

Developing Soft and Parallel Programming Skills Using Project-Based Learning

Justin Choi, Adam Henrie, Jean-Pierre Sacha, Titilayo Shonuyi, Shanza Siddiqi

Spring 2019
Group Name: Qubits

Planning and Scheduling

Team name: qubits

Name	Email	Task	Duration	Dependency	Due Date	Note
Justin Choi	jchoi34@student.gsu.edu	Report	3 hours	Task 3 and Task 4	3/7/19	Finish report a day before the due date
Adam Henrie	ahenrie1@student.gsu.edu	Video, support with parallel questions and arm programming	4 hours	Video	3/6/19	Please assist in answering questions and with arm programming as needed
JP Sacha	jsacha2@student.gsu.edu	Parallel programming and parallel questions	4 hours	None	3/4/19	Finish by Monday to run through the code with group
Titilayo Shonuyi	tshonuyi1@student.gsu.edu	Arm assembly programming	4 hours	None	3/4/19	Finish by Monday to run through code with group
Shanza Siddiqi	Ssiddiqi2@student.gsu.edu	Create To do/In Progress/Done columns and cards	3 hours	None	3/4/19	Create new cards on github and provide screenshots of the tasks

Parallel Programming Skills

Foundation

- 1. Define the following: Task, Pipelining, Shared Memory, Communications, Synchronization. (in your own words)**
 - o A task is defined as a completely separate set of instructions that can be executed by a processor.
 - o Pipelining is defined as breaking up a program into specific sets of task that can be computed by specific units within the processor.
 - o Shared Memory is a computer architecture that allows for any processor in a system to access the same main memory over a bus.
 - o Communications in concurrent and parallel processes and threads is used to be able to express how each process is to collaborate with each other. Also, it allows for light weight parallel threads to not be executing their code in a critical section at the same time as another thread. This communication can happen over shared memory or via dedicated sockets via a network in distributed computing applications.
 - o Synchronization is used when a task waits for another task or tasks to reach the same established synchronization point in an application in order to proceed. The wall clock execution time of a parallel application can be increased due to tasks having to wait.

- 2. Classify parallel computers based on Flynn's taxonomy. Briefly describe every one of them.**

SISD - Single instruction stream single data stream. Totally sequential.

SIMD - Single instruction stream, multiple data streams. Single instruction operating on multiple data streams in parallel.

MISD - Multiple instruction streams, single data stream. Multiple different instructions simultaneously operating on one data stream.

MIMD - Multiple instruction streams, multiple data streams. Simultaneously executing multiple different instructions on multiple different data streams.

- 3. What are the Parallel Programming Models?**

- Shared Memory (without threads)
- Threads
- Distributed Memory / Message Passing
- Data Parallel
- Hybrid
- Single Program Multiple Data (SPMD)
- Multiple Program Multiple Data (MPMD)

4. List and briefly describe the types of Parallel Computer Memory Architectures.**What type is used by OpenMP and why?**

Shared Memory: Multiple processors can operate independently but share the same memory resources

UMA - Equal access and access times to memory

NUMA - Not all processors have equal access time to all memories.

