Developing Soft and Parallel Programming Skills Using Project-Based Learning

02.10.2020

How About Bits

Team: Ta Riq Singleton, Ilka Jean, Rayaan Chowdhury, Brian Williams, Mamadou Diallo

Planning and Scheduling

Introduction

Our team name is How About Bits, for the purpose of the first project we had six tasks to accomplish as a team. During our first meeting, we selected a team coordinator and discussed how task will be delegated, which source of communication will be used, and the details of planning and scheduling future meetings. Below is a breakdown of our work structure for the first project, with all details pertaining to each members' responsibilities.

Work Breakdown Structure

Assignee Name	Email	Task	Duration (hours)	Dependency	Due date	Notes
Mamadou Diallo	mdiallo20@student.gsu.edu	Editing and uploading the video on YouTube.	2 hours	YouTube account must be created first.	02/07/2020	Please send report section to coordinator to be added to report. 100%
Rayaan Chowdhury	rchowdhury5@student.gsu.edu	Creating and managing team GitHub account.(Task 2)	2 hours	none	02/02/2020	Please send everyone the link. All project docs must be uploaded. Take screenshot of project page. 100%
Ilka Jean (coordinator)	ijean1@student.gus.edu	Task 1 Technical writing(getting the report ready)	5 hours	none	01/31/2020 02/09/2020	Send report to teammates before typing final draft. 100%
Brian Williams	bwilliams198@student.gsu.edu	Task 3: Learning Teamwork Basics. (Task will be answered as a group, assignee will type and edit for readability)	1 hour	none	02/04/2020	Send to team members. 100%

Ta Riq Singleton-Peters	tpeters8@student.gsu.edu	Create YouTube and Slack channel as described in assignment.	1 hour	none	01/31/2020	Please send invitation and YouTube link to teammates. Take screenshot of main screen showing member’s messages. 100%
All team members	Must have Raspberry Pi running Complete ARM assembly Programming and send observation/screenshots to coordinator to be added to the report				02/02/2020 02/04/2020	

Teamwork Basics

Ta Riq Singleton

1. **What to do to get the task accomplished the team members' satisfaction high?**
   Get to know each other. Have rules. Someone to facilitate. Communicate. Create ways to avoid or solve the same problems.
2. **Answer all the questions in the Work Norms, Facilitator Norms, Communication Norms using your own words and your own context.**
   Work Norms: Work will be distributed based off of strengths. If the final deadline is missed, credit is lost. Otherwise warnings and checkups are issued. Work is reviewed as a group. If quality is an issue we will refer to the rubric and see if the standards are met. If not met work must be done over until it meets those standards. The people who like to do things first will have tasks that take more time. The procrastinators will have assignments that can be done in shorter periods of times.
   Facilitator Norms: Yes. We used a randomizer since each member will have the opportunity to be the facilitator. To make sure work is done. Group meetings are productive. Assignments are turned in. Communicate with the professor.
   Communication Norms: Slack and GroupMe. When people have questions when an assignment is complete.
   Meeting Norms: We will have schedules after a time is made that everyone can make. Meet after classes during the week. Weekends if necessary. Students will be updated if the meeting is missed. If this becomes a habit come together as a group and talk to that person and get to the root of the problem.
   Consideration Norms: Yes . No. Direct questions specific people. Have suggestions of improvement after each project.
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: Direct question their way. Get to know them so they feel comfortable opening up.
   Argues: Set up an objective debate where everyone can speak their mind and have a decision voted on at the end(By the group).
4. **When making decisions, If the team is having trouble reaching consensus, what should you do?(use your own words and your own context).**
   Slow it down and have them defend why it should be done looking at possibilities. Decide as a group an individual should not have control.
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 we are halted by making a decision list pros and cons of both and pick the one that hurts the grade the least and have a vote.
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?**
   Ask them why B is enough and then explain what an A means to the other members.

Rayaan Chowdhury

1. **What to do to get the task accomplished and the team members' satisfaction high?**
   Work together with clear communication for all tasks provided, and be willing to lend a hand to anyone that might be struggling.
2. **Answer all the questions in the Work Norms, Facilitator Norms, Communication Norms using your own words and your own context.**

Work Norms:

a) Equally so everyone gets a fair share.
b) Team leader.
c) Looked over before sending to the team leader, who also reviews it .
d) Try to reach a conclusion that best fits the group.
e) Try to work with them, as long as it gets done correctly and accordingly.

Facilitator Norms:

a) Yes(Team Coordinator).
b) Chosen different time for each assignment.
c) Yes.
d) To organize the group, have schedule set for everyone to get work done and to be the final say for any and all conflicts.
e) Keep team focused on the task at hand(project/in meetings).
f) Try to come up with compromises when

Communication Norms:

Whenever necessary, through a group texting app(GroupMe or Slack)

Meeting Norms:

a) Everyone's schedules should be known to the rest of the team so meetings can be planned
b) Library study rooms
c) Ideally one person will request a meeting if he/she feels the group needs to hold a meeting in order to further grasp a certain part of the project.
d) Ideally if someone is late/missing they can be called/facetimed in to still receive all the information the rest of us agree upon and participate in the meeting
e) Have a word with them and see what's going on, clearly something is going on behind the scenes that's either stopping them from attending or possible personal conflicts

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: I believe is something that can be handled quite easily depending on the rest of the group. If a member is too quiet and not vocal enough in meetings and over text/slack then the case most likely is that the person is not comfortable enough with the group to speak freely just yet. So the best thing to do in the scenario is to make him/her feel welcome and let them know it's a safe space to speak your mind.

   Argues: I believe is something slightly more difficult to deal with I would say. If someone is arguing often then most likely it might be that the person might not fully comprehend the situation and is assuming certain cases which are mistaken. So the best way to handle that would be to clear up any sort of misconceptions that person might have and then listen to what that person has to say because it might be useful information that the rest of the group might not have got.

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

   Have a facilitator that can choose a path to take or make a compromise between team members. And if that doesn't work then a simple vote of majority rules could be implemented to have a fair outcome.

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?**

   Check and make sure with the group to see if it's a decision that everyone can agree on, and then continue as the group comes to a consensus on how fast we should continue our work.

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?**
   In my opinion, the group's efforts should be matched by everyone in the group. So if one person wants a B while everyone else wants a B, then the group should speak to the person calmly and collectively and let them know that if everyone else wants an A then the group should be striving for that together. Ideally the person will understand and raise their own expectations. If that doesn't work then possibly let them know that these projects will be very useful going forward for job endeavors as the work we're doing here is very similar to industry. So if they can grasp that it will be beneficial for them as well to put in more effort than they might change their mind.

Ilka Jean

1. **What to do to get the task accomplished the team members' satisfaction high?**
   They are many imperative steps to take to ensure that all task are accomplished and that each members' satisfaction is high. A great first step is to get together as a group and get to know each other. Learning about each other's skills, strengths and weaknesses I believe will help ease the process of work distributions and task assignment. Setting up ground rules, will help each members know what to expect as team and will help us stay accountable. Another fundamental thing is to have a direct line of communication for the whole team, it is impossible to achieve our goals an solve problems that we may have as team without open communication
2. **Answer all the questions in the Work Norms, Facilitator Norms, Communication Norms using your own words and your own context.**
   Work Norms: Work will be distributed based on the initial meeting, where we discussed about each other skills, strengths and weaknesses and as well on a voluntary basis. The team coordinator will set deadlines in cooperation with rest of the team. If a team member miss a deadline, I would discuss it with them. To try to discover the reason and why and if we can do something as a team so that it doesn't happen again. I would also remind about the expectations we have as team and what we do affect each other. I would let them know that is important to communicate ahead of time if they realize that they might miss a deadline. Work will be reviewed by all team member before final report is submitted at least 24-48 hours before submission. If they are disagreement about the quality of work, the work will be reviewed and all suggestions will be taken under considerations based on most to least significant. Working with people with different work habits, I would say to break down the work load and create short deadlines instead of piling them on with distant deadlines.
   Facilitator Norms: For this project the facilitator was also the team coordinator. I would say that the role be rotated similar to the team coordinator.
   Communication Norms: Communication will be taken place through GroupMe and Slack. Communication should take place at least every two days to keep the team updated on task or more often if there is a certain challenge that we are facing.
   Meeting Norms: The team coordinator is responsible to schedule meeting with cooperation with the rest of the team. They are preferences for when meetings are held based on everyone's school schedule and other responsibilities. The best and most convenient place to have meetings is at the Georgia State Library. When a group member is late to a meeting or completely miss a meeting and did not discuss it ahead of time with the team, the team coordinator will speak with them and get to know the reason why they are late or did not show up. They will be reminded about the expectations for the group and how it affect the rest of the team. If a team member miss multiple meetings, and they were notified by the team coordinator or by the whole team about how their actions will affect the team, it will be reported on the report to the professor.

Consideration Norms: Team members are allowed to eat during meetings but they are not allowed to do anything that would be distracting or affect a team member in any negative way, like smoking for example. If someone is dominating the conversation, I would let them finish with what they are saying and I will jump in and ask if the team or a specific person on their thoughts about the matter. Or I would jump in say, although I appreciate his/her ideas that I would like to hear everyone else ideas as well. To change norms, it is best first to discuss with the person on why they are not comfortable with what is going on to see if It can be resolve at that moment, else discuss it with the group how what we can do as team to improve norms if they are reasonable complaints.

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: If someone is too quiet, it might just be that they are shy not that they are not hard working. Sometimes people like that take some time to open up but are great at following directions and getting their task done. The best way to handle someone like that is to be kind and assure them that they opinions matter and that they are important to the team. It might sometime take some time outside of team work to get to know them.
   Complaints: I would listen to the person and see whether their complaints are legitimate. Some people complain because they do not want to get work done. However, if their complaints are legitimate, it is important to discuss it as a group and get their thoughts on it. Maybe the person is seeing a mistake in ways we are not seeing as whole, it might an opportunity to better improve our work.
4. **When making decisions, If the team is having trouble reaching consensus, what should you do?(use your own words and your own context).**
   When reaching decisions if there is difficulty reaching a consensus, I would suggest as team for us to take a vote. Based on what the majority voted we will make our decision. If there is a tie and there seem to be no chance of compromise, and we cannot reach a conclusion at all and it is causing issues within the group, maybe then we will bring in an outside party, perhaps a professor or TA.
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?**
   We will talk with the person, and let them know although we appreciate their decision. We would like to wait first on everyone else's decision before we make a final decision. I would emphasize about how important it is for us to work as a team so that we can get things done fairly and efficiently.
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?**
   I would talk to them and maybe try to understand why it is okay with them to get a B instead of an A. Perhaps they might have other priorities, so I will to talk to them about maybe what we can do as team for everyone to achieve the maximum result. If it is something that the team can't help with or motivate to do. I would just let them know that they need to make sure to work diligently on the task that will affect the group as team. Letting them know how important it is for him to work us, because it does not only affect him but what he does affects the whole group.

Brian Williams

1. **What to do to get the task accomplished the team members' satisfaction high?**
   For full completion of a task and high satisfaction among all group members, I think it is important to find a common, concrete goal and work toward that goal diligently. Going the

extra mile, albeit not always possible, almost guarantees high satisfaction. Being able to compromise effectively is an important skill as well.

