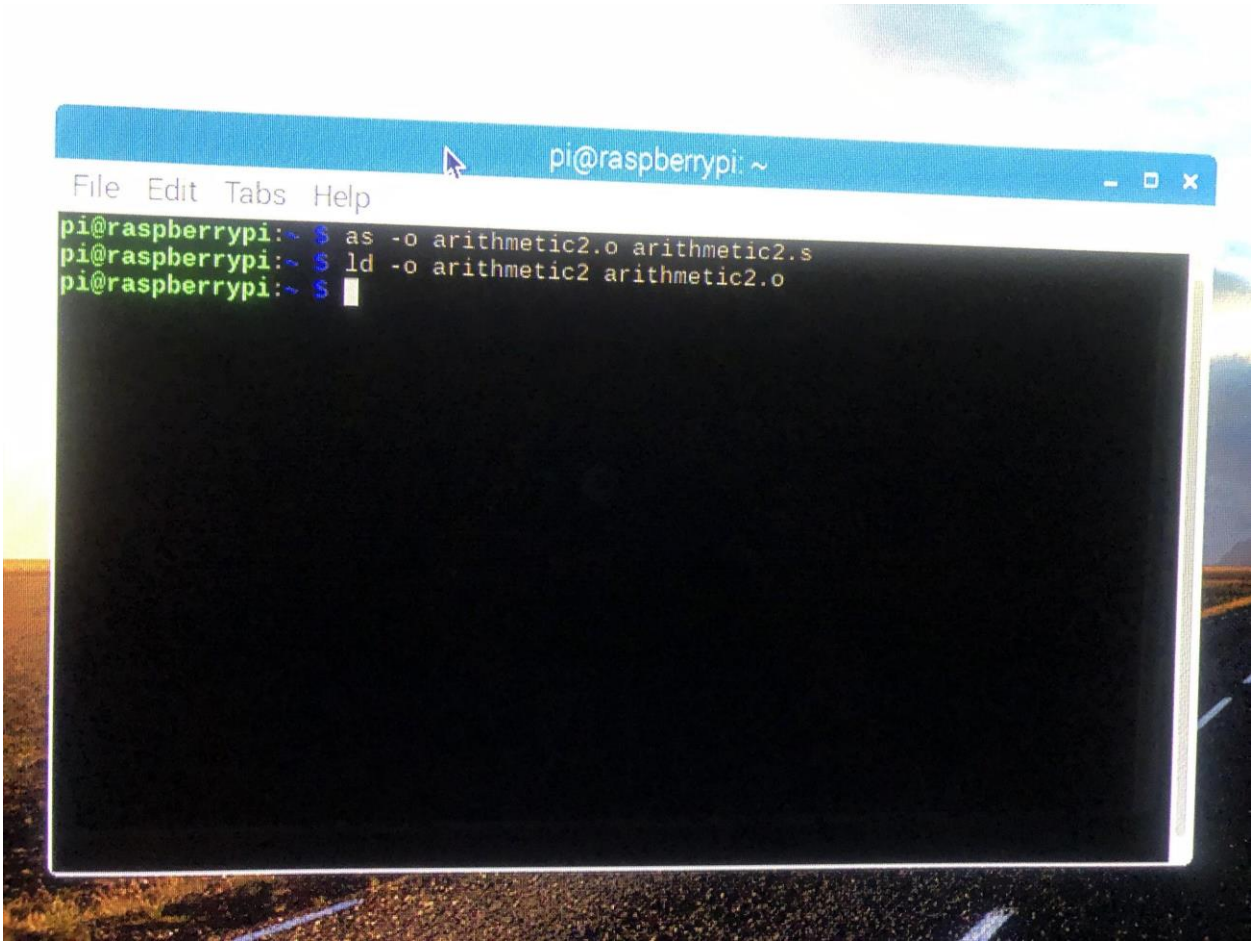
Writing the code on ARM assembly with registers part 2



