

Developing Soft and Programming Skills Using Project – Based Learning

Spring 2019

The Pi-Rates

Omolola Solaru

Crystle Yi

Zachary Combs

Benjamin Heppard

Michael Kellam

Report:

Our project proved to be an overall success. There were some obstacles along the way, but through coordination and effective planning, we efficiently solved our problems and completed the task.

Firstly, we had to create a general plan and schedule that we changed based on anything that came up, whether it be our availability, a problem with our hardware, or anything else that could change how often we would need to meet up or how much we had to do. During our first meeting, only Omolola, Crystle and I (Michael) were able to meet up. To make up for the fact that only three of us could attend that first meeting, the group decided that Zach and Nate would work on setting up our accounts, which consisted of a Gmail account, a GitHub account, and a YouTube channel. This was good cooperation from all group members because even though we didn't have everyone present at the time, we were able to find a way to distribute the workflow and keep the workload fair for all five members. Fortunately, everyone was able to attend the rest of the meetings afterwards. During the first meeting, we decided to plan the tasks for everyone to do. Zach and Nate created the media accounts; Crystle's experience with video editing prompted her to do said task. Omolola worked on getting the Pi to work (which was a lot harder than we thought it'd be due to many complications), and I oversaw the write-up. I was in charge with doing the write-up. Tuesdays and Fridays were our main days of meeting up for this project, and everybody was on time and attended the crucial meetings.

Our team was cohesive and cooperative from the very beginning. We all knew we had a task, and got it done as we saw fit. Nobody wanted to settle for a grade lower than an A, so we didn't have to worry about discourse in the group or anything along those lines. While we did have our tasks assigned to us, it didn't mean that we were restricted to only doing that task and doing it independently. This is demonstrated from the amount of troubleshooting we had to do to fix our Raspberry Pi not booting properly, which will be covered later. Every step and task were revised by each of our group members, which consisted of ensuring our media accounts were created properly and appropriately, reviewing each draft of the full video, ensuring our written report was articulate, on-topic, and reflective. And ensuring the actual task itself was completed correctly and effectively. Nobody was afraid to speak up if they believed there was something wrong with our result or our method of completing a certain task. Additionally, we all tried to stay on the same pace. This meant we had to always communicate where we were on our tasks, sending our documents and files to GitHub appropriately which we essentially had to keep version control in check. We agreed that even though we had all worked in a group environment before, this experience felt different in that every person's role felt pivotal, and it didn't seem like anybody was doing most of the work. The fact that this group assignment *required* us organize our documents, thoughts, and files for everybody to access. We believe that this is a fantastic way to introduce the class to proper group work without overwhelming. Fortunately, it wasn't at all difficult to get in the mindset that we were all equal in this task and were held accountable for our work.

The installation of our Raspberry Pi was by far our most difficult obstacle during this project. For the first few meetings, we had a lot of trouble figuring out why the OS on the SD card

would not boot properly inside of the Pi. We tried reformatting the SD card, reinstalling RASPBIAN, consulting forum posts of our errors (namely the “kernel panic” error), and nothing seemed to work. We tried for two hours straight and could not find out what was causing this issue. Omolola decided to contact Mussa, and he was able to help fix the problem. There was an issue with the way the SD card was formatted which prevented the OS from properly loading onto the Pi. After this problem was fixed though, RASPBIAN was running flawlessly on our Pi; our peripherals (mouse, keyboard, and monitor) were all able to seamlessly connect and communicate with our Pi. The programming itself was primitive and simple, so we were able to finish that in a decent amount of time without much problem and got the right results. Overall, after the installation, the execution and coding were very easy tasks.

In conclusion, our group is very satisfied with the final project. This project was the perfect introduction to team-based planning and execution. We are now prepared to take on a larger task now that we’ve got a feel for how to work as a team, our strengths, our weaknesses, and how to work effectively without wasting time. We now know our schedules, so planning will not be an issue either. We all look forward to working with one-another throughout this semester.

