Developing Soft and Parallel Programming Skills Using Project-Based Learning

By Assembly Chefs

Team Coordinators:
Andy Lee

Members:
Mezemir Gebre
Nathan Heckman
Bryanna Hardy
Rahul Sunkara

Contents

Introduction:

Project Cover Page pg.1

Contents Page 1 pg.2

Contents Page 2 pg.3

Planning and Scheduling pg.4

Bryanna Hardy Report Section:

Bryanna Hardy Title Page pg.5

Bryanna Hardy Teamwork Basics Answer 1 pg.6

Bryanna Hardy Teamwork Basics Answer 2 pg.7

ARM assembly Programming 1 pg.8

ARM assembly Programming 2 pg.9

Andy Lee Report Section:

Andy Lee Title Page pg.10

Andy Lee Teamwork Basics Answer 1 pg.11

Andy Lee Teamwork Basics Answer 2 pg.12

ARM assembly Programming 1 pg.13

ARM assembly Programming 2 pg.14

ARM assembly Programming 3 pg.15

Rahul Sunkara Report Section:

Rahul Sunkara Title Page pg.16

Rahul Sunkara Teamwork Basics Answer 1 pg.17

Rahul Sunkara Teamwork Basics Answer 2 pg.18

ARM assembly Programming 1 pg.19

ARM assembly Programming 2 pg.20

Nathan Heckman Report Section:

Nathan Heckman Title Page pg.21

Nathan Heckman Teamwork Basics Answer 1 pg.22

ARM assembly Programming 1 pg.23

ARM assembly Programming 2,3,4,5 pg.24 ~ pg.27

Mezemir Gebre Report Section:

Mezemir Gebre Title Page pg.28

ARM assembly Programming pg.?

Nathan Heckman Title Page pg.?

Extra Info:

Appendix: links pg.?

MORE pg.?

REPORT: Planning and Scheduling

Team: Assembly Chefs

Member's name	Email	Task	Due Date
Andy Lee	alee162@student.gsu.edu	Create Team Schedule Table, summarize all the report, Install Raspberry Pi and connect to the PC.	2/3/20
Rahul Sunkara	rsunkara3@student.gsu.edu	Create Slack Account, Screenshot the introduction, Install Raspberry Pi, and connect to the PC.	2/3/20
Bryanna Hardy	bhardy11@student.gsu.edu	Create GitHub Account, Install Raspberry Pi and connect to the PC.	2/3/20
Nathan Heckman	nheckman1@student.gsu.edu	Create YouTube Account, Install Raspberry Pi and connect to the PC.	2/3/20
Mezemir Gebre	mgebre1@student.gsu.edu	Record and Edit presentation video, Install Raspberry Pi and connect to the PC.	2/3/20

Member's name	Duration	Dependency	Note
Andy Lee	Meeting: 1:30 hours Task: 1:00 hours	Microsoft Words, Raspberry Pi	Done all task that was assigned 100% Grade
Rahul Sunkara	Meeting: 1:30 hours Task: 1:00 hours	Slack, Raspberry Pi	Done all task that was assigned 100% Grade
Bryanna Hardy	Meeting: 1:30 hours Task: 1:00 hours	GitHub, Raspberry Pi	Done all task that was assigned 100% Grade
Nathan Heckman	Meeting: 1:30 hours Task: 1:00 hours	YouTube, Raspberry Pi	Done all task that was assigned 100% Grade
Mezemir Gebre	Meeting: 1:30 hours Task: 1:00 hours	Movie Editor, Raspberry Pi	Done all task that was assigned 100% Grade

Summary:

The Project A1 was required to communicate with team members to successfully achieved the final goal. The mains tasks of Project A1 were 1) create the schedule table, 2) answer the "TEAM BASICS" as a team, 3) Create necessary web-accounts, and 4) record the video for presentation. Among many difficult tasks, our team members had equally divided the work to ensure everyone in our group participates the project in some way.

As written in planning table, **Andy Lee**, the team coordinator, takes responsibility on creating the planning table, and making report. **Rahul Sunkara** was responsible to create the "slack" accounts along with other tasks. **Bryanna Hardy** takes responsibility on creating "GitHub" account, and answer "team-base basics" questions. **Nathan Heckman** was responsible to create "YouTube" accounts and prepare for the task 6: presentation. Last but not least, **Mezemir Gebre** was responsible to edit/modify the video we took for presentation.

BRYANNA HARDY INDIVIDUAL REPORT

THIS PAGE IS PURPOSELY LEVAVED AS BLANK

REPORT: Team Basics

!!!Important!!!
The Answer Below is from "Team Basics"

Document Start from Page Here.
Task 3: Teamwork Basics

Andy Lee, Bryanna Hardy, Mezemir Gebre, Nathan Heckman, & Rahul Sunkara

1. What to do to get the task accomplished and the team members' satisfaction high?
   a. Get to know other members of your group and their strengths.
   b. Set ground rules.
   c. Use a facilitator.
   d. Keep lines of communication open.
   e. Know how to avoid (or solve) common problems.
2. How will work be distributed?
   a. We will distribute the work evenly and using the factor based off of individual strengths.
3. Who will set deadlines?
   a. The team coordinator will set deadlines.
4. What happens who does not follow through the commitment?
   a. Communicate with them.
5. How will the work be reviewed?
   a. On our Monday's meetings.
6. What happens if people have different opinions?
   a. Democracy and we will find a common ground.
7. Different work habits?
   a. Communicate with tea m member
8. Will you use a facilitator?
   a. Yes
9. Will you rotate the position?
   a. Yes
10. What are the responsibilities?
    a. Create the table and turn in report to the professor
    b. *LOOK AT THE BOTTOM OF PDF*
11. When should communication takes place and through what medium?
    a. We will use our GroupMe group chat and use Slack that we created.
12. What is everyone's schedule?
    a. Look at groupMe
13. Should one person be responsible for coordinating meetings?
    a. Yes, the team coordinator will be responsible for coordinating meetings.
14. Do people have a preference for when meetings…
    a. Yes.

15. Where is a good place to hold the meeting?
    a. Library
16. What happens if people are late to a meeting?
    a. Give a warning for the first time. After the first time, we will communicate with them.
17. What happens if a group member misses a meeting?
    a. Update them about what happened and communicate.
18. Serval meetings?
    a. Communicate with professor and communication.
19. Can people eat at meetings?
    a. Yes, we're hungry kids.
20. Smoke?
    a. No.
21. Dominating the discussion?
    a. Ask for the team member to let everyone get a chance to communicate their idea(s).
22. How can norms be changed if someone is not comfortable?
    a. Find a common ground.

23. As a team, select two cases out of the four mentioned in Handling Difficult Behavior.
    a. ARGUES: when argument occurs, figure out the problem and try to find common solution
    b. TOO QUIET: talk to the person, try to draw them into the group discussion
24. When making decisions, If the team is having trouble reaching consensus, what should you do?
    a. Find a common ground; don't try to linger around on one decision….
25. 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?
    a. Discuss regardless with group.
26. What happens if most people on the team want
    a. Let everyone do their best

Document Ended Here

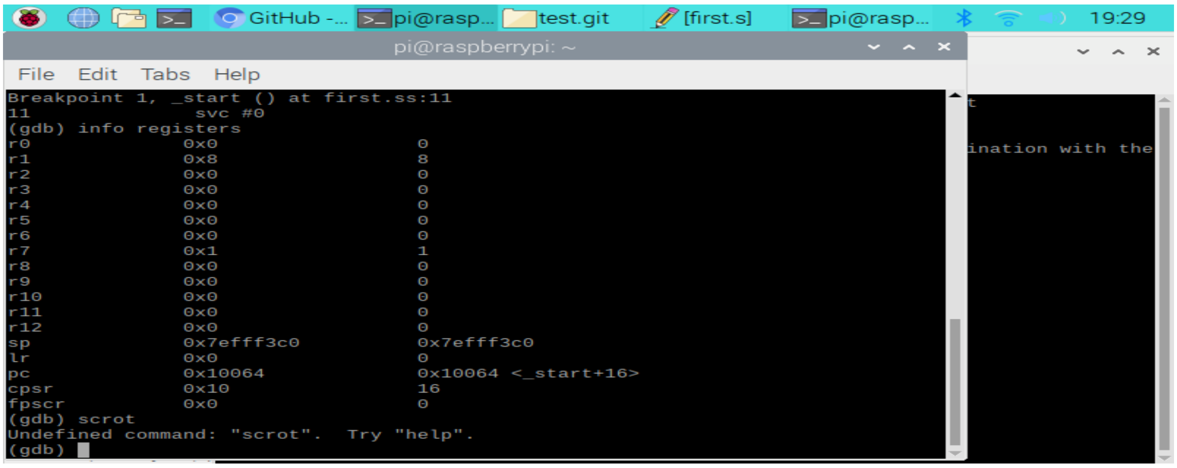
REPORT: ARM Assembly Programming.