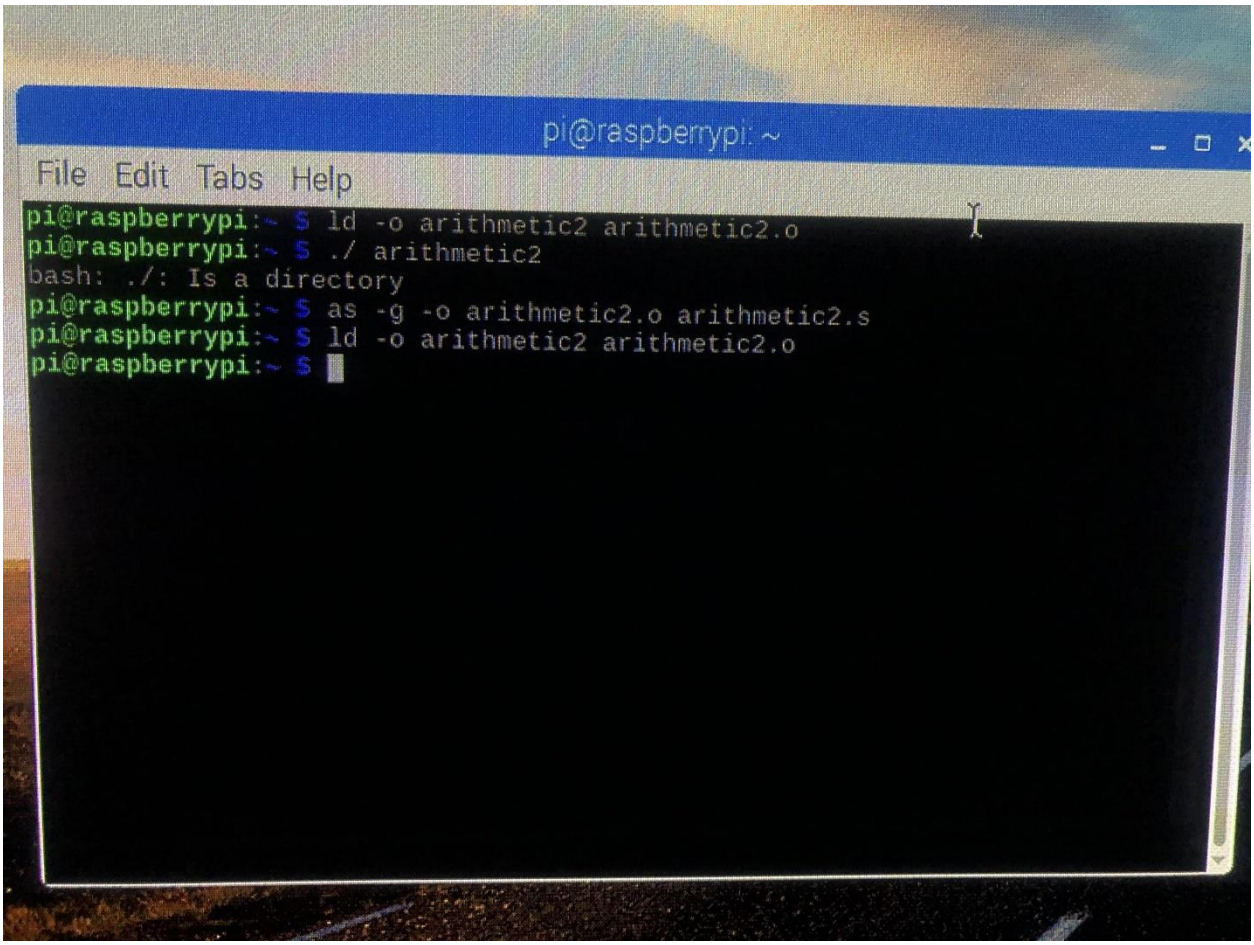
Assemble program. arithmetic2.o create file object.