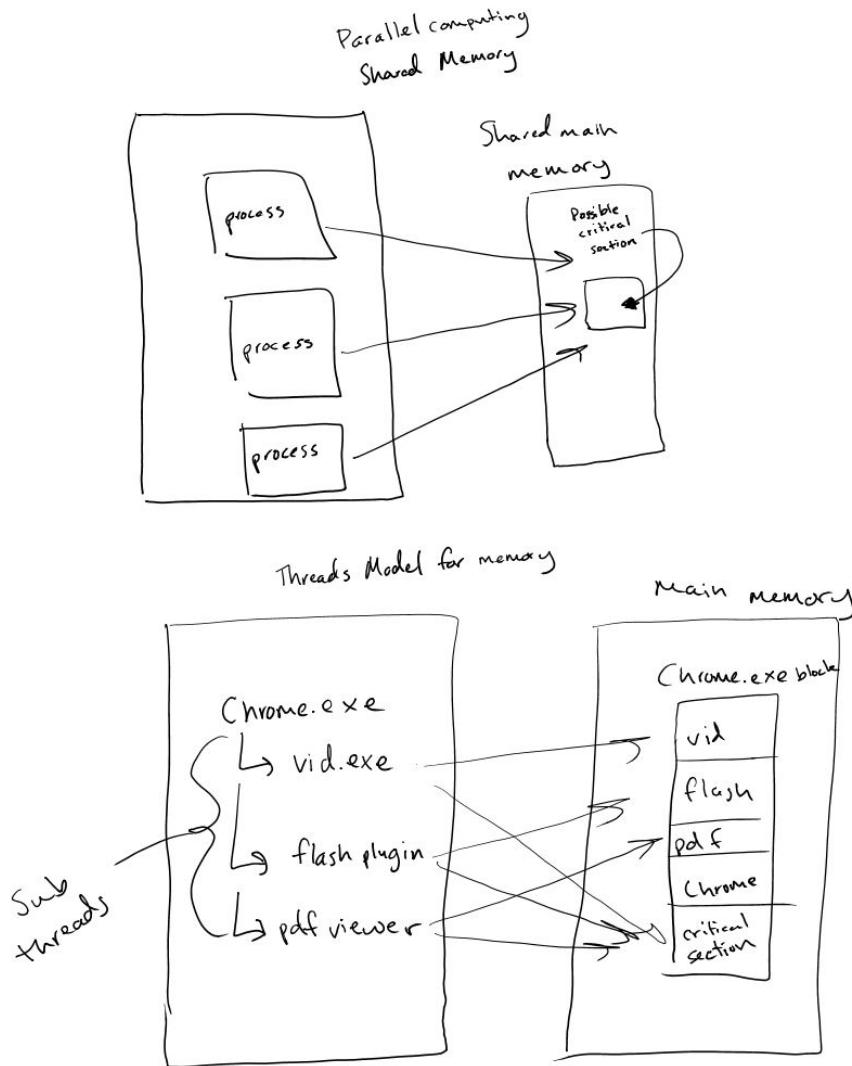
Memory access across link is slower

Distribute Memory: Processors have their own local memory. No global memory space.

OpenMP uses a shared memory architecture to exploit parallelism for multi-core and multi-thread processing.

5. Compare Shared Memory Model with Threads Model? (in your own words and show pictures)

The shared memory model is similar to the Von Neumann architecture in that parallelized processes share the same data from within main memory. These processes are independent of each other but share the same critical sections of data in main memory. The threads model is based on another layer of abstraction where there is shared memory between the sub-process/threads of a main heavy weight process which is responsible for forking and joining sub-parallel threads as it sees fit.



6. What is Parallel Programming? (in your own words)

Parallel programming is programming that allows for multiple simultaneous instructions on the same set of data to allow for faster overall computing.

7. What is system on chip (SoC)? Does Raspberry PI use system on SoC?

System on a Chip integrates the CPU, GPU, RAM, and other aspects onto a single PCB to allow for computing on a single compact package.

8. Explain what the advantages are of having a System on a Chip rather than separate CPU, GPU and RAM components.

SOC is cheaper, optimized, portable, lower power. Having separate components can cause problems with compatibility and performance. With a SOC you can guarantee every component won't bottleneck the other component and that they all work well together.