Document Started Here

Bryanna Hardy

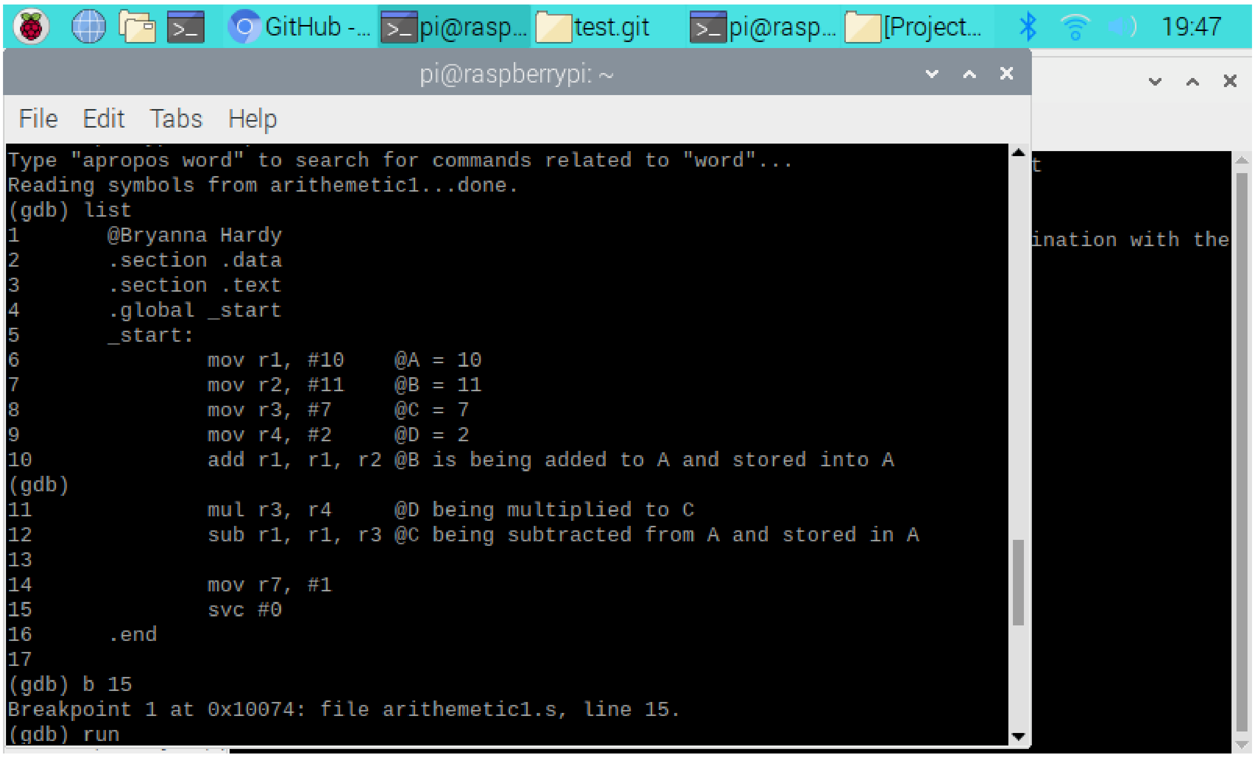
05 February 2020

ARM Programming Detail Report



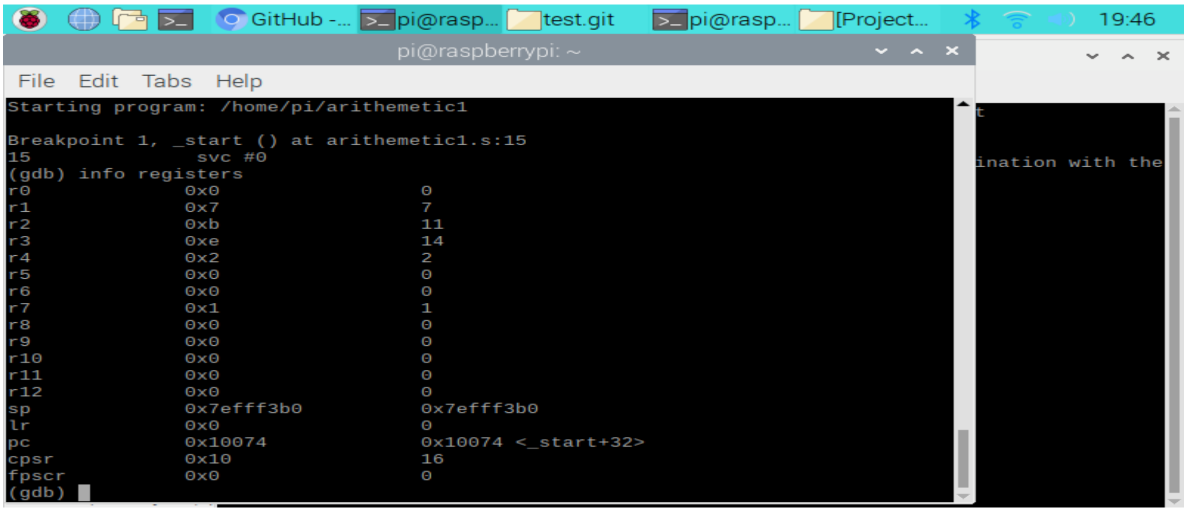
```
Breakpoint 1, _start () at first.ss:11
11              svc #0
(gdb) info registers
r0             0x0                 0
r1             0x8                 8
r2             0x0                 0
r3             0x0                 0
r4             0x0                 0
r5             0x0                 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) scrot
Undefined command: "scrot".  Try "help".
(gdb) 
```

This is the tutorial register information. Eight is stored into register 1.



```
Type "apropos word" to search for commands related to "word"...
Reading symbols from arithemetic1...done.
(gdb) list
1       @Bryanna Hardy
2       .section .data
3       .section .text
4       .global _start
5       _start:
6               mov r1, #10     @A = 10
7               mov r2, #11     @B = 11
8               mov r3, #7      @C = 7
9               mov r4, #2      @D = 2
10              add r1, r1, r2 @B is being added to A and stored into A
(gdb)
11              mul r3, r4      @D being multiplied to C
12              sub r1, r1, r3 @C being subtracted from A and stored in A
13
14              mov r7, #1
15              svc #0
16      .end
17
(gdb) b 15
Breakpoint 1 at 0x10074: file arithemetic1.s, line 15.
(gdb) run
```

