Developing Soft and Parallel Programming Skills Using Project Based Learning

Fall-2019: Jose Diaz, Austin Yuille, Matt Hayes, Nabeeha Ashfaq, Micah Robins

Name	Email	Task	Duration (hours)	Dependency
Jose Diaz	jdiaz28@student.gsu.edu	Create the Slack, organize meetings, create task sheet	6	Slack, Google Sheets
Austin Yuille	ayuille1@student.gsu.edu	Install OS onto Raspberry PI, typed up part of ARM programming	6	Etcher, Rasbian OS
Micah Robins	mrobins1@student.gsu.edu	Create YouTube channel, and direct video presentation, typed up part of ARM programming	6	YouTube
Matt Hayes	mhayes37@student.gsu.edu	Create the GitHub, typed up part of ARM programming	6	GitHub, nano text editor
Nabeeha Ashfaq	nashfaq1@student.gsu.edu	Facilitator, type up Teamwork sheet answers	6	Google Docs

- What to do to get the task accomplished and the team members' satisfaction high?
    1. Get to know other members of your group and their strengths
    2. Set ground rules
    3. Use a facilitator
    4. Keep lines of communication open
    5. Know how to avoid (or solve) common problems

- 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?

-Work will be distributed by the team coordinator and at the beginning of each assignment

Who will set deadlines?

-The team coordinator will set the deadlines. The general rule will be to set deadlines so that the assignment is done four days before the due date.

What happens if someone doesn't follow through on his/her commitment (for example, misses a deadline)?

-Contact them to see what happened. If it seems like they are not going to finish their commitment, someone else will do it or the work will be divided between the remaining members. The person who failed their commitment will have points deducted.

How will the work be reviewed?

-All individual commitments will be reviewed at meetings and all members will review the work when the assignment is completed.

What happens if people have different opinions about the quality of the work?

-The group will take a vote to see what course of action to take.

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).

-As long as each individual task is done by the deadline and everyone shows up to meetings, everything is fine. If a member is unreasonable about a deadline, then points will be deducted.

Facilitator Norms

Will you use a facilitator?

- A facilitator will be chosen if a problem arises that the group cannot vote on.

How will the facilitator be chosen?

-If the group cannot find a solution amongst ourselves, then we will approach an outside party such as a TA or the professor to help us resolve the problem. Otherwise we will rotate the position.

Will you rotate the position?

-Yes

What are the responsibilities of the facilitator?

-They will help make sure that everyone is doing their part and has a say.

Communication Norms

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

-Communication will primarily take place through Slack.

- As a team, select two cases out of the four mentioned in Handling Difficult Behavior.

  Overly Talkative

  If someone is talking too much, wait until they are done and direct the conversation to someone who hasn't spoken.

  Too quiet

  Make sure that before making a decision, everyone's opinion and comments are heard. If a person is being too quiet, an effort to appreciate their efforts will be made.

- When making decisions, If the team is having trouble reaching consensus, what should you do? (use your own words and your own context)

  -The group is having trouble reaching a consensus then voting is the next step. If voting does not help solve the problem then that is when we reach out of the group for a facilitator.

- 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?

-The group is to make decisions as a whole, and if a person reaches theirs more quickly than the others, they will have to wait for the others so everyone's opinion can be included.

- 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?

  -As a general rule, the group is aiming for an "A". If someone does not want to put in the effort, then they will have points deducted and the rest of the group will pick up the slack when reviewing the assignment when it is declared finished.

Setting Up the Raspberry Pi:

The first step of using the Raspberry Pi was installing the Raspbian operating system onto the Raspberry Pi. To do this, we took the given micro SD card and inserted it into one of our laptops that had an SD card reader. We then followed the link provided to us and began downloading the operating system on the computer. While the Raspbian was downloading we also downloaded the program Etcher. Etcher is used to create bootable USB and SD drives, meaning that it will turn the USB or SD drive into the drive that holds the operating system used for booting a computer. When the operating system was finished downloading, we used Etcher to flash Raspbian on the micro SD card making it a boot drive suitable for our Raspberry Pi. We inserted the card into the Pi, and upon seeing the green light flash on we continued onto our next task.

The next step of the set-up process was connecting the Raspberry Pi to a mouse, keyboard, and monitor, or in other words our primary input and output devices. When we first connected the Pi to the monitor via an HDMI cable, the monitor was telling us that there was no HDMI connection found. We began to search for solutions on the internet and found that we simply had to uncomment a line of code in the operating system to fix our issue. After uncommenting the line, "hdmi_force_hotplug=1" from the source code, we reconnected the Pi to the monitor and it worked! We then connected our mouse and keyboard, through the USB ports on Pi, and our set up was complete.

Part 1: First Program:

After setting up the Pi, we were able to begin coding. The first thing we wrote was "first.exe"(see Appendix B5). This program consisted of us doing simple assembly commands such as moving values to different registers, and performing certain arithmetic operations with the numbers stored in the registers. For example, in lines six and seven of our code we stored the integer five into the register, r1, and then subtracted one from the number stored in r1, and stored the resulting value, four, in r1(See Appendix ARM 1, 2 and 5). After writing all of the code for the program we then attempted to assemble and link the program, but encountered an error for "bad instruction"(See Appendix ARM 1, 3) We found that we accidentally used a semicolon instead of a colon after "_start" in the beginning of the code, so it was easily fixed and we could successfully assemble and link our program now(See Appendix ARM 1, 3).

We then began to use GDB, or GNU debugger, to see what exactly our file was doing. We found that we could not use the list command in GDB to display our code because we forgot to add the "-g" flag in the assemble command, so the debugger could not access our source code from the executable file(See Appendix ARM 1, 4). After adding the flag, the list command worked within GDB and we were able to see our source code in the debugger, and we added a breakpoint at line 11 by using the command "b 11"(See Appendix ARM 1, 6). This breakpoint stops breaks the computer's execution cycle(fetch, decode, execute) just before it reaches line 11. This allows us to see what is happening at certain points in the program, which allows for us to test and fix the program. Since line eleven is just before the end of the program, all operations have been executed and the final result of the program, 8, should be stored in register r1. Since the program is stopped at the breakpoint, we can use the command "info registers" to see what is stored in each register. We can see that the result 8 is stored in r1 as we expected(See Appendix ARM 1, 7).

Part 2: Arithmetic Program:

To complete part 2, we had a group meeting in one of the GSU library conference rooms. After hooking up the Raspberry Pi to the monitor, and the keyboard and mouse (provided by Austin Yuille), we were up and running. Using the "First" program as a guide, we sat down as a group and created the code for arithmetic1.s (see Appendix ARM 2, 1). Micah Robins typed out our code, but everyone in the group actively participated by reading through the instructions in iCollege and vocalizing instructions and recommendations about how to use the commands and registers.

We started by analyzing the expression we were asked to program:

*A = (A + B) - (C * D), where A=10, B=11, C=7, and D=2*

Our instructions explicitly stated that we should only use registers to store the variables. So our next step was to load registers with the specified values. Once all 4 variables had been stored in their respective registers (r0, r1, r2, and r3), we added code to perform the operations. Starting within the parentheses first, then performing the subtraction (see Appendix ARM 2, 2). The final result is stored in A (r0). Originally, we had missed the termination calls at the end so we had to go back and add the lines "mov r7, #1" and "svc #0"

After saving our code in the "arithmetic1" program, we assembled, linked, and ran the debugger on our code (see Appendix ARM 2, 3). We then added a breakpoint at line 10 in our code. We ran the "list" command to look at our code, and then ran the program (see Appendix ARM 2, 4). With the "arithmetic1" program running in the GDB debugger, we used the "stepi" command to incrementally work through our program until we reached the final calculation (see Appendix ARM 2, 5). We invoked the "info registers" command twice while stepping through the program; once after the parentheses integers had been calculated (see Appendix ARM 2, 6), and once again after all calculations were complete. As expected, memory A (register r0) has stored a value of 7 (see Appendix ARM 2, 7). Since (10 + 11) - (7 * 2) = 7, we are happy with our result.