Parallel Programming Basics - Running Loops in Parallel

Part 2.x

- Code for parallelLoopEqualChunks.c

```

File Edit Tabs Help
pLoopEqual... runTab Screenshots
GNU nano 2.7.4 File: parallelLoopEqualChunks.c

#include <stdio.h> //printf()
#include <stdlib.h> //atoi()
#include <omp.h> //OpenMP

int main(int argc, char** argv) {
const int REPS = 16;

printf("\n");
if(argc > 1) {
omp_set_num_threads(atoi(argv[1]));
}

#pragma omp parallel for
for (int i = 0; i <REPS; i++) {
int id = omp_get_thread_num();
printf("Thread %d performed iteration %d\n", id, i);
}

printf("\n");
return 0;
}

pi@raspberrypi:~ $ gcc parallelLoopEqualChunks.c -o pLoop -fopenmp
pi@raspberrypi:~ $ ./pLoop 4

Thread 2 performed iteration 8
Thread 2 performed iteration 9
Thread 2 performed iteration 10
Thread 2 performed iteration 11
Thread 3 performed iteration 12
Thread 3 performed iteration 13
Thread 3 performed iteration 14
Thread 3 performed iteration 15
Thread 1 performed iteration 4
Thread 1 performed iteration 5
Thread 1 performed iteration 6
Thread 1 performed iteration 7
Thread 0 performed iteration 0
Thread 0 performed iteration 1
Thread 0 performed iteration 2
Thread 0 performed iteration 3

pi@raspberrypi:~ $ █
pi@raspberrypi:~ $ ./pLoop 3

Thread 1 performed iteration 6
Thread 1 performed iteration 7
Thread 1 performed iteration 8
Thread 1 performed iteration 9
Thread 1 performed iteration 10
Thread 2 performed iteration 11
Thread 2 performed iteration 12
Thread 2 performed iteration 13
Thread 2 performed iteration 14
Thread 2 performed iteration 15
Thread 0 performed iteration 0
Thread 0 performed iteration 1
Thread 0 performed iteration 2
Thread 0 performed iteration 3
Thread 0 performed iteration 4
Thread 0 performed iteration 5

pi@raspberrypi:~ $ █

```

- Initialized with 3 threads, the number of iterations (16) is not evenly divisible between the threads. Thread 1 and 2 perform 5 iterations while thread 0 performs 6.

Part 3.x

4. Code for
parallelLoop
ChunkOf1.c

-In this program
chunks are 1
instead of 4 .

5. When the first
for loop is
scheduled
statically, it
mimics the loop
without
scheduling;
thread 0
performs
iteration 0,
thread 1
performs
iteration 1,
thread 2
performs
iteration 2, and
so on. This is
because the
iteration order is
defined before
running.

```

File Edit Tabs Help
pLoopEqual... × runTab × Screenshots × pLoopChunk... ×
GNU nano 2.7.4 File: parallelLoopChunksOf1.c Modified
#include <stdio.h> //printf()
#include <stdlib.h> //atoi()
#include <omp.h> //OpenMP
int main(int argc, char** argv) {
    const int REPS = 16;
    printf("\n");
    if(argc > 1) {
        omp_set_num_threads(atoi(argv[1]));
    }
    #pragma omp parallel for schedule(static,1)
    for (int i = 0; i < REPS; i++) {
        int id = omp_get_thread_num();
        printf("Thread %d performed iteration %d\n", id, i);
    }
    printf("\n---\n");
    #pragma omp parallel
    {
        int id = omp_get_num_threads();
        for (int i = id; i < REPS; i+= numThreads) {
            printf("Thread %d performed iteration %d\n", id,i);
        }
        printf("\n");
    }
    return 0;
}

pi@raspberrypi:~ $ ./pLoop2 4
Thread 1 performed iteration 1
Thread 1 performed iteration 5
Thread 1 performed iteration 9
Thread 1 performed iteration 13
Thread 3 performed iteration 3
Thread 3 performed iteration 7
Thread 3 performed iteration 11
Thread 3 performed iteration 15
Thread 0 performed iteration 0
Thread 0 performed iteration 4
Thread 0 performed iteration 8
Thread 0 performed iteration 12
Thread 2 performed iteration 2
Thread 2 performed iteration 6
Thread 2 performed iteration 10
Thread 2 performed iteration 14
---
Thread 1 performed iteration 1
Thread 1 performed iteration 5
Thread 1 performed iteration 9
Thread 1 performed iteration 13
Thread 3 performed iteration 3
Thread 3 performed iteration 7
Thread 3 performed iteration 11
Thread 3 performed iteration 15
Thread 0 performed iteration 0
Thread 0 performed iteration 4
Thread 0 performed iteration 8
Thread 0 performed iteration 12
Thread 2 performed iteration 2
Thread 2 performed iteration 6
Thread 2 performed iteration 10
Thread 2 performed iteration 14

pi@raspberrypi:~ $ []

