Developing Soft and Parallel Programming Skills Using Project-Based Learning

LZKEA

Spring - 2020

Logan Luque

Zejneb Al-Aboudy

Kemper Schlaak

Emilio Enriquez

Ashlynne Matta

Planning and Scheduling

Team Name: LZKEA

Work Breakdown Structure

Assignee Name	Email	Task	Duration (Hours)	Dependency	Due date	Notes
Zejneb Al-Aboudy	Zalaboudy1@student.gsu.edu	Task 5 & Task 1: Technical writing	4 hours	Slack, Github must be done	2/3	
Logan Luque	lluque1@student.gsu.edu	Task 2: Creating Slack and Git. Providing screenshots	4 hours	Team name and usernames required	1/29	
Kemper Schlaak	kschlaak1@student.gsu.edu	Task 3: Learning Teamwork Basics	4 hours	Everyone needs to read teamwork basics and meet together	2/1	Must be ready on due date so I can read over it
Emilio Enriquez	eenriquez3@student.gsu.edu	Task 4: Raspberry PI installation	5 hours	Github must be ready	2/1	Must be ready before YouTube video
Ashlynne Matta	amatta1@student.gsu.edu	Task 6: create YouTube channel, direct	5 hours	Meeting time must be discussed	2/4	

Teamwork Basics

1. Kemper:

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

First, we held an introductory meeting to familiarize ourselves with each other and we talked about our goals for this project. We established some guidelines that could help us in future projects. We concluded that it will be important for us to demonstrate discipline and attention to detail. We also emphasized that asking for help is encouraged as there may be times a task is too much for an individual member. We then decided that the current project's coordinator will serve as the facilitator. To communicate effectively, we will use group chat applications such as slack, and recommend that each member can be reached privately if needed. Us all being on the same page is the most important factor to keeping our satisfaction high.

Work norms:

The coordinator will distribute work and deadlines for this project. We will be rated on our performance with a percentage grade based on the work that has been completed. The coordinator will review the work quality. The group will review and discuss any disputes on opinions of quality of work to see if the work in question meets expectations. The coordinator will assign roles to members that fit what needs to be completed first.

Facilitator norms:

Our group will have a facilitator. This role will be rotated through different members for each project based on who the coordinator is. The facilitator cycle will be chosen based on the order of who joined the group first. The facilitator will be responsible for focusing on the team task and acquiring participation from the by members making sure the team meets the deadline, assisting in solving any problems that may slow the project's completion, helping the group resolve issues, and submitting the report to the professor.

Communication norms:

We have decided to hold at least one meeting a week after class or after lab in order to maintain a consistent flow of communication and collaboration. We also plan to use Slack and GitHub to keep up with responsibilities and the group's priorities.

Handling Difficult Behavior

- **Too quiet** - There can be many reasons why some of us may get quiet. There can be times where one of us may be confused in a meeting, bored of the meeting, or is just distracted. If this were to happen, we should put forth the effort to simplify the content of the meeting and make the meeting more engaging and entertaining.

- **Complains** - If there are complaints being made within the group, the issue should be acknowledged and solved before the group can move on. The group can come to a reasonable solution that works for everyone. Some tactics we came up with were to ask that individual what is bothering them and find out what we can do together, or to take some time for the issue to take work itself out.

When making decisions, if the team is having trouble reaching consensus, what should you do?

We will put the problem in question to a vote, where a simple majority wins. In the instance that there is a tie in voting, the decision falls on the vote of the team coordinator.

What should you do if a person may reach a decision more quickly than others and pressure people to move on before it is a good idea to do so?

Some of us are quicker to reach conclusions than others, so we should not pressure others into the same decision, rather we should give them time and space to focus on the current task. If one of us moves quicker than the rest of the group, they will be required to maintain patience as the tasks get completed and maybe ask if they can be a part of helping get other tasks done in the meantime. Part of being a team is learning how to maintain patience and consistent with our work and behavior.

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?

We realized that all of us are targeting for an A+ in our first meeting. Knowing this, we feel as if we can be organized and trusting of each other as a team and don't have to worry about underperformance and carelessness.

2. **Logan:**

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

For me it's all about communication. I think that the best way to ensure all the tasks get accomplished and team satisfaction stays high is just simple communication. For example, we all met in the library and introduced ourselves, got to know each other, talked about our strengths and weaknesses, and assigned tasks from there. If we were to just assign tasks straight away with no communication before, some people may have felt uncomfortable with their task or like the assignments were unfair. In addition, if you are having a problem completing your task, we encouraged one another to ask the group questions and see if we could all figure it out. If you were to just keep your problem quiet and not communicate, the level of your work could suffer causing the team's grade to fall as well.

Work norms:

Work is distributed at the start of every assignment (after team discussion) by the team coordinator for that specific project. The deadlines are also set by the team coordinator after talking with each member about their task. If someone doesn't follow through on a commitment, the group will decide how much(if any) punishment there will be. If the missed commitment is on a key part of the project, a percentage of the member's grade will be deducted as decided by the rest of the group. The work will be reviewed by the team coordinator once all tasks have been completed and will decide if more work needs to be done. If there is a case where there are differences in opinion about the quality of the work, the group as a whole will discuss it and determine if more work needs to be done, and how much. If people have different work habits, then it is up to the coordinator to determine the best solution for each individual. For example, if one group member likes to get work done early, then the coordinator should assign a task to them that has to be completed first to ensure the rest of the project can be started.

Facilitator norms:

Yes, our group will use a "facilitator"(coordinator). The facilitator will be rotated on a per-project basis and is chosen on the order of which the group members joined the group-me chat. For example, for this project Zejneb is the coordinator and I will be the coordinator on the next project because I joined the chat second, and so on. The facilitator responsibilities include: assigning tasks to each group member, ensuring participation on assigned tasks from each member, setting deadlines, resolving conflicts, setting up the meet times for the group members, being the main communicator between the group and the Professor on any problems, and turning in the team's final product.

Communication norms:

The team communication (when not together) takes place on slack or group-me which are apps that can be downloaded on a phone or computer and used anywhere to ensure we can be reached at any time. The team communication (for in-person meet ups) has been decided to be in a study room at the GSU library on Fridays to ensure focus and productivity.

Handling Difficult Behavior

- **Too quiet** - Fortunately, all of our group members are comfortable putting out their opinions to the team. However, it can be easy for someone to feel unwelcome in a group, uncomfortable sharing their thoughts, and being quiet for a multitude of other reasons. At that point the responsibility is on the other team members to try and find a solution to the problem. There are many ways to go about this but I have found the most effective to be making jokes, including them in conversation, and asking their opinions on things relating to the project so that you let them know their opinion matters.

- **Complains** - Personally I am not a big fan of when someone complains because I feel it adds very little to the group output. However, on the other hand at least it means they are sharing their opinion which can be useful. So I think that as a team we should take a look at their complaint and come to a conclusion about it as soon as possible so that we can continue. One way of dealing with this is to simply listen to the person's complaint instead of just instantly ignoring it and seeing if they have a legitimate reason for what they're saying. If we decide it has some truth behind it, then we will act accordingly and try to be as open and fair as possible moving forward.

When making decisions, if the team is having trouble reaching consensus, what should you do?

We decided that the best way to make decisions when we cannot reach common ground is to just put the disagreement to a vote. The side with the most votes will be the route we take.

What should you do if a 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 someone reaches a decision before the rest of the group and tries to pressure others into moving on too then we must put the group before the individual and be sure to clearly communicate that it's not an option. We understand everyone works at different speeds, but the flow as a whole should never be rushed by an individual trying to rush things. We must be sure to put the success of the group first, and that may mean compromising with each other to work at a relative speed.

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?

Fortunately, we have all decided that our goal is an A+, and anything below an A is unacceptable. This means that if any work is not up to the group's standard, it will have to be redone until it meets what were looking for. No compromises.

3. **Zejneb:**

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

In order to get tasks accomplished in a timely manner, we have given designated tasks to everyone to oversee. For example, someone assigned with task one will have a certain amount of time to get that completed. In order to keep team member satisfaction high, we will make sure to review everyone's work as a group and make sure it meets the requirements of the project.

Work norms:

The work will be distributed by the coordinator and if someone doesn't follow through with their commitments then the coordinator will rate them using the percentage scale and post it on the work breakdown chart. The work quality will be rated by the coordinator and if the member disagrees then it can be put up for vote with the team members. The coordinator's job is to accommodate work habits so that the most important work gets completed first. The members will be given roles that fit the need of the work required.

Facilitator norms:

Our group will have a facilitator and the position will be rotated in the group based on who joined the group first. The facilitator will be in charge of assigning tasks, scheduling meetings, keeping the members on track of deadlines, and resolving issues within the group and finally submitting the assignment to the professor during office hours on the deadline.

Communication norms:

We have decided that in order to maintain consistent communication and prevent misunderstandings, our group will meet once a week on Fridays after our labs. We will also maintain communication throughout the week using slack and post responsibilities on Github. The coordinator will also be responsible for keeping the group updated on tasks and maintaining clear goals.