Appendix A: Links

Slack: https://app.slack.com/client/TN5SVBQEL/CMUD7K7HR

Github Project: https://github.com/orgs/thesnakes-csc3210/projects/1

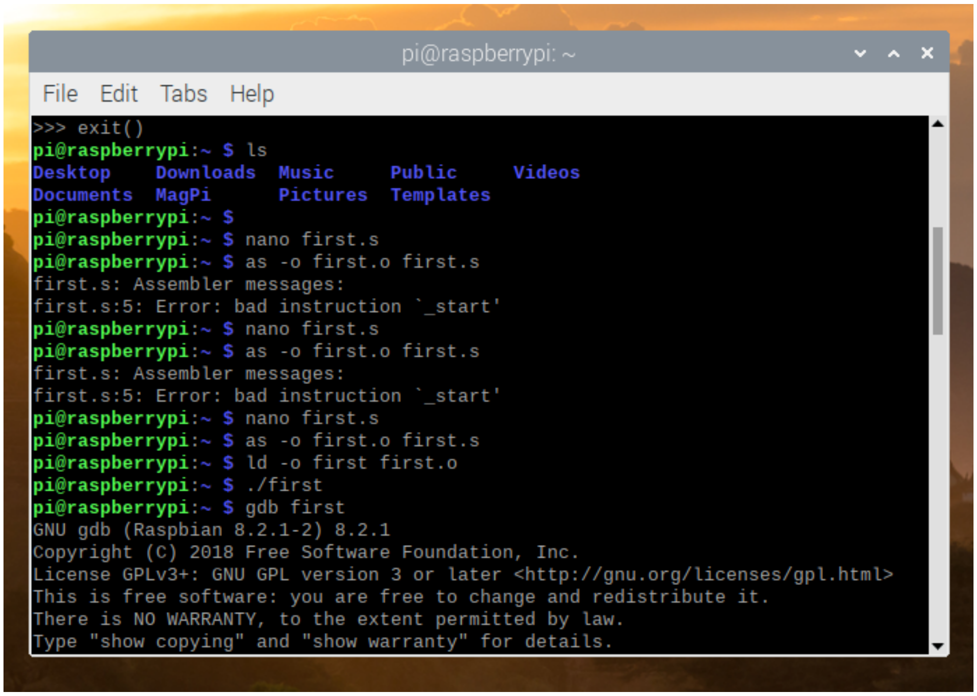
Github Repository: https://github.com/thesnakes-csc3210/projectA1

Video Presentation:

Appendix

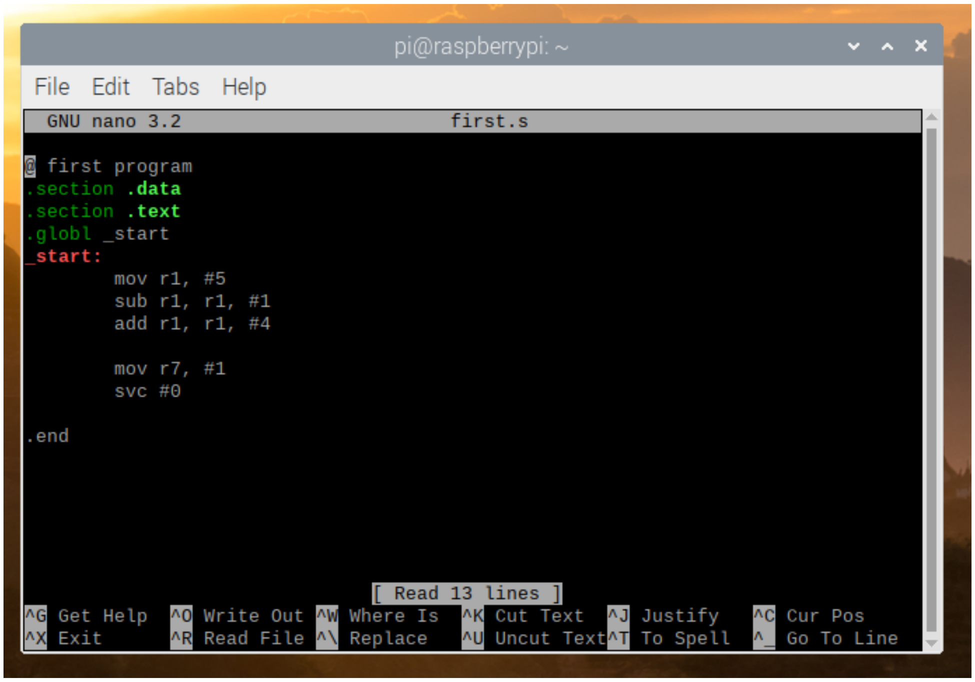
Appendix ARM 1

(ARM 1, 1)



```
>>> exit()
pi@raspberrypi:~ $ ls
Desktop    Downloads  Music     Public     Videos
Documents  MagPi      Pictures  Templates
pi@raspberrypi:~ $
pi@raspberrypi:~ $ nano first.s
pi@raspberrypi:~ $ as -o first.o first.s
first.s: Assembler messages:
first.s:5: Error: bad instruction `_start'
pi@raspberrypi:~ $ nano first.s
pi@raspberrypi:~ $ as -o first.o first.s
first.s: Assembler messages:
first.s:5: Error: bad instruction `_start'
pi@raspberrypi:~ $ nano first.s
pi@raspberrypi:~ $ as -o first.o first.s
pi@raspberrypi:~ $ ld -o first first.o
pi@raspberrypi:~ $ ./first
pi@raspberrypi:~ $ gdb first
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 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.
```

(ARM 1, 2)