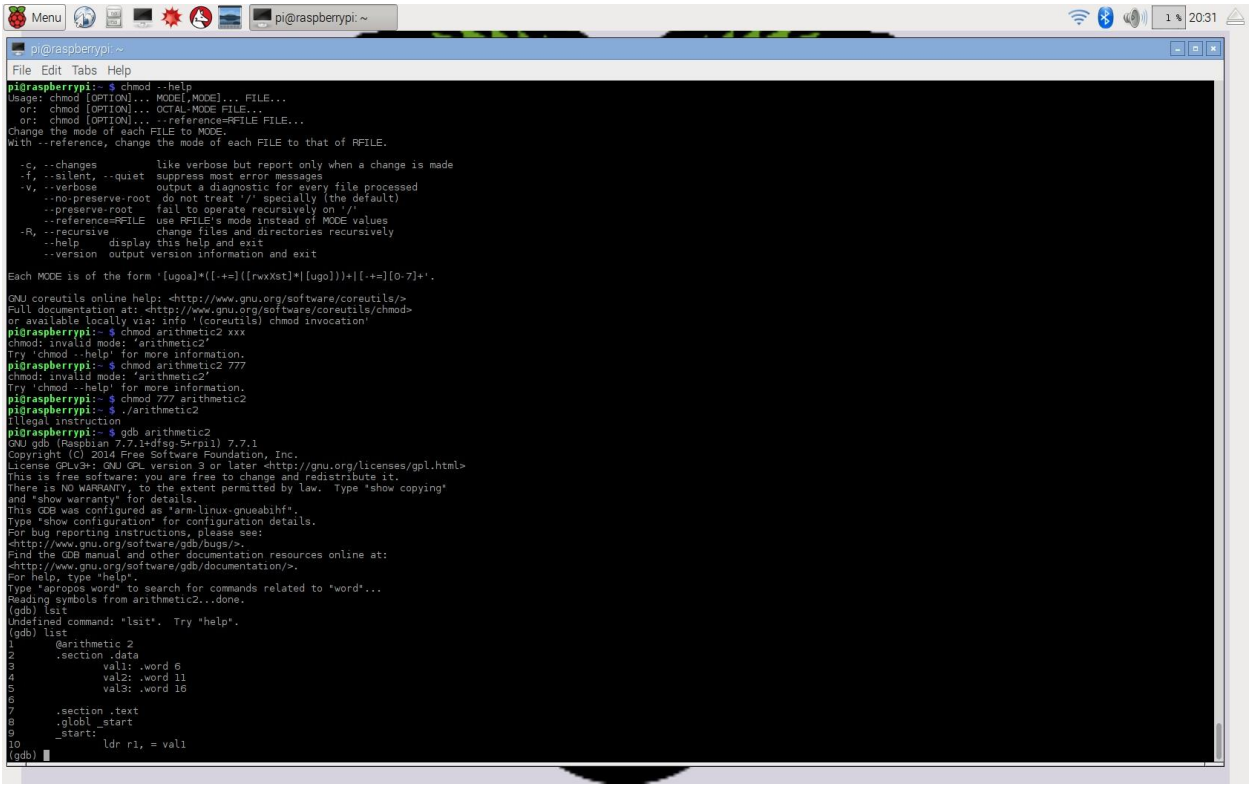
Link file in the second line of code.



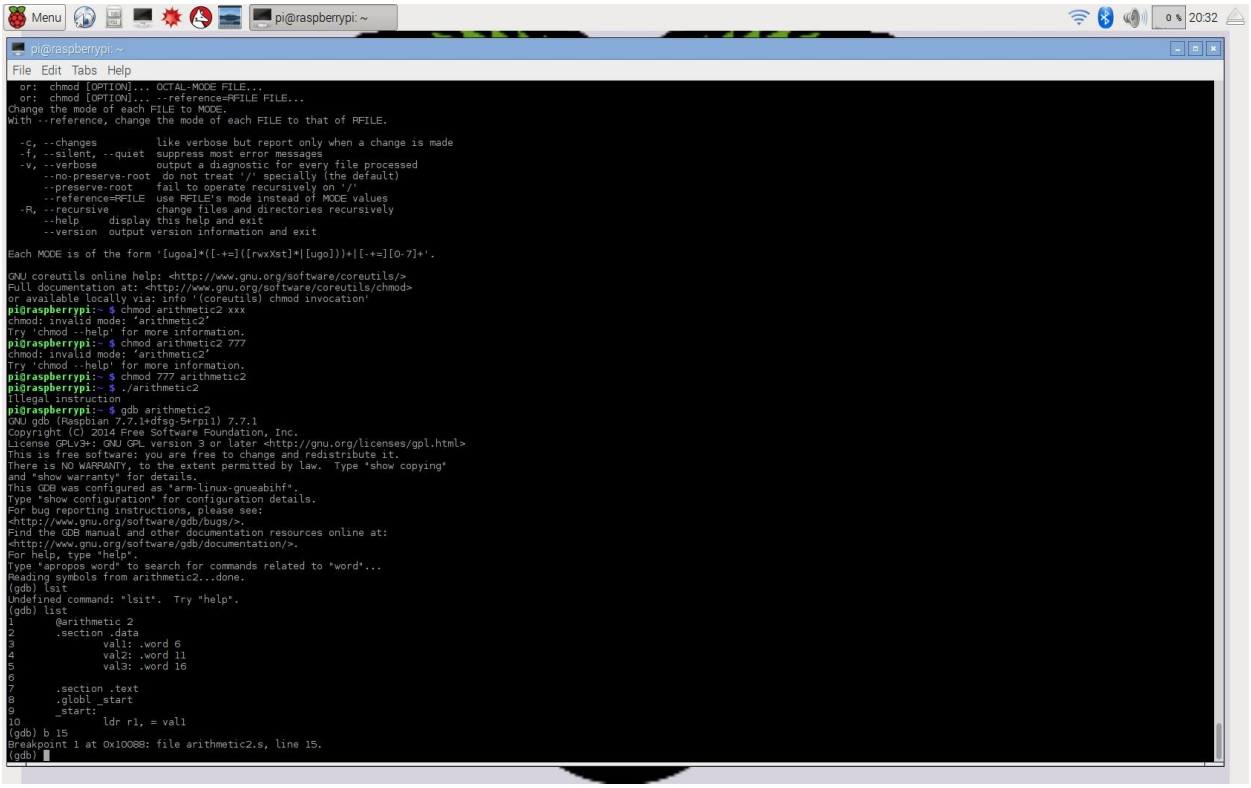
The second line of command trying to run the program, but there is an error message (bash: ./: is a directory). Then I try to debug by adding (-g) to third line of the program.