Handling Difficult Behavior

• **Too quiet** - A team member can be too quiet for a variety of reasons and it should be the job of the coordinator to make sure everyone is responding in group messages, meetings, and emails in order to make sure everyone is voicing their opinion and a decision from the group is based on what everyone agrees on.

- **Complains** - If there is one or more members complaining about a decision or situation. It will be addressed immediately so that the group may move forward with other plans. The coordinator should address the complaints to the group and work on coming up with a solution.

When making decisions, if the team is having trouble reaching consensus, what should you do?

Since our group is a team of 5 members, a simple majority vote will allow us to resolve discrepancies and come to an official agreement. However in a situation where there is an even number of votes, it is the role of the coordinator to come up with an informed decision that will be in the best interest of the group.

What should you do if a person may reach a decision more quickly than others and pressure people to move on before it is a good idea to do so?

While it is encouraged to make decisions in a timely manner, rushing to a decision is not in the best interest of the group. If a member has reached a decision , they may be asked to compromise with the rest of the group in order to come to an agreement. It should be the role of the team coordinator to talk to the member of the team and explain why the group cannot move with their decision. Members are discouraged from pressuring others to make a decision before the group is ready to move on. Ultimately, the coordinator has the final say on what decisions the group will make. However that decision is based on the best interest of the group and how the team members feel about a decision.

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

As a group we have already discussed what we expect to receive for the assignment and we unanimously agreed that we will be striving for an A+. With that being said, if there is a member who thinks a B is acceptable, the group will still try to aim for an A+. As the semester progresses, the group will reevaluate goals and determine what is still possible with grades we have received so far.

4. Emilio:

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

I believe that cooperation and time-management are key to our success. It helped greatly that this group was able to get plans set early in terms of setting tasks, sending docs, and making the groupme despite having some complications in terms of who's in our group at the beginning. It also helped that this group of people had some great chemistry together. We were able to openly ask for help and everyone was more than willing to help. Not to mention, this kept everyone's satisfaction high because it made meetings more fun while also remaining very productive.

Work norms:

The project is distributed into separate tasks and each task will be assigned to a different person which is decided by the team coordinator. Once everyone finishes the tasks they are to be sent to the coordinator so he/she can review it and approve it. Each person will be given a percentage grade on their task by the coordinator. From 0%, meaning they just didn't do it, up to 100% meaning that they did the task fully and the coordinator approves.

Facilitator Norms:

Each project the group will designate a new facilitator. For us, we have decided on the rotation based on the order who joined the group. The facilitator is in charge of assigning tasks, evaluating everyone's works, getting the group to meet.

Communication Norms:

The group maintains steady communication over groupme as well as Slack. We keep each other updated on our progress on tasks and tell each other any other news that we need to know. We also try to meet once a week in the library to update each other and also help each other with whatever.

Handling Difficult Behavior

Too quiet - If a team member is too quiet, we will just have to try and talk to that individual in person, probably in class and tell them that it is crucial to the success of the group if we all communicate better. And we will try to reach out to him personally instead of just saying stuff in the group chat to ensure that he is communicating with at least one of us

Complains - If a team member is complaining too much, I think we should discuss it as a group during the meeting. We would hear the complaints and discuss it as a group on how to go further. Essentially just talking out the problems and resolving any issues we may have as a team.

When making decisions, if the team is having trouble reaching consensus, what should you do?

If the team is having trouble coming to a consensus we decided to leave it up to a vote and majority wins.

What should you do if a person may reach a decision more quickly than others and pressure people to move on before it is a good idea to do so?

Although it is good to have people who can make decisions decisively and quickly, the priority should be that we make the best decision for the team. The time that it takes to make a decision is less important than making the best and most logical one. Therefore, we should make it clear that no one is to be rushed into making any decisions and everyone can take their time.

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?

No matter what people deem to be an acceptable grade to them, every group member should try and do their best on the assignment. Luckily, this group all shares the same goal which helps us work better as a team. If this were not the case, I think it'd be incredibly rude to not put in your best effort even if you think that a B is an acceptable grade.

5. Ashlynne:

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

To keep the tasks accomplished, we plan to create a groupME and discuss which task will go to who. The coordinator will determine if the split work is fair and will help in the event of any disagreements. We will communicate and come to a final decision on what we want our outcome to look like and set deadlines that should be strictly followed.

Work norms:

We expect every team member to have their work done by the set deadline or before and if there are issues with following through with a deadline then we expect full communication and discussion with the coordinator. If the work is still not completed in a timely manner and done correctly then the group will decide how much percentage they will lose on the assessment.

Facilitator norms:

We decided to pick the facilitator based on order of who joined the group, but in the event of a disagreement, we will just vote on who will coordinate each assignment. The coordinator must effectively communicate with the group and set deadlines and review over all work for that assignment before turning it in.

Communication norms:

We have decided that in order to maintain consistent communication that we must reply to each other within 24 hours, excluding Saturday and Sunday. We have also agreed to meet once a week in order to prevent confusion and to be able to prepare better for the assignments so no one gets left behind.

Handling Difficult Behavior

- **Too quiet** - if a team member is not speaking up enough and their ideas are not getting heard then it is the job of the coordinator to get them to be proactive. If they refuse, then points will be taken off the assessment.

- **Complains** - If a team member is not satisfied with a decision then we will immediately rectify the situation because we want to make sure we are producing the best work possible and make sure everyone is happy with our outcome.

When making decisions, if the team is having trouble reaching consensus, what should you do?

We have a group of 5 people so we could vote, however the coordinator gets final say. We can respectfully discuss why we cannot agree and come to a compromise so our grade and work is not affected

What should you do if a 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 want every team member to be satisfied with our work, so if one person comes to a conclusion quicker than others then we must give the member time to think and decide what they think is best. We want to produce the best quality A+ work so while working quickly is ideal. We do not want to sacrifice the quality of our work

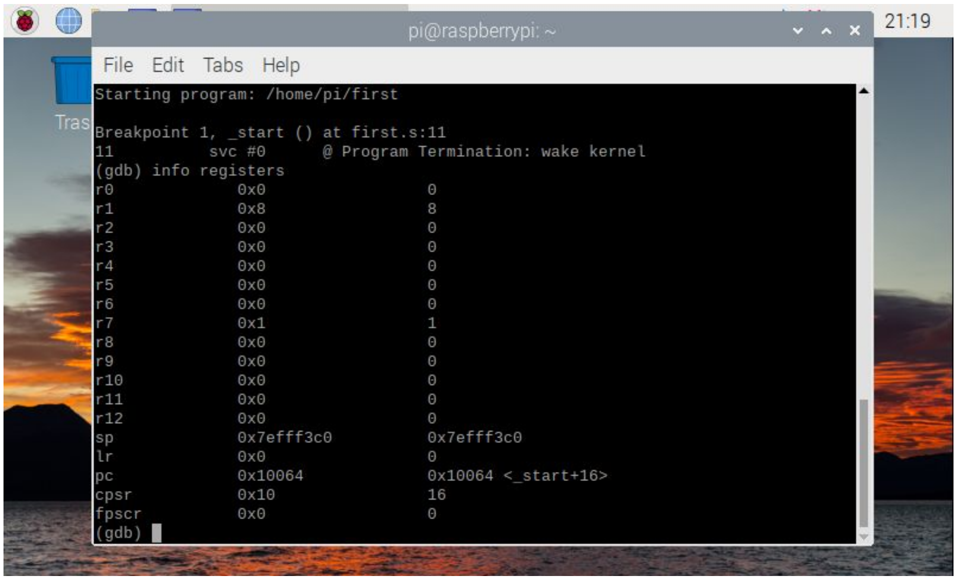
What happens if most people on the team want to get an "A" on the assignment, but another person decides that a "B" will be acceptable?

One of our main goals as a group is to attain an A+, so we diligently read the assignment and the rubric to make sure that our work is of quality. We agreed on day 1, that nothing less than A work is acceptable and we are working very hard to achieve this goal and will continue to do so throughout the semester.

Raspberry PI Installation and ARM Assembly Programming

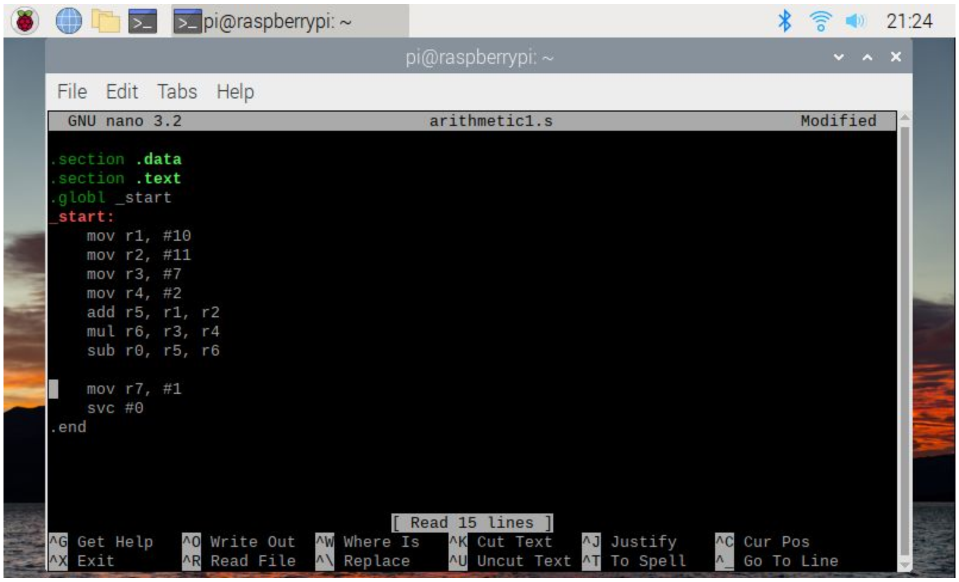
1. Kemper

Debugging the first.s program reveals the value that is stored in the r1 register after it goes through the process of having the value 5 loaded onto it, then having 1 taken away from it, and finally having 4 added to the register. Looking at the info registers reveals that r1 is left with the value of 8.



