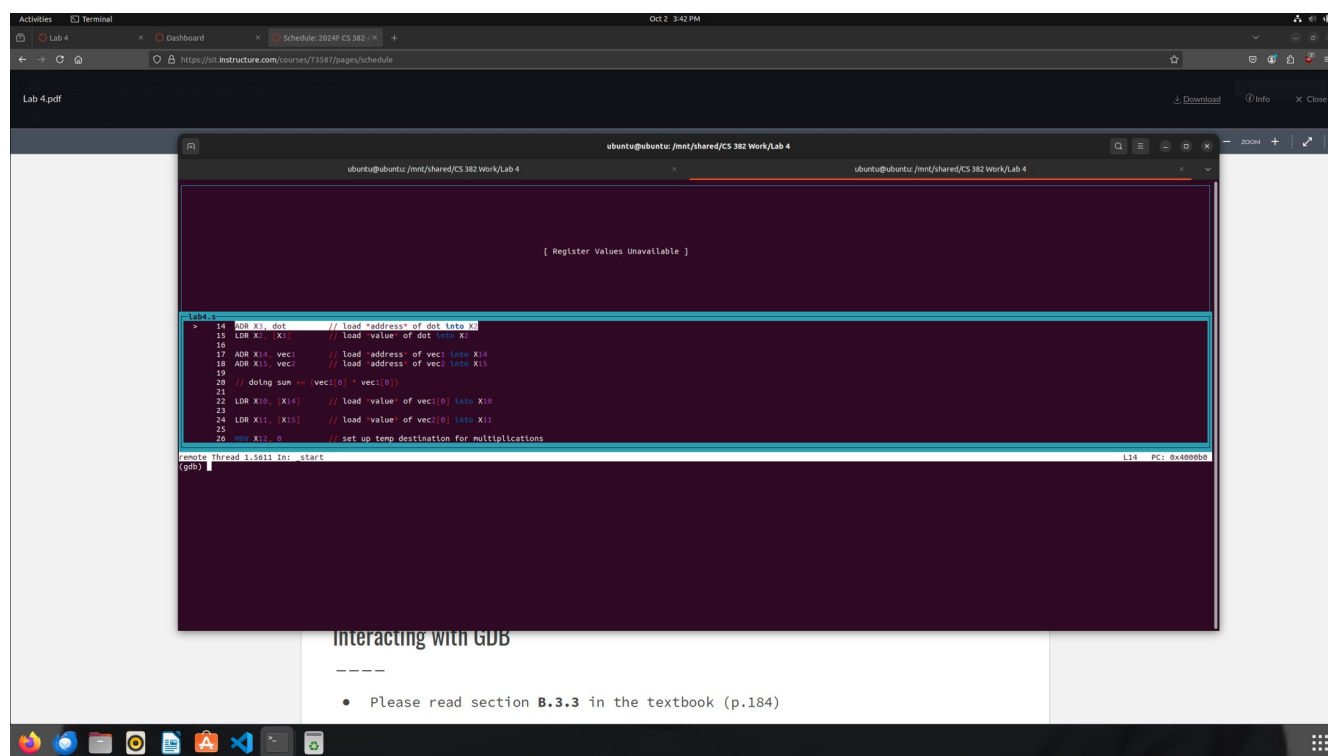


Author(s): Elliott McKelvey

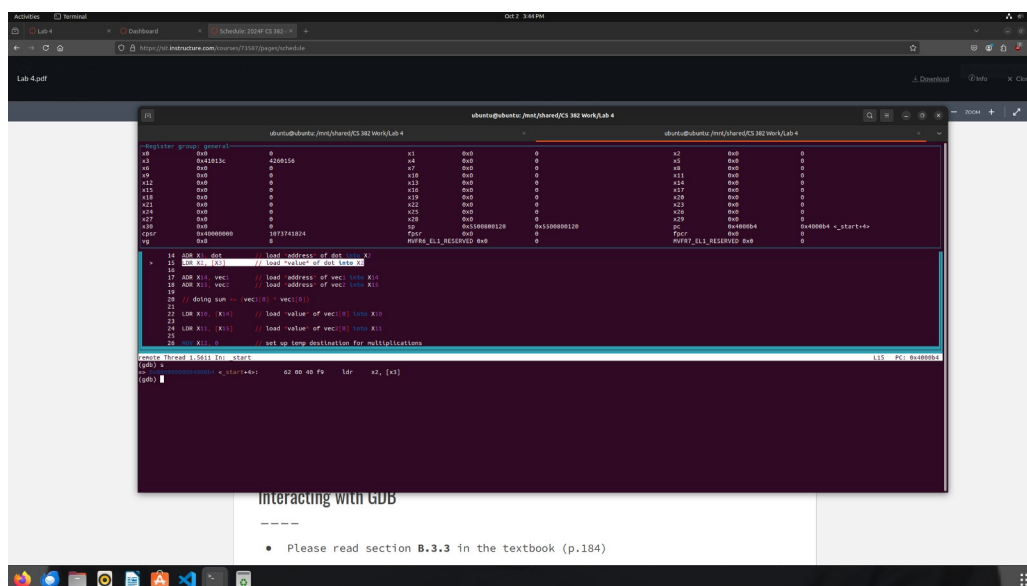
Pledge: I pledge my honor that I have abided by the Stevens Honor System.