This is the ARM Assembly program I had to create. It is basically stating that 10 is being stored in register 1. Eleven is being stored into register 2. Seven is being stored into register 3. Two is being stored into register 4. Now that I have stored my integers into my registers, I am adding register 2 to register 1 and letting that sum be stored into register 1. Next, I multiplied register 4 to register 3. Lastly, I am subtracting register 3 from register 1, and letting that difference be stored into register 1. Down below, there is a screenshot of the register information. It should show that register 1 (r1) stores 7, which should be our answer from the equation, (A+B) – (C*D).



```
Starting program: /home/pi/arithemetic1

Breakpoint 1, _start () at arithemetic1.s:15
15              svc #0
(gdb) info registers
r0             0x0                 0
r1             0x7                 7
r2             0xb                 11
r3             0xe                 14
r4             0x2                 2
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             0x7efff3b0          0x7efff3b0
lr             0x0                 0
pc             0x10074             0x10074 <_start+32>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

Document Ended Here

ANDY LEE INDIVIDUAL REPORT

THIS PAGE IS PURPOSELY LEVAVED AS BLANK

REPORT: TEAMWORK BASICS

DOCUMENT STARTED HERE
Individual Report: Team Basics by Andy

Q1.1 : **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)?**

A. The work will be disturbed by the team coordinator equally and fairly. However, we decided that the report will be written and submitted by Team Coordinator. The deadlines will also set by the team coordinator, and team members must complete their task before the due date. If the team member does not follow the commitment, then we warn them first, but if he/she still does not follow, we decided to report to TA or instructor.

Q1.2. **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**

A. The work will be reviewed during the weekly meeting.

If people have different opinion, we discuss each other, then choose the best one out of it. If there is disagreement, then we vote. Last but not least, we people have different work habits, we will try hard to adjust each other to go with same phase.

Q3. **Will you use a facilitator? How will the facilitator be chosen? Will you rotate the position? What are the responsibilities of the facilitator?**

A. Facilitator will be a team coordinator. The team coordinator will be rotated each project. The team coordinator will be assigned as a volunteer, but everyone has to be the team coordinator at least once.

Q4.1 **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?**

A. We, as a team, decided to have a meeting on Monday 6:30 to 7:00. The purpose of the meeting is the check each other’s work and record the process for presentation. The meeting time is set; thus no person will not be responsible for meeting time. The meeting time is known to be a Library, but may change on team’s discretion.

Q4.2 **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?**

A. If person/people late to the meeting, we would start the meeting regardless. Individuals are responsible to attend meeting every Monday. If he/she misses several meeting, then team coordinator will write it in at the “planning table.”

Q5**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?

A. People can eat, can’t smoke.

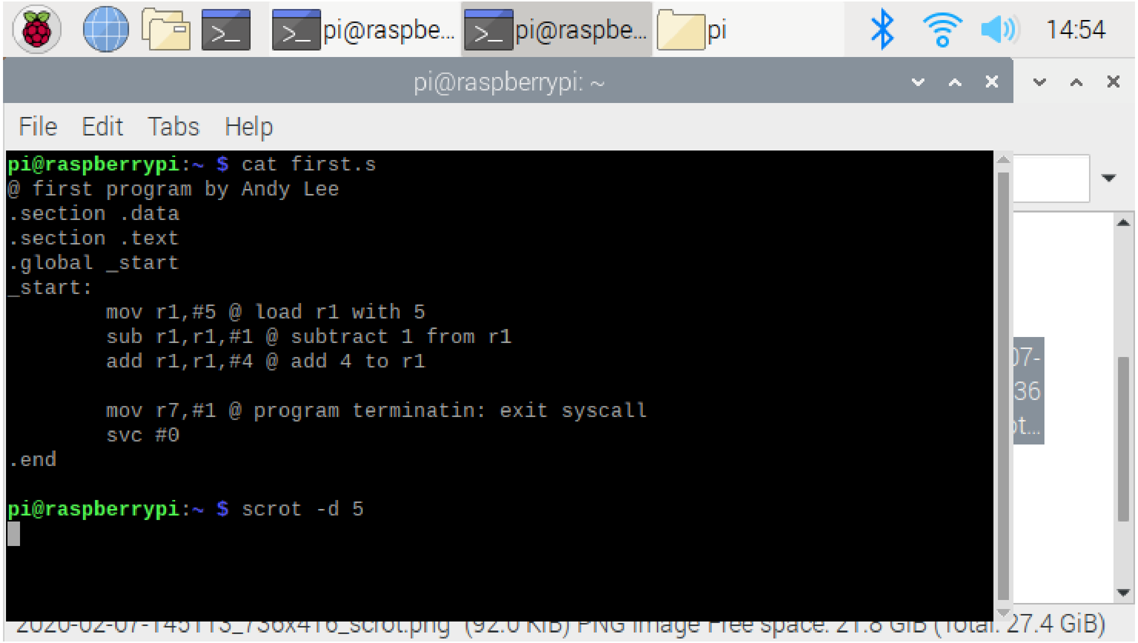
If one person dominating the discussion, team coordinator will ask to another people as well for their idea. If there is uncomfortable language, or behavior occurs during the meeting, we would write it in the note at “Planning table”.

DOCUMENT ENDED HERE

REPORT: ASSMEBLY PROGRAM

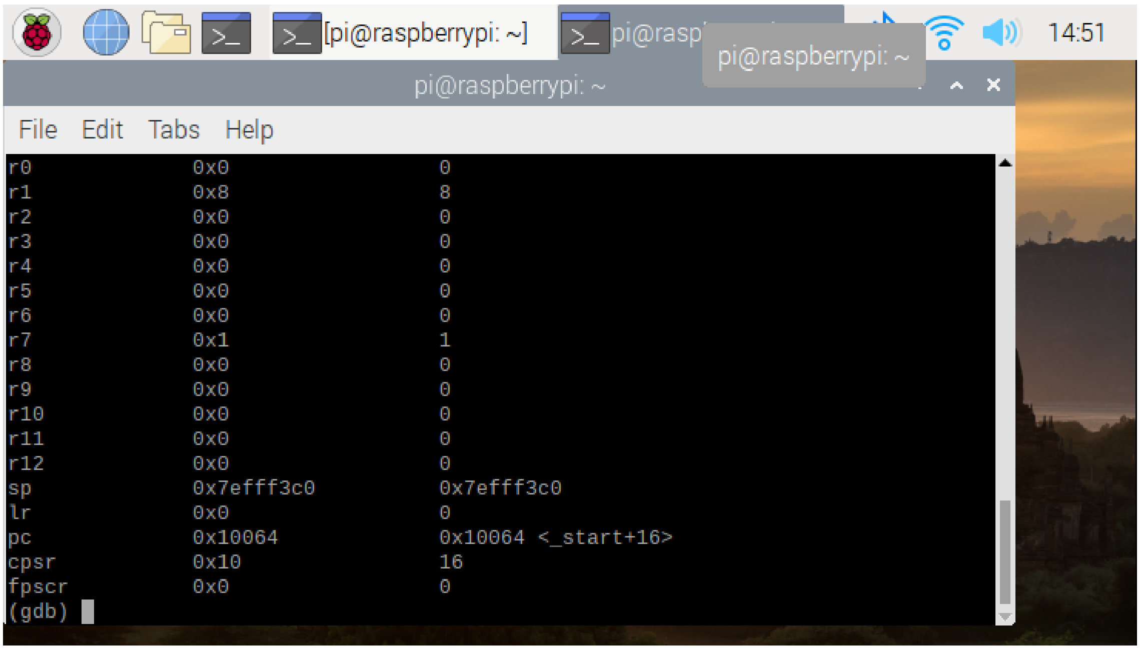
Document Start Here
Andy Individual Report

EX 1 : first.s


pi@raspbe...
pi@raspbe...
pi
14:54
pi@raspberrypi: ~
File Edit Tabs Help
2020-02-07-145113_736x416_scrot.png (92.0 KiB) PNG image Free space: 21.8 GiB (Total: 27.4 GiB)