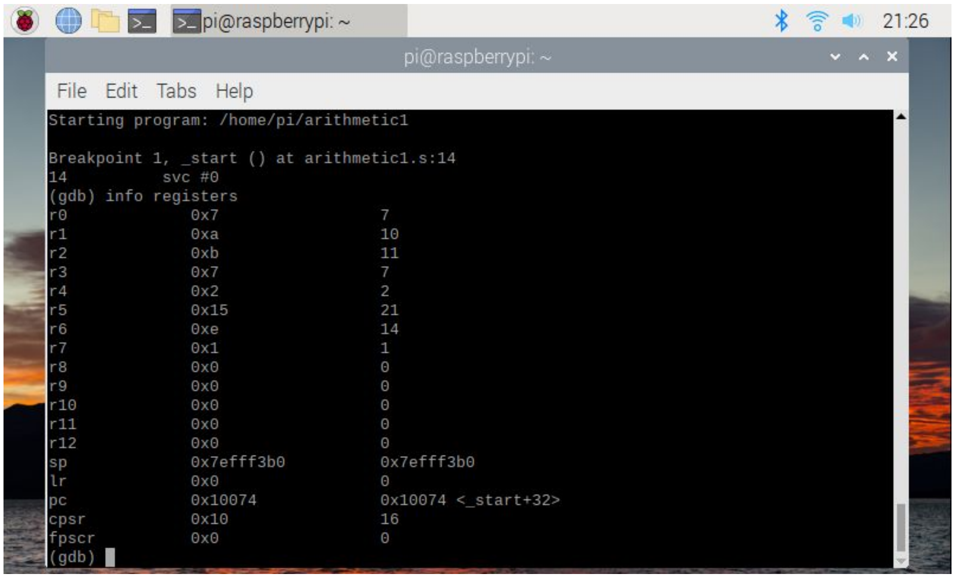
```
Starting program: /home/pi/first

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             0x7efff3c0          0x7efff3c0
lr             0x0                 0
pc             0x10064             0x10064 <_start+16>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

The individual values for A, B, C and D were loaded into individual registers at the start of artithmetic1.s. Then the add instruction was used to combine the values of r1 and r2 into a new register r5, and the register r6 was used to store the output of the mul instruction on r3 and r4. The last instruction was to subtract r6 from r5 and store its output in the r0 register. The code had to be written in this order to maintain the order-of-operations hierarchy.



```
  GNU nano 3.2                    arithmetic1.s                    Modified

.section .data
.section .text
.globl _start
_start:
    mov r1, #10
    mov r2, #11
    mov r3, #7
    mov r4, #2
    add r5, r1, r2
    mul r6, r3, r4
    sub r0, r5, r6

    mov r7, #1
    svc #0
.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
```

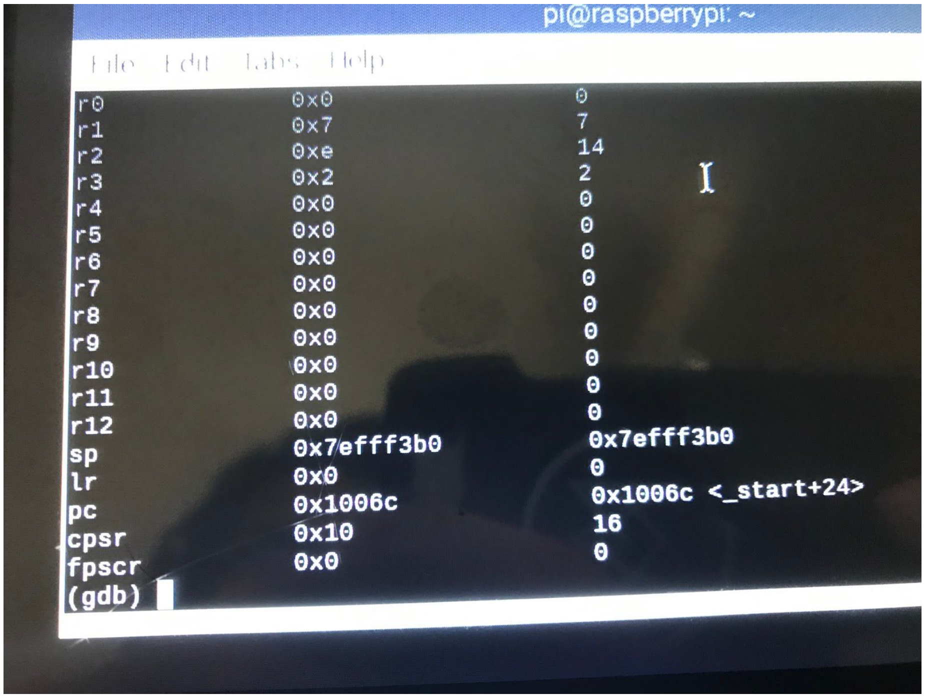
Running the debugger shows the value of the registers used. Multiple registers were used in order to easily see the data at each step of the process. Executing ‘info registers’ in the debugger reveals that r0 contains the value 7, which is the correct output of the operation.



```
Starting program: /home/pi/arithmetic1

Breakpoint 1, _start () at arithmetic1.s:14
14            svc #0
(gdb) info registers
r0             0x7                 7
r1             0xa                 10
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             0x7efff3b0          0x7efff3b0
lr             0x0                 0
pc             0x10074             0x10074 <_start+32>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

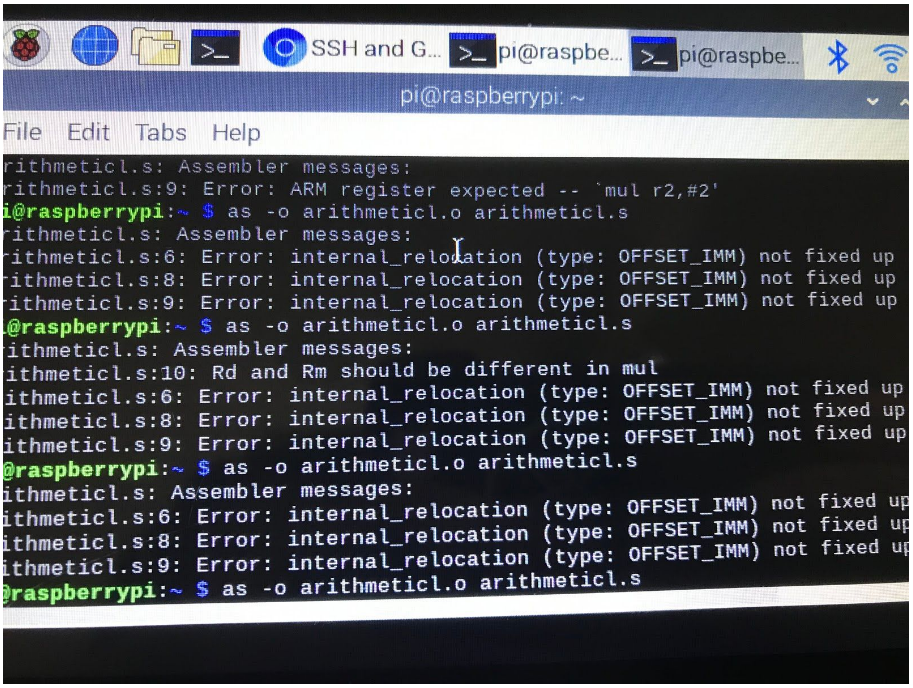
2. Emilio:

For this task, I had trouble in the beginning with the wording of the first task. I did not interpret it correctly at first. I wasn’t sure where to do the assembling, linking, and debugging. My thoughts were that they would be done using nano, which was where I write the program. But after many tries, I finally figured out the correct way. Once I figured it out, following the instructions was pretty straight forward. Once I got to the end and opened the registers, I observed the value 8 in r1.