Lab 4 Report

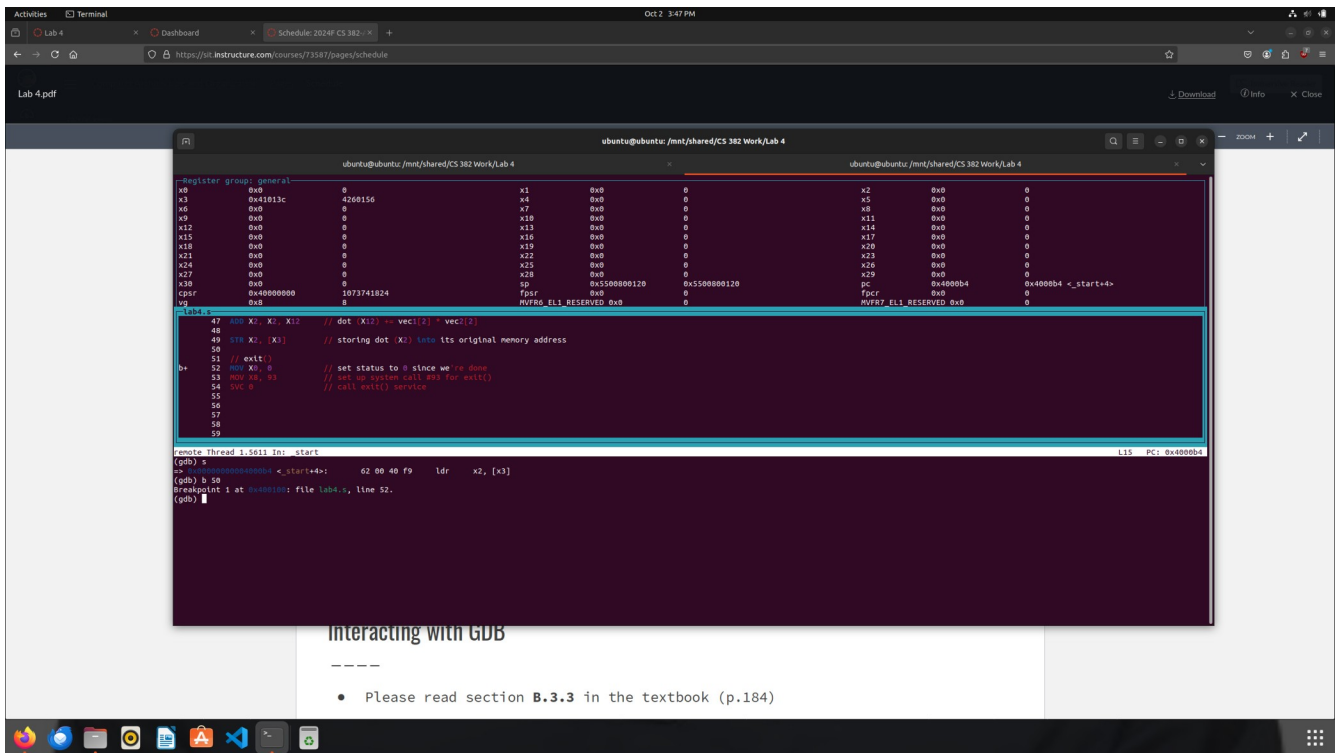
1. Here is the view of the terminal upon beginning to debug. Register values are not yet available because we have not stepped into the program yet.



2. Performing the *s* command once steps us into the program. Here, we are about to load the address of dot into register X2.



- Before we proceed any further, we set a breakpoint (command *b*) at line 50 to view what the resultant of our dot product operation will be.