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

Work norms: (1) Distribution of work depends on the type of work and the type of group, but playing to teammates' strengths is a good place to start. (2) If the group has a scheduler or someone who is capable of managing deadlines, I think that it is fair to allow that person to control the work flow from a group perspective. (3) Dealing with unfulfilled promises is challenging, but sometimes you have to pick up from where someone else left off, and other times you can talk things out and help that person to get back on track. (4) I think it is important for at least the majority of the group to revise and critique the group's output. (5) Disputes about quality of work cannot always be settled, but seeking to improve the quality further is a decent solution; taking a group survey might also be tactical. (6) Allowing group members to work in their comfort zones is acceptable within reason, in my opinion. For very important tasks, for example, extreme procrastination is not acceptable because of the possibility of unexpected outcomes.

Facilitator norms: (1) Although I am not a fan of the idea of a facilitator, let alone being a facilitator myself, I think that someone who can be trusted when the group becomes stagnant is a valuable asset. (2) For a group with no chemistry, I would consider picking someone who has administrative gifts or desires to be in charge. I do not think that selecting someone who does not want administrative power is a good idea. (3) I recommend rotating positions to circumvent tiring, exhaustion, or boredom. (4) I would say that the facilitator is most needed to make sure that the group adheres to the end goals that it committed to.

Communication norms: (1) I think that face-to-face communication is essential for any serious concern. It is far too easy to misinterpret the meaning of text without the non-verbal elements of communication, including facial expressions and gestures. My personal favorite electronic medium is Discord, as it allows for convenient group discussion and its interface is very code-friendly.

Meeting norms: (1) Finding compatible time slots is very difficult for students with variable schedules, but I would imagine that congregating at the corporate level would be much easier. (2) In groups of ten or fewer, I believe that having one person to coordinate meetings may work, but any more than that calls for multiple meetings that categorize themselves by the subjects at hand. (3) I think it is likely that almost all people have meeting time preferences. For example, I am usually busy on Monday and Wednesday after I come home from school because I serve at a church, and I would prefer not to meet at night because I function better during the morning. (4) Places that are quiet and inexpensive are highest priority for me. (5) Being late is usually fine, in my opinion, as long as it is not habitual. (6) If someone chooses not to show up for meetings, I would closely monitor their work to verify the quality and lack of face-to-face time.

Consideration norms: (1) I am fine with eating in most cases. (2) I find smoking to be far more disruptive and inappropriate than eating during a meeting. (3) I do not think it is difficult to balance the discussion among group members as long as the group is larger than three people. The group can politely address the issue together if private discussion fails to make change. (4) I would say to be "above reproach" in group settings in the first place, but I understand that that phrase means different things to different people. I suggest talking out discomfort in private.

3. **As a team, select two cases out of the four mentioned in Handling Difficult Behavior. (use your own words and your own context).**
   Arguing: I condone and even support arguing as long as a literal argument is presented and the parties involved are not fighting or disrespecting one another. I think that all group members should be involved, given that the argument is civil, because all members can learn from the positions that each member takes. Yelling at others or personal insults cannot be permitted, in my opinion.
   Too quiet: As a natural introvert, I completely understand this position and respect it. First, I would attempt to find out why someone is being quiet—uncomfortable, disinterested, stressed, naturally quiet, etc. If the person is uncomfortable or stressed, reaching out privately and encouraging is helpful. In the case of other causes, I would consider monitoring the group member and making sure he or she understands the assignment and the group's goals.

4. **When making decisions, If the team is having trouble reaching consensus, what should you do?(use your own words and your own context).**
   I think that rushing consensus on important decisions is almost never good, but if a decision is urgent and progress is stagnated, simply trying some of the ideas presented may make the decision easier. Great discoveries arise from trying lofty ideas, whether the outcome was intended or not.

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?**
   I would tell such a person to remember that an idea is not a decision, and that people need time to process. I would consider going through pros and cons with that person and the group to help reach an agreement on the potential benefit or risk.

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?**
   I would do my best to convince the person that working for the A is worth it, but even if that person is not convinced, a full effort is mandatory and any less cannot be tolerated. On any aspect of the project where the entire group's grade or report is affected, I would assert that the greatest effort is required, whereas any individual portions of the project can be of whatever quality the person deems appropriate.

Mamadou Diallo

1. **What to do to get the task accomplished the team members' satisfaction high?**
   I believe a way that we can get the task accomplished, while having the approval of group members is by getting to know one another and understanding the strengths of our teammates before delegating tasks. We should be aware of what responsibilities our team members feel comfortable with, and if they're not comfortable with certain tasks, we should figure out ways to aid them in order for them to feel confident in contributing to the end product.
2. **Answer all the questions in the Work Norms, Facilitator Norms, Communication Norms using your own words and your own context.**
   Work Norms: Work will be distributed by the team having a group discussion and choosing assignments that fit our respective strengths. If someone has already completed a certain task, the team coordinator should be trusted to assign that person another task. The team coordinator should be tasked with setting deadlines while taking into account the approval of the group members. If someone doesn't follow through on his/her commitment, as a group, we would

have to communicate with that team member, and let the person know our feelings about the situation, and how it makes us feel. I believe work should be reviewed as a group through the group-me that we have, and group members should be able to give their take on all the tasks before the final submission. If people have different opinions on the quality of the work, I think we should seek counsel from the professor on what he's expecting, and then as a group discuss if we've reached the mark or missed it. If people have different work habits, I believe that it would be best if we discussed the pros and cons of certain habits, and also trying different practices as a group would be helpful to make sure everyone is heard.

Facilitator Norms: I think it would be essential to use a facilitator to make sure that we are having productive meetings and hitting all of our marks. The facilitator can be chosen by volunteering but it's important to make sure that everyone gets a chance to facilitate during the semester. The responsibilities of the facilitator should be engaging all the team members and leading an advantageous discussion that makes sure we're staying on tasks and taking advantage of our time.

Communication Norms: I believe daily communication is necessary to ensure a great end product. It also can help in forming a great relationship with your fellow team members. A group chat, such as "group-me" is a great medium to communicate with your teammates, since it works on any phone the same and there isn't a disconnect. Group members should also provide their phone numbers, since having a conversation on the phone sometimes can accomplish a lot more than texts.

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

   Overly Talkative: An overly talkative person isn't always bad, since they may be bringing important information to the table, but it should definitely be discussed if it begins to take away from the participation of other group members. The group facilitator can be very helpful in this situation since they can point the conversation to other members, and ensure everyone is getting a chance to speak their mind.

   Too quiet: If a person in a group is too quiet, group members should sympathize with this person, but also take measures to receive some feedback and responses from that person. Consistently asking this person their opinions and takes on certain things can go a long way toward improving this person's interaction with the group. An example could be asking the person how they feel about every decision before it's made.

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

   If the team is having trouble reaching consensus, I believe a round table that allows every member to plead their case and their reasoning for backing a certain opinion would allow for the group to understand each other more and gauge which decision would benefit the group the most. If there still isn't an agreement on what direction the team should take, the selected team coordinator or facilitator should lead a voting session that finds a decision that the majority agrees with.

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 reaching decisions quickly, and pressuring others to do the same, the facilitators or the team coordinator should ensure that everyone has had enough time to decide by asking them questions. An example would be, "Does anyone need more time before deciding?" Questions like these ensure that each group member is allowed enough time to gauge their decision.

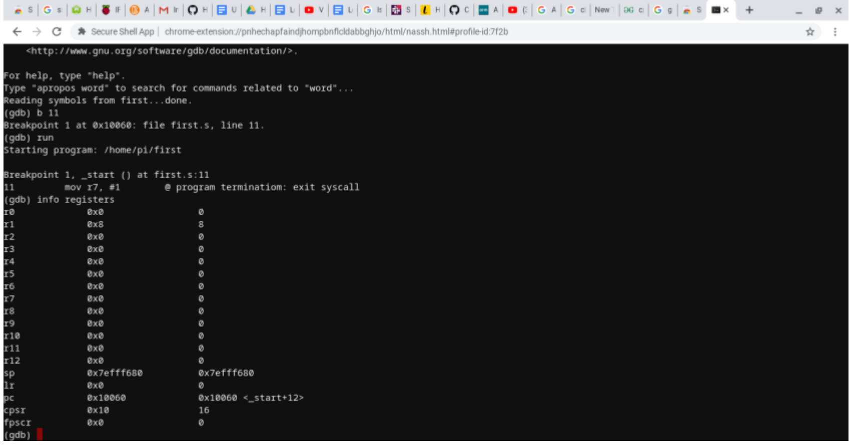
6. **What happens if most people on the team want to get an "A" on the assignment, but another person decides that a "B" will be acceptable?**

If most people on the team want an A, and another member decides a B is acceptable, then the group should talk to this person, since the majority has decided that they want to reach toward an A. The group communicating with this person and letting him/her know how you feel and your motive behind getting an A should do the trick for the most part. If it's made clear that this person's behavior has an impact on every other member, then the situation could be resolved.

Raspberry Pi Installation and ARM Assembly Programming

Ta Riq Singleton

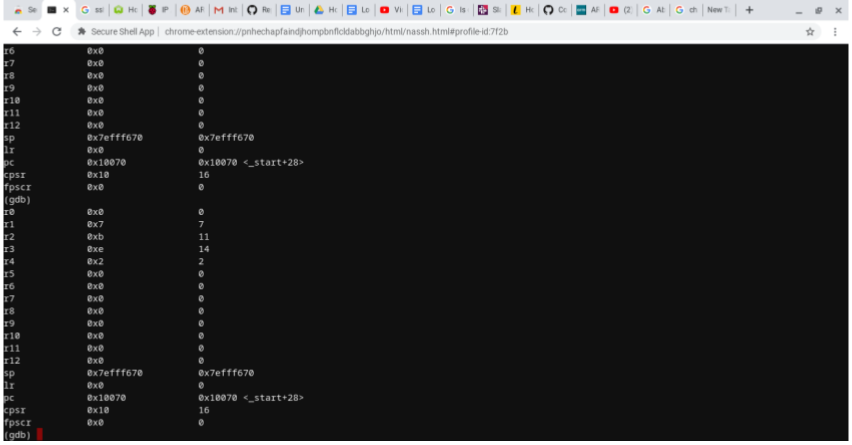
Part 1


Secure Shell App | chrome-extension://pnhechapfaindjhompbnflcldabbghjo/html/nassh.html#profile-id:7f2b

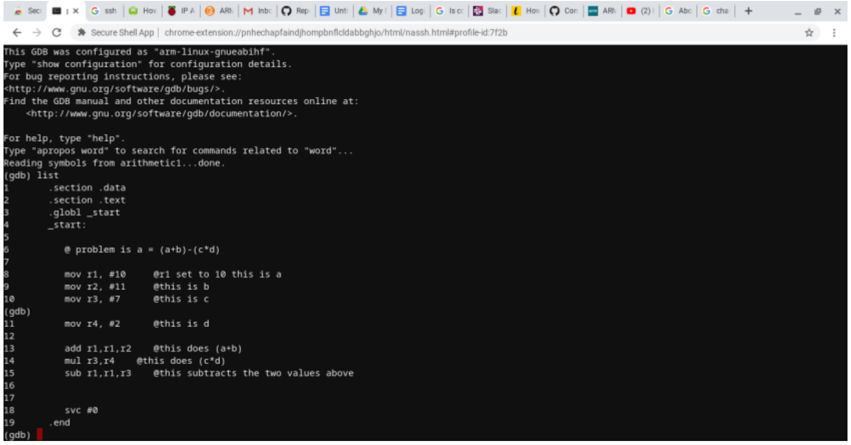
```
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from first...done.
(gdb) b 11
Breakpoint 1 at 0x10060: file first.s, line 11.
(gdb) run
Starting program: /home/pi/first

Breakpoint 1, _start () at first.s:11
11          mov r7, #1          @ program terminatiom: exit syscall
(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             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff680          0x7efff680
lr             0x0                 0
pc             0x10060             0x10060 <_start+12>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

Similar to x86 destination comes first and then the source for mov/add/sub/mul methods. The only difference was that add/sub took three operands instead of two. First thing I noticed is that the register names are different while I was in the gdb debugging. What I didn't see is the register that held the ALU status flag values.

Part 2


Secure Shell App | chrome-extension://pnhechapfaindjhompbnflcldabbghjo/html/nassh.html#profile-id:7f2b