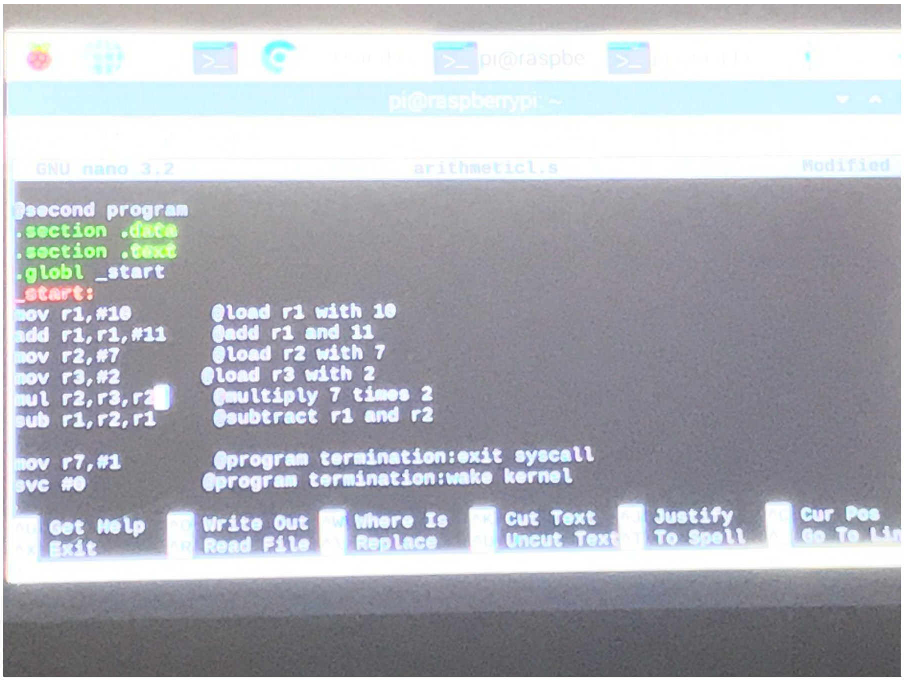
```
r0             0x0                 0
r1             0x7                 7
r2             0xe                 14
r3             0x2                 2
r4             0x0                 0
r5             0x0                 0
r6             0x0                 0
r7             0x0                 0
r8             0x0                 0
r9             0x0                 0
r10            0x0                 0
r11            0x0                 0
r12            0x0                 0
sp             0x7efff3b0          0x7efff3b0
lr             0x0                 0
pc             0x1006c             0x1006c <_start+24>
cpsr           0x10                16
fpscr          0x0                 0
(gdb)
```

The second part of the ARM Assembly task also gave me difficulties. At the beginning, I continuously got an error message that said “Error: ARM register expected – ‘mul r2, #2’”. I was confused for about 10 mins, until I realized that the formatting in my code was wrong. Once I properly read the instructions and did research on the proper syntax that error message went away. After this, I began to get a new error message as seen on the third picture shown it said “Error: internal_relocation not fixed up”.



```
rithmeticl.s: Assembler messages:
rithmeticl.s:9: Error: ARM register expected -- `mul r2,#2'
i@raspberrypi:~ $ as -o arithmeticl.o arithmeticl.s
rithmeticl.s: Assembler messages:
rithmeticl.s:6: Error: internal_relocation (type: OFFSET_IMM) not fixed up
rithmeticl.s:8: Error: internal_relocation (type: OFFSET_IMM) not fixed up
rithmeticl.s:9: Error: internal_relocation (type: OFFSET_IMM) not fixed up
@raspberrypi:~ $ as -o arithmeticl.o arithmeticl.s
ithmeticl.s: Assembler messages:
ithmeticl.s:10: Rd and Rm should be different in mul
ithmeticl.s:6: Error: internal_relocation (type: OFFSET_IMM) not fixed up
ithmeticl.s:8: Error: internal_relocation (type: OFFSET_IMM) not fixed up
ithmeticl.s:9: Error: internal_relocation (type: OFFSET_IMM) not fixed up
@raspberrypi:~ $ as -o arithmeticl.o arithmeticl.s
ithmeticl.s: Assembler messages:
ithmeticl.s:6: Error: internal_relocation (type: OFFSET_IMM) not fixed up
ithmeticl.s:8: Error: internal_relocation (type: OFFSET_IMM) not fixed up
ithmeticl.s:9: Error: internal_relocation (type: OFFSET_IMM) not fixed up
raspberrypi:~ $ as -o arithmeticl.o arithmeticl.s
```

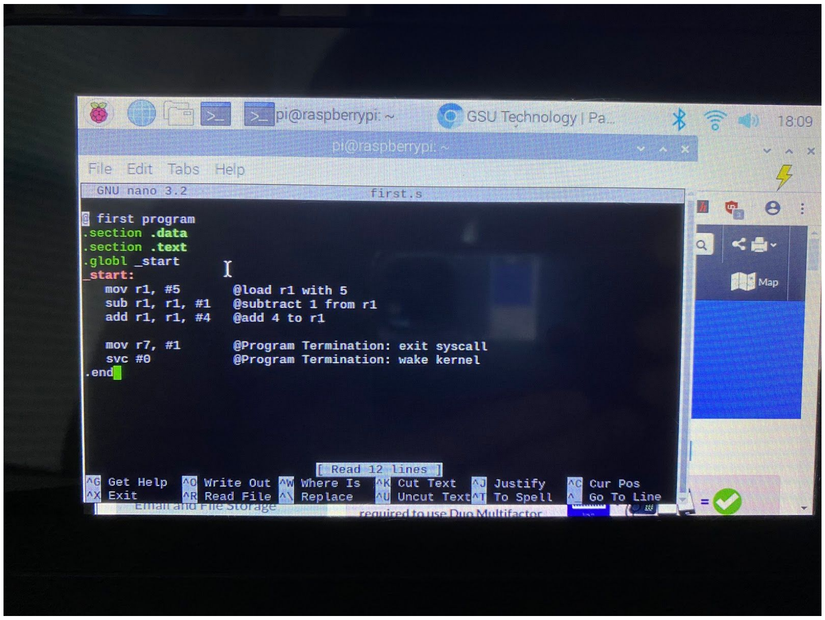
I was using ldr to load the initial values on to registers, which is what I believed to cause the internal relocation error. When I changed my code from ldr to mov, which can be seen in second picture, it was fixed. After researching, I still am slightly unclear on what the error message meant. Despite this, I got the value that I wanted. Looking at the first picture, the value for (A+B)-(C*D) was 7 in r1.



```
GNU nano 3.2                arithmeticl.s                    Modified

@second program
.section .data
.section .text
.globl _start
_start:
mov r1,#10          @load r1 with 10
add r1,r1,#11       @add r1 and 11
mov r2,#7           @load r2 with 7
mov r3,#2          @load r3 with 2
mul r2,r3,r2        @multiply 7 times 2
sub r1,r2,r1        @subtract r1 and r2

mov r7,#1           @program termination:exit syscall
svc #0             @program termination:wake kernel
```

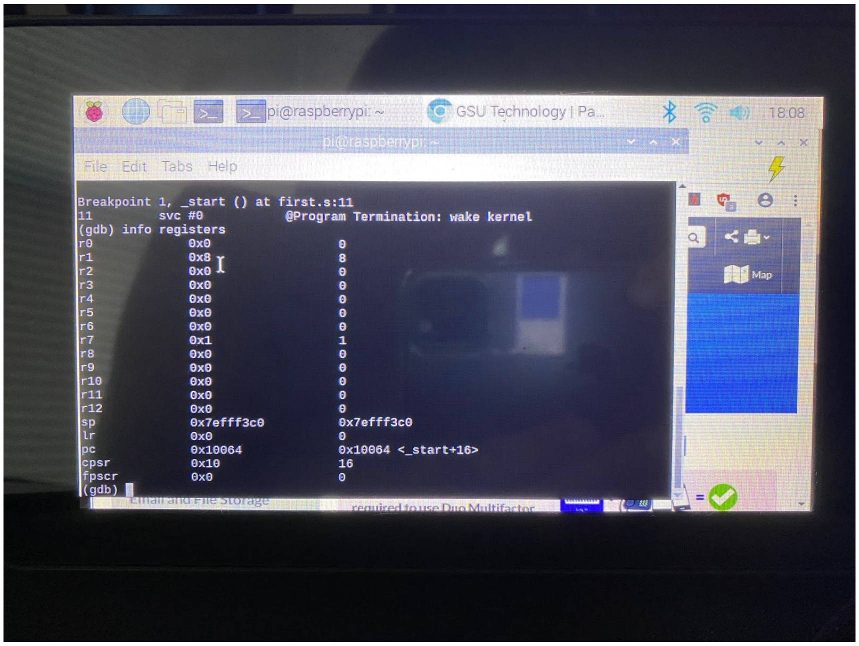
3. Logan

For this task I began by first writing the assembly code to a file we named first.s using a text editor called nano. This was relatively easy because it was just copying from the directions exactly what to write.