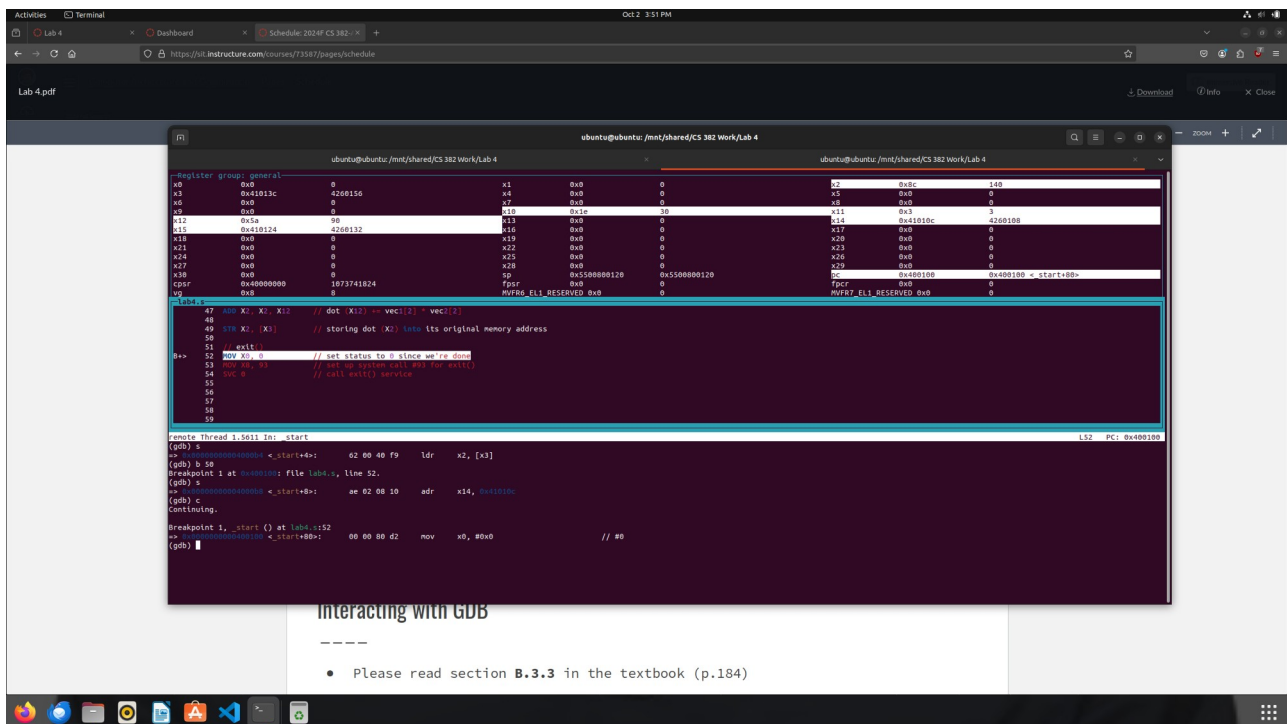


```
ubuntu@ubuntu: /mnt/shared/CS 382 Work/Lab 4
--Register group: general--
x0 0x0 0 x1 0x0 0 x2 0x0 0
x3 0x41013c 4268156 x4 0x0 0 x5 0x0 0
x6 0x0 0 x7 0x0 0 x8 0x0 0
x9 0x0 0 x10 0x0 0 x11 0x0 0
x12 0x0 0 x13 0x0 0 x14 0x0 0
x15 0x0 0 x16 0x0 0 x17 0x0 0
x18 0x0 0 x19 0x0 0 x20 0x0 0
x21 0x0 0 x22 0x0 0 x23 0x0 0
x24 0x0 0 x25 0x0 0 x26 0x0 0
x27 0x0 0 x28 0x0 0 x29 0x0 0
x30 0x0 0 sp 0x5508000120 0x5508000120 pc 0x4000b4 0x4000b4 <_start+4>
cpsr 0x40000000 8 fpcr 0x0 0
v0 0x0 0 mvfr0 EL1: RESERVED 0x0 0
mvfr7 EL1: RESERVED 0x0 0

47 mov x2, x12 // dot (x12) += vec1[2] * vec1[2]
48
49 str x2, [x1] // storing dot (x2) into its original memory address
50
51 // exit()
52 mov x0, 0 // set status to 0 since we're done
53 mov x8, 93 // set up system call #93 for exit()
54 svc 0 // call exit() service
55
56
57
58
59

Remote Thread 1.5611 In: _start
(gdb) s
0x4000b4: 0x4000b4 <_start+4>: 02 00 40 f9 ldr x2, [x1]
(gdb) b 50
Breakpoint 1 at 0x4000b4: file lab4.s, line 52.
(gdb)
```