```

6. When the first for loop is scheduled dynamically, the outputs differ. You can see that thread 3 performs 13 of the 16 iterations, because the chunk decomposition is defined when the program runs and assigns each iteration as the threads are available.

```
pi@raspberrypi:~ $ gcc parallelLoopChunksOf1.c -o pLoop2 -fopenmp
pi@raspberrypi:~ $ ./pLoop2 4

Thread 3 performed iteration 3
Thread 3 performed iteration 4
Thread 3 performed iteration 5
Thread 3 performed iteration 6
Thread 3 performed iteration 7
Thread 3 performed iteration 8
Thread 3 performed iteration 9
Thread 3 performed iteration 10
Thread 3 performed iteration 11
Thread 3 performed iteration 12
Thread 3 performed iteration 13
Thread 3 performed iteration 14
Thread 3 performed iteration 15
Thread 2 performed iteration 0
Thread 1 performed iteration 2
Thread 0 performed iteration 1
---
Thread 0 performed iteration 0
Thread 0 performed iteration 4
Thread 0 performed iteration 8
Thread 0 performed iteration 12
Thread 2 performed iteration 2
Thread 2 performed iteration 6
Thread 2 performed iteration 10
Thread 2 performed iteration 14
Thread 3 performed iteration 3
Thread 3 performed iteration 7
Thread 3 performed iteration 11
Thread 3 performed iteration 15
Thread 1 performed iteration 1
Thread 1 performed iteration 5
Thread 1 performed iteration 9
Thread 1 performed iteration 13

pi@raspberrypi:~ $ []
```

Part 4.x

7. Code for reduction.c

-Returns the sum of integers in an array sequentially and in parallel.

8. The first run is sequential.
Changing the number of threads did not change the output.

9. The second and third calls are ran in parallel, but output the wrong sum.

The threads are trying to change the sum simultaneously, giving a false result.

10. After adding the reduction clause

the output is correct since the sum for each thread is joined at the end

```

GNU nano 2.7.4                               File: reduction.c

#include <stdio.h> //printf()
#include <stdlib.h> //rand()
#include <omp.h> //OpenMP

void initialize(int* a, int n);
int sequentialSum(int* a, int n);
int parallelSum(int*a, int n);

#define SIZE 1000000

int main(int argc, char** argv) {
int array[SIZE];

if(argc>1) {
    omp_set_num_threads(atoi(argv[1]));
}
initialize(array,SIZE);
printf("\nSequential sum: %d\nParallel sum: %d\n\n",
sequentialSum(array,SIZE),
parallelSum(array, SIZE));

return 0;
}

void initialize(int* a, int n) {
int i;
for(i=0;i<n;i++) {
    a[i] = rand() %1000;
}
}

int sequentialSum(int* a, int n) {
int sum = 0;
int i;
for(i=0;i<n;i++) {
    sum += a[i];
}
return sum;
}

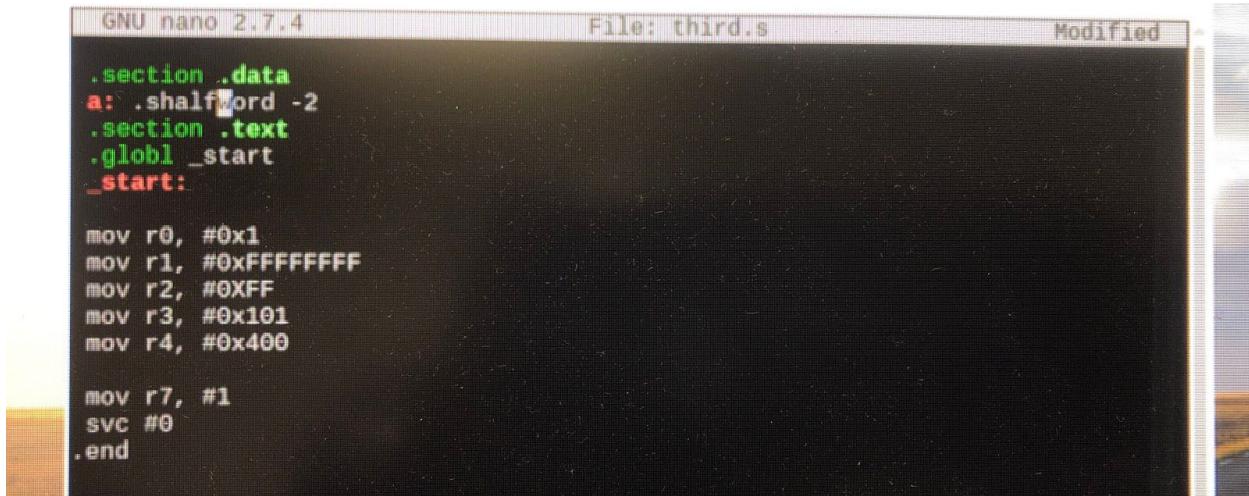
int parallelSum(int* a, int n) {
int sum = 0;
int i;
// #pragma omp parallel for // reduction(+:sum)
for(i=0;i<n;i++) {
    sum += a[i];
}
return sum;
}

pi@raspberrypi:~ $ ./reduction
Sequential sum: 499562283
Parallel sum: 499562283
pi@raspberrypi:~ $ ./reduction
Sequential sum: 499562283
Parallel sum: 158359623
pi@raspberrypi:~ $ ./reduction
Sequential sum: 499562283
Parallel sum: 158206868
pi@raspberrypi:~ $ ./reduction
Sequential sum: 499562283
Parallel sum: 499562283