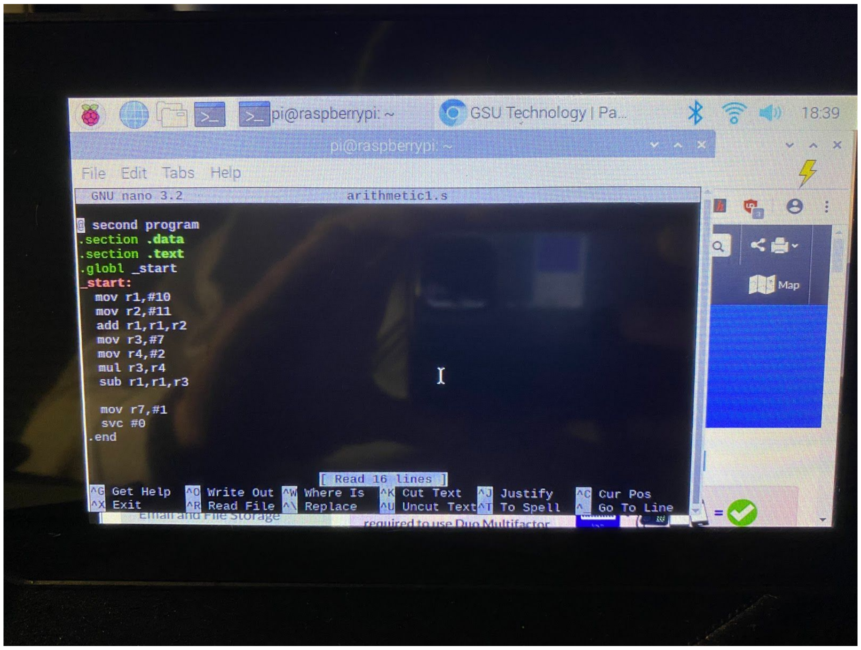
```
GNU nano 3.2                first.s

@ 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

                        [ Read 12 lines ]
```

Then I assembled the first.s file and created an object file from it. Once there, I was able to link the object file to get an executable that could be run in the GDB (GNU Debugger) to set a breakpoint and see the results of our program in the registers. As expected, the register r1 contained the value 8. (Seen below)



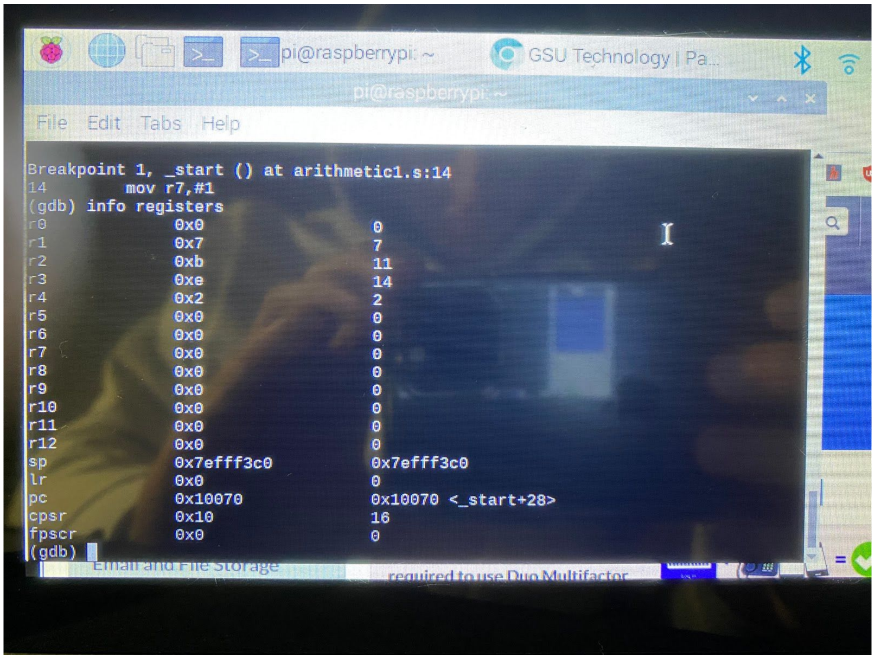
```
Breakpoint 1, _start () at first.s:11
11          svc #0          @Program Termination: wake kernel
(gdb) info registers
r0             0x0                 0
r1             0x8                 8
r2             0x0                 0
r3             0x0                 0
r4             0x0         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

For the second part of the project, we had to create our own assembly program that would do simple arithmetic without any multiplication. The part that was the most difficult to me was trying to create the code without declaring any variables and only using registers to keep track of the data. However, the lab for the class helped me to learn how to use the registers and the following program was my result:



```
  GNU nano 3.2                   arithmetic1.s

@ second program
.section .data
.section .text
.globl _start
_start:
  mov r1,#10
  mov r2,#11
  add r1,r1,r2
  mov r3,#7
  mov r4,#2
  mul r3,r4
  sub r1,r1,r3

  mov r7,#1
  svc #0
.end

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

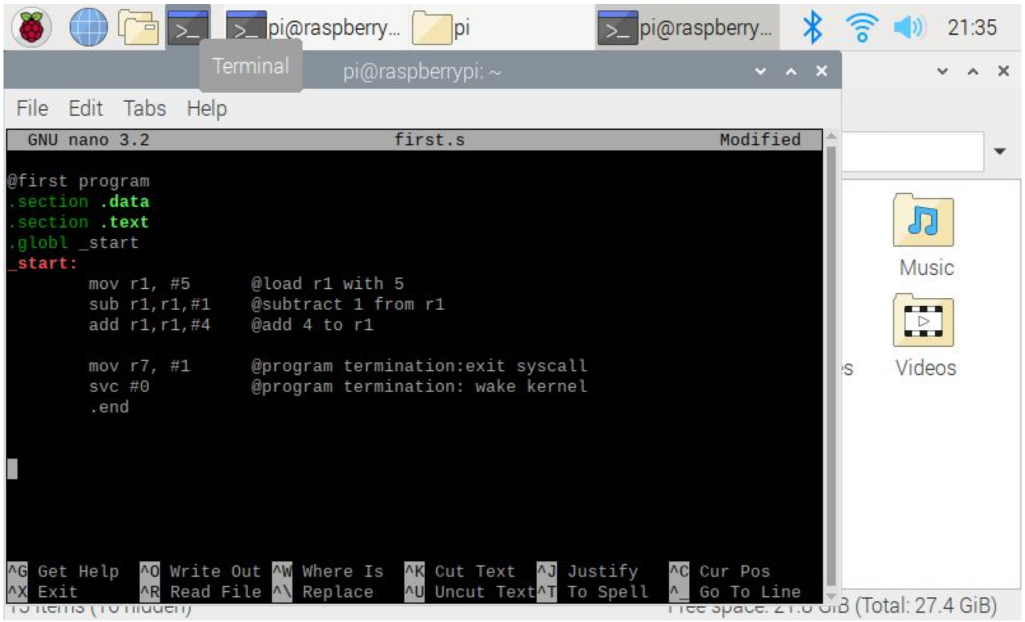
I then ran through the exact same process of assembling it and creating an object file, linking it to get an executable, and using the GDB to be able to look at the registers. The value of 7 in r1 was exactly what we were looking for, so the program was a success.



```
Breakpoint 1, _start () at arithmetic1.s:14
14              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             0x7efff3c0          0x7efff3c0
lr             0x0                 0
pc             0x10070             0x10070 <_start+28>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) 
```

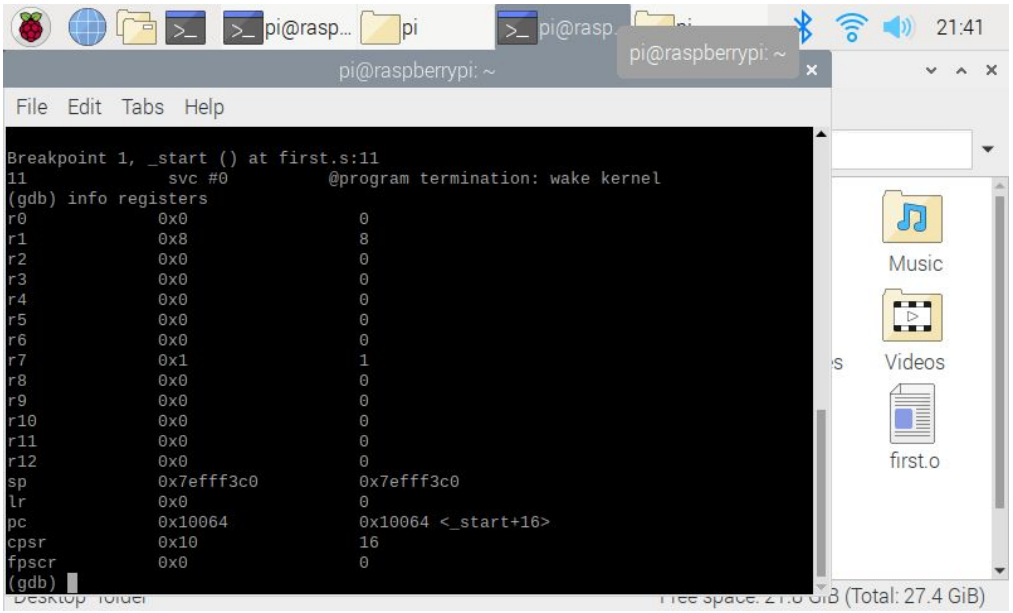
4. Zejneb

I started the assignment by writing the program given below. There was no output because the values are going into the registers. The way we check registers through the terminal is by typing (gdb) info registers.