```
r6             0x0                 0
r7             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff670          0x7efff670
lr             0x0                 0
pc             0x10070             0x10070 <_start+28>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
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             0x7efff670          0x7efff670
lr             0x0                 0
pc             0x10070             0x10070 <_start+28>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

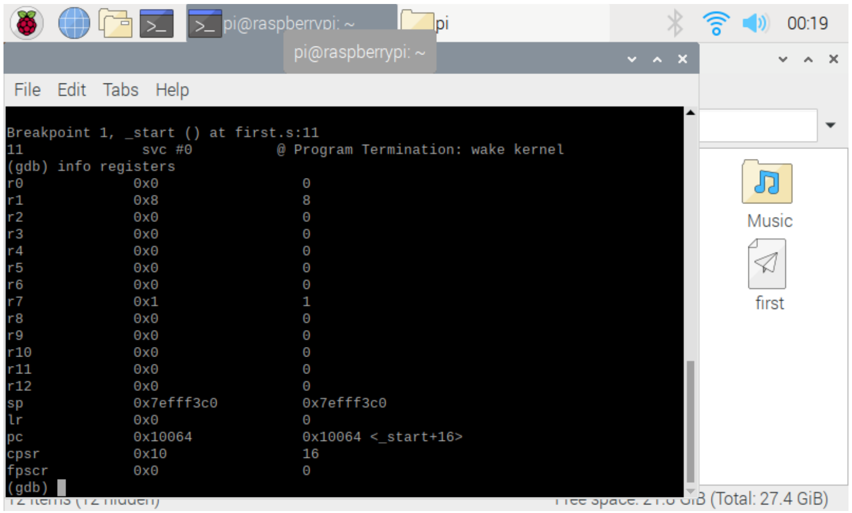


```
This GDB was configured as "arm-linux-gnueabihf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from arithmetic1...done.
(gdb) list
1       .section .data
2       .section .text
3       .globl _start
4       _start:
5
6           @ problem is a = (a+b)-(c*d)
7
8           mov r1, #10     @r1 set to 10 this is a
9           mov r2, #11     @this is b
10          mov r3, #7      @this is c
(gdb)
11          mov r4, #2      @this is d
12
13          add r1,r1,r2    @this does (a+b)
14          mul r3,r4    @this does (c*d)
15          sub r1,r1,r3    @this subtracts the two values above
16
17
18          svc #0
19      .end
(gdb)
```

I learned that I could use two operands to multiply numbers together instead of three. Immediate (integers) have a major difference since I had to put the '#' symbol in front of the number. The continue command allows you to proceed after setting multiple breakpoints (Looked up). Learned that cspr is the equivalent to eflags. Also the results in the register are in decimal instead of hex.

Rayaan Chowdhury

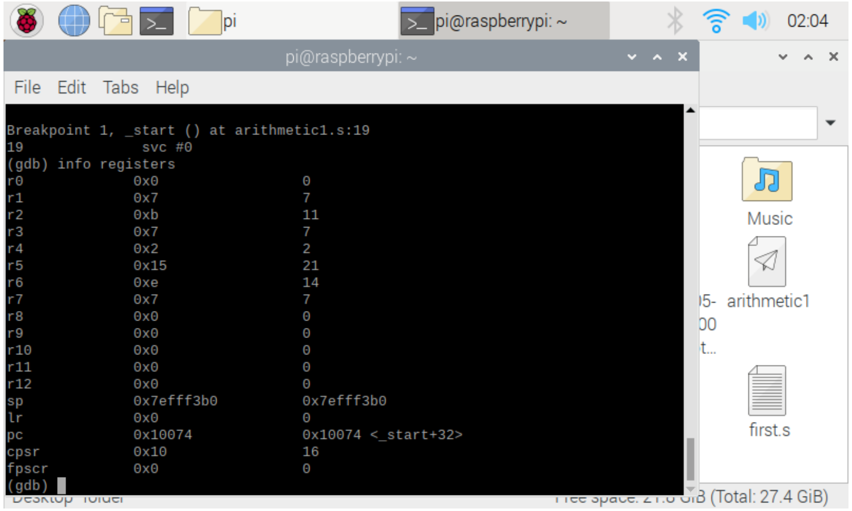
Part 1



```
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)
```

Registers show that the original source code had r1 as 8 because 5 was loaded onto r1 and then the instructions sub r1, r1, #1 and add r1, r1, #4 subtracted 1 from r1 and then added 4 to r1. Equaling 8 which still shows at r1.

Part 2

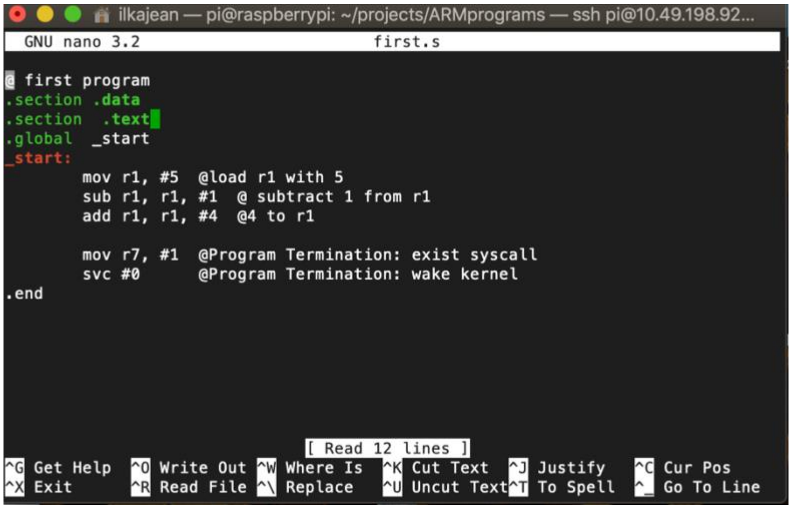


```
Breakpoint 1, _start () at arithmetic1.s:19
19                svc #0
(gdb) info registers
r0             0x0                 0
r1             0x7                 7
r2             0xb                 11
r3             0x7                 7
r4             0x2                 2
r5             0x15                21
r6             0xe                 14
r7             0x7                 7
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)
```

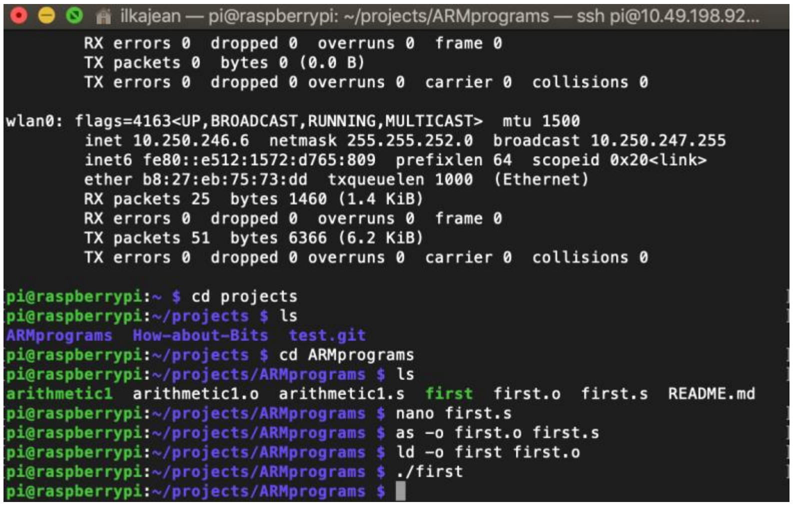
The registers this time around have different values because of the given equation that had to be replicated. So now we are using more than one register. R1 through r7 are now given values. The important values being r1 = 10 , r2 = 11, r3 = 7 , r4 = 2. The registers r5 and r6(and r1) were modified to add and multiply(and subtract) values according to the formula. And r1 was changed from the original value to equal the final answer after all the given arithmetic

Ilka Jean

Part 1

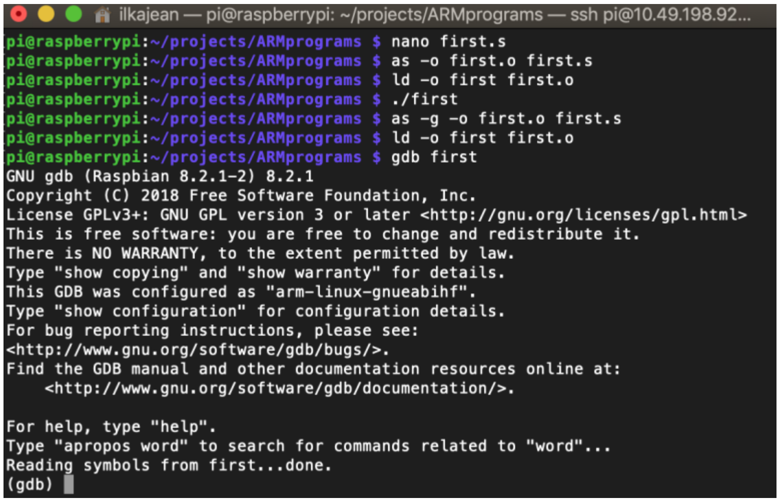

ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...

```
  GNU nano 3.2                           first.s

@ first program
.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  @4 to r1

        mov r7, #1  @Program Termination: exist syscall
        svc #0      @Program Termination: wake kernel
.end

                               [ Read 12 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
```

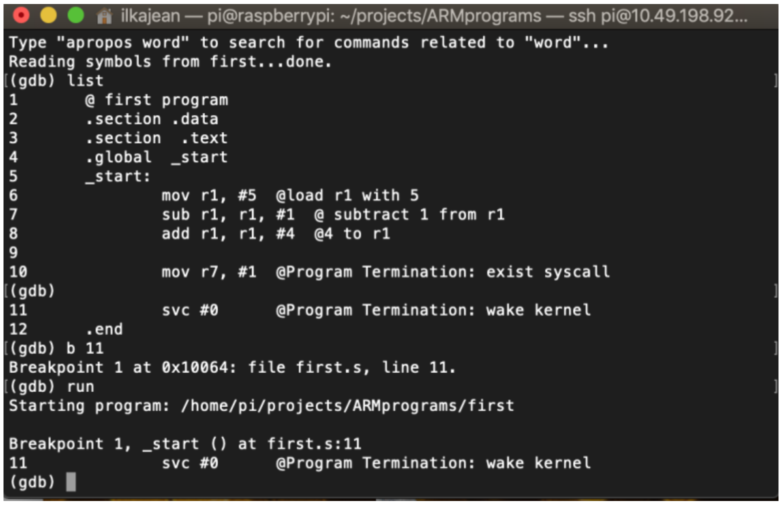

ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...

```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.250.246.6  netmask 255.255.252.0  broadcast 10.250.247.255
        inet6 fe80::e512:1572:d765:809  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:75:73:dd  txqueuelen 1000  (Ethernet)
        RX packets 25  bytes 1460 (1.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 51  bytes 6366 (6.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

pi@raspberrypi:~ $ cd projects
pi@raspberrypi:~/projects $ ls
ARMprograms  How-about-Bits  test.git
pi@raspberrypi:~/projects $ cd ARMprograms
pi@raspberrypi:~/projects/ARMprograms $ ls
arithmetic1  arithmetic1.o  arithmetic1.s  first  first.o  first.s  README.md
pi@raspberrypi:~/projects/ARMprograms $ nano first.s
pi@raspberrypi:~/projects/ARMprograms $ as -o first.o first.s
pi@raspberrypi:~/projects/ARMprograms $ ld -o first first.o
pi@raspberrypi:~/projects/ARMprograms $ ./first
pi@raspberrypi:~/projects/ARMprograms $ 
```