Planning and Scheduling:

Assigned Name	Email	Task	Duration(hours)	Dependency	Due Date	Note
Omolola Solaru(Group Coordinator)	osolaru1@student.gsu.edu	Raspberry PI Installation, Written Report, Setting Up Slack Account, TEAMWORK BASICS document	1 hour	Showing the members how to use a Raspberry Pi and how to Install it, Along with the ARM Assembly Programming	2/7/19	
Crystle Yi	cyi8@student.gsu.edu	Raspberry PI Installation, Will Work on Presentation Video, TEAMWORK BASICS document	1 hour	Will help produce and edit the video in order to make sure the video can be uploaded onto YouTube in a timely matter	2/7/19	
Zachary Combs	zcombs1@student.gsu.edu	Setting up Github Account, TEAMWORK BASICS document		Will Set the GitHub Account, Must learn or have an Idea of how to set up Raspberry Pi in case they have coordinator role next	2/7/19	
Benjamin Heppard	bheppard1@student.gsu.edu	Setting up Github Account, TEAMWORK BASICS document		Will Set the GitHub Account, Must learn or have an Idea of how to set up Raspberry Pi in case they have coordinator role next	2/5/19	
Michael Kellam	mkellam1@student.gsu.edu	Raspberry PI Installation, Written Report, TEAMWORK BASICS document	1 hour	Will Work on the Report with Group Coordinator in order to have the information accurately written down in a proper format	2/5/19	

Learning Teamwork Basics:

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

- Get to know other members of your group and their strengths
- Set ground rules
- Use a facilitator
- Keep lines of communication open
- Know how to avoid (or solve) common problem

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: Attempt to engage them into the group discussion
- Argues: Talk about it and try to work it as a team.

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

Set up a Pro and Con lists, do a majority votes, or try to be open minded and combine everyone's ideas into a more effective approach to the problem.

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?

Try to ask these three questions:

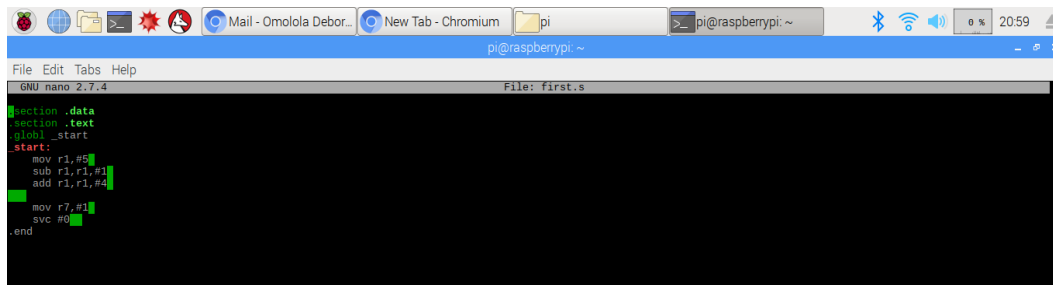
- "Are we already to make a decision on this?"
- "What needs to be done on this before we can move ahead?"
- "Let's check and see where everyone stands on this?"

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 they want do "B" work they can do what feel is acceptable but when we need to detail what was done on the planning and schedule form, they will have a smaller task compared to the others. And will receive an acceptable grade due to their smaller task compared the others.

Raspberry PI Installation and ARM Assembly Programming:

First.s Code:

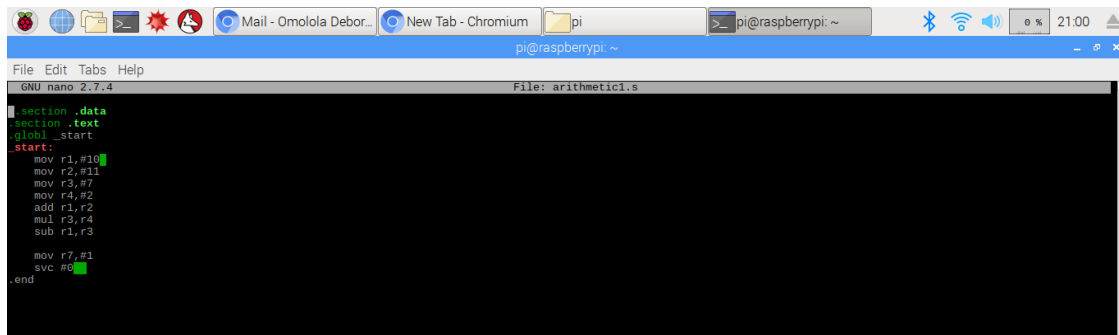


```
GNU nano 2.7.4 File: first.s
section .data
section .text
global _start
_start:
    mov r1,#5
    sub r1,r1,#1
    add r1,r1,#4
    mov r7,#1
    svc #0
.end
```

First.s Code Description:

This is the first program of code that was created in the terminal windows of the Raspberry pi. This code only used one general purpose register for the arithmetic operations of immediate values. *First.s* had a total of 3 immediate values for executing and storing.