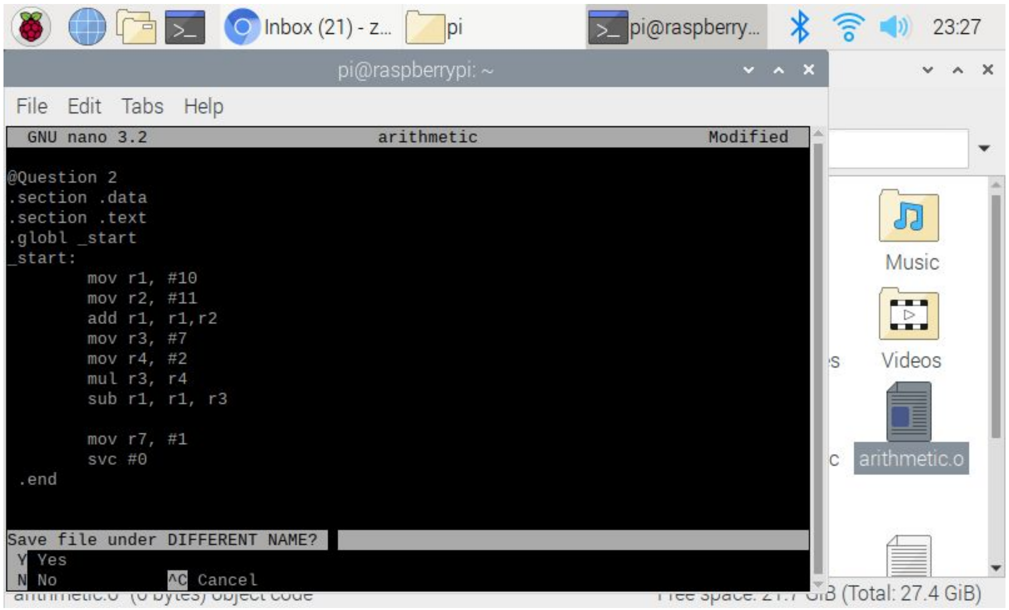
```
 GNU nano 3.2                        first.s                         Modified

@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



^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 the next step, I assembled the code with the assemble command, used the linker command and launched the GNU debugger. This was where I could set a breakpoint and view the registers. Register r1 contained the value of 8.



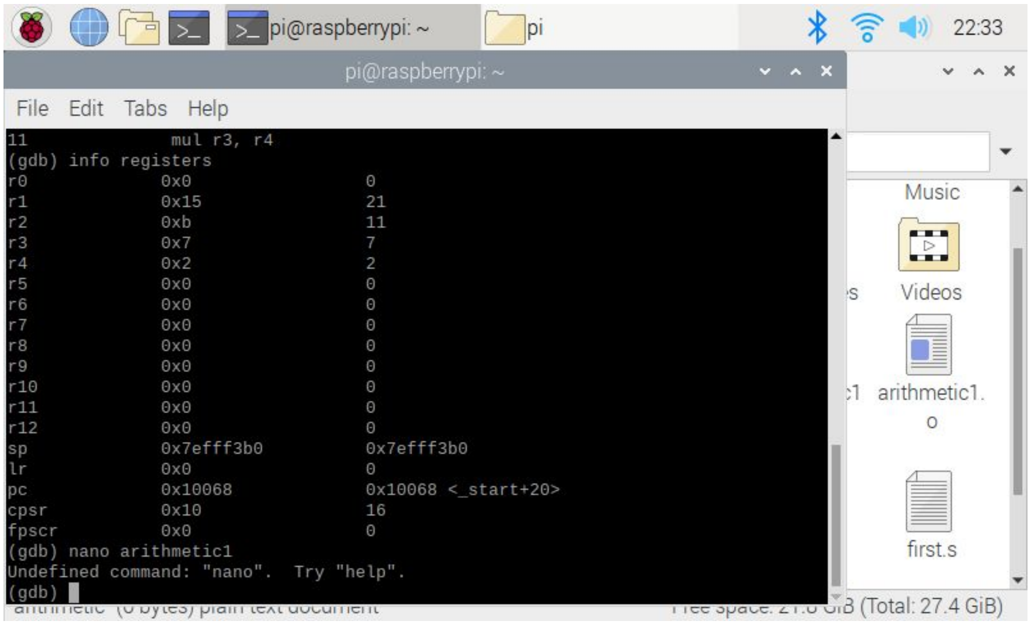
```
Breakpoint 1, _start () at first.s:11
11              svc #0          @program termination: wake kernel
(gdb) info registers
r0             0x0                 0
r1             0x8                 8
r2             0x0                 0
r3             0x0                 0
r4             0x0     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

For the next part of the assignment, I wrote an assembly code that would place values in registers without using a memory variable. I did so by using the mov instruction to place numerical values in separate registers. The value of 10 is moved to r1, the value of 11 moved to r2. I then added r2 and r1 and placed the value in r1. I moved the value of 7 into r3 and 2 into r4. I multiplied r4 and r3. Then subtracted r3 from r1 and placed the value of 7 into r3



```
  GNU nano 3.2                    arithmetic                     Modified

@Question 2
.section .data
.section .text
.globl _start
_start:
        mov r1, #10
        mov r2, #11
        add r1, r1,r2
        mov r3, #7
        mov r4, #2
        mul r3, r4
        sub r1, r1, r3

        mov r7, #1
        svc #0
 .end

Save file under DIFFERENT NAME?
 Y Yes
 N No              ^C Cancel
```

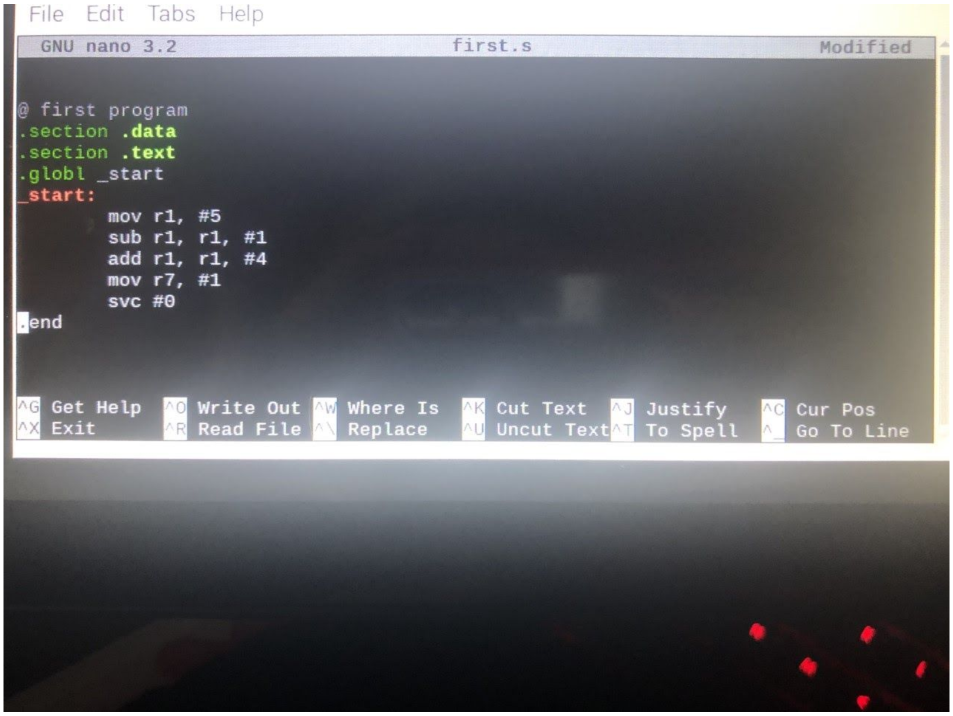
After assembling, linking and performing the debugging process, the value of the registers is shown below. It states that 21 is in r1, 11 is in r2, 7 is in r3 and finally 2 is in r4.



```
11              mul r3, r4
(gdb) info registers
r0             0x0                 0
r1             0x15                21
r2             0xb                 11
r3             0x7                 7
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             0x10068             0x10068 <_start+20>
cpsr           0x10                16
fpscr          0x0                 0
(gdb) nano arithmetic1
Undefined command: "nano".  Try "help".
(gdb)
```

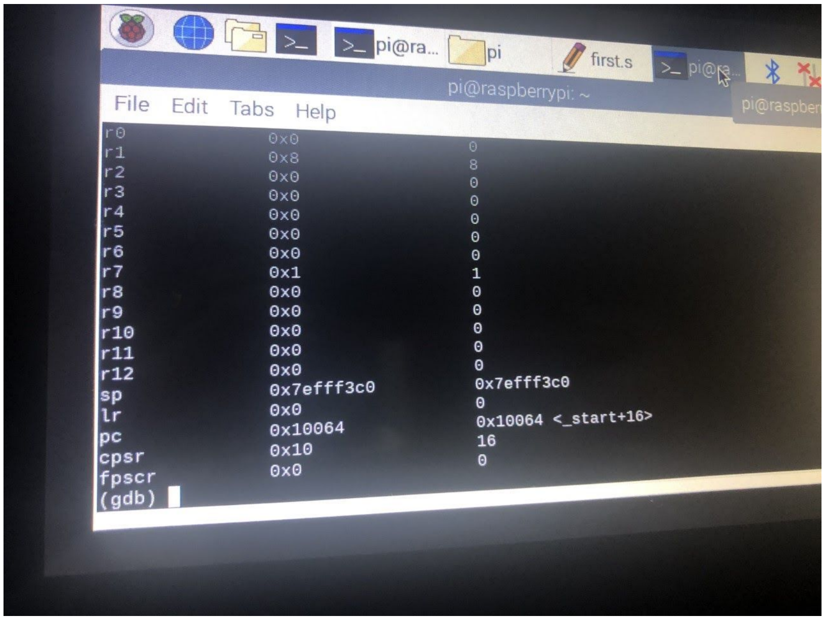
5. **Ashlynne:**

I started with this code using nano, which was just copying from the assignment. I showed no output because we used registries so you needed to assemble and link the program. Then find out more about the registers to view an output.