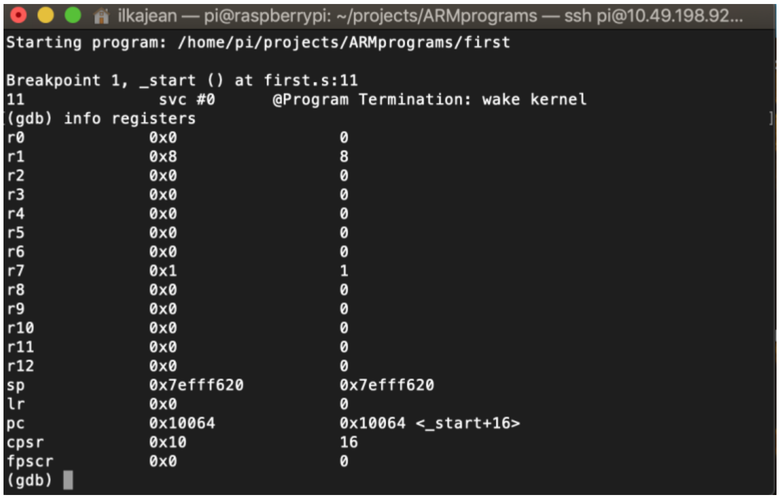
There’s no output shown, although we assembled the code we did not do any action that will produce in output. We execute the program no instruction was given to print in output. There was no debugger/breakpoint used so we are not stepping into the program to see the values. We only assembled and linked but did not use the gdb.


ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...

```
pi@raspberrypi:~/projects/ARMprograms $ nano first.s
pi@raspberrypi:~/projects/ARMprograms $ as -o first.o first.s
pi@raspberrypi:~/projects/ARMprograms $ ld -o first first.o
pi@raspberrypi:~/projects/ARMprograms $ ./first
pi@raspberrypi:~/projects/ARMprograms $ as -g -o first.o first.s
pi@raspberrypi:~/projects/ARMprograms $ ld -o first first.o
pi@raspberrypi:~/projects/ARMprograms $ 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...done.
(gdb) 
```


ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...

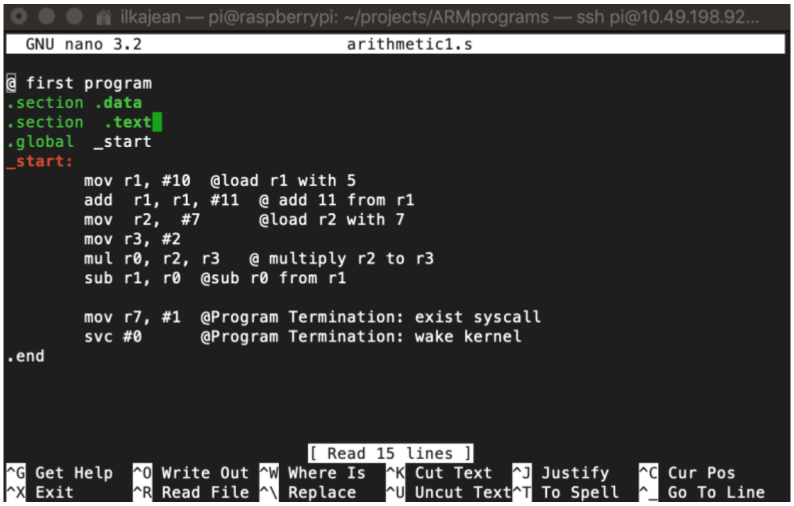
```
Type "apropos word" to search for commands related to "word"...
Reading symbols from first...done.
(gdb) list
1       @ first program
2       .section .data
3       .section  .text
4       .global  _start
5       _start:
6               mov r1, #5  @load r1 with 5
7               sub r1, r1, #1  @ subtract 1 from r1
8               add r1, r1, #4  @4 to r1
9
10              mov r7, #1  @Program Termination: exist syscall
(gdb)
11              svc #0      @Program Termination: wake kernel
12      .end
(gdb) b 11
Breakpoint 1 at 0x10064: file first.s, line 11.
(gdb) run
Starting program: /home/pi/projects/ARMprograms/first

Breakpoint 1, _start () at first.s:11
11              svc #0      @Program Termination: wake kernel
(gdb) 
```



```
ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...
Starting program: /home/pi/projects/ARMprograms/first

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             0x7efff620          0x7efff620
lr             0x0                 0
pc             0x10064             0x10064 <_start+16>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

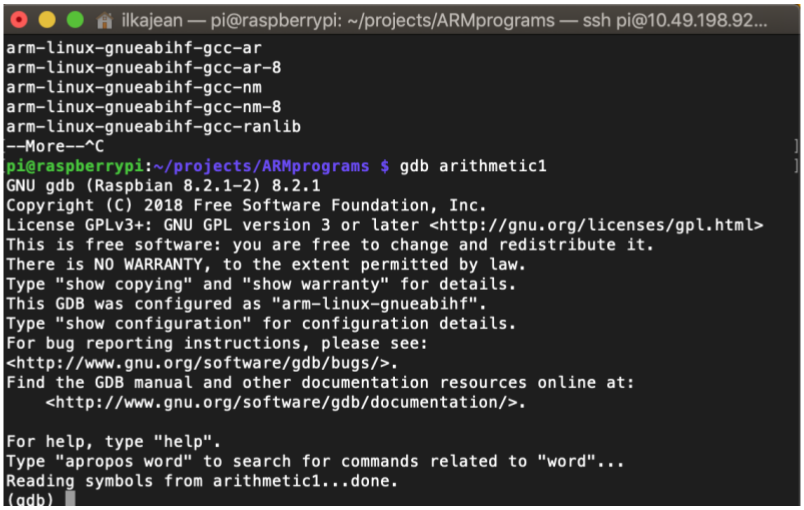
We, first, load the value 5 into register (r1), then we subtract the value 1 from r1 then add the value 4 to register 1. We can see the final value 8 loaded/updated into the r1 register.

Part 2

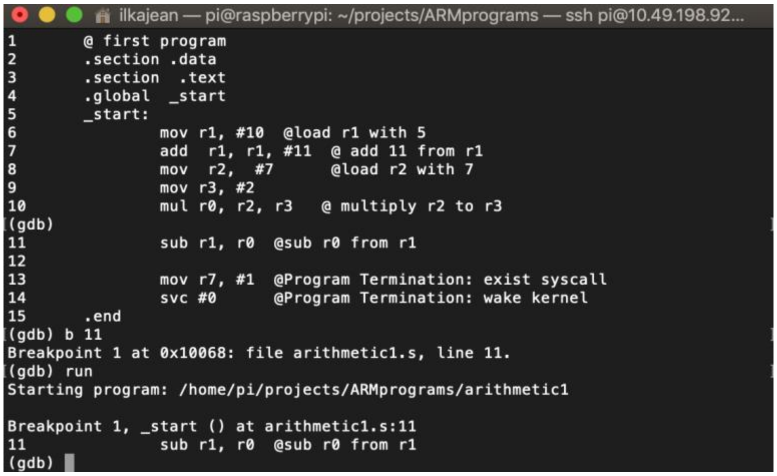


```
ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...
  GNU nano 3.2                          arithmetic1.s

@ first program
.section .data
.section  .text
.global  _start
_start:
        mov r1, #10  @load r1 with 5
        add  r1, r1, #11  @ add 11 from r1
        mov  r2,  #7      @load r2 with 7
        mov r3, #2
        mul r0, r2, r3   @ multiply r2 to r3
        sub r1, r0  @sub r0 from r1

        mov r7, #1  @Program Termination: exist syscall
        svc #0      @Program Termination: wake kernel
.end

                               [ Read 15 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
```

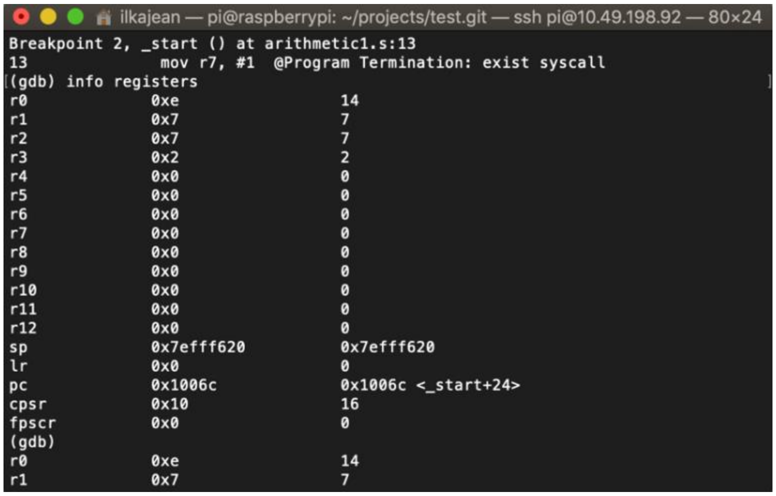
For this program we are using multiple registers to compute A = (A+B) – (C*D). We load the value 5 into register r1 then we add the value 11 into r1. In register r2, we load the value 7 and in register r3 we load the value 2. Using register r0, we multiply r2 to r3. Finally in r1, we subtract r0 from r1. The result is stored in r1. Something I noticed when I was working on this section is that no other register seems to work to multiply, other than r0.


ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...