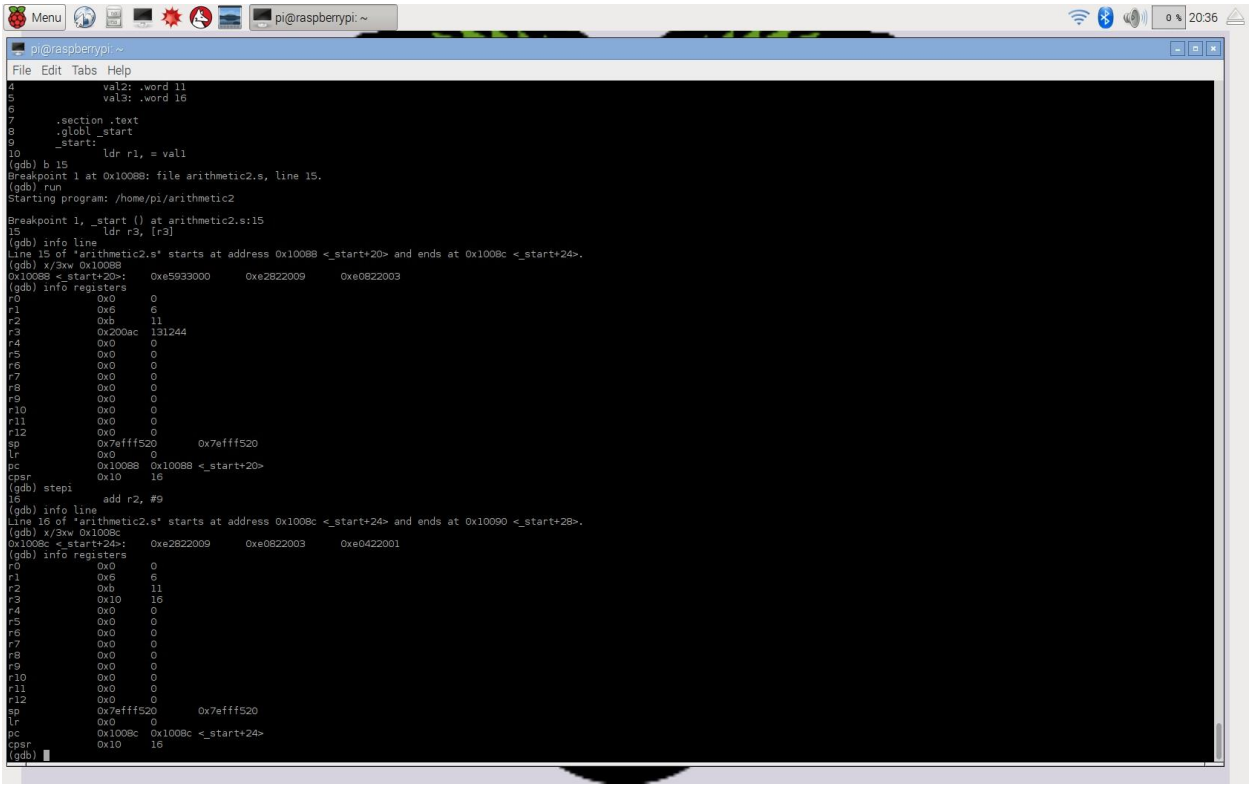
Found out the file did not end with svc #0 and .end which was causing run errors.
Next we use the list command.