```
pi@raspberrypi:~ $ cat first.s
@ first program by Andy Lee
.section .data
.section .text
.global _start
_start:
        mov r1,#5 @ load r1 with 5
        sub r1,r1,#1 @ subtract 1 from r1
        add r1,r1,#4 @ add 4 to r1

        mov r7,#1 @ program terminatin: exit syscall
        svc #0
.end

pi@raspberrypi:~ $ scrot -d 5
```

Result



```
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) 
```

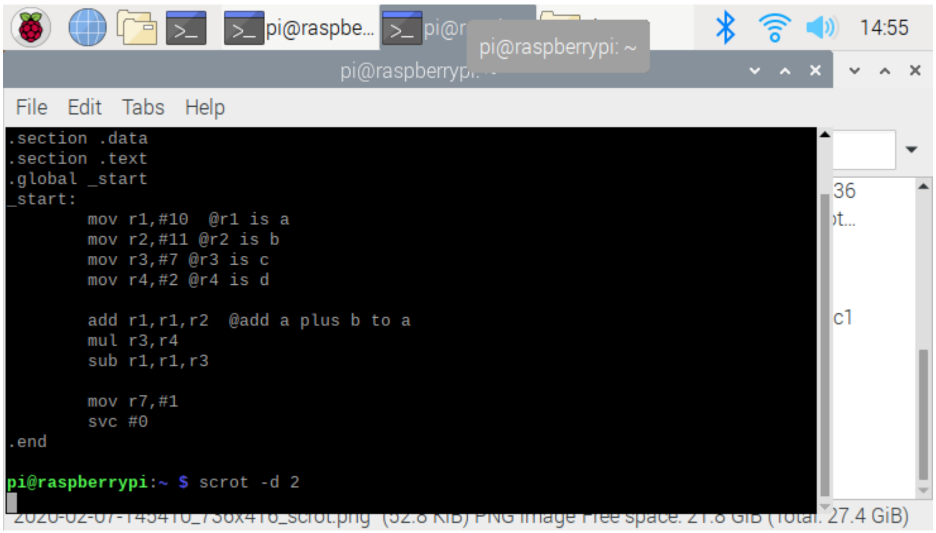
Summary

The assign task for first.s was to copy the code that instruction provide, and explain why result come out like the second photo.

To start with, we have loaded the register 1 and assigned the numerical number of 5. By loading the register, now computer have assigned the address of r1, which we could do a arithmetic operation using this path. On second line, as u could see from the top photo, I declared the subtraction function that subtract “1” from r1. Therefore, the value for r1 is now become 4. Follow by second, on third line, I have declared another functions “addition” with 4. This will add the numerical value of 4 in to r1. Since r1 is 4 before adding, the result of overall program will be 8.

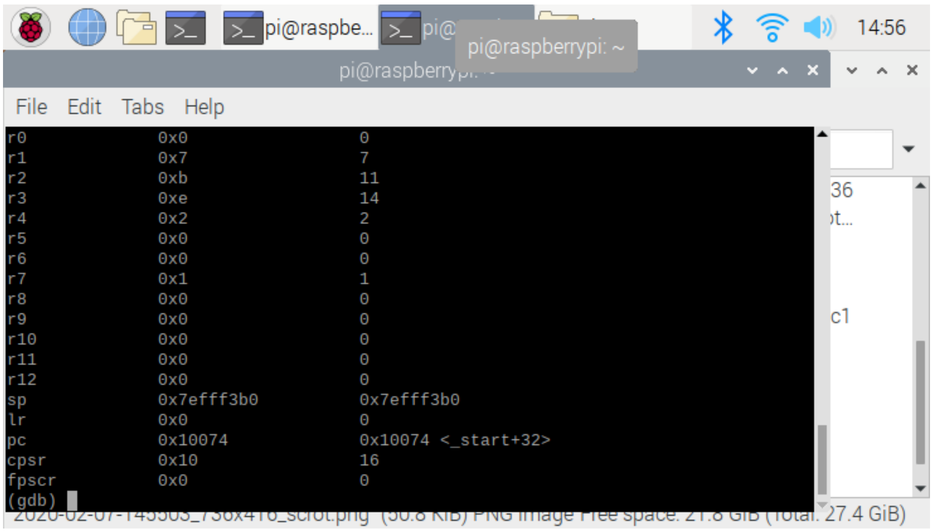
Now going to the debugger, we could see the r1 have decimal value of 8, thus the program work perfectly.

EX 2 arithmatrics1.s



```
.section .data
.section .text
.global _start
_start:
        mov r1,#10  @r1 is a
        mov r2,#11 @r2 is b
        mov r3,#7 @r3 is c
        mov r4,#2 @r4 is d

        add r1,r1,r2  @add a plus b to a
        mul r3,r4
        sub r1,r1,r3

        mov r7,#1
        svc #0
.end

pi@raspberrypi:~ $ scrot -d 2
```

Result



```
r0             0x0                 0
r1             0x7                 7
r2             0xb                 11
r3             0xe                 14
r4             0x2                 2
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             0x7efff3b0          0x7efff3b0
lr             0x0                 0
pc             0x10074             0x10074 <_start+32>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

Summary

The task for arithmatrics1.s is to find equation of “A = (A+B) – (C* D) using ARM assembly.

The first this I do was load 4 registers, since we need 4 variable. I have used the comment to indicates which register represents which variable. (i.e A represent r1). After the loading, I have started with function “add” that add r1 and r2. This addition will be stored in r1. In another word, the added value of two number will be store in to “A”. On second operation “mul”, I have multiplied r3 and r4, then store the operated value to the r3. In another word, the “C” will not have “14”.

So far the values for A is 21 , C is 14. Now we only have one other operation, which is subtraction. We simply said subtract A -C and store it to A. Overall the final answer of A is now 7

After coding, I have go to the debugger mode, and the answer we got is correct thus the program accomplished the achievement.

Document Ended Here.

RAHUL SUNKARA INDIVIDUAL REPORT

THIS PAGE IS PURPOSELY LEVAVED AS BLANK

REPORT : TEAM BASICS

DOCUMENT START HERE

Teamwork Basics

1. How will the work be distributed?
    a. The work will be divided evenly among us and while distributing the work we will take into account everyone’s strengths and experience.
2. Who will set deadlines?
    a. We each will have our own deadlines which will be before the final deadline set by our team coordinator.
3. What happens if someone does not follow through on their commitment?
    a. If anyone falls behind on their task we will communicate with them and help them get back on track.
4. How will the work be reviewed?
    a. We plan on reviewing our work during our weekly meetings.
5. What happens if people have different opinions about the quality of the work?
    a. We will discuss with everyone and come to a common ground that will benefit majority of the group
6. What happens if people have different work habits?
    a. If people have different work habits, we will use that to our advantage, which benefits the group as a whole
7. Will you use a facilitator?
    a. Yes, we do plan on using a facilitator, it will likely be our team leader/coordinator
8. How will the facilitator be chosen?
    a. We plan on asking for volunteer, if no one volunteers then we will pick one person
9. Will you rotate the position?
    a. Yes, we will rotate our roles
10. What are the responsibilities of the facilitator?
    a. As a team facilitator their tasks will be make sure everyone in the team stays on task and focused. Make sure that everyone is participating. Keep encouraging the team to get their tasks completed by the deadline. Help come up with an solution if the team is struck. Encourage the team to open up if they are having any problems. Lastly, to clearly explain what the final goal for the team is.
11. When should communication take place and through what medium?
    a. Communication should be taking place all the time, and we plan on using GroupMe and Slack to communicate
12. What is everyone’s schedule?
    a. On Monday’s we are mostly free around 6PM/7PM and Friday’s we are free around 11AM/1PM. Other days, we are all busy with classes and other commitment(s).
13. Should one person be responsible for coordinating meetings?
    a. Yes and No, no in a sense that we all need to communicate and coordinate the meeting, and yes that the team leader should be managing to make sure everyone is attending
14. Do people have a preference for when meetings are held?
    a. Yes, we plan on having them on Monday evenings
15. Where is a good place to hold meetings?
    a. The library is the best place to have our meetings
16. What happens if people are late to a meeting?
    a. If someone is late for a meeting, we will catch them on what they have missed either in person or through GroupMe
17. What happens if a group member misses a meeting?

   a. If someone misses a meeting, we will catch them on what they have missed either in person or through GroupMe