```
arm-linux-gnueabihf-gcc-ar
arm-linux-gnueabihf-gcc-ar-8
arm-linux-gnueabihf-gcc-nm
arm-linux-gnueabihf-gcc-nm-8
arm-linux-gnueabihf-gcc-ranlib
--More--^C
pi@raspberrypi:~/projects/ARMprograms $ 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) 
```


ilkajean — pi@raspberrypi: ~/projects/ARMprograms — ssh pi@10.49.198.92...

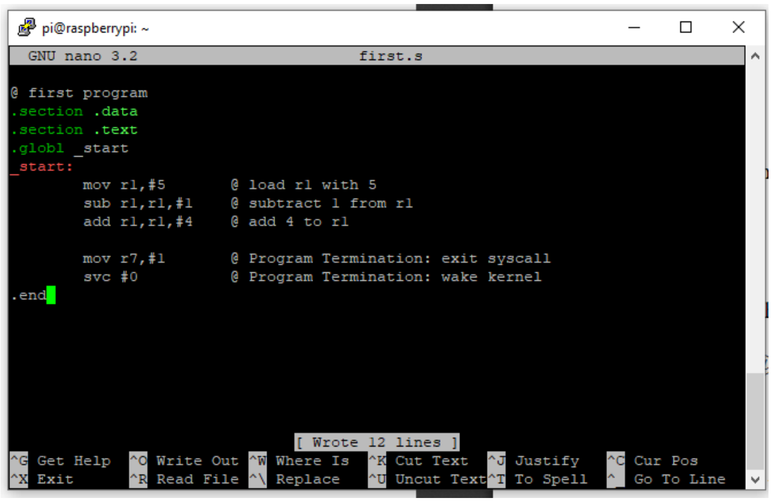
```
1       @ first program
2       .section .data
3       .section  .text
4       .global  _start
5       _start:
6               mov r1, #10  @load r1 with 5
7               add  r1, r1, #11  @ add 11 from r1
8               mov  r2,  #7      @load r2 with 7
9               mov r3, #2
10              mul r0, r2, r3   @ multiply r2 to r3
(gdb)
11              sub r1, r0  @sub r0 from r1
12
13              mov r7, #1  @Program Termination: exist syscall
14              svc #0      @Program Termination: wake kernel
15      .end
(gdb) b 11
Breakpoint 1 at 0x10068: file arithmetic1.s, line 11.
(gdb) run
Starting program: /home/pi/projects/ARMprograms/arithmetic1

Breakpoint 1, _start () at arithmetic1.s:11
11              sub r1, r0  @sub r0 from r1
(gdb) 
```


ilkajean — pi@raspberrypi: ~/projects/test.git — ssh pi@10.49.198.92 — 80×24

```
Breakpoint 2, _start () at arithmetic1.s:13
13              mov r7, #1  @Program Termination: exist syscall
(gdb) info registers
r0             0xe                 14
r1             0x7                 7
r2             0x7                 7
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             0x7efff620          0x7efff620
lr             0x0                 0
pc             0x1006c             0x1006c <_start+24>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
r0             0xe                 14
r1             0x7                 7
```

We see that each value is loaded into a register, including their hexadecimal equivalent(14 = 0xe). The final result is stored/updated in r1 because after all the arithmetic, r0 was subtracted from r1.

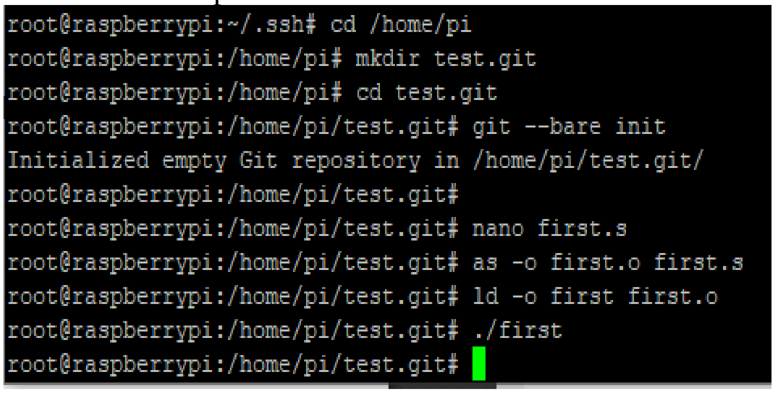
Brian Williams



```
  GNU nano 3.2                          first.s

@ first program
.section .data
.section .text
.globl _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 Termination: exit syscall
        svc #0          @ Program Termination: wake kernel
.end
```

[Wrote 12 lines]

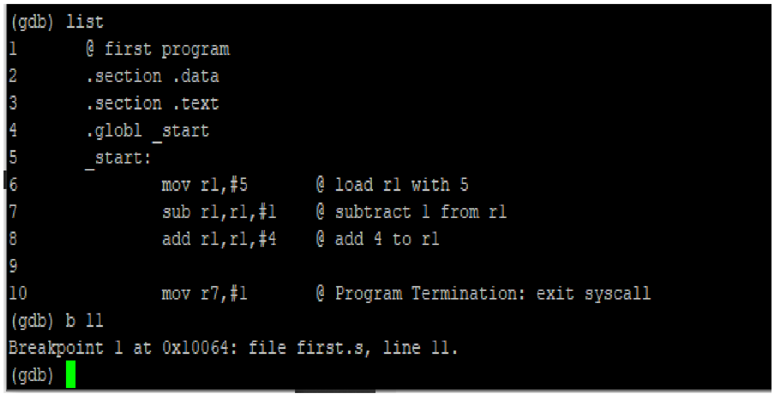
Command Prompt



```
root@raspberrypi:~/.ssh# cd /home/pi
root@raspberrypi:/home/pi# mkdir test.git
root@raspberrypi:/home/pi# cd test.git
root@raspberrypi:/home/pi/test.git# git --bare init
Initialized empty Git repository in /home/pi/test.git/
root@raspberrypi:/home/pi/test.git#
root@raspberrypi:/home/pi/test.git# nano first.s
root@raspberrypi:/home/pi/test.git# as -o first.o first.s
root@raspberrypi:/home/pi/test.git# ld -o first first.o
root@raspberrypi:/home/pi/test.git# ./first
root@raspberrypi:/home/pi/test.git#
```

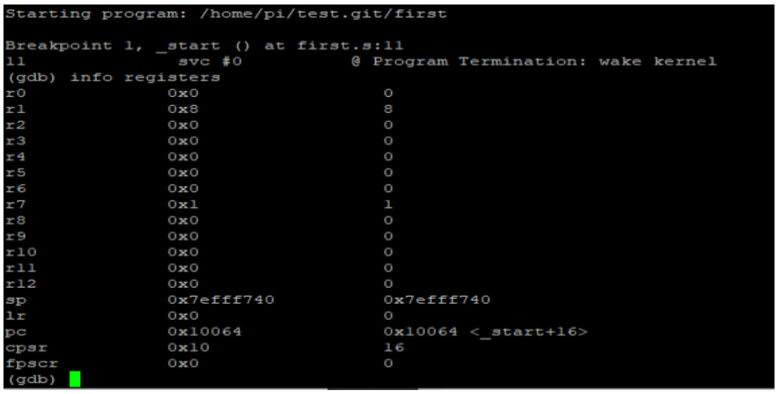
First, I wrote, assembled, and linked the Assembly program Upon running the program, no output appeared in the terminal. I was not expecting an output of any sort because the assembly code written was merely register arithmetic.

GDB debugger:



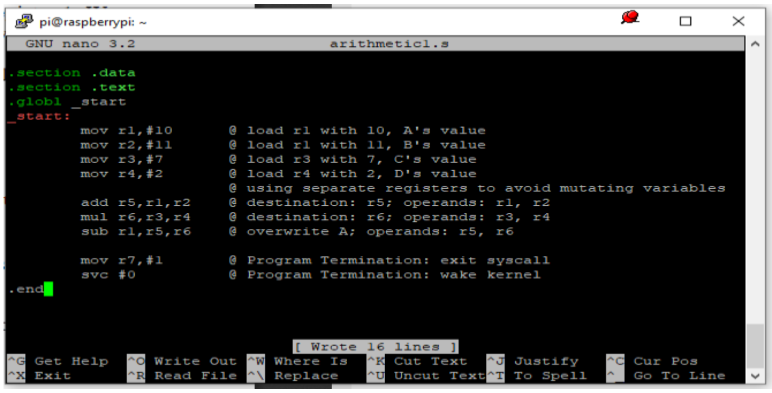
```
(gdb) list
1       @ first program
2       .section .data
3       .section .text
4       .globl _start
5       _start:
6               mov r1,#5       @ load r1 with 5
7               sub r1,r1,#1    @ subtract 1 from r1
8               add r1,r1,#4    @ add 4 to r1
9
10              mov r7,#1  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

Registers at the breakpoint prior to termination: r1 is 8(5-1+4); r7 is 1, remaining arithmetic registers are 0



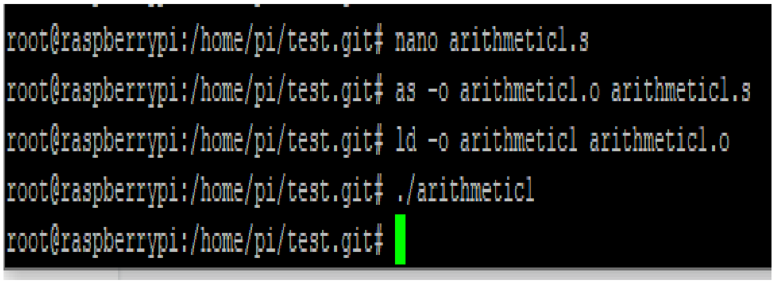
```
Starting program: /home/pi/test.git/first

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             0x7efff740          0x7efff740
lr             0x0                 0
pc             0x10064             0x10064 <_start+16>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

Second program:



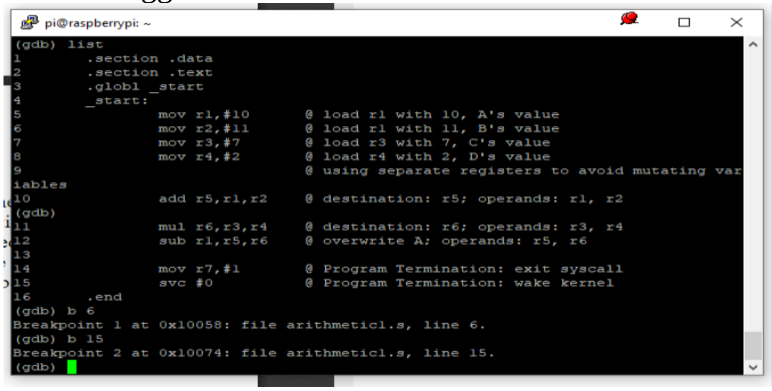
```
  GNU nano 3.2                           arithmetic1.s

.section .data
.section .text
.globl _start
_start:
        mov r1,#10        @ load r1 with 10, A's value
        mov r2,#11        @ load r1 with 11, B's value
        mov r3,#7         @ load r3 with 7, C's value
        mov r4,#2         @ load r4 with 2, D's value
                          @ using separate registers to avoid mutating variables
        add r5,r1,r2      @ destination: r5; operands: r1, r2
        mul r6,r3,r4      @ destination: r6; operands: r3, r4
        sub r1,r5,r6      @ overwrite A; operands: r5, r6