```

ARM Assembly Programming

Writing the code for third.s on raspberry pi

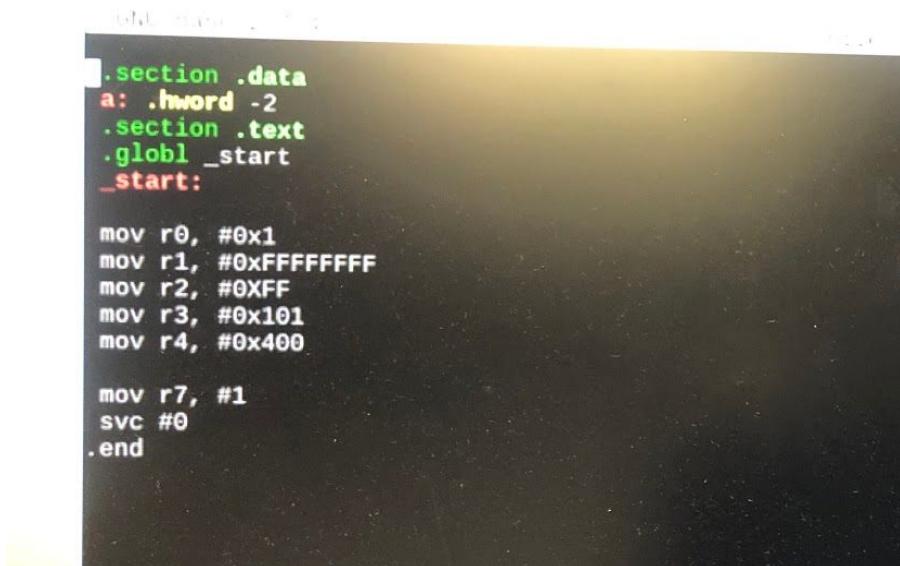


```
GNU nano 2.7.4                               File: third.s                                Modified
.section .data
a: .halfword -2
.section .text
.globl _start
_start:

    mov r0, #0x1
    mov r1, #0xFFFFFFFF
    mov r2, #0xFF
    mov r3, #0x101
    mov r4, #0x400

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

After writing the code, I was unable to assemble it. There is an error code pseudo-op ‘halfword’. halfword is a data type that should be denoted as h or sh.

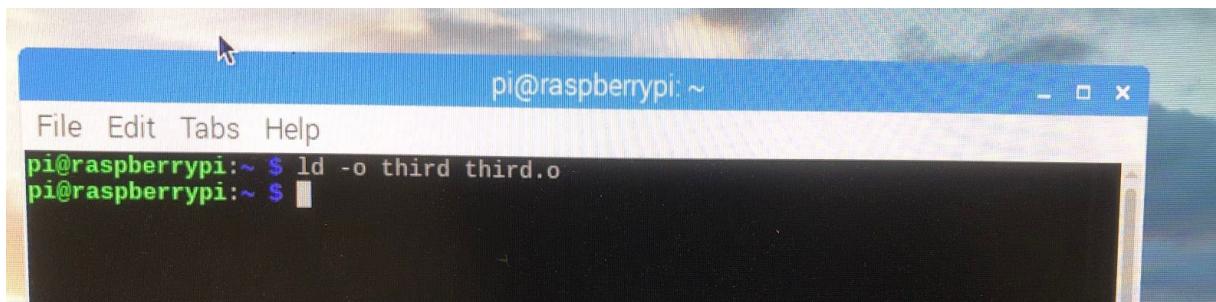