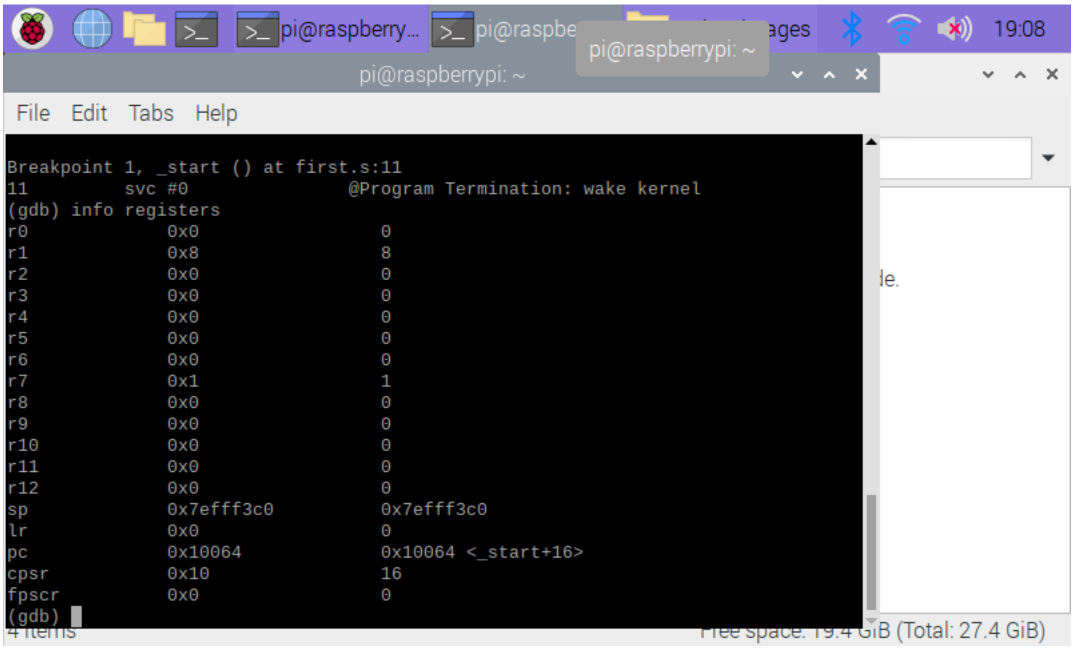
18. What if they miss several meetings?
    a. If one happens to miss several meetings then we talk to them and see what is causing them miss the meetings and either help them or find another time that will work for them and everyone
19. Can people eat at meetings?
    a. Yes
20. Smoke?
    a. No
21. What happens if someone is dominating the discussion?
    a. We will ask them to wait for other members opinions so that we can make one consensus decision as a whole
22. How can norms be changed if someone is not comfortable with what is going on in the team?
    a. Communicate and find a common ground

DOCUMENT ENDED HERE

REPORT: ASSMEBLY CODE

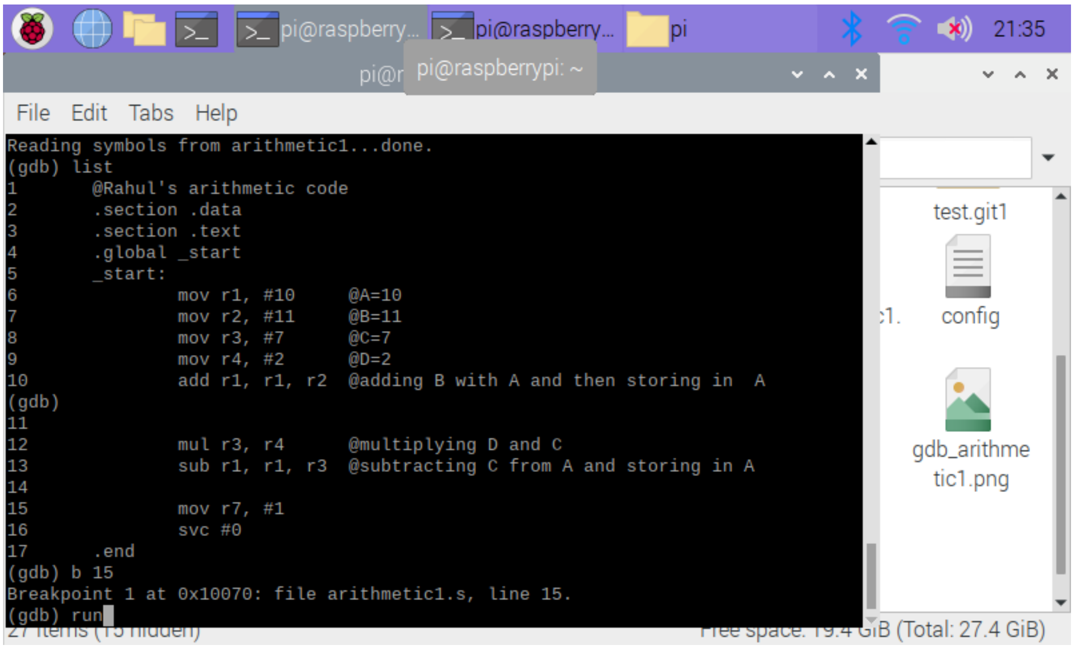
DOCUMENT START HERE

ARM Program Report



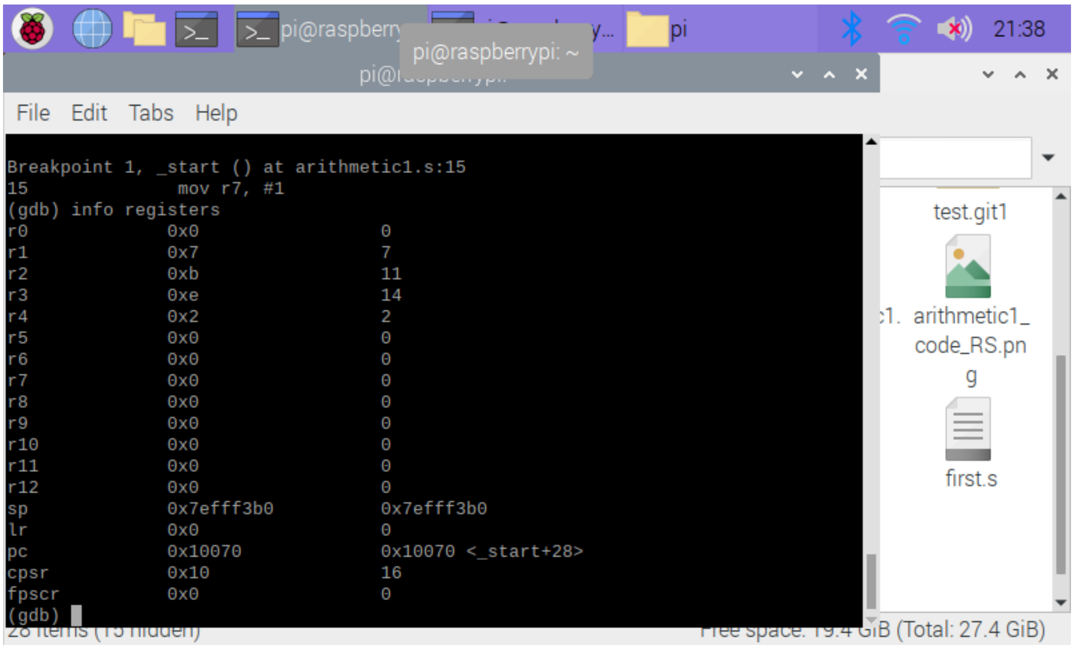
```
Breakpoint 1, _start () at first.s:11
11          svc #0              @Program Termination: wake kernel
(gdb) info registers
r0             0x0                 0
r1             0x8                 8
r2             0x0                 0
r3             0x0                 0
r4             0x0                 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) 
```