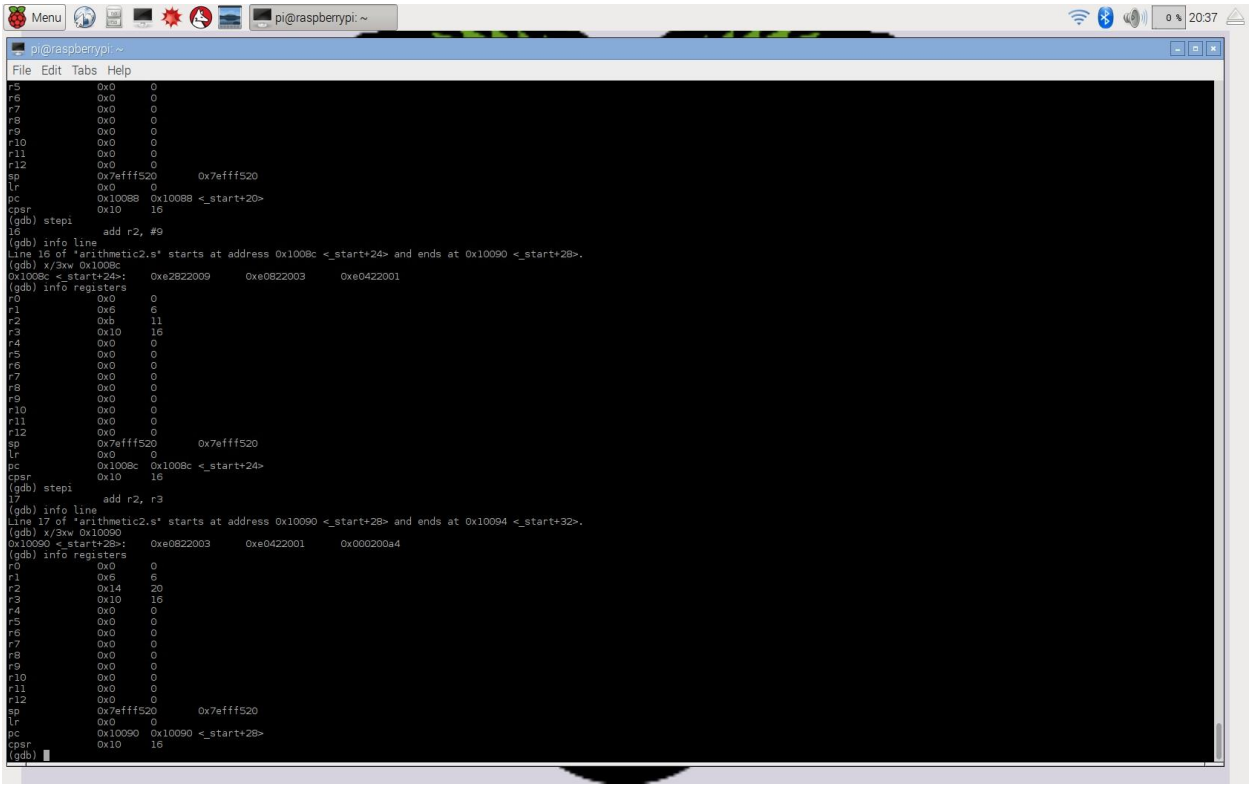
Here we set a breakpoint



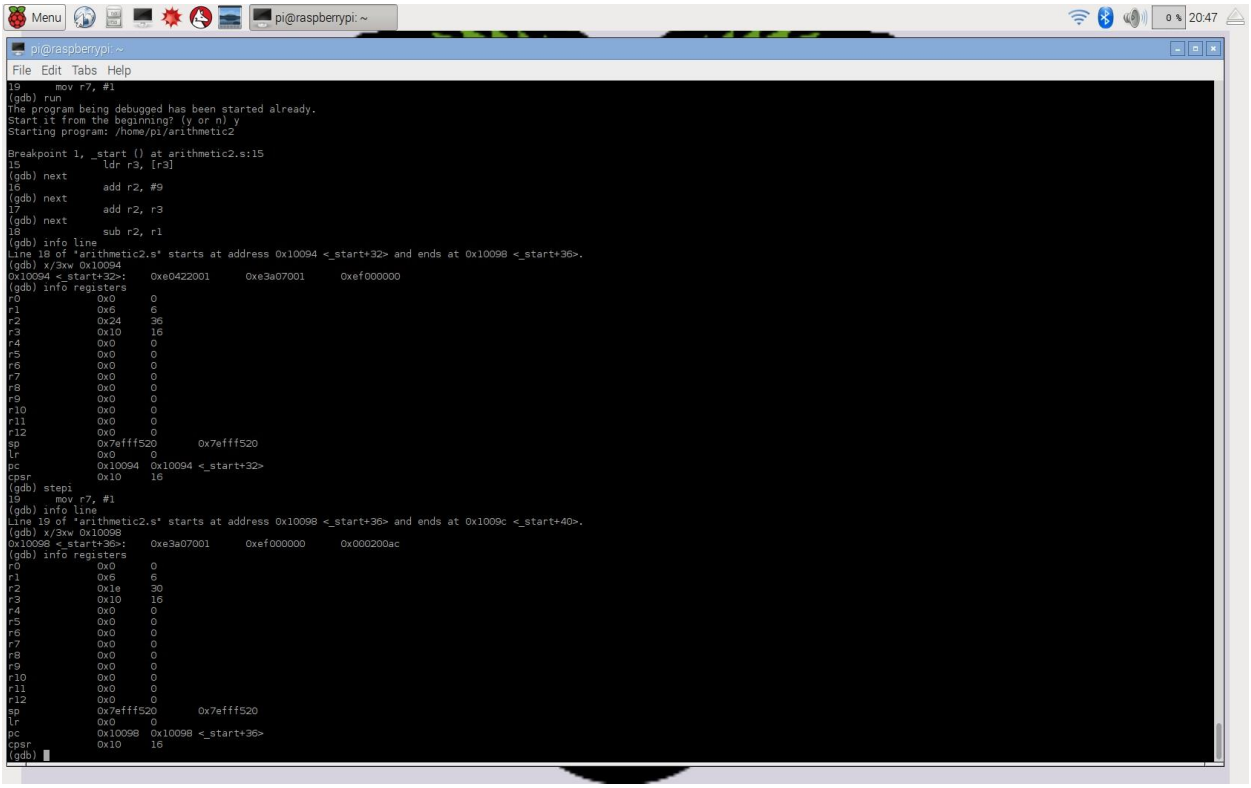
Next we see lines 15-19 executed and then the