```
.section .data
a: .hword -2
.section .text
.globl _start
_start:

    mov r0, #0x1
    mov r1, #0xFFFFFFFF
    mov r2, #0xFF
    mov r3, #0x101
    mov r4, #0x400

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

Linking third.o file



```
pi@raspberrypi:~ $ ld -o third third.o
pi@raspberrypi:~ $
```

Third.s license preview.

```
pi@raspberrypi:~ $ gdb third
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 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 third...done.
(gdb) ■
```

Source code gdb list

```
pi@raspberrypi:~ $ ld -o third third.o
pi@raspberrypi:~ $ gdb third
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 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 third...done.
(gdb) list
1      .section .data
2      a: .hword -2
3      .section .text
4      .globl _start
5      _start:
6
7      mov r0, #0x1
8      mov r1, #0xFFFFFFFF
9      mov r2, #0xFF
10     mov r3, #0x101
(gdb) ■
```

Setting break point in line 7 with gdb b 7

```
pi@raspberrypi:~ $ ld -o third third.o
pi@raspberrypi:~ $ gdb third
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 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 third...done.
(gdb) list
1         .section .data
2         a: .hword -2
3         .section .text
4         .globl _start
5         _start:
6
7             mov r0, #0x1
8             mov r1, #0xFFFFFFFF
9             mov r2, #0xFF
10            mov r3, #0x101
(gdb) b 7
Breakpoint 1 at 0x10078: file third.s, line 7.
(gdb) ■
```

gdb run the program

```
pi@raspberrypi:~ $ ld -o third third.o
pi@raspberrypi:~ $ gdb third
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 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 third...done.
(gdb) list
1         .section .data
2         a: .hword -2
3         .section .text
4         .globl _start
5         _start:
6
7             mov r0, #0x1
8             mov r1, #0xFFFFFFFF
9             mov r2, #0xFF
10            mov r3, #0x101
(gdb) b 7
Breakpoint 1 at 0x10078: file third.s, line 7.
(gdb) run
Starting program: /home/pi/third

Breakpoint 1, _start () at third.s:8
8             mov r1, #0xFFFFFFFF
(gdb) ■
```