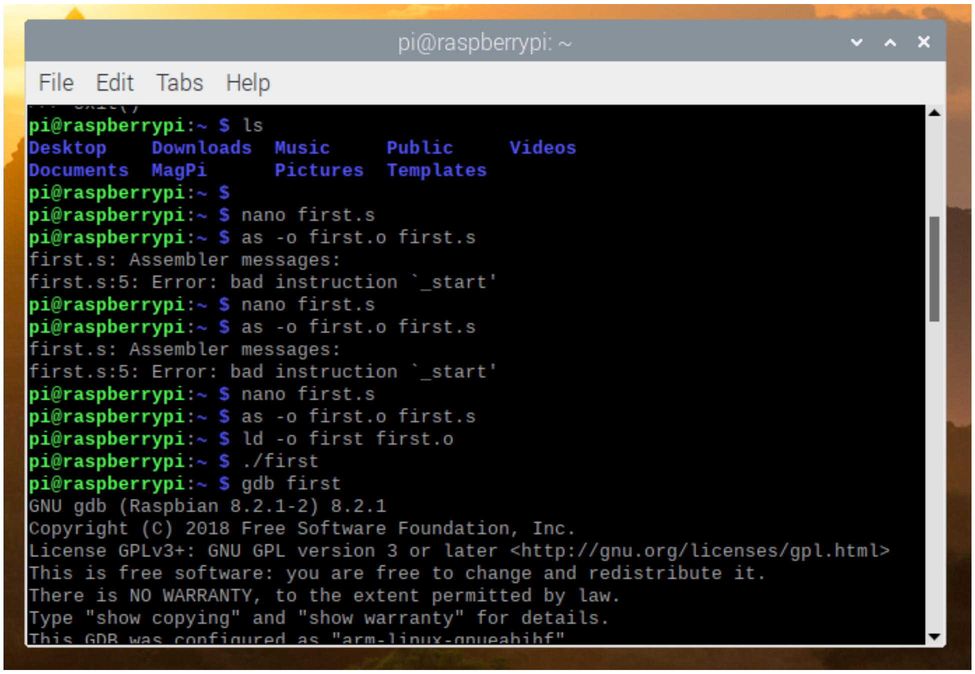
```
  GNU nano 3.2                        first.s

@ first program
.section .data
.section .text
.globl _start
_start:
        mov r1, #5
        sub r1, r1, #1
        add r1, r1, #4

        mov r7, #1
        svc #0

.end
```

```
                         [ Read 13 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Appendix

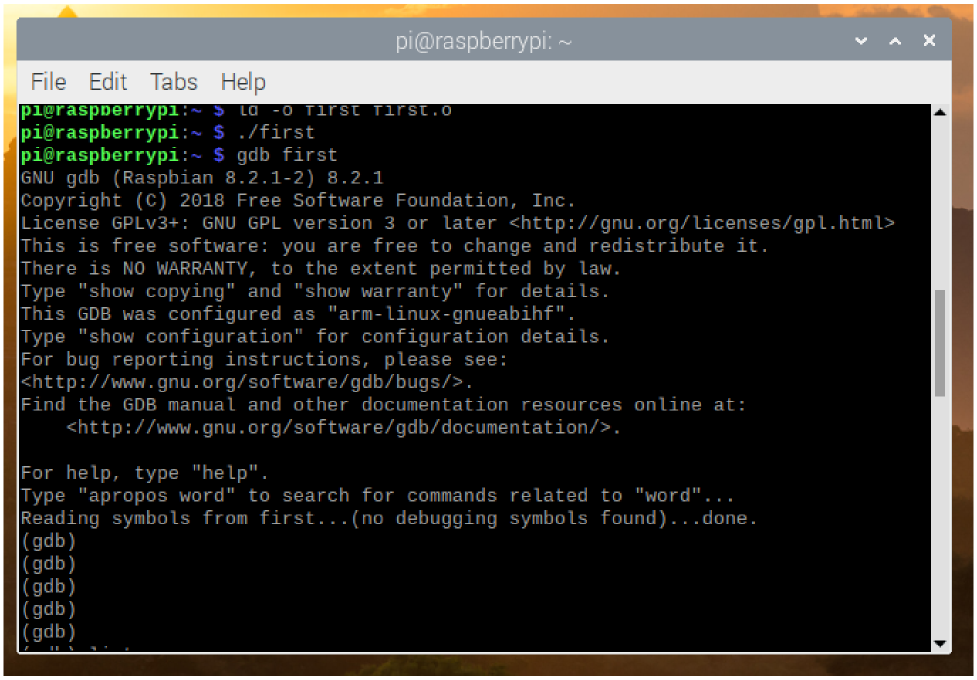
Appendix ARM 1

(ARM 1, 3)



```
pi@raspberrypi:~ $ ls
Desktop    Downloads  Music     Public     Videos
Documents  MagPi      Pictures  Templates
pi@raspberrypi:~ $
pi@raspberrypi:~ $ nano first.s
pi@raspberrypi:~ $ as -o first.o first.s
first.s: Assembler messages:
first.s:5: Error: bad instruction `_start'
pi@raspberrypi:~ $ nano first.s
pi@raspberrypi:~ $ as -o first.o first.s
first.s: Assembler messages:
first.s:5: Error: bad instruction `_start'
pi@raspberrypi:~ $ nano first.s
pi@raspberrypi:~ $ as -o first.o first.s
pi@raspberrypi:~ $ ld -o first first.o
pi@raspberrypi:~ $ ./first
pi@raspberrypi:~ $ gdb first
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 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".
```

(ARM 1, 4)

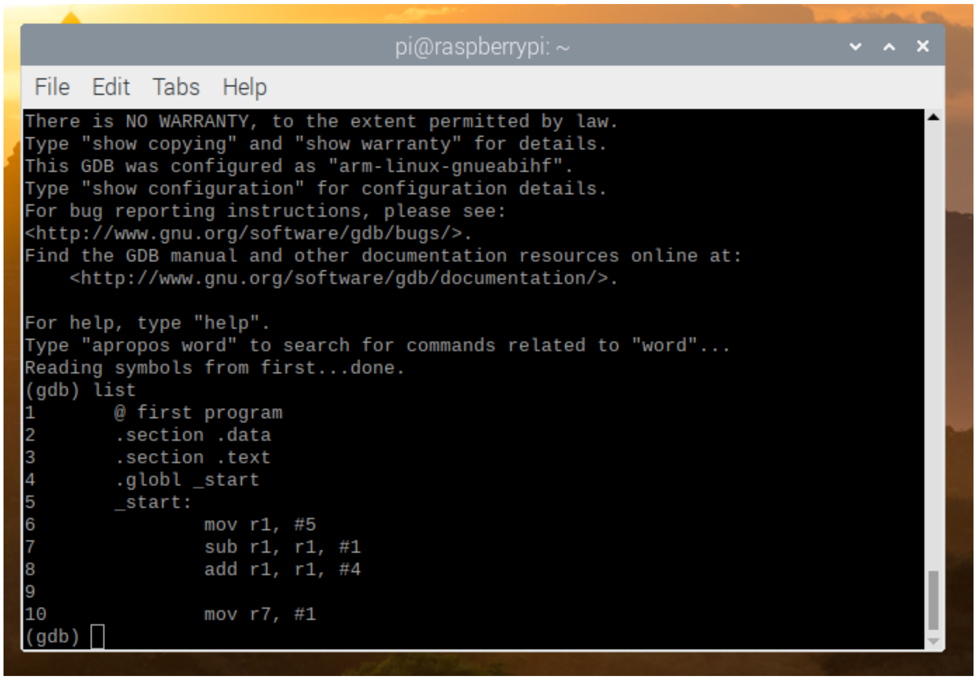


```
pi@raspberrypi:~ $ ld -o first first.o
pi@raspberrypi:~ $ ./first
pi@raspberrypi:~ $ gdb first
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change 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"...
Reading symbols from first...(no debugging symbols found)...done.
(gdb)
(gdb)
(gdb)
(gdb)
(gdb)
(gdb) list
```

Appendix

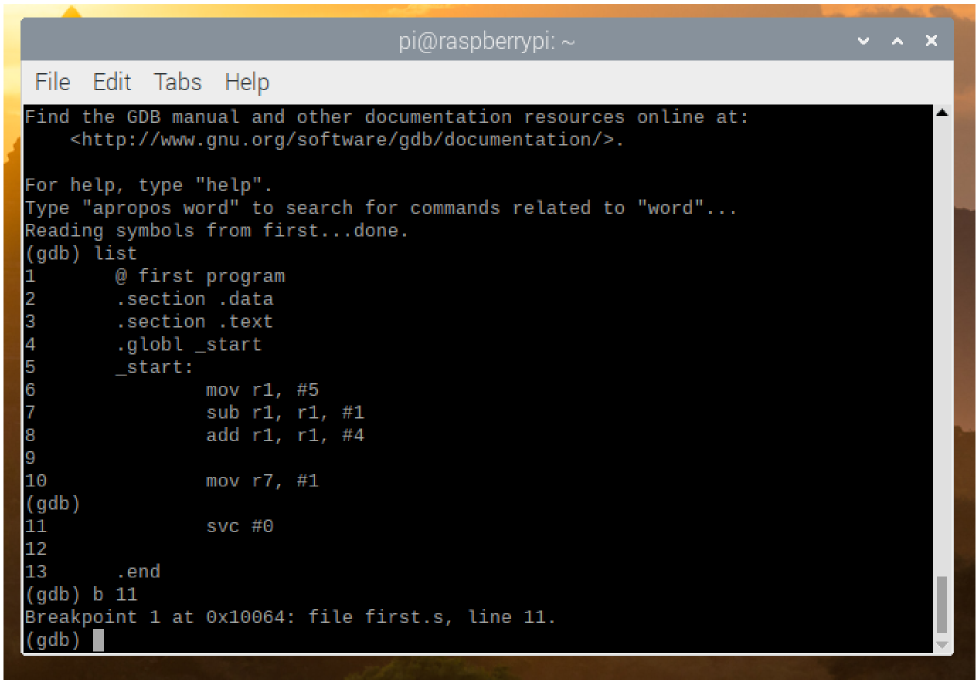
Appendix ARM 1

(ARM 1, 5)