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


                                 [ Wrote 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
```

Command Prompt:



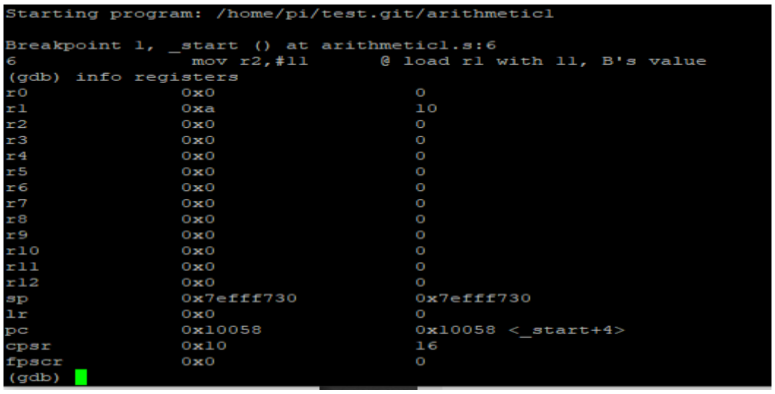
```
root@raspberrypi:/home/pi/test.git# nano arithmetic1.s
root@raspberrypi:/home/pi/test.git# as -o arithmetic1.o arithmetic1.s
root@raspberrypi:/home/pi/test.git# ld -o arithmetic1 arithmetic1.o
root@raspberrypi:/home/pi/test.git# ./arithmetic1
root@raspberrypi:/home/pi/test.git#
```

“arithmetic1” was saved, assembled, and linked. After it was run, again, no output.
GDB debugger interface:



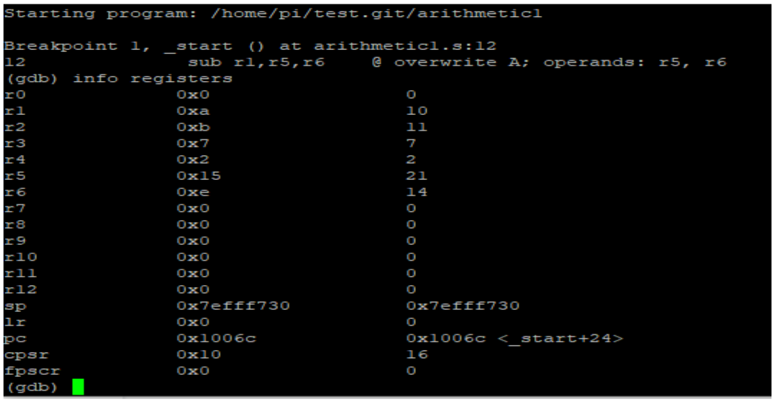
```
(gdb) list
1       .section .data
2       .section .text
3       .globl _start
4       _start:
5               mov r1,#10        @ load r1 with 10, A's value
6               mov r2,#11        @ load r1 with 11, B's value
7               mov r3,#7         @ load r3 with 7, C's value
8               mov r4,#2         @ load r4 with 2, D's value
9                                 @ using separate registers to avoid mutating var
iables
10              add r5,r1,r2      @ destination: r5; operands: r1, r2
(gdb)
11              mul r6,r3,r4      @ destination: r6; operands: r3, r4
12              sub r1,r5,r6      @ overwrite A; operands: r5, r6
13
14              mov r7,#1         @ Program Termination: exit syscall
15              svc #0            @ Program Termination: wake kernel
16      .end
(gdb) b 6
Breakpoint 1 at 0x10058: file arithmetic1.s, line 6.
(gdb) b 15
Breakpoint 2 at 0x10074: file arithmetic1.s, line 15.
(gdb)
```

Registers before line 6 (most importantly, before r1 is overwritten): r1 is 10 (as expected), and all other arithmetic registers are 0



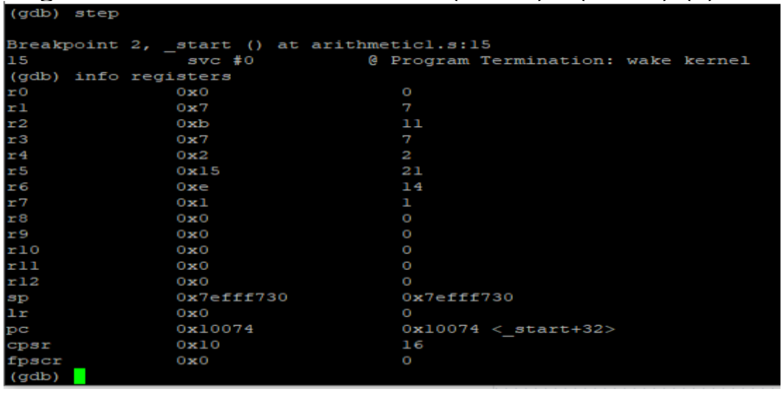
```
Starting program: /home/pi/test.git/arithmetic1

Breakpoint 1, _start () at arithmetic1.s:6
6               mov r2,#11      @ load r1 with 11, B's value
(gdb) info registers
r0             0x0                 0
r1             0xa                 10
r2             0x0                 0
r3             0x0                 0
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             0x7efff730          0x7efff730
lr             0x0                 0
pc             0x10058             0x10058 <_start+4>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

Registers before A is overwritten: r1 is A (10), r2 is B (11), r3 is C (7), r4 is D (2), r5 is A + B (21), r6 is C * D (7 * 2); all values are correct



```
Starting program: /home/pi/test.git/arithmetic1

Breakpoint 1, _start () at arithmetic1.s:12
12              sub r1,r5,r6    @ overwrite A; operands: r5, r6
(gdb) info registers
r0             0x0                 0
r1             0xa                 10
r2             0xb                 11
r3             0x7                 7
r4             0x2                 2
r5             0x15                21
r6             0xe                 14
r7             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff730          0x7efff730
lr             0x0                 0
pc             0x1006c             0x1006c <_start+24>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

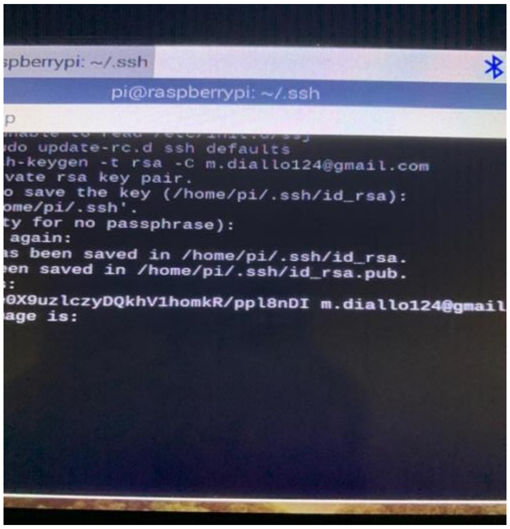
Registers before termination: r1 is (A + B) - (C * D) (7)



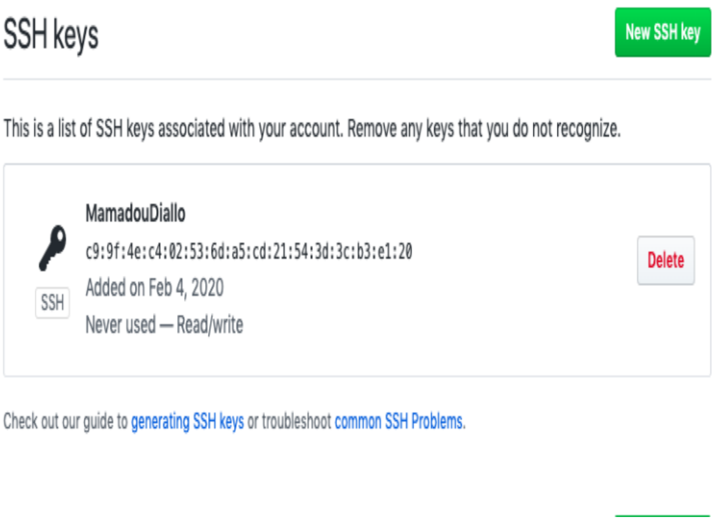
```
(gdb) step

Breakpoint 2, _start () at arithmetic1.s:15
15              svc #0          @ Program Termination: wake kernel
(gdb) info registers
r0             0x0                 0
r1             0x7                 7
r2             0xb                 11
r3             0x7                 7
r4             0x2                 2
r5             0x15                21
r6             0xe                 14
r7             0x1                 1
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff730          0x7efff730
lr             0x0                 0
pc             0x10074             0x10074 <_start+32>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

Mamadou Diallo

Connecting to GitHub with SSH:
I began by installing Git on the raspberry pi, installed SSH, and then I created the SSH key locally on the raspberry pi. Afterwards, I used the key that was generated by the pi, and on GitHub I created an SSH key with that key. Then, I created an empty git repository, and started working on code inside that file.



```
spberrypi: ~/.ssh
pi@raspberrypi: ~/.ssh
do update-rc.d ssh defaults
h-keygen -t rsa -C m.diallo124@gmail.com
vate rsa key pair.
o save the key (/home/pi/.ssh/id_rsa):
ome/pi/.ssh'.
ty for no passphrase):
 again:
s been saved in /home/pi/.ssh/id_rsa.
en saved in /home/pi/.ssh/id_rsa.pub.
:
0X9uzlczyDQkhV1homkR/ppl8nDI m.diallo124@gmail
age is:
```



SSH keys

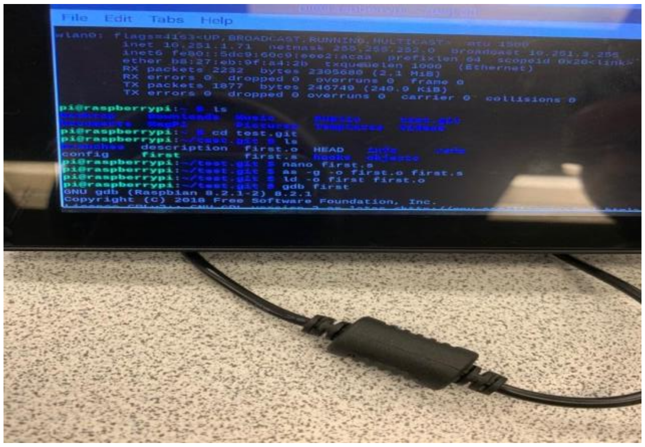
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

MamadouDiallo
c9:9f:4e:c4:02:53:6d:a5:cd:21:54:3d:3c:b3:e1:20
Added on Feb 4, 2020
Never used — Read/write

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

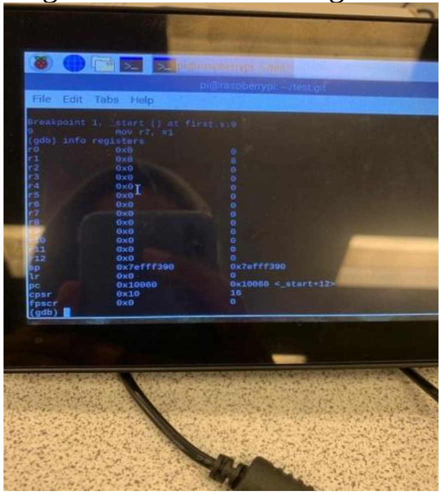
ARM assembler in Raspberry Pi:

In the terminal of the raspberry pi, I began writing my first program by typing “nano first.s”. After writing the program and saving it, I assembled the link, and linked it, so it could become an executable file.

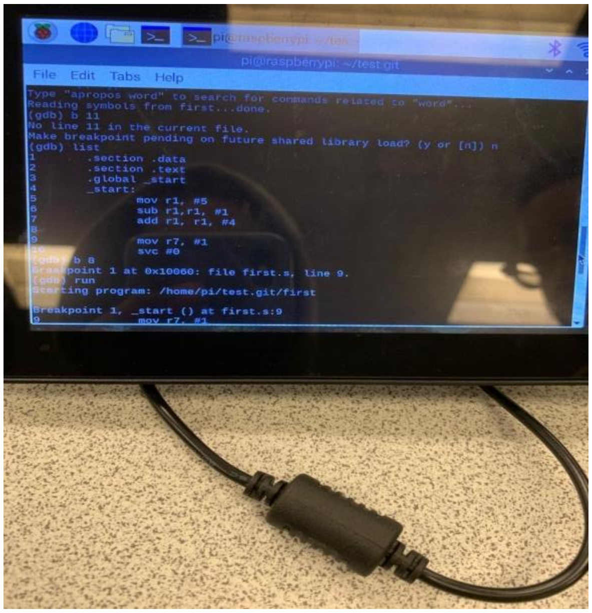


```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.251.1.71  netmask 255.255.252.0  broadcast 10.251.3.255
        inet6 fe80::5dc9:60c9:eee2:acaa  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:9f:a4:2b  txqueuelen 1000  (Ethernet)
        RX packets 2232  bytes 2305680 (2.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1877  bytes 246749 (240.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
pi@raspberrypi:~ $ ls
pi@raspberrypi:~ $ cd test.git
pi@raspberrypi:~/test.git $ ls
branches  description  first.o  HEAD   info   refs
config    first        first.s  hooks  objects
pi@raspberrypi:~/test.git $ nano first.s
pi@raspberrypi:~/test.git $ as -g -o first.o first.s
pi@raspberrypi:~/test.git $ ld -o first first.o
pi@raspberrypi:~/test.git $ gdb first
GNU gdb (Raspbian 8.2.1-2) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
```

Then, I ran the program, and afterwards, I launched a “GNU Debugger” using various commands. I also learned how to set a breakpoint by typing (gdb) b “11”. After running the program, I also was taught how to view the registers.