gdb stepi

```
Reading symbols from third...done.
(gdb) list
1      .section .data
2      a: .hword -2
3      .section .text
4      .globl _start
5      _start:
6
7      mov r0, #0x1
8      mov r1, #0xFFFFFFFF
9      mov r2, #0xFF
10     mov r3, #0x101
(gdb) b 7
Breakpoint 1 at 0x10078: file third.s, line 7.
(gdb) run
Starting program: /home/pi/third

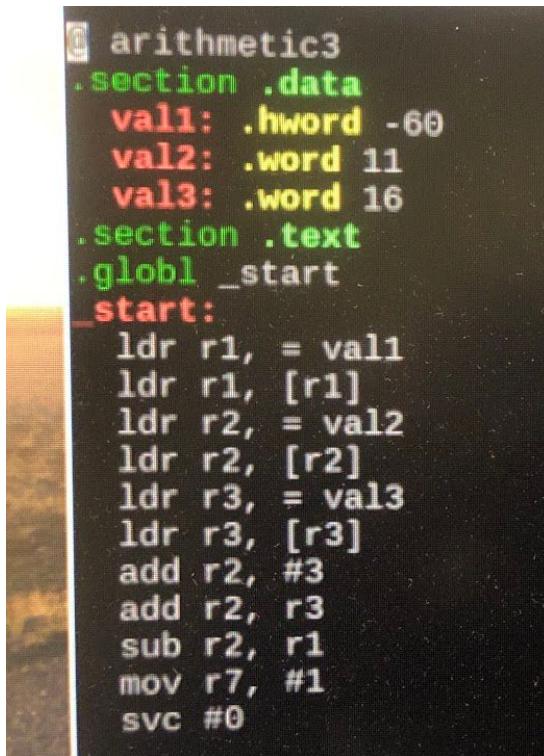
Breakpoint 1, _start () at third.s:8
8      mov r1, #0xFFFFFFFF
(gdb) stepi
9      mov r2, #0xFF
(gdb) stepi
10     mov r3, #0x101
(gdb) █
```

gdb x/1xh 0x10078

```
1      .section .data
2      a: .hword -2
3      .section .text
4      .globl _start
5      _start:
6
7      mov r0, #0x1
8      mov r1, #0xFFFFFFFF
9      mov r2, #0xFF
10     mov r3, #0x101
(gdb) b 7
Breakpoint 1 at 0x10078: file third.s, line 7.
(gdb) run
Starting program: /home/pi/third

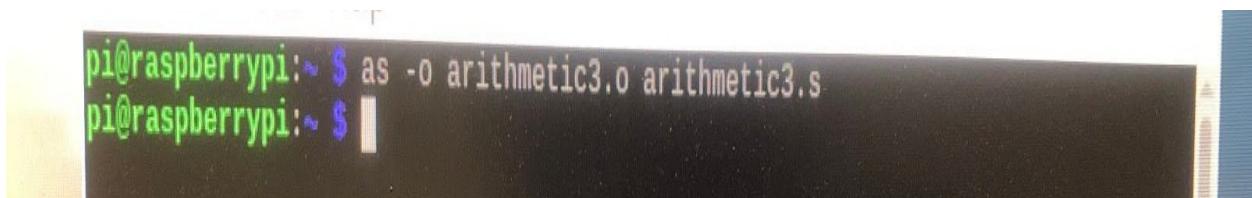
Breakpoint 1, _start () at third.s:8
8      mov r1, #0xFFFFFFFF
(gdb) stepi
9      mov r2, #0xFF
(gdb) stepi
10     mov r3, #0x101
(gdb) x/1xh 0x10078
0x10078 <_start+4>:    0x1000
(gdb) █
```

Writing the arithmetic.3 program



```
assembly3
.section .data
val1: .hword -60
val2: .word 11
val3: .word 16
.section .text
.globl _start
_start:
    ldr r1, = val1
    ldr r1, [r1]
    ldr r2, = val2
    ldr r2, [r2]
    ldr r3, = val3
    ldr r3, [r3]
    add r2, #3
    add r2, r3
    sub r2, r1
    mov r7, #1
    svc #0
```

At first when I try to run the program, there was an error message (pseudo-op ‘shalfword’). I corrected it by removing shalfword with hworld and the program work and I was able to assemble the program.



```
pi@raspberrypi:~ $ as -o arithmetic3.o arithmetic3.s
pi@raspberrypi:~ $
```