The tutorial register’s information, once the program completes it is storing 8 into register 1.



```
Reading symbols from arithmetic1...done.
(gdb) list
1       @Rahul's arithmetic code
2       .section .data
3       .section .text
4       .global _start
5       _start:
6               mov r1, #10     @A=10
7               mov r2, #11     @B=11
8               mov r3, #7      @C=7
9               mov r4, #2      @D=2
10              add r1, r1, r2  @adding B with A and then storing in  A
(gdb)
11
12              mul r3, r4      @multiplying D and C
13              sub r1, r1, r3  @subtracting C from A and storing in A
14
15              mov r7, #1
16              svc #0
17      .end
(gdb) b 15
Breakpoint 1 at 0x10070: file arithmetic1.s, line 15.
(gdb) run
```

The above image is the code that we had to compute for our first ARM Assembly program. This program wants us to compute (A + B) – (C * D) and store the final result into A.   I start off by storing the value 10 into register 1, then the value 11 into register 2, then value 7 into register 3 and finally value 2 into register 4.   To compute the equation, we first have to do the parenthesis, so we initially add register 2 to register 1 and store the result in register 1. Next, I multiplied register 3 by register 4. Finally, I subtracted register 3 from register 1 and storing the result into register 1. So, the final value stored in register 1 should be 7.



```
Breakpoint 1, _start () at arithmetic1.s:15
15              mov r7, #1
(gdb) info registers
r0             0x0                 0
r1             0x7                 7
r2             0xb                 11
r3             0xe                 14
r4             0x2                 2
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) 
```

The above picture shows the values stored in each of the register, and as stated above the result of the equation (A + B) – (C * D), which is 7 is stored into register 1, while the values in the other register are unchanged except register 3 which has the value of previous register 3 multiplied by register 4.

DOCUMENT END HERE

NATHAN HECKMAN INDIVIDUAL REPORT

THIS PAGE IS PURPOSELY LEVAVED AS BLANK

REPORT: TEAM BASICS

DOCUMENT START HERE

Nathan Heckman

February 6, 2020

Teamwork Basics

My team members and I recently discussed our strategy to work together on this project which are outlined below.

Work will be distributed as evenly as possible and everyone's skills and experience will be taken into account as well. We all set our own personal deadlines and the team coordinator helps manage them. If someone doesn't follow through on their end, we will talk to them and figure out what wrong and get them back on the right track. Work will be reviewed at our weekly group meetings. If people have different opinions of work quality, we will listen to all group members and come to a consensus as to what benefits the most group members. If people have different work habits, we will use them to our advantage and assign tasks accordingly.

We will use a facilitator (group coordinator). They will be chosen on a single assignment basis and the same person will not be allowed to be coordinator again until everyone else has gotten the chance to. The coordinator keeps everyone on track, creates the final project assignment report and submits it to iCollege.

For communication, we plan on utilizing GroupMe and I personally created a group email which we used to create our GitHub and YouTube channel. This can also be used for group communication if need be.

We are all responsible for meeting coordination. Meetings will be held in Library North on Mondays at 6pm. If people are late or miss the meeting, we will catch them up through communication on GroupMe. If they miss several meetings, we will sit down with them and ask how we can help them attend.

If someone is dominating the discussion, we will ask them to let others speak so we can agree on decisions as a group.

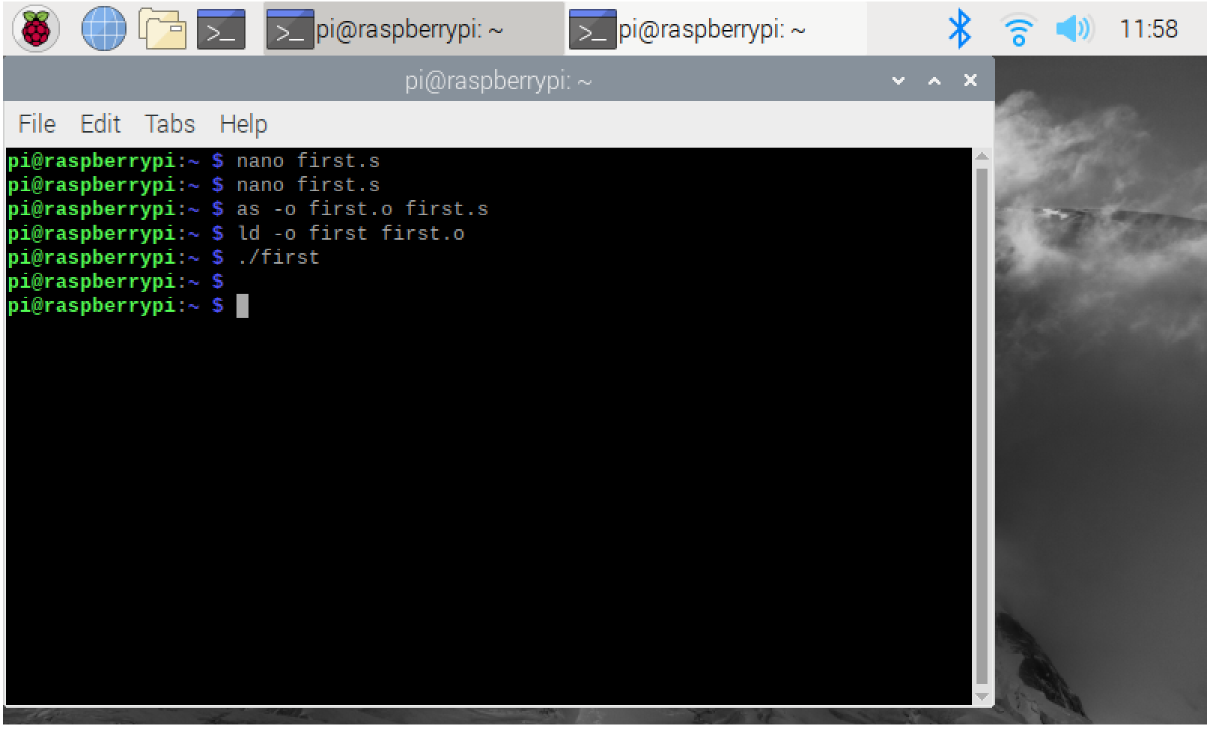
This is our basic teamwork strategy for the semester.

DOCUMENT END HERE

REPORT: ASSEMBLY CODE

Nathan Heckman

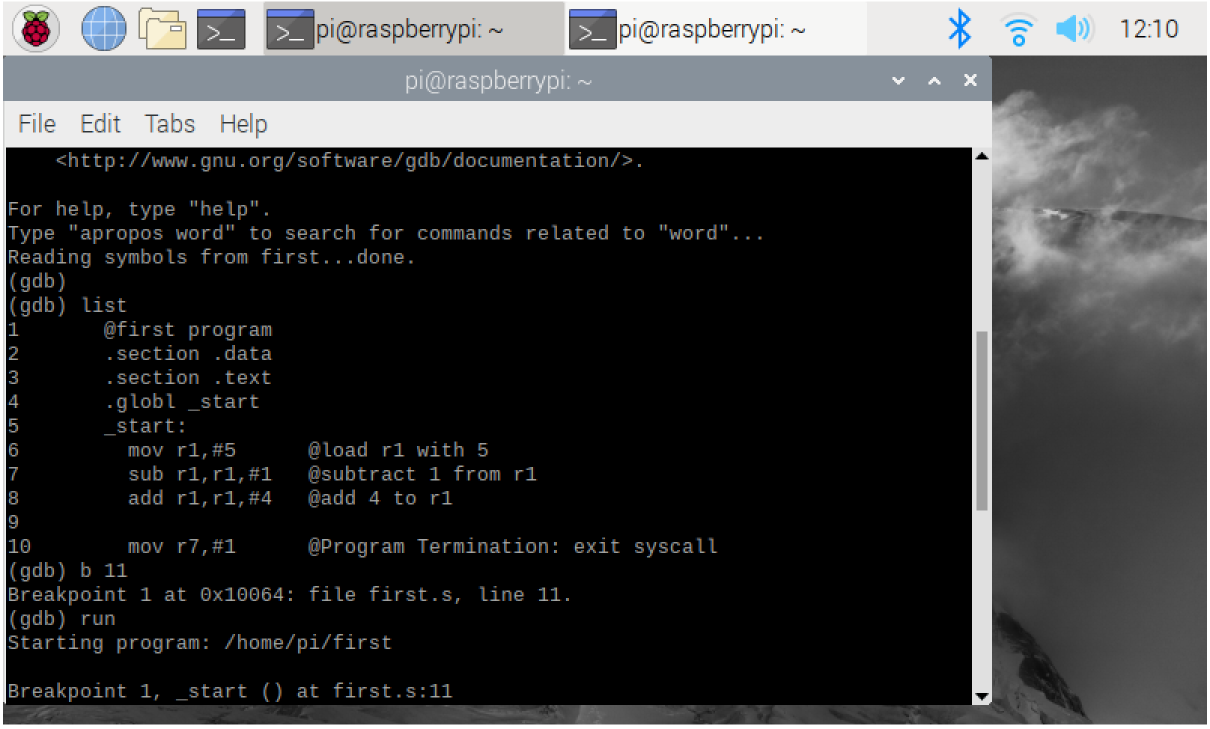
ARM Assembly Programming Question 6:



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

Here I attempt to run the first program using ./first and no output is given. This is because the program isn't in debug mode and no program stop point is set. The info registers command also hasn't been written and will therefore produce no output. I then went into debug mode and was able to run the program.