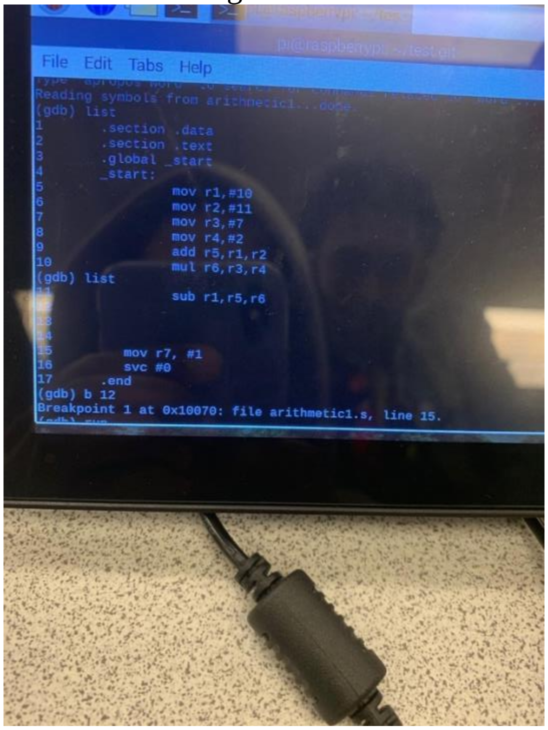
```
Breakpoint 1, _start () at first.s:9
9               mov r7, #1
(gdb) info registers
r0             0x0                 0
r1             0x0                 0
r2             0x0                 0
r3             0x0                 0
r4             0x0                 0
r5             0x0                 0
r6             0x0                 0
r7             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff390          0x7efff390
lr             0x0                 0
pc             0x10060             0x10060 <_start+12>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```



```
Type "apropos word" to search for commands related to "word"...
Reading symbols from first...done.
(gdb) b 11
No line 11 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) list
1           .section .data
2           .section .text
3           .global _start
4           _start:
5                   mov r1, #5
6                   sub r1,r1, #1
7                   add r1, r1, #4
8
9                   mov r7, #1
10                  svc #0
(gdb) b 8
Breakpoint 1 at 0x10060: file first.s, line 9.
(gdb) run
Starting program: /home/pi/test.git/first

Breakpoint 1, _start () at first.s:9
9                   mov r7, #1
```

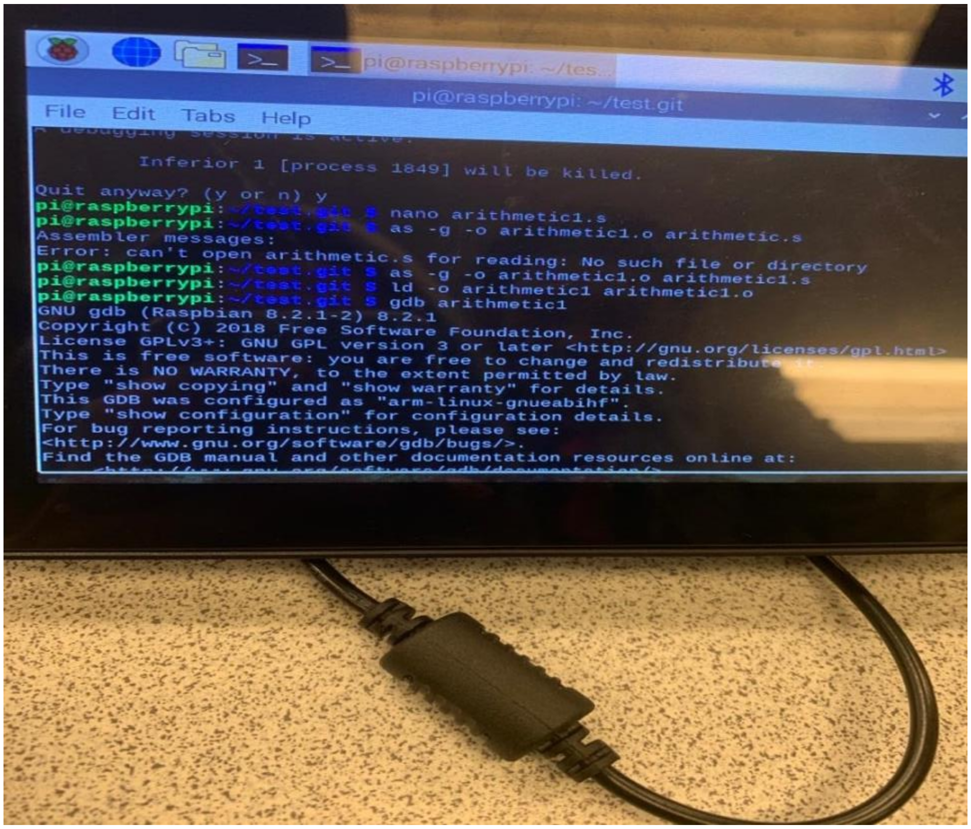
Part 2

Next, I wrote another program called "arithmetic1.s that calculated the expression "A = (A+B) – (C*D)". I used the same steps to create another program, and I used several instruction operands and stored the values in several registers, before calculating them. In the end, it stored the final value in a register.



```
Reading symbols from arithmetic1...done.
(gdb) list
1           .section .data
2           .section .text
3           .global _start
4           _start:
5                   mov r1,#10
6                   mov r2,#11
7                   mov r3,#7
8                   mov r4,#2
9                   add r5,r1,r2
10                  mul r6,r3,r4
(gdb) list
11
12                  sub r1,r5,r6
13
14
15              mov r7, #1
16              svc #0
17          .end
(gdb) b 12
Breakpoint 1 at 0x10070: file arithmetic1.s, line 15.
```

I also assembled and linked the program, before turning it into an executable file, which is shown below.

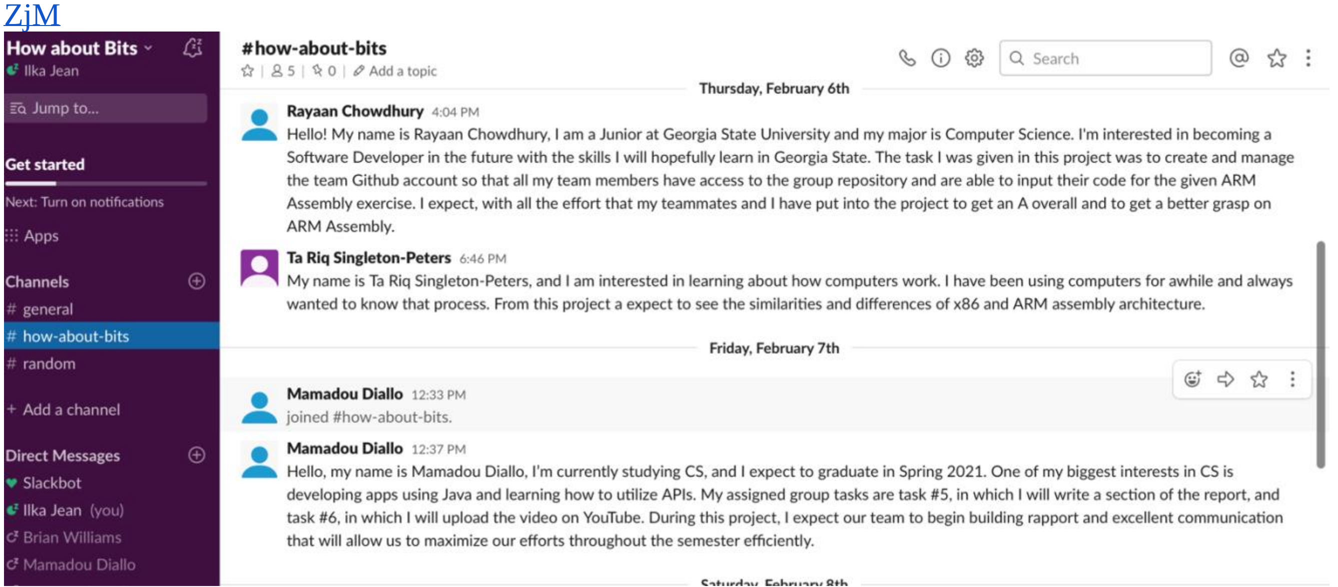


```
A debugging session is active.

        Inferior 1 [process 1849] will be killed.