Arithmetic1.s Code:



```
GNU nano 2.7.4 File: arithmetic1.s
section .data
section .text
global _start
_start:
    mov r1,#10
    mov r2,#11
    mov r3,#7
    mov r4,#2
    add r1,r2
    mul r3,r4
    sub r1,r3
    mov r7,#1
    svc #0
.end
```

Arithmetic1.s Code Description:

The code used following expression in the programming:

$$A = (A + B) - (C * D),$$

where A=10, B=11, C=7, and D=2

This is the second program of code that was created in the terminal windows of the Raspberry pi. This code used 4 general purpose register for the arithmetic operations of its immediate values. It had utilized four immediate values in its mathematical operations.

```

pi@raspberrypi: ~
File Edit Tabs Help
This GDB was configured as "arm-linux-gnueabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from first...done.
(gdb) list
Function "list" not defined.
(gdb) list
warning: Source file is more recent than executable.
1      .section .data
2      .section .text
3      .globl _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)

pi@raspberrypi: ~
File Edit Tabs Help
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-gnueabi".
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          mov r1,#10
6          mov r2,#11
7          mov r3,#7
8          mov r4,#2
9          add r1,r2
10         mul r3,r4
(gdb)

```

For each code we had to insert a breakpoint at the wake kernel to see the results for the program. This could only be done in the *gdb* operating system. But for this to happen we needed to see the full code before we can start debugging and running the file.

```

pi@raspberrypi: ~
File Edit Tabs Help
(gdb) run
Starting program: /home/pi/first
Breakpoint 1, _start () at first.s:10
10      svc #0
(gdb) info registers
r0          0x0      0
r1          0x8      8
r2          0x0      0
r3          0x0      0
r4          0x0      0
r5          0x0      0
r6          0x0      0
r7          0x1      1
r8          0x0      0
r9          0x0      0
r10         0x0      0
r11         0x0      0
r12         0x0      0
sp          0x7efff0c0 0x7efff0c0
lr          0x0      0
pc          0x10064 0x10064 <_start+16>
cpsr       0x10     16
(gdb)

pi@raspberrypi: ~
File Edit Tabs Help
(gdb) run
Starting program: /home/pi/arithmetic1
Breakpoint 1, _start () at arithmetic1.s:13
13      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          0x7efff0c0 0x7efff0c0
lr          0x0      0
pc          0x10070 0x10070 <_start+28>
cpsr       0x10     16
(gdb)

```

Info registers:

The codes info registers are displaying their general-purpose registers, along with the contents that have both hexadecimal and decimal. The *sp* register seems to store the address from the last program call (which is usually the *svc #0*) in a formatted stack.

Appendix:

Slack: <https://csc3210groupies.slack.com/messages/CFQKPD55E/team/UFSDSH3PZ/>

GitHub: <https://github.com/ThePi-rates/CSC3210---The-Pi-rates>

ThePi-rates Add files via upload		Latest commit eab7dbf 2 hours ago
Planning+Schedule.docx	Add files via upload	5 hours ago
Project_A1_Report.Updated.docx	Add files via upload	2 hours ago
Project_A1_Report.docx	Add files via upload	5 hours ago
README.md	Update README.md	3 days ago
TW Form.docx	Add files via upload	4 hours ago

README.md

CSC3210---The-Pi-rates

Team Coordinator: Omolola Solaru

Team Members: Crystle Yi, Zachary Combs, Benjamin Heppard, Michael Kellam

Search or jump to... Pull requests Issues Marketplace Explore

ThePi-rates / CSC3210---The-Pi-rates

<> Code 0 Issues 0 Pull requests 0 Projects 1 Wiki Insights Settings

CSC3210 - The Pi-rates Updated 2 hours ago

0 To Do + ...


0 In Progress + ...

7 Done + ...


- GitHub Account Setup
Added by ThePi-rates
- Presentation Video + Upload
Added by ThePi-rates
- Slack Account Setup
Added by ThePi-rates
- YouTube Account Setup
Added by ThePi-rates
- Raspberry Pi Installation
Added by ThePi-rates
- TEAMWORK BASICS Document
Added by ThePi-rates
- Written Report
Added by ThePi-rates

7

YouTube Video: <https://www.youtube.com/watch?v=3aIasX9be8A&feature=youtu.be>



Search



0:14 / 5:48

KE A PIC. IAG US.

pirate 1

No views

0 0 SHARE SAVE ...