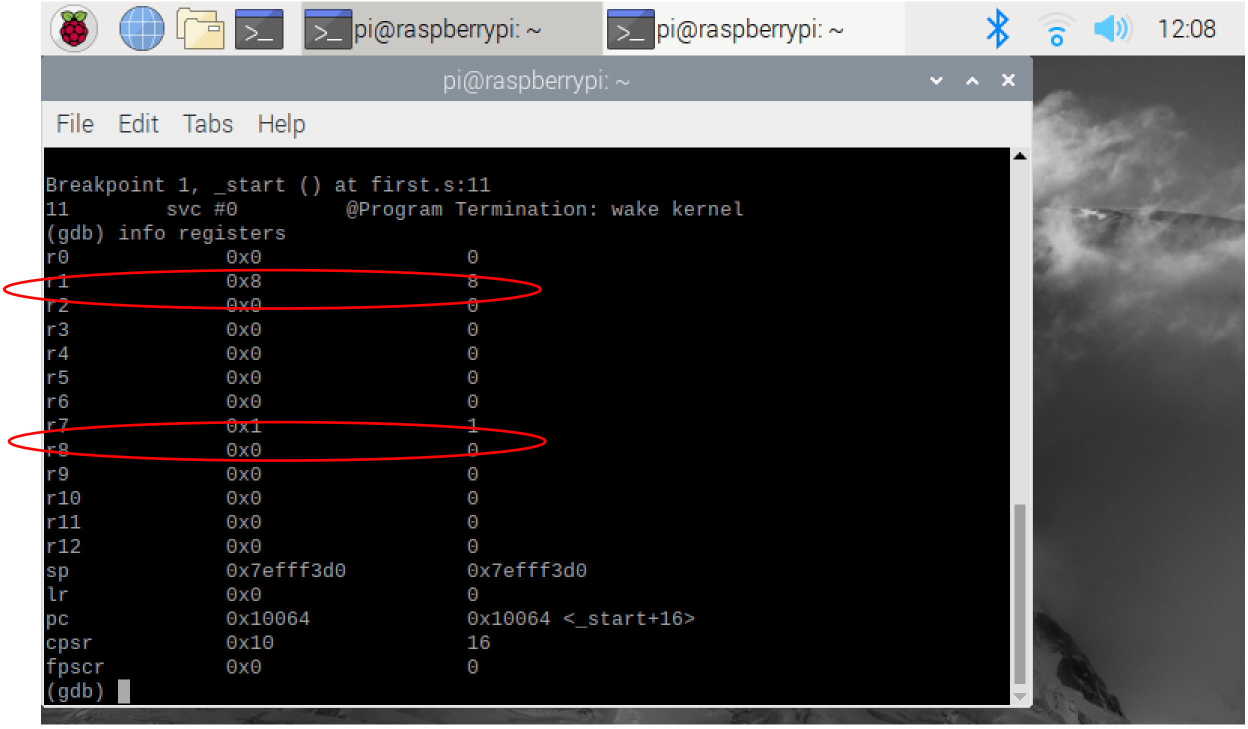
ARM Assembly Programming First Program:



```
    <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)
(gdb) list
1       @first program
2       .section .data
3       .section .text
4       .globl _start
5       _start:
6         mov r1,#5      @load r1 with 5
7         sub r1,r1,#1   @subtract 1 from r1
8         add r1,r1,#4   @add 4 to r1
9
10        mov r7,#1      @Program Termination: exit syscall
(gdb) b 11
Breakpoint 1 at 0x10064: file first.s, line 11.
(gdb) run
Starting program: /home/pi/first

Breakpoint 1, _start () at first.s:11
```

This is the code for the first program which was outlined in the assignment instructions. I used this template to build the arithmetic1.s program.