programs exits and give control to the kernal.
R3 loaded with value



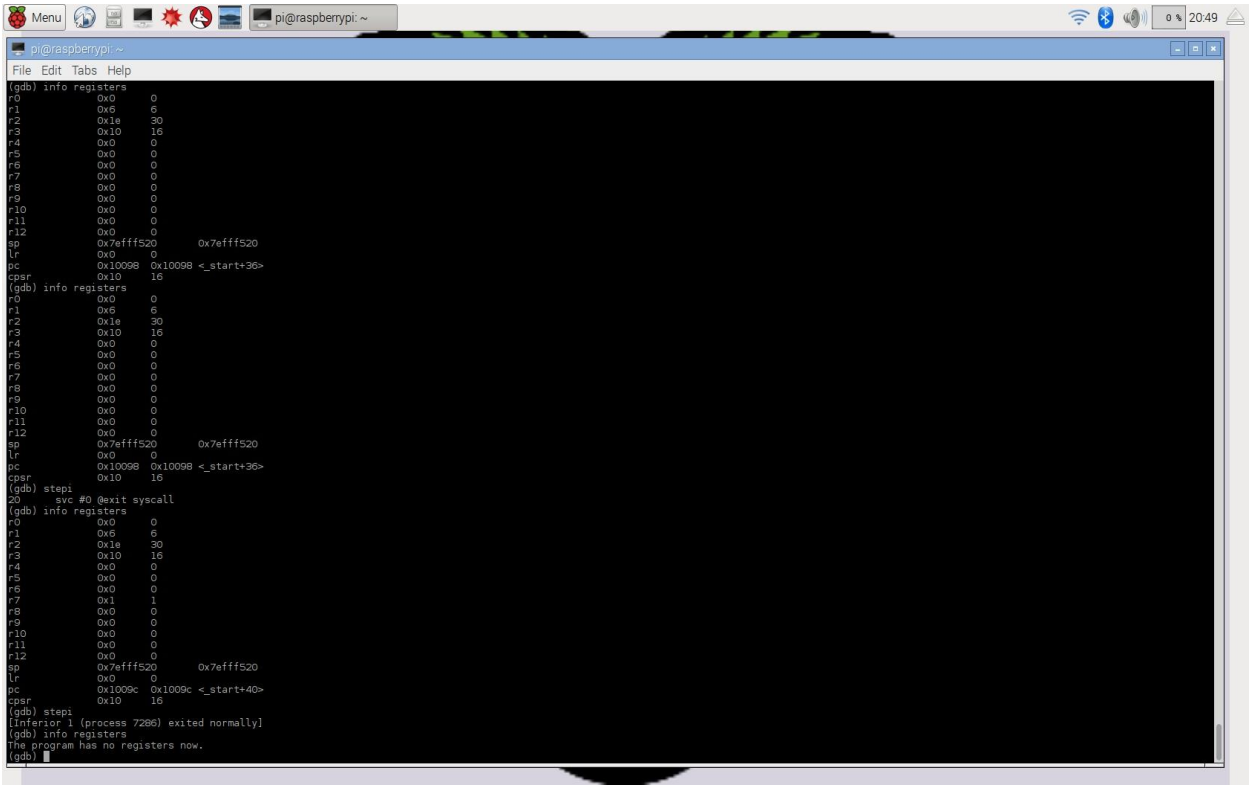
Line 17 at bottom of picture is #9 added to r2