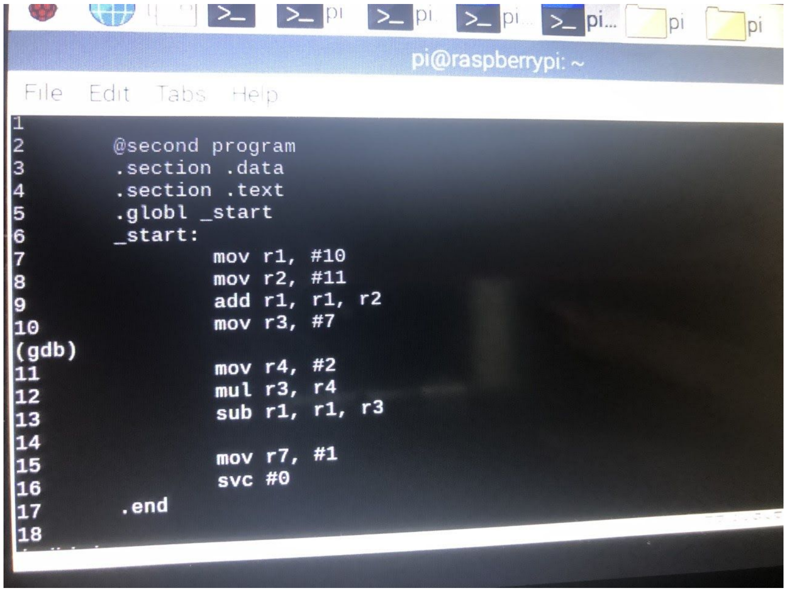
```
@ first program
.section .data
.section .text
.globl _start
_start:
        mov r1, #5
        sub r1, r1, #1
        add r1, r1, #4
        mov r7, #1
        svc #0
.end
```

After using (gdb) info registers and setting a breakpoint you are able to view them, we got 8 in r1 as expected.



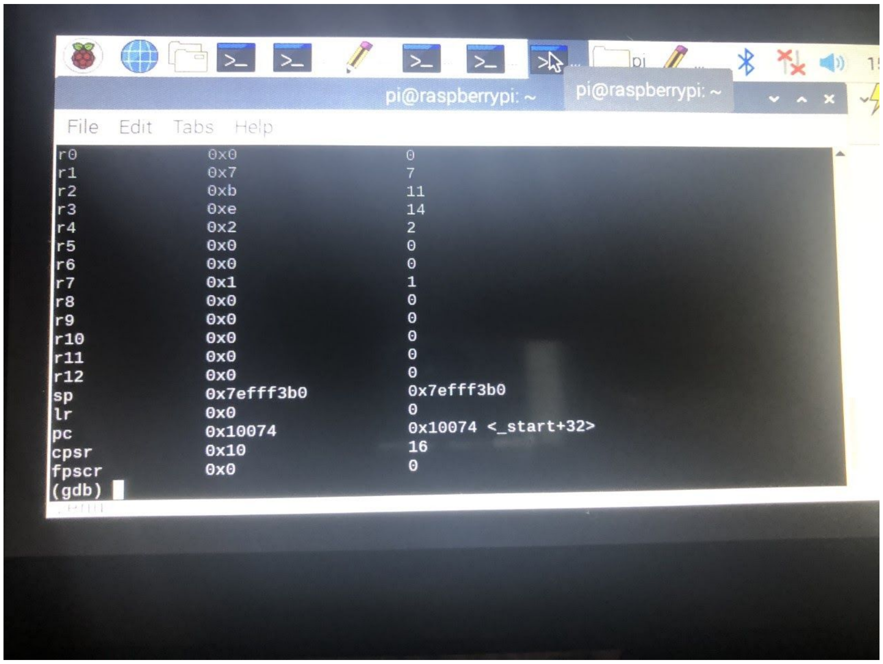
```
r0             0x0                 0
r1             0x8                 8
r2             0x0                 0
r3             0x0                 0
r4             0x0                 0
r5             0x0                 0
r6             0x0                 0
r7             0x1                 1
r8             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)
```

This part of the assignment was a challenge for me because I have to learn a new way of thinking. Instead of thinking in memory, we have to use registers. I kept getting an error for my multiplication line because i was writing it as mul r3,r3,r4 but i discovered the first r3 was not needed to save the output in the r3 register. Following the first question as an example, I loaded the correct variables for A, B, C, D into registers named r1, r2, r3 and r4.



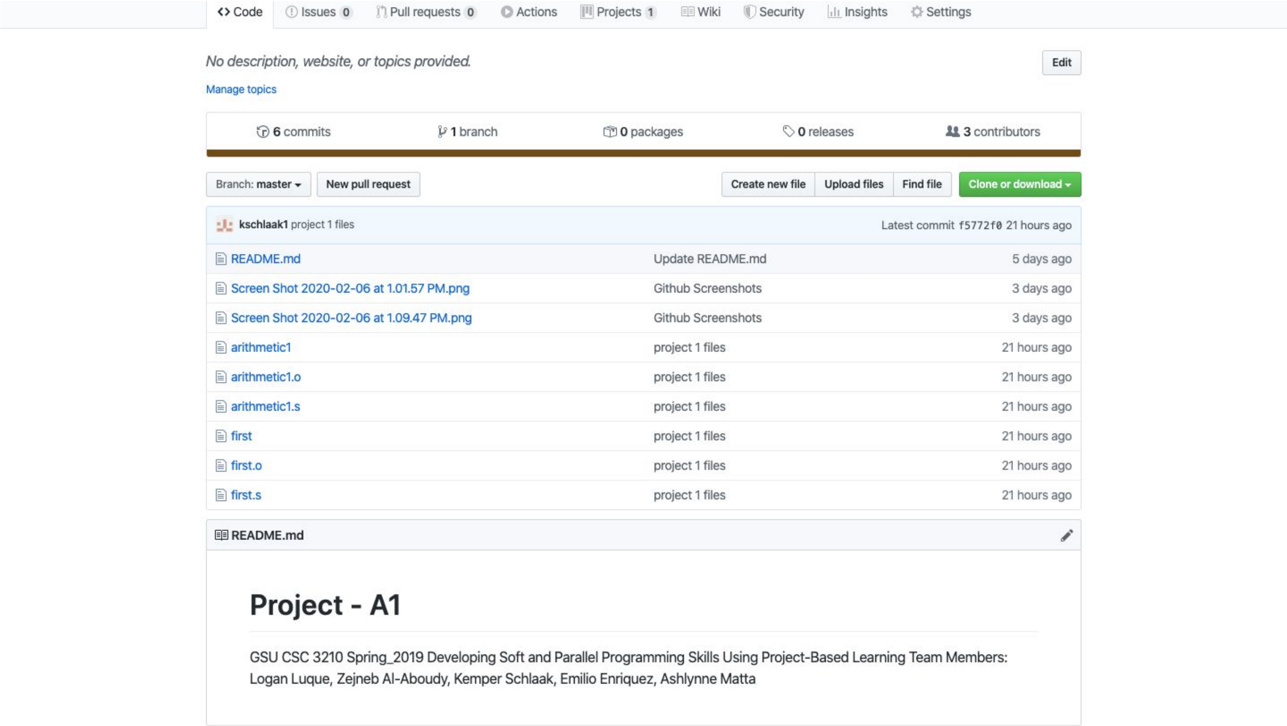
```
        @second program
        .section .data
        .section .text
        .globl _start
        _start:
                mov r1, #10
                mov r2, #11
                add r1, r1, r2
                mov r3, #7
(gdb)
                mov r4, #2
                mul r3, r4
                sub r1, r1, r3

                mov r7, #1
                svc #0

        .end
```

After following the same steps as before and creating a .o file and then linking, assembling and debugging, I got the register outputs and we have 7 in r1 which is what we were looking for



```
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)
```

Appendix



No description, website, or topics provided.