```
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"...
Reading symbols from first...done.
(gdb) list
1       @ first program
2       .section .data
3       .section .text
4       .globl _start
5       _start:
6               mov r1, #5
7               sub r1, r1, #1
8               add r1, r1, #4
9
10              mov r7, #1
(gdb) 
```

(ARM 1, 6)

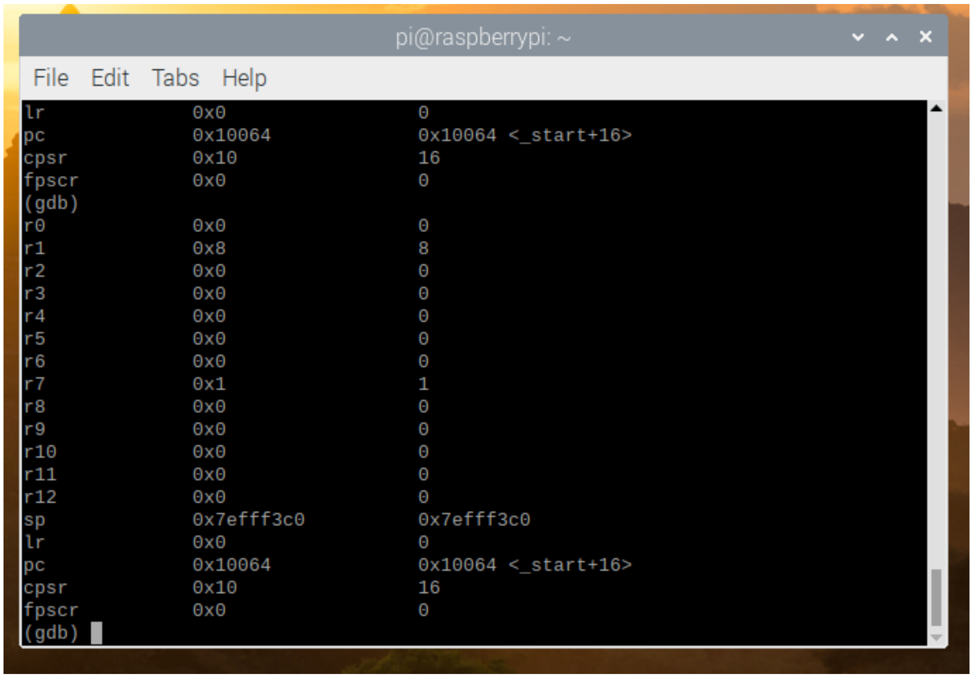


```
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from first...done.
(gdb) list
1       @ first program
2       .section .data
3       .section .text
4       .globl _start
5       _start:
6               mov r1, #5
7               sub r1, r1, #1
8               add r1, r1, #4
9
10              mov r7, #1
(gdb)
11              svc #0
12
13      .end
(gdb) b 11
Breakpoint 1 at 0x10064: file first.s, line 11.
(gdb) 
```

Appendix

Appendix ARM 1

(ARM 1, 7)

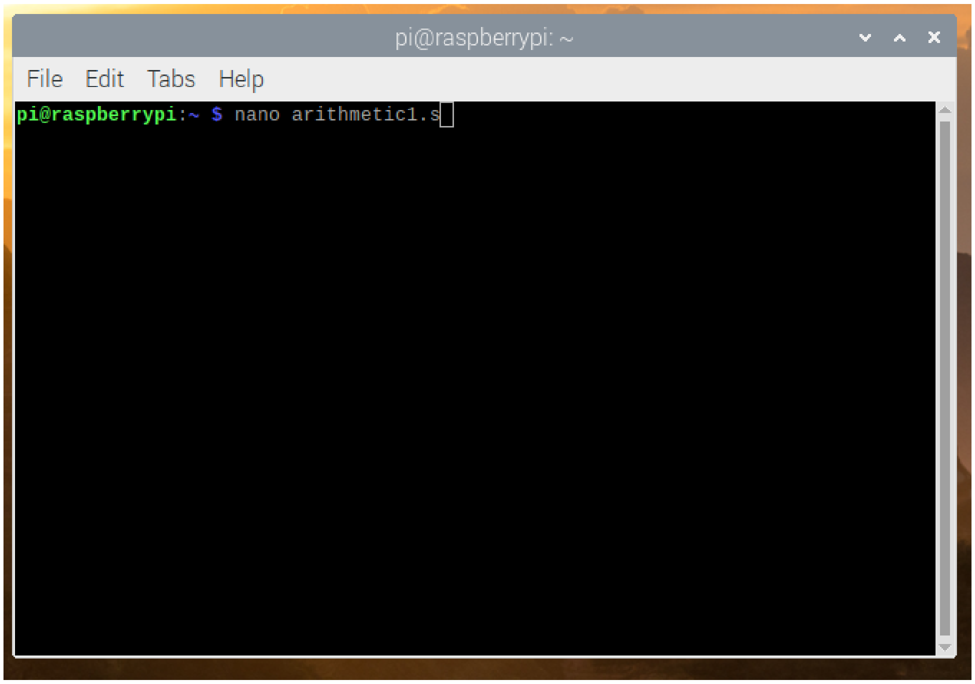


```
lr             0x0                 0
pc             0x10064             0x10064 <_start+16>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
r0             0x0                 0
r1             0x8                 8
r2             0x0                 0
r3             0x0                 0
r4             0x0                 0
r5             0x0                 0
r6             0x0                 0
r7             0x1                 1
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff3c0          0x7efff3c0
lr             0x0                 0
pc             0x10064             0x10064 <_start+16>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

Appendix

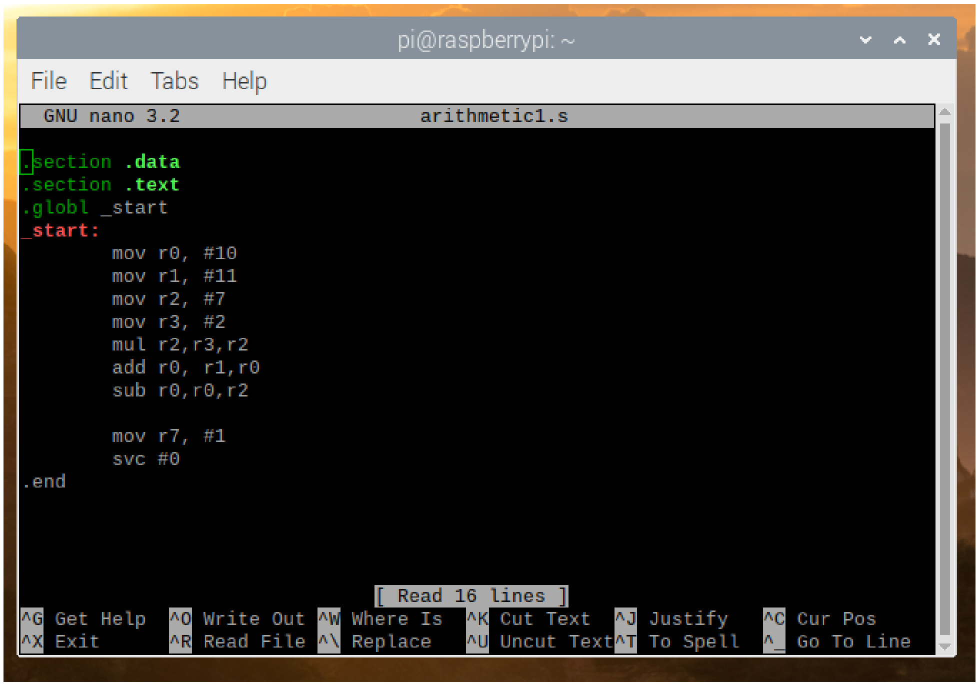
Appendix ARM 2

(ARM 2, 1)



```
pi@raspberrypi:~ $ nano arithmetic1.s
```

(ARM 2, 2)