Quit anyway? (y or n) y
pi@raspberrypi:~/test.git $ nano arithmetic1.s
pi@raspberrypi:~/test.git $ as -g -o arithmetic1.o arithmetic.s
Assembler messages:
Error: can't open arithmetic.s for reading: No such file or directory
pi@raspberrypi:~/test.git $ as -g -o arithmetic1.o arithmetic1.s
pi@raspberrypi:~/test.git $ ld -o arithmetic1 arithmetic1.o
pi@raspberrypi:~/test.git $ 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/>.
```

Appendix

Slack

https://join.slack.com/t/howaboutbits/shared_invite/enQtOTQxNTM2NjI3NjIyLTc2NjJkZDY2ZjkwOWFiMzRlM2ZjNTgwNzgxOGYzOGRkMzQ4OThjMGYxMmY5YTBjNWI0NjhjNTUzZWEyZTY2ZjM



Thursday, February 6th

Rayaan Chowdhury 4:04 PM

Hello! My name is Rayaan Chowdhury, I am a Junior at Georgia State University and my major is Computer Science. I'm interested in becoming a Software Developer in the future with the skills I will hopefully learn in Georgia State. The task I was given in this project was to create and manage the team Github account so that all my team members have access to the group repository and are able to input their code for the given ARM Assembly exercise. I expect, with all the effort that my teammates and I have put into the project to get an A overall and to get a better grasp on ARM Assembly.

Ta Riq Singleton-Peters 6:46 PM

My name is Ta Riq Singleton-Peters, and I am interested in learning about how computers work. I have been using computers for awhile and always wanted to know that process. From this project a expect to see the similarities and differences of x86 and ARM assembly architecture.

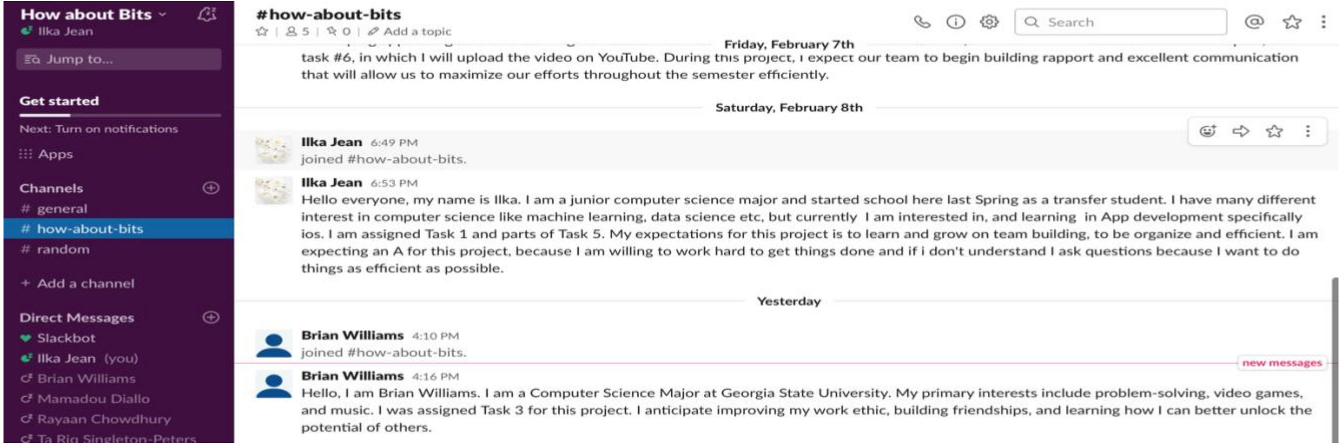
Friday, February 7th

Mamadou Diallo 12:33 PM

joined #how-about-bits.

Mamadou Diallo 12:37 PM

Hello, my name is Mamadou Diallo, I'm currently studying CS, and I expect to graduate in Spring 2021. One of my biggest interests in CS is developing apps using Java and learning how to utilize APIs. My assigned group tasks are task #5, in which I will write a section of the report, and task #6, in which I will upload the video on YouTube. During this project, I expect our team to begin building rapport and excellent communication that will allow us to maximize our efforts throughout the semester efficiently.



Friday, February 7th

task #6, in which I will upload the video on YouTube. During this project, I expect our team to begin building rapport and excellent communication that will allow us to maximize our efforts throughout the semester efficiently.

Saturday, February 8th

Ilka Jean 6:49 PM

joined #how-about-bits.

Ilka Jean 6:53 PM

Hello everyone, my name is Ilka. I am a junior computer science major and started school here last Spring as a transfer student. I have many different interest in computer science like machine learning, data science etc, but currently I am interested in, and learning in App development specifically ios. I am assigned Task 1 and parts of Task 5. My expectations for this project is to learn and grow on team building, to be organize and efficient. I am expecting an A for this project, because I am willing to work hard to get things done and if i don't understand I ask questions because I want to do things as efficient as possible.

Yesterday

Brian Williams 4:10 PM

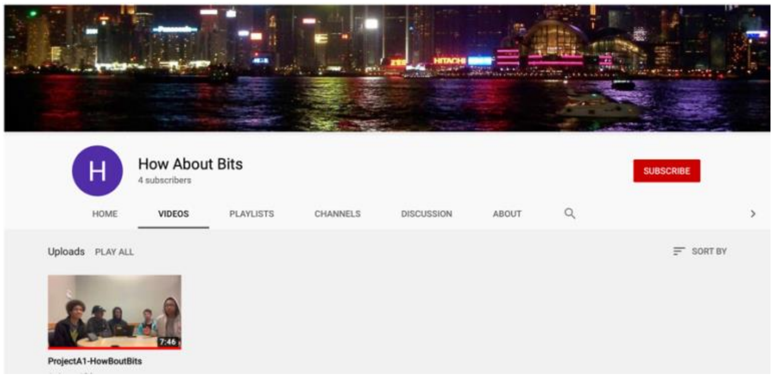
joined #how-about-bits.

Brian Williams 4:16 PM

Hello, I am Brian Williams. I am a Computer Science Major at Georgia State University. My primary interests include problem-solving, video games, and music. I was assigned Task 3 for this project. I anticipate improving my work ethic, building friendships, and learning how I can better unlock the potential of others.

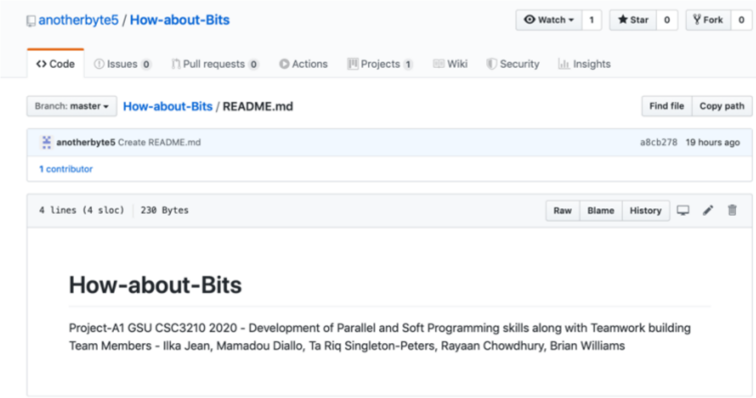
YouTube

https://www.youtube.com/channel/UCKE7zQPcMU6x5kz0OtkaOMg/videos?view_as=subscriber



GitHub

https://github.com/anotherbyte5/How-about-Bits



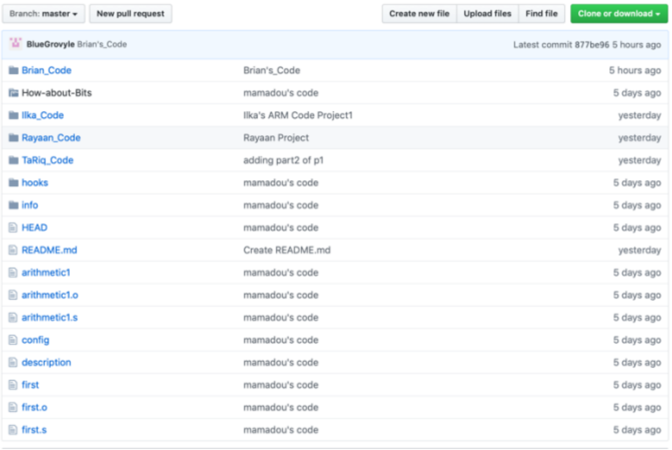
anotherbyte5 Create README.md a8cb278 19 hours ago

1 contributor

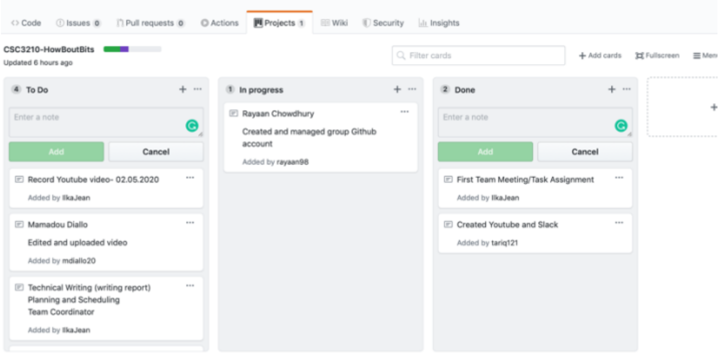
4 lines (4 sloc) | 230 Bytes

How-about-Bits

Project-A1 GSU CSC3210 2020 - Development of Parallel and Soft Programming skills along with Teamwork building
Team Members - Ilka Jean, Mamadou Diallo, Ta Riq Singleton-Peters, Rayaan Chowdhury, Brian Williams



BlueGrovyle Brian's_Code		Latest commit 877be96 5 hours ago
Brian_Code	Brian's_Code	5 hours ago
How-about-Bits	mamadou's code	5 days ago
Ilka_Code	Ilka's ARM Code Project1	yesterday
Rayaan_Code	Rayaan Project	yesterday
TaRiq_Code	adding part2 of p1	yesterday
hooks	mamadou's code	5 days ago
info	mamadou's code	5 days ago
HEAD	mamadou's code	5 days ago
README.md	Create README.md	yesterday
arithmetic1	mamadou's code	5 days ago
arithmetic1.o	mamadou's code	5 days ago
arithmetic1.s	mamadou's code	5 days ago
config	mamadou's code	5 days ago
description	mamadou's code	5 days ago
first	mamadou's code	5 days ago
first.o	mamadou's code	5 days ago
first.s	mamadou's code	5 days ago



CSC3210-HowBoutBits
Updated 6 hours ago

4 To Do

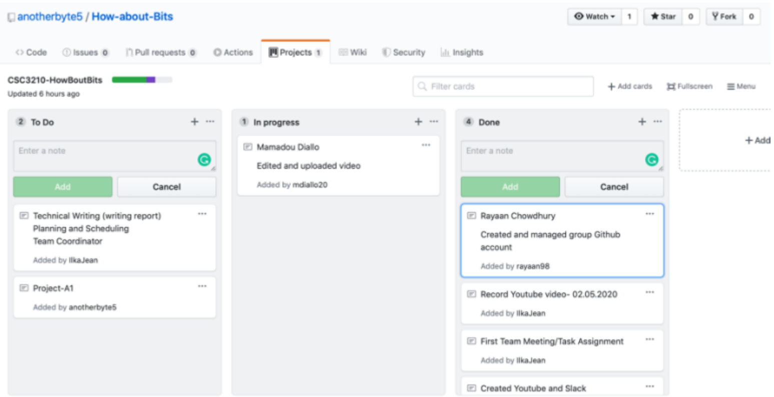
- Record Youtube video- 02.05.2020
  Added by IlkaJean
- Mamadou Diallo
  Edited and uploaded video
  Added by mdiallo20
- Technical Writing (writing report)
  Planning and Scheduling
  Team Coordinator
  Added by IlkaJean

1 In progress

- Rayaan Chowdhury
  Created and managed group Github account
  Added by rayaan98

2 Done

- First Team Meeting/Task Assignment
  Added by IlkaJean
- Created Youtube and Slack
  Added by tariq121



anotherbyte5 / How-about-Bits

CSC3210-HowBoutBits
Updated 6 hours ago

2 To Do

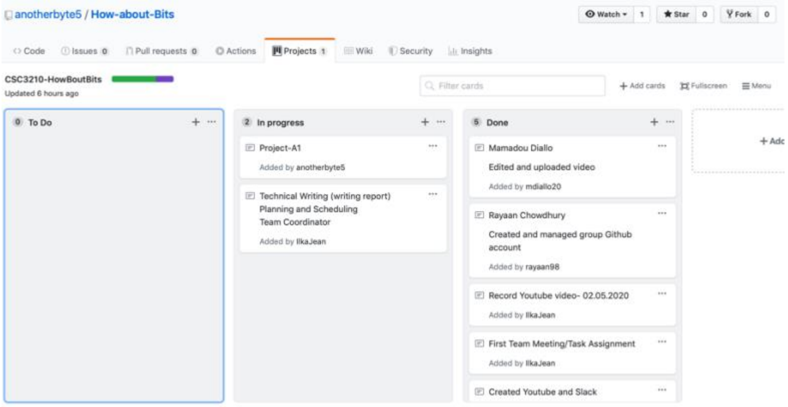
- Technical Writing (writing report)
  Planning and Scheduling
  Team Coordinator
  Added by IlkaJean
- Project-A1
  Added by anotherbyte5

1 In progress

- Mamadou Diallo
  Edited and uploaded video
  Added by mdiallo20

4 Done

- Rayaan Chowdhury
  Created and managed group Github account
  Added by rayaan98
- Record Youtube video- 02.05.2020
  Added by IlkaJean
- First Team Meeting/Task Assignment
  Added by IlkaJean
- Created Youtube and Slack



anotherbyte5 / How-about-Bits

CSC3210-HowBoutBits
Updated 6 hours ago

0 To Do

2 In progress

- Project-A1
  Added by anotherbyte5
- Technical Writing (writing report)
  Planning and Scheduling
  Team Coordinator
  Added by IlkaJean

5 Done

- Mamadou Diallo
  Edited and uploaded video
  Added by mdiallo20
- Rayaan Chowdhury
  Created and managed group Github account
  Added by rayaan98
- Record Youtube video- 02.05.2020
  Added by IlkaJean
- First Team Meeting/Task Assignment
  Added by IlkaJean
- Created Youtube and Slack