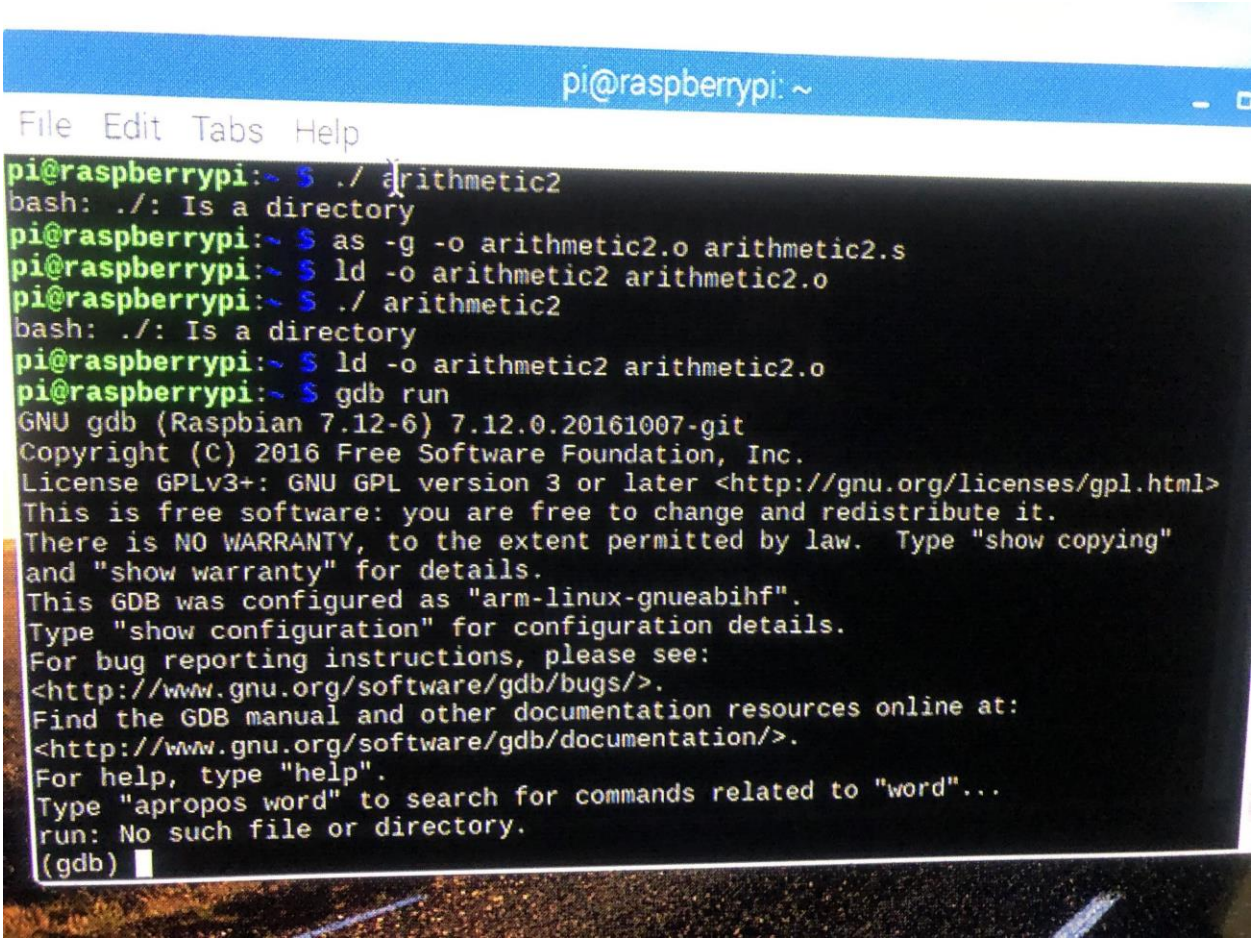
Finally we see a subtraction of r1 from r2 which gives a final answer of 30



We see the program exit after moving 1 to r7.