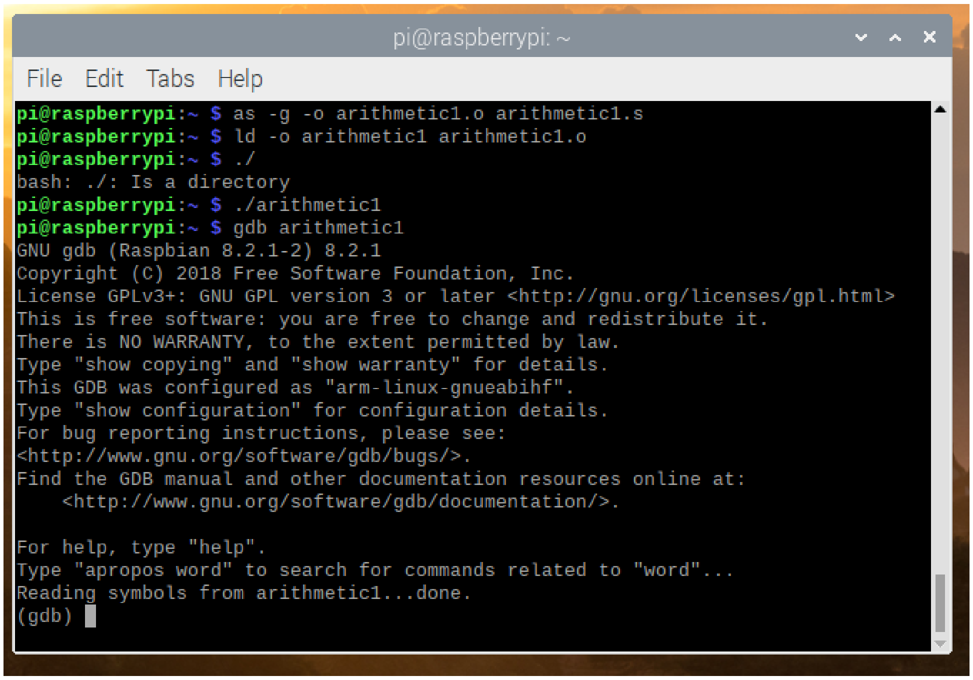


```
  GNU nano 3.2                          arithmetic1.s

.section .data
.section .text
.globl _start
_start:
        mov r0, #10
        mov r1, #11
        mov r2, #7
        mov r3, #2
        mul r2,r3,r2
        add r0, r1,r0
        sub r0,r0,r2

        mov r7, #1
        svc #0
.end

                             [ Read 16 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Appendix

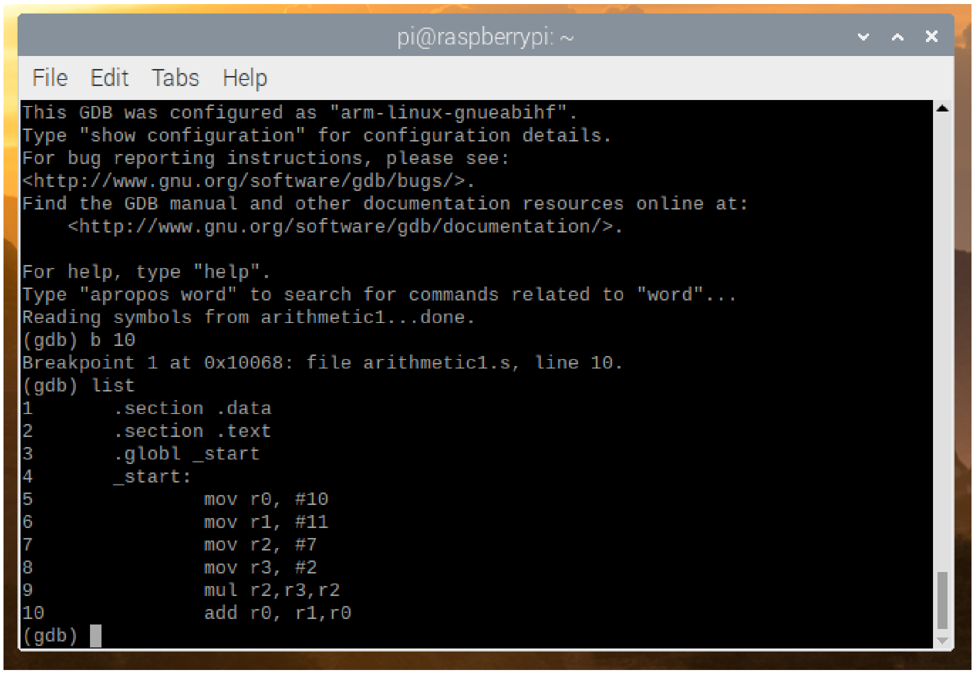
Appendix ARM 2

(ARM 2, 3)



```
pi@raspberrypi:~ $ as -g -o arithmetic1.o arithmetic1.s
pi@raspberrypi:~ $ ld -o arithmetic1 arithmetic1.o
pi@raspberrypi:~ $ ./
bash: ./: Is a directory
pi@raspberrypi:~ $ ./arithmetic1
pi@raspberrypi:~ $ gdb arithmetic1
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 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"...
Reading symbols from arithmetic1...done.
(gdb) 
```

(ARM 2, 4)

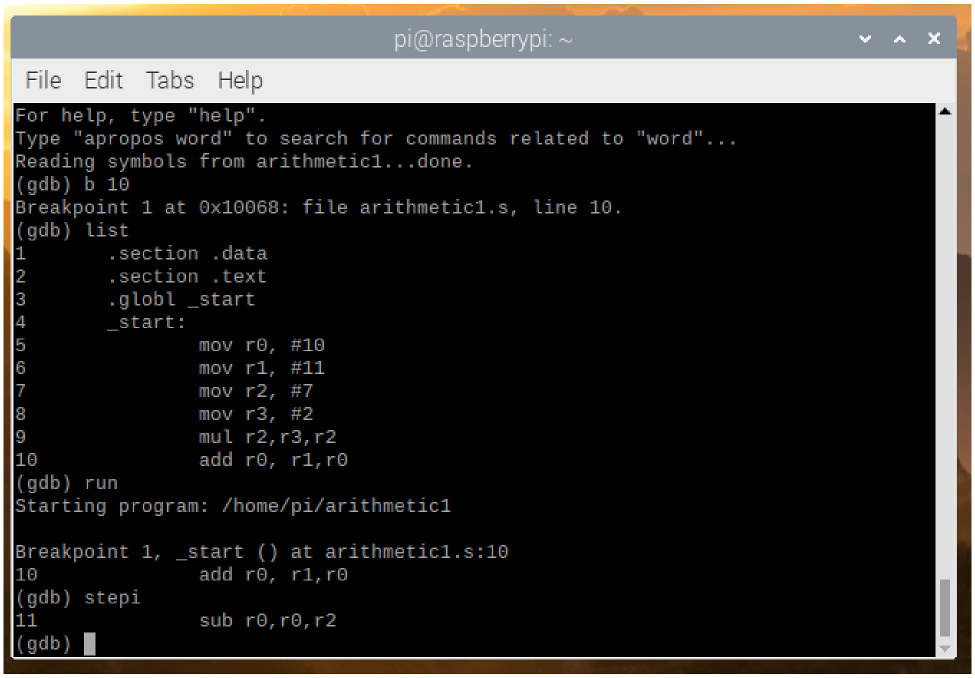