6 commits | 1 branch | 0 packages | 0 releases | 3 contributors

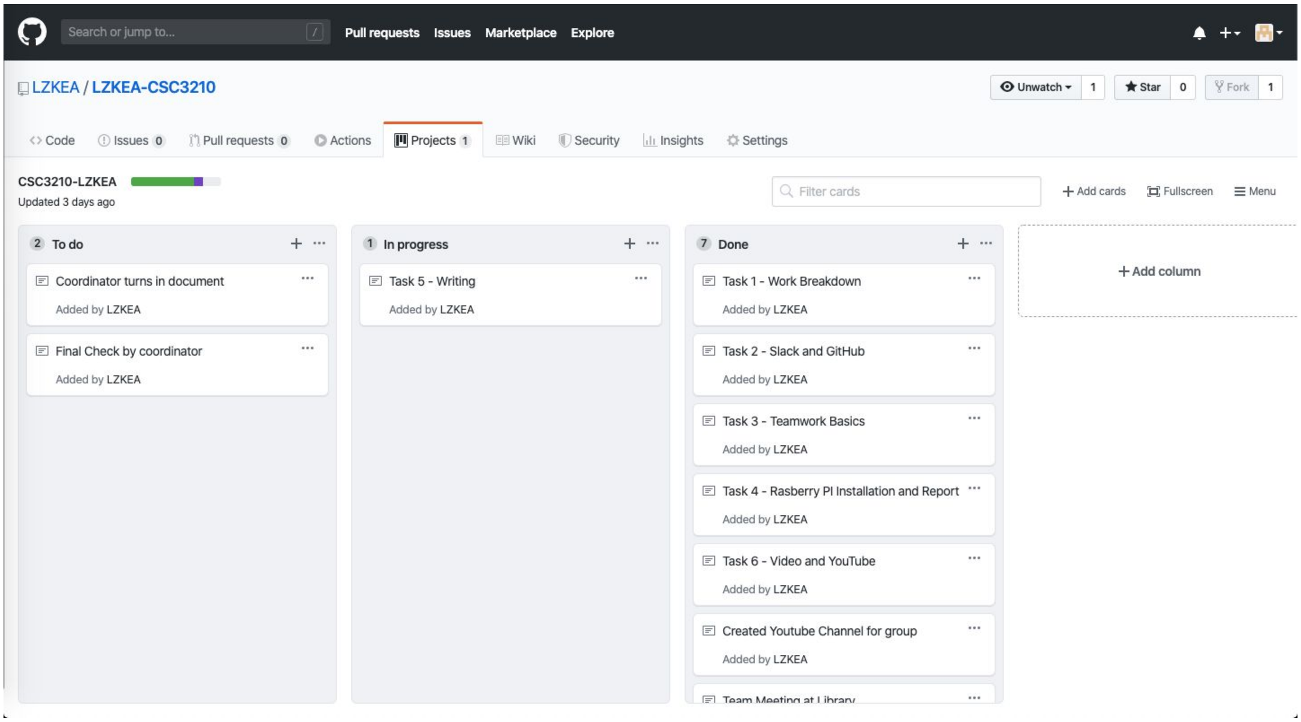
kschlaak1 project 1 files — Latest commit f5772f0 21 hours ago

README.md	Update README.md	5 days ago
Screen Shot 2020-02-06 at 1.01.57 PM.png	Github Screenshots	3 days ago
Screen Shot 2020-02-06 at 1.09.47 PM.png	Github Screenshots	3 days ago
arithmetic1	project 1 files	21 hours ago
arithmetic1.o	project 1 files	21 hours ago
arithmetic1.s	project 1 files	21 hours ago
first	project 1 files	21 hours ago
first.o	project 1 files	21 hours ago
first.s	project 1 files	21 hours ago

README.md

Project - A1

GSU CSC 3210 Spring_2019 Developing Soft and Parallel Programming Skills Using Project-Based Learning Team Members: Logan Luque, Zejneb Al-Aboudy, Kemper Schlaak, Emilio Enriquez, Ashlynne Matta



LZKEA / LZKEA-CSC3210

CSC3210-LZKEA

Updated 3 days ago

2 To do

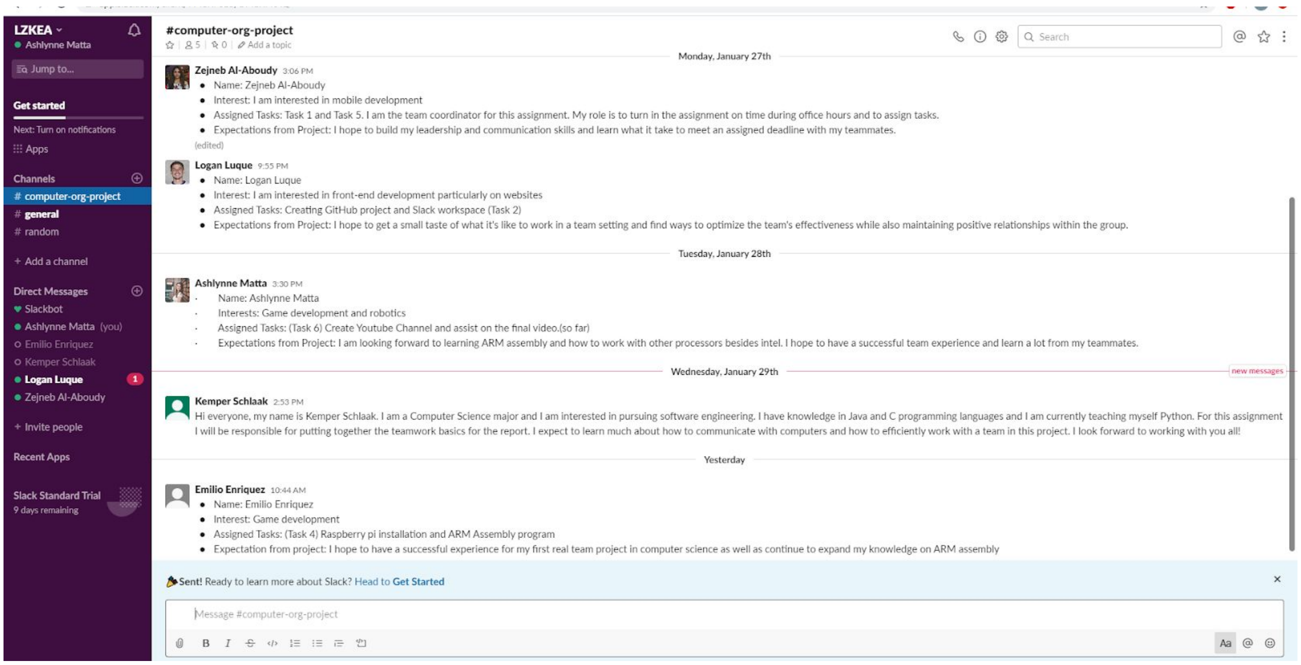
- Coordinator turns in document — Added by LZKEA
- Final Check by coordinator — Added by LZKEA

1 In progress

- Task 5 - Writing — Added by LZKEA

7 Done

- Task 1 - Work Breakdown — Added by LZKEA
- Task 2 - Slack and GitHub — Added by LZKEA
- Task 3 - Teamwork Basics — Added by LZKEA
- Task 4 - Rasberry PI Installation and Report — Added by LZKEA
- Task 6 - Video and YouTube — Added by LZKEA
- Created Youtube Channel for group — Added by LZKEA
- Team Meeting at Library



#computer-org-project

Monday, January 27th

Zejneb Al-Aboudy 3:06 PM

- Name: Zejneb Al-Aboudy
- Interest: I am interested in mobile development
- Assigned Tasks: Task 1 and Task 5. I am the team coordinator for this assignment. My role is to turn in the assignment on time during office hours and to assign tasks.
- Expectations from Project: I hope to build my leadership and communication skills and learn what it take to meet an assigned deadline with my teammates.

(edited)

Logan Luque 9:55 PM

- Name: Logan Luque
- Interest: I am interested in front-end development particularly on websites
- Assigned Tasks: Creating GitHub project and Slack workspace (Task 2)
- Expectations from Project: I hope to get a small taste of what it's like to work in a team setting and find ways to optimize the team's effectiveness while also maintaining positive relationships within the group.

Tuesday, January 28th

Ashlynne Matta 3:30 PM

· Name: Ashlynne Matta
· Interests: Game development and robotics
· Assigned Tasks: (Task 6) Create Youtube Channel and assist on the final video.(so far)
· Expectations from Project: I am looking forward to learning ARM assembly and how to work with other processors besides intel. I hope to have a successful team experience and learn a lot from my teammates.

Wednesday, January 29th

Kemper Schlaak 2:53 PM

Hi everyone, my name is Kemper Schlaak. I am a Computer Science major and I am interested in pursuing software engineering. I have knowledge in Java and C programming languages and I am currently teaching myself Python. For this assignment I will be responsible for putting together the teamwork basics for the report. I expect to learn much about how to communicate with computers and how to efficiently work with a team in this project. I look forward to working with you all!

Yesterday

Emilio Enriquez 10:44 AM

- Name: Emilio Enriquez
- Interest: Game development
- Assigned Tasks: (Task 4) Raspberry pi installation and ARM Assembly program
- Expectation from project: I hope to have a successful experience for my first real team project in computer science as well as continue to expand my knowledge on ARM assembly

Sent! Ready to learn more about Slack? Head to Get Started

GitHub Link: https://github.com/LZKEA/LZKEA-CSC3210 Workspace slack URL: LZKEA

Slack Link: https://app.slack.com/client/TT4BXP8LU/CT4BXPKAQ

YouTube Link: https://youtu.be/e9QfPMOU2ms