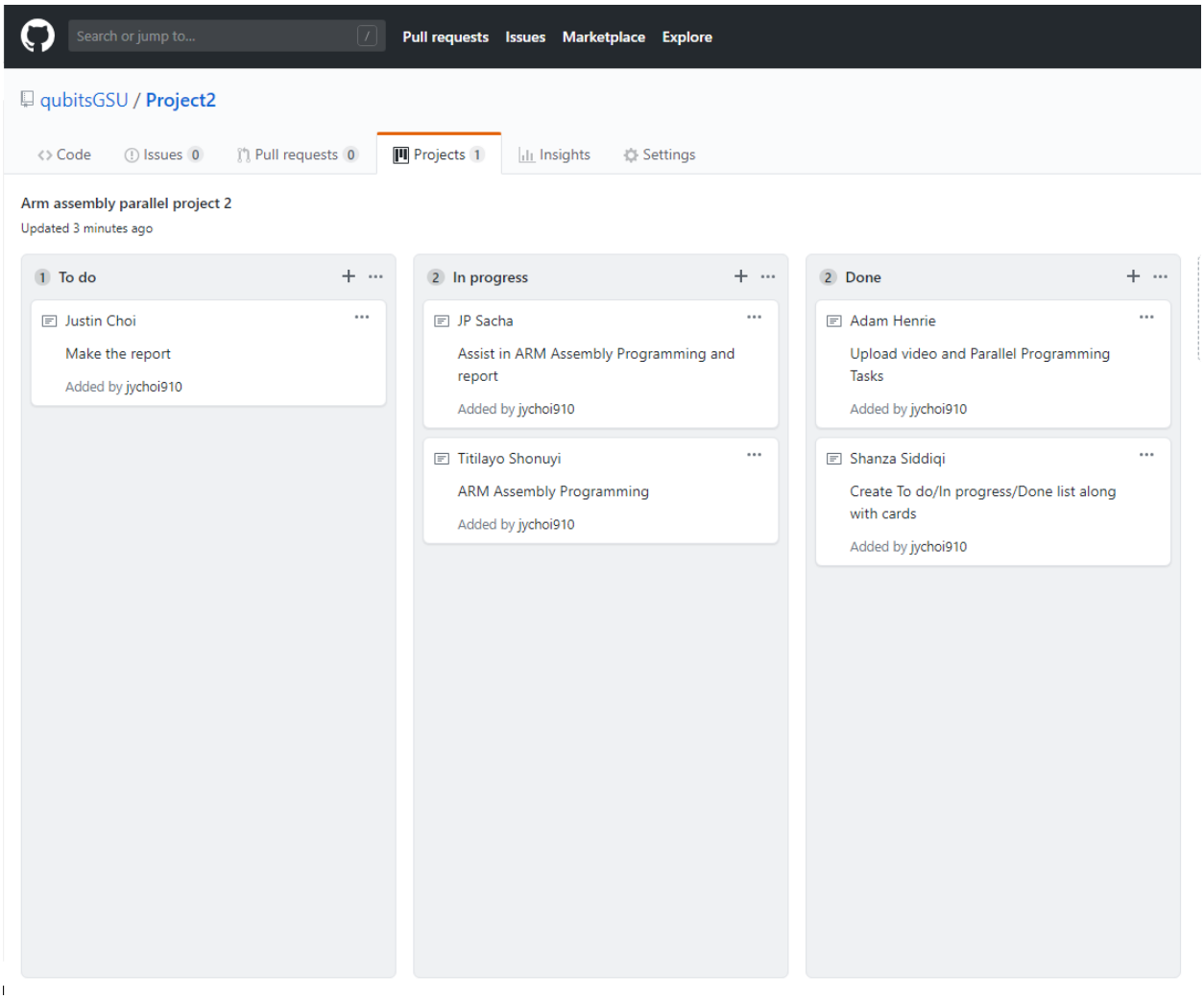


gdb run show the version of the raspberry pi and the license of the software.


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ./ arithmetic2
bash: ./: Is a directory
pi@raspberrypi:~ $ as -g -o arithmetic2.o arithmetic2.s
pi@raspberrypi:~ $ ld -o arithmetic2 arithmetic2.o
pi@raspberrypi:~ $ ./ arithmetic2
bash: ./: Is a directory
pi@raspberrypi:~ $ ld -o arithmetic2 arithmetic2.o
pi@raspberrypi:~ $ gdb run
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabihf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
run: No such file or directory.
(gdb)

Appendix:

GitHub



Important Links:

- **Slack:** computerorganizespr19.slack.com
- **GitHub:** https://github.com/qubitsGSU
- **YouTube: https://www.youtube.com/watch?v=um1jBlF9TRI**