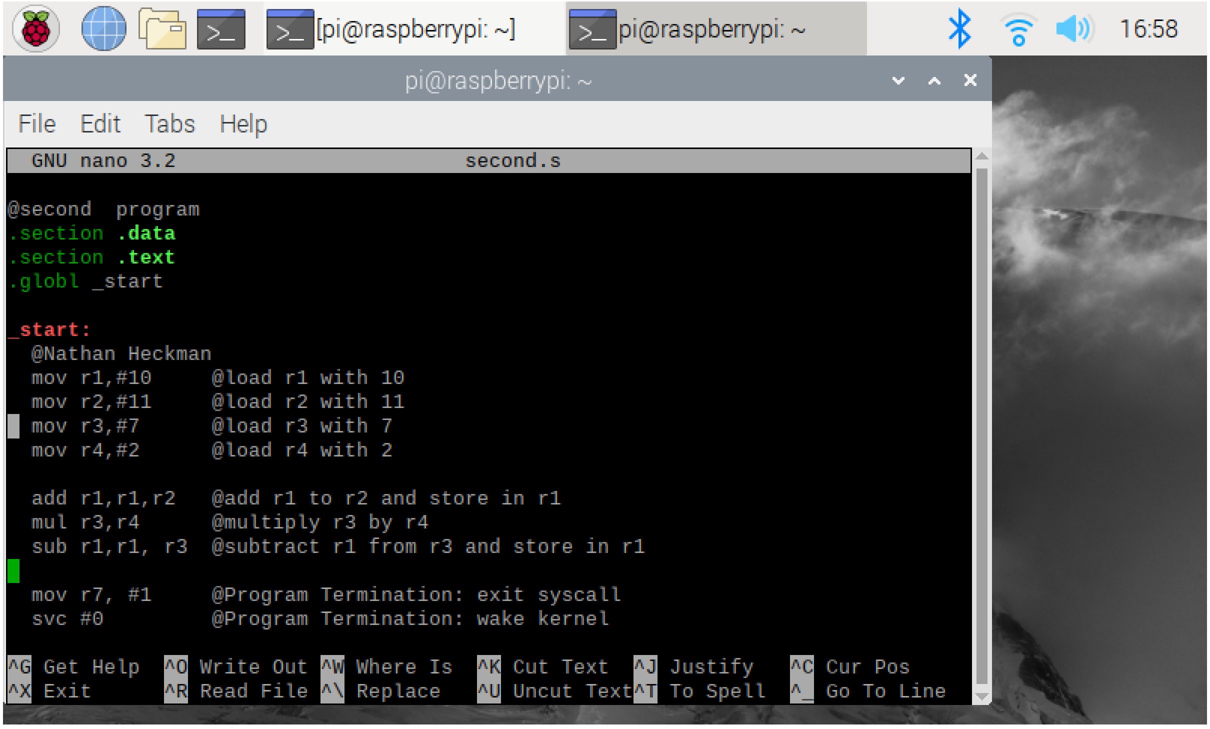
ARM Assembly Programming First Program Registers:



```
Breakpoint 1, _start () at first.s:11
11          svc #0          @Program Termination: wake kernel
(gdb) info registers
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             0x7efff3d0          0x7efff3d0
lr             0x0                 0
pc             0x10064             0x10064 <_start+16>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

These are the registers after the first program is executed, shown by using the info registers command inside the debugger. R1 holds a value of 8 since 5-1+4 = 8. R7 holds a value of 1 since we assigned it the value at the end of the program.

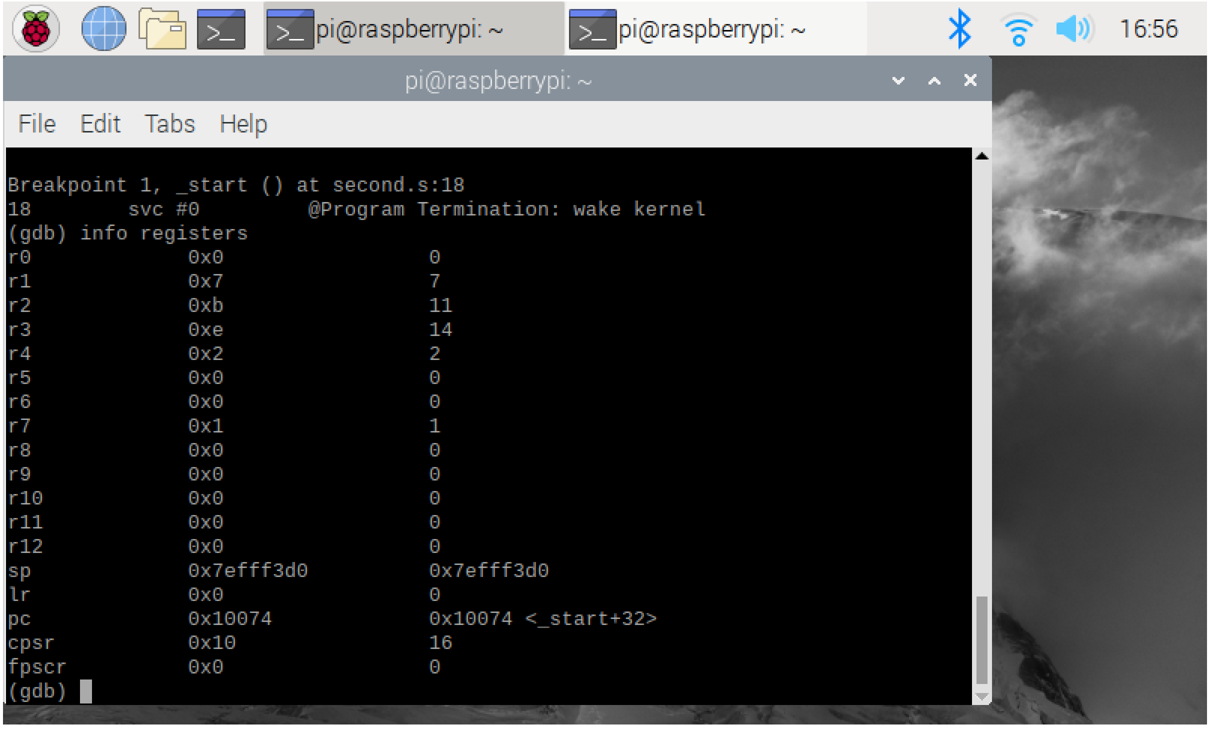
ARM Assembly Programming Arithmetic1 Code:



```
  GNU nano 3.2                               second.s

@second  program
.section .data
.section .text
.globl _start

_start:
  @Nathan Heckman
  mov r1,#10     @load r1 with 10
  mov r2,#11     @load r2 with 11
  mov r3,#7      @load r3 with 7
  mov r4,#2      @load r4 with 2

  add r1,r1,r2   @add r1 to r2 and store in r1
  mul r3,r4      @multiply r3 by r4
  sub r1,r1, r3  @subtract r1 from r3 and store in r1

  mov r7, #1     @Program Termination: exit syscall
  svc #0         @Program Termination: wake kernel

^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
```

This is the code for the arithmetic1.s program. I followed the same steps as the last program to assemble, link, run, and debug.

ARM Assembly Programming Arithmetic1 Info Registers:



```
Breakpoint 1, _start () at second.s:18
18          svc #0          @Program Termination: wake kernel
(gdb) info registers
r0             0x0                 0
r1             0x7                 7
r2             0xb                 11
r3             0xe                 14
r4             0x2                 2
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             0x7efff3d0          0x7efff3d0
lr             0x0                 0
pc             0x10074             0x10074 <_start+32>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

These are the registers after the second program is executed. R1 holds a value of 7 since (10+11) – (7 * 2) = 7. R2, R3, R4 hold their values because that is the value we assigned to them in order to eventually store the operation in R1. R7 holds a value of 1 since we assigned it the value at the end of the program.

DOCUMENT END HERE

MEZEMIR GEBRE INDIVIDUAL REPORT

THIS PAGE IS PURPOSELY LEVAVED AS BLANK

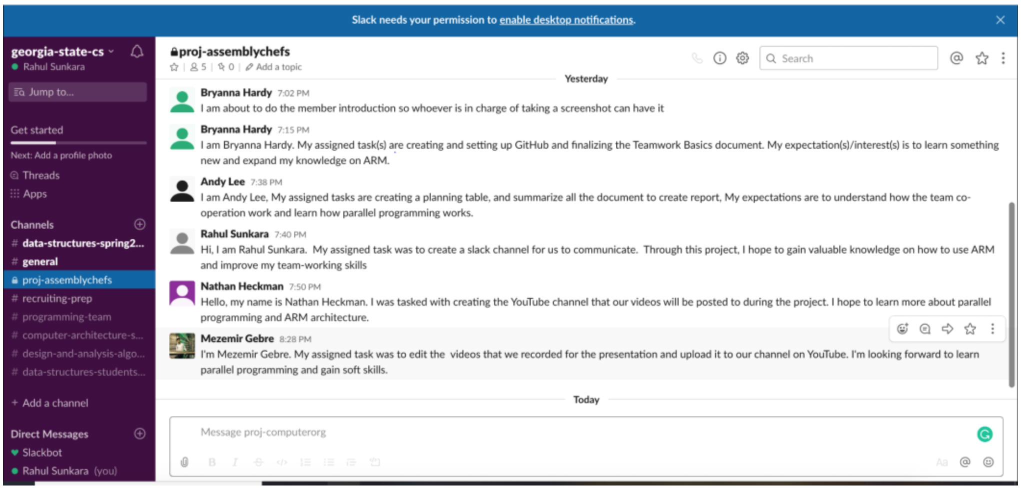
REPORT : APPENDIX

YouTube Video: Not Yet Uploaded

Slack : ?????

GitHub : https://github.com/AssemblyChefs/CSC3210-AssemblyChefs

Slack Photo



Yesterday

Bryanna Hardy 7:02 PM
I am about to do the member introduction so whoever is in charge of taking a screenshot can have it

Bryanna Hardy 7:15 PM
I am Bryanna Hardy. My assigned task(s) are creating and setting up GitHub and finalizing the Teamwork Basics document. My expectation(s)/interest(s) is to learn something new and expand my knowledge on ARM.

Andy Lee 7:38 PM
I am Andy Lee, My assigned tasks are creating a planning table, and summarize all the document to create report, My expectations are to understand how the team co-operation work and learn how parallel programming works.

Rahul Sunkara 7:40 PM
Hi, I am Rahul Sunkara. My assigned task was to create a slack channel for us to communicate. Through this project, I hope to gain valuable knowledge on how to use ARM and improve my team-working skills

Nathan Heckman 7:50 PM
Hello, my name is Nathan Heckman. I was tasked with creating the YouTube channel that our videos will be posted to during the project. I hope to learn more about parallel programming and ARM architecture.

Mezemir Gebre 8:28 PM
I'm Mezemir Gebre. My assigned task was to edit the videos that we recorded for the presentation and upload it to our channel on YouTube. I'm looking forward to learn parallel programming and gain soft skills.

Today