```
This GDB was configured as "arm-linux-gnueabihf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from arithmetic1...done.
(gdb) b 10
Breakpoint 1 at 0x10068: file arithmetic1.s, line 10.
(gdb) list
1       .section .data
2       .section .text
3       .globl _start
4       _start:
5               mov r0, #10
6               mov r1, #11
7               mov r2, #7
8               mov r3, #2
9               mul r2,r3,r2
10              add r0, r1,r0
(gdb) 
```

Appendix

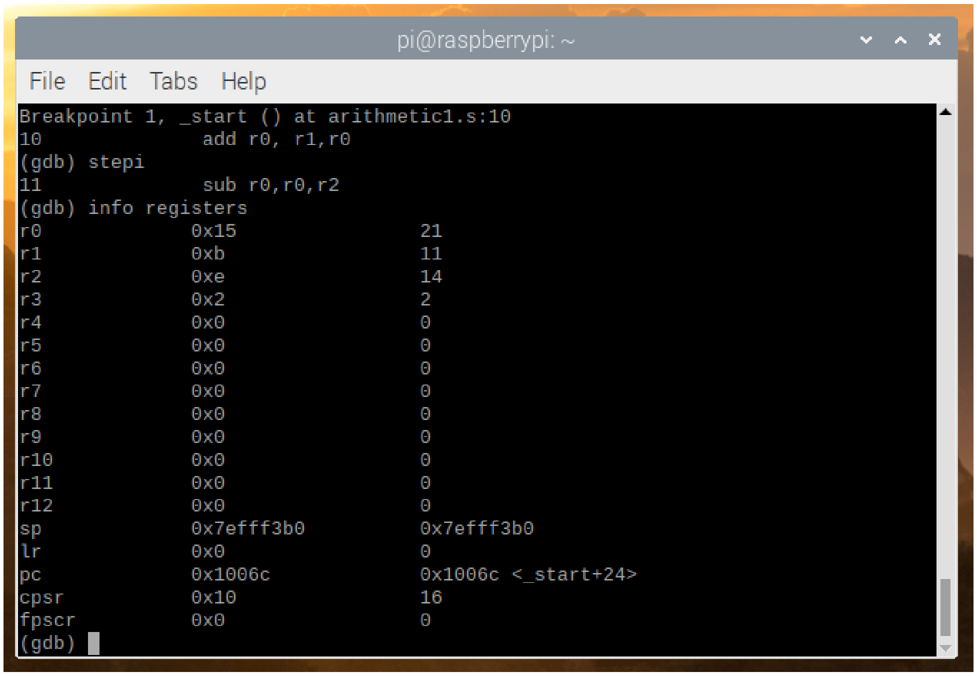
Appendix ARM 2

(ARM 2, 5)



```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from arithmetic1...done.
(gdb) b 10
Breakpoint 1 at 0x10068: file arithmetic1.s, line 10.
(gdb) list
1       .section .data
2       .section .text
3       .globl _start
4       _start:
5               mov r0, #10
6               mov r1, #11
7               mov r2, #7
8               mov r3, #2
9               mul r2,r3,r2
10              add r0, r1,r0
(gdb) run
Starting program: /home/pi/arithmetic1

Breakpoint 1, _start () at arithmetic1.s:10
10              add r0, r1,r0
(gdb) stepi
11              sub r0,r0,r2
(gdb) 
```

(ARM 2, 6)

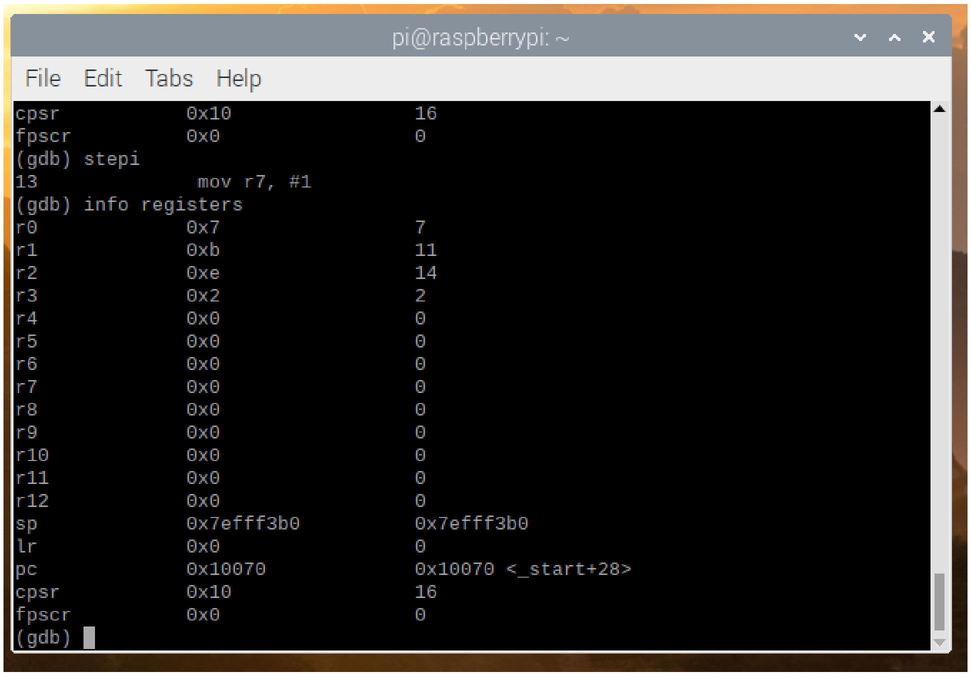


```
Breakpoint 1, _start () at arithmetic1.s:10
10              add r0, r1,r0
(gdb) stepi
11              sub r0,r0,r2
(gdb) info registers
r0             0x15                21
r1             0xb                 11
r2             0xe                 14
r3             0x2                 2
r4             0x0                 0
r5             0x0                 0
r6             0x0                 0
r7             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff3b0          0x7efff3b0
lr             0x0                 0
pc             0x1006c             0x1006c <_start+24>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

Appendix

Appendix ARM 2

(ARM 2, 7)