Linking the file

```
pi@raspberrypi:~ $ ld -o third third.o
pi@raspberrypi:~ $ ld -o arithmetic3 arithmetic3.o
ld: cannot find arithmetic3.o: No such file or directory
pi@raspberrypi:~ $ nano arithmetic3.s
pi@raspberrypi:~ $ nano third.s
pi@raspberrypi:~ $ nano arithmetic3.s
pi@raspberrypi:~ $ ld -o arithmetic3 arithmetic3.o
ld: cannot find arithmetic3.o: No such file or directory
pi@raspberrypi:~ $ nano arithmetic3.s
pi@raspberrypi:~ $ as -o arithmetic3.o arithmetic3.s
pi@raspberrypi:~ $ as -g -o arithmetic3.o arithmetic3.s
pi@raspberrypi:~ $ ld -o arithmetic3 arithmetic3.o
pi@raspberrypi:~ $ █
```

arithmetic3 GNU debugger and raspberry pi license display.

```
pi@raspberrypi:~ $ gdb arithmetic3
GNU gdb (Raspbian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 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 arithmetic3...done.
(gdb) █
```

gdb list for arithmetic3

```

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 arithmetic3...done.
(gdb) arithmetic3
Undefined command: "arithmetic3". Try "help".
(gdb) list
1      @ arithmetic3
2      .section .data
3          val1: .word -60
4          val2: .word 11
5          val3: .word 16
6      .section .text
7      .globl _start
8      _start:
9          ldr r1, = val1
10         ldr r1, [r1]
(gdb)

```

Setting break point in line 7

```

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 arithmetic3...done.
(gdb) arithmetic3
Undefined command: "arithmetic3". Try "help".
(gdb) list
1      @ arithmetic3
2      .section .data
3          val1: .word -60
4          val2: .word 11
5          val3: .word 16
6      .section .text
7      .globl _start
8      _start:
9          ldr r1, = val1
10         ldr r1, [r1]
(gdb) b 7
Breakpoint 1 at 0x10078: file arithmetic3.s, line 7.
(gdb)

```

arithmetic.3 gdb run and it show break start in line 10

```

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from arithmetic3...done.
(gdb) arithmetic3
Undefined command: "arithmetic3". Try "help".
(gdb) list
1      @ arithmetic3
2      .section .data
3          val1: .word -60
4          val2: .word 11
5          val3: .word 16
6      .section .text
7      .globl _start
8      _start:
9          ldr r1, = val1
10         ldr r1, [r1]
(gdb) b 7
Breakpoint 1 at 0x10078: file arithmetic3.s, line 7.
(gdb) run
Starting program: /home/pi/arithmetic3

Breakpoint 1, _start () at arithmetic3.s:10
10         ldr r1, [r1]
(gdb) █

```

Showing the stepi of arithmetic.3 program

```

(gdb) b 7
Breakpoint 1 at 0x10078: file arithmetic3.s, line 7.
(gdb) run
Starting program: /home/pi/arithmetic3

Breakpoint 1, _start () at arithmetic3.s:10
10         ldr r1, [r1]
(gdb) stepi
11         ldr r2, = val2
(gdb) stepi
12         ldr r2, [r2]
(gdb) stepi
13         ldr r3, = val3
(gdb) stepi
14         ldr r3, [r3]
(gdb) stepi
15         add r2, #3
(gdb) stepi
16         add r2, r3
(gdb) stepi
17         sub r2, r1
(gdb) stepi
18         svc #0
(gdb) █

```

Showing gdb x/1xh 0x10078

```
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1601 1602
```

Appendix:

GitHub

The screenshot shows a GitHub project named 'Project3'. The top navigation bar includes 'Code', 'Issues 0', 'Pull requests 0', 'Projects 1' (which is selected), 'Wiki', and 'Insights'. The project summary indicates it was updated 9 days ago.

To Do List: 0 items

In Progress: 2 items

- Justin: Help set up the report (Added by qubitsGSU)
- JP: Work on the Pi's programming and report (Added by qubitsGSU)

Done: 3 items

- Adam: Help with the programming and help make and upload youtube video (Added by qubitsGSU)
- Titi: Additional help in report and programming (Added by qubitsGSU)
- Shanza: Help upload docs and manage GitHub (Added by qubitsGSU)

Important Links:

- **Slack:** computerorganizespr19.slack.com
- **GitHub:** <https://github.com/qubitsGSU>
- **YouTube:** <https://www.youtube.com/watch?v=R15X40qjR40>