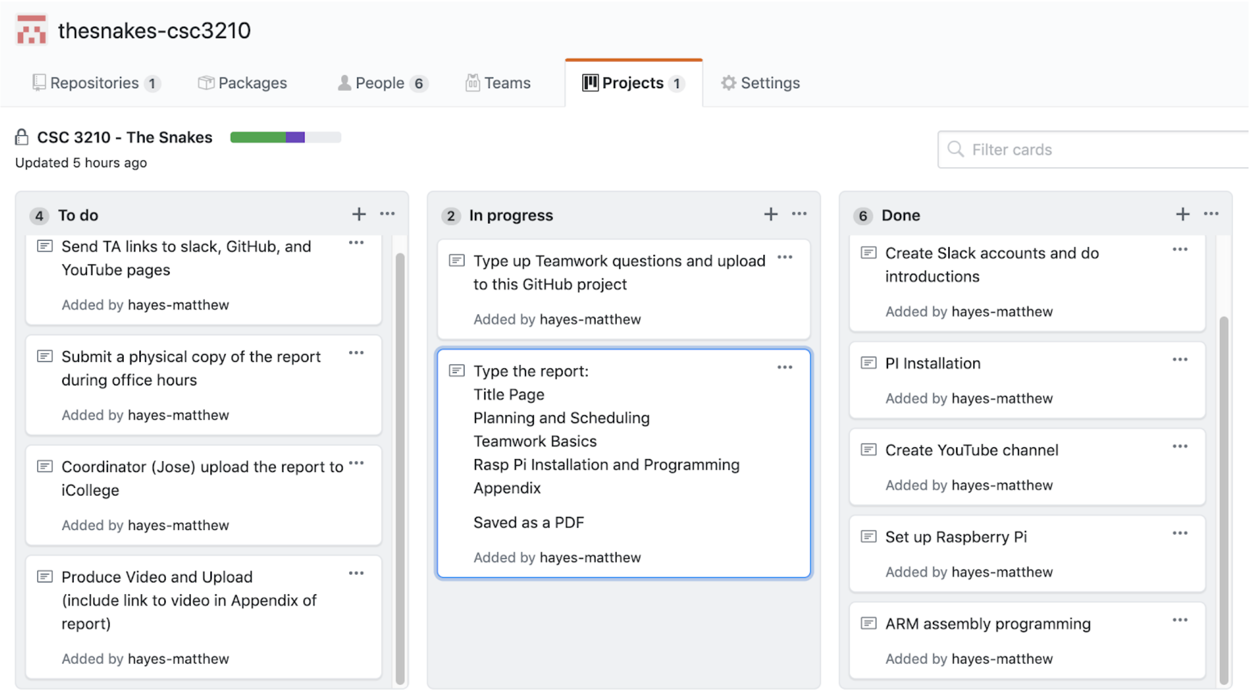


```
cpsr           0x10                16
fpscr          0x0                 0
(gdb) stepi
13              mov r7, #1
(gdb) info registers
r0             0x7                 7
r1             0xb                 11
r2             0xe                 14
r3             0x2                 2
r4             0x0                 0
r5             0x0                 0
r6             0x0                 0
r7             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff3b0          0x7efff3b0
lr             0x0                 0
pc             0x10070             0x10070 <_start+28>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

Appendix

Appendix B: Screenshots

(B1) Github Project Screenshot



CSC 3210 - The Snakes
Updated 5 hours ago

4 To do

Send TA links to slack, GitHub, and YouTube pages
Added by hayes-matthew

Submit a physical copy of the report during office hours
Added by hayes-matthew

Coordinator (Jose) upload the report to iCollege
Added by hayes-matthew

Produce Video and Upload (include link to video in Appendix of report)
Added by hayes-matthew

2 In progress

Type up Teamwork questions and upload to this GitHub project
Added by hayes-matthew

Type the report:
Title Page
Planning and Scheduling
Teamwork Basics
Rasp Pi Installation and Programming
Appendix

Saved as a PDF

Added by hayes-matthew

6 Done

Create Slack accounts and do introductions
Added by hayes-matthew

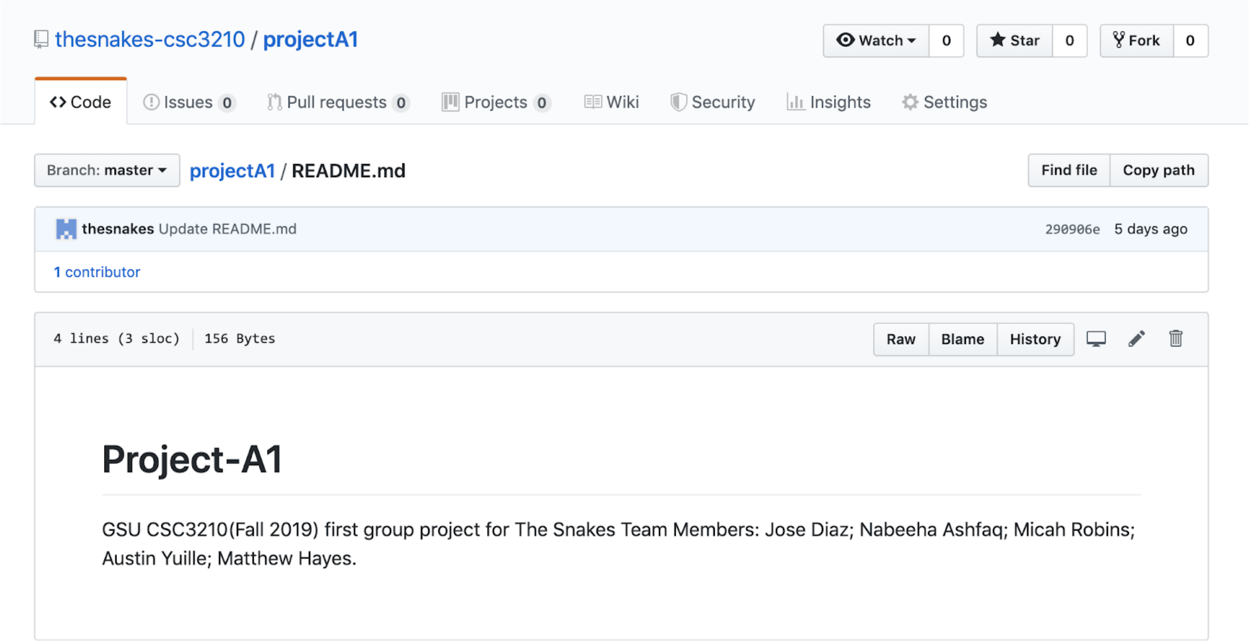
PI Installation
Added by hayes-matthew

Create YouTube channel
Added by hayes-matthew

Set up Raspberry Pi
Added by hayes-matthew

ARM assembly programming
Added by hayes-matthew

(B2) Github README screenshot



projectA1 / README.md

thesnakes Update README.md 290906e 5 days ago

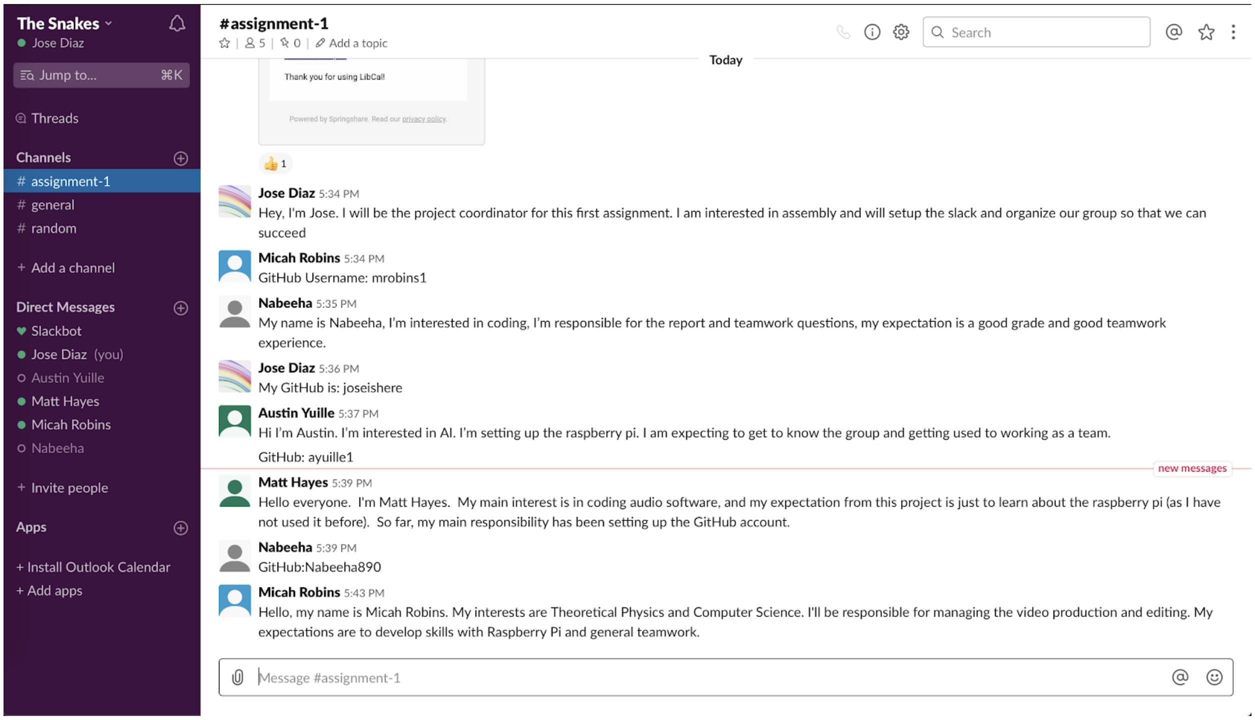
1 contributor

4 lines (3 sloc) | 156 Bytes

Project-A1

GSU CSC3210(Fall 2019) first group project for The Snakes Team Members: Jose Diaz; Nabeeha Ashfaq; Micah Robins; Austin Yuille; Matthew Hayes.

(B3) Slack Screenshot



Jose Diaz 5:34 PM
Hey, I'm Jose. I will be the project coordinator for this first assignment. I am interested in assembly and will setup the slack and organize our group so that we can succeed

Micah Robins 5:34 PM
GitHub Username: mrobins1

Nabeeha 5:35 PM
My name is Nabeeha, I'm interested in coding, I'm responsible for the report and teamwork questions, my expectation is a good grade and good teamwork experience.

Jose Diaz 5:36 PM
My GitHub is: joseishere

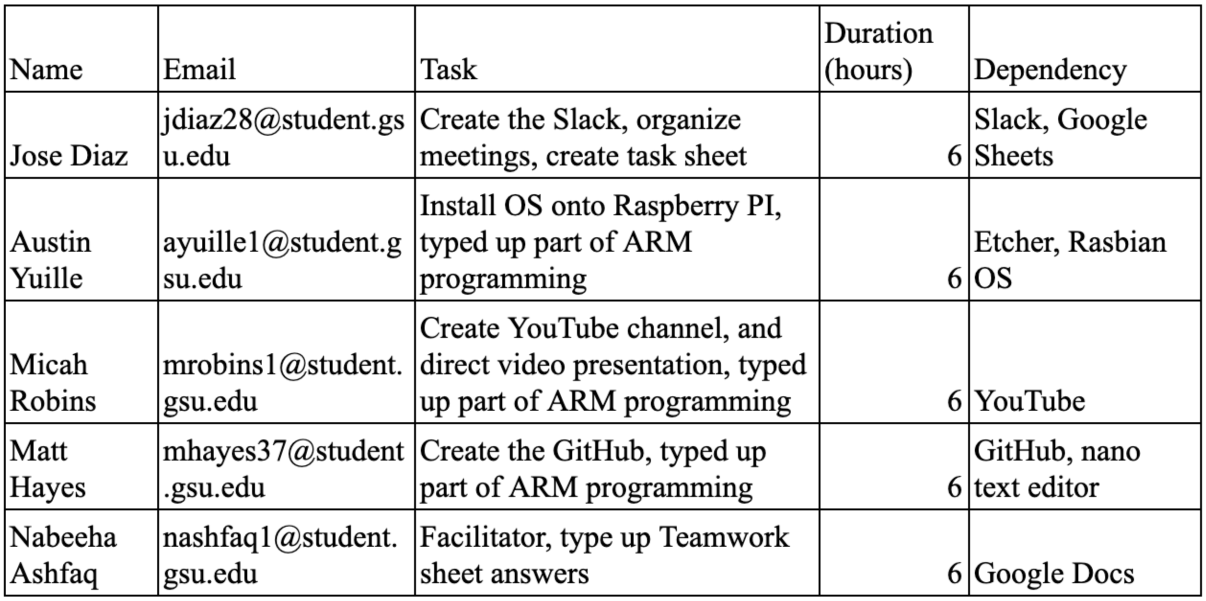
Austin Yuille 5:37 PM
Hi I'm Austin. I'm interested in AI. I'm setting up the raspberry pi. I am expecting to get to know the group and getting used to working as a team.
GitHub: ayuille1

Matt Hayes 5:39 PM
Hello everyone. I'm Matt Hayes. My main interest is in coding audio software, and my expectation from this project is just to learn about the raspberry pi (as I have not used it before). So far, my main responsibility has been setting up the GitHub account.

Nabeeha 5:39 PM
GitHub:Nabeeha890

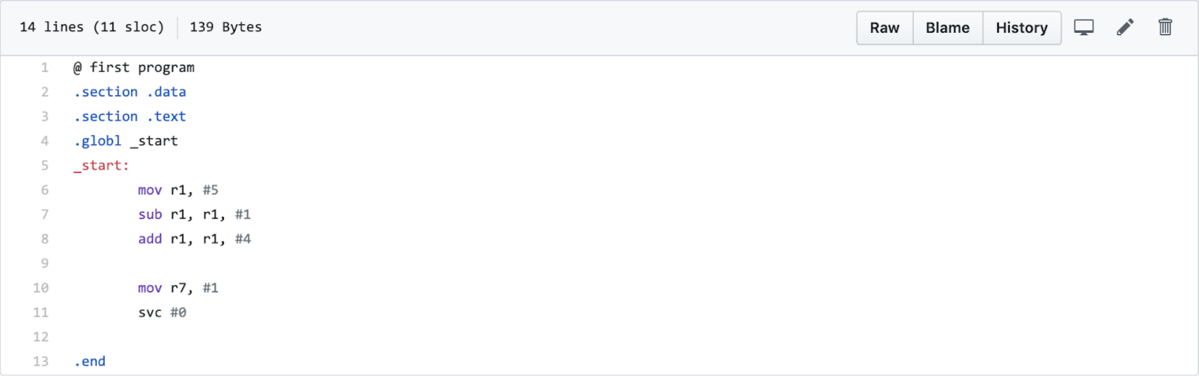
Micah Robins 5:43 PM
Hello, my name is Micah Robins. My interests are Theoretical Physics and Computer Science. I'll be responsible for managing the video production and editing. My expectations are to develop skills with Raspberry Pi and general teamwork.

(B4) Task Sheet Screenshot



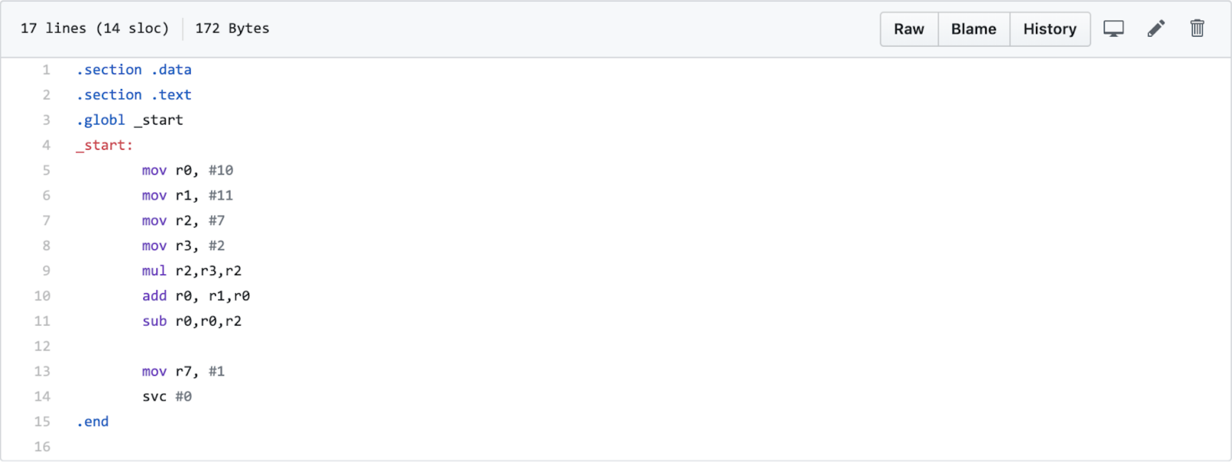
Name	Email	Task	Duration (hours)	Dependency
Jose Diaz	jdiaz28@student.gsu.edu	Create the Slack, organize meetings, create task sheet	6	Slack, Google Sheets
Austin Yuille	ayuille1@student.gsu.edu	Install OS onto Raspberry PI, typed up part of ARM programming	6	Etcher, Rasbian OS
Micah Robins	mrobins1@student.gsu.edu	Create YouTube channel, and direct video presentation, typed up part of ARM programming	6	YouTube
Matt Hayes	mhayes37@student.gsu.edu	Create the GitHub, typed up part of ARM programming	6	GitHub, nano text editor
Nabeeha Ashfaq	nashfaq1@student.gsu.edu	Facilitator, type up Teamwork sheet answers	6	Google Docs

(B5) first.s screenshot



```
@ first program
.section .data
.section .text
.globl _start
_start:
        mov r1, #5
        sub r1, r1, #1
        add r1, r1, #4

        mov r7, #1
        svc #0

.end
```

(B6) arithmetic1.s screenshot



```
.section .data
.section .text
.globl _start
_start:
        mov r0, #10
        mov r1, #11
        mov r2, #7
        mov r3, #2
        mul r2,r3,r2
        add r0, r1,r0
        sub r0,r0,r2

        mov r7, #1
        svc #0
.end

```