- Next, we run *c* to continue execution of the program. Because of our actions in step 3, the program halts at the next possible instruction after line 50, which is the start of our exit script. As the program has copied the vectors over, we set the status to 0.



```
ubuntu@ubuntu: /mnt/shared/CS 382 Work/Lab 4
--Register group: general--
x0 0x0 0 x1 0x0 0 x2 0x0 140
x3 0x41013c 4268156 x4 0x0 0 x5 0x0 0
x6 0x0 0 x7 0x0 0 x8 0x0 0
x9 0x0 0 x10 0x0 30 x11 0x3 3
x12 0x0 0 x13 0x0 0 x14 0x41013c 4268156
x15 0x410124 4268132 x16 0x0 0 x17 0x0 0
x18 0x0 0 x19 0x0 0 x20 0x0 0
x21 0x0 0 x22 0x0 0 x23 0x0 0
x24 0x0 0 x25 0x0 0 x26 0x0 0
x27 0x0 0 x28 0x0 0 x29 0x0 0
x30 0x0 0 sp 0x5508000120 0x5508000120 pc 0x400100 0x400100 <_start+80>
cpsr 0x40000000 8 fpcr 0x0 0
v0 0x0 0 mvfr0 EL1: RESERVED 0x0 0
mvfr7 EL1: RESERVED 0x0 0

47 mov x2, x12 // dot (x12) += vec1[2] * vec1[2]
48
49 str x2, [x1] // storing dot (x2) into its original memory address
50
51 // exit()
52 mov x0, 0 // set status to 0 since we're done
53 mov x8, 93 // set up system call #93 for exit()
54 svc 0 // call exit() service
55
56
57
58
59

Remote Thread 1.5611 In: _start
(gdb) s
0x4000b4: 0x4000b4 <_start+4>: 02 00 40 f9 ldr x2, [x1]
(gdb) b 50
Breakpoint 1 at 0x4000b4: file lab4.s, line 52.
(gdb) s
0x4000b4: 0x4000b4 <_start+4>: 02 00 40 f9 ldr x2, [x1]
(gdb) s
0x4000b4: 0x4000b4 <_start+4>: 02 00 40 f9 ldr x2, [x1]
(gdb) c
Continuing.
Breakpoint 1, _start () at lab4.s:52
0x4000b4: 0x4000b4 <_start+80>: 00 00 80 d2 mov x0, #0
(gdb)
```

5. We step one more time to prepare for exit. As we can see from the circled portion, the program was able to give us the correct answer of 140 for the dot product output, which was copied to X2.

```
Register group: general
x0 0x0 0 0 x1 0x0 0 0 x2 0x8c 140
x3 0x401013c 4260156 x4 0x0 0 0 x5 0x0 0 0
x6 0x0 0 0 x7 0x0 0 0 x8 0x0 0 0
x9 0x0 0 0 x10 0x1e 30 0 x11 0x3 3 0
x12 0x5a 90 x13 0x0 0 0 x14 0x401010c 4260188
x15 0x4010124 4260132 x16 0x0 0 0 x17 0x0 0 0
x18 0x0 0 0 x19 0x0 0 0 x20 0x0 0 0
x21 0x0 0 0 x22 0x0 0 0 x23 0x0 0 0
x24 0x0 0 0 x25 0x0 0 0 x26 0x0 0 0
x27 0x0 0 0 x28 0x0 0 0 x29 0x0 0 0
x30 0x0 0 0 sp 0x5500000120 0x5500000120
cpsr 0x40000000 1073741824 fpscr 0x0 0 0 fpcr 0x400104 0x400104 <- start+84>
vfpfr EL1_RESERVED 0x0 0 0 vfpfr EL1_RESERVED 0x0 0 0

lab4.s
47: add x2, x1, x1 // dot (x12) = vec1[2] * vec1[2]
48:
49: str x2, [x1] // storing dot (x1) into its original memory address
50:
51: // exit()
52: mov x0, 0 // set status to 0 since we're done
53: mov x8, #3 // set up system call #93 for exit()
54: // call exit() service
55:
56:
57:
58:
59:

remote Thread 1.5011 in: _start
(gdb) s
=> 0x0000000000000000 <_start+40>: 62 00 40 f9 ldr x2, [x1]
(gdb) b 50
Breakpoint 1 at 0x0000000000000000: file lab4.s, line 52.
(gdb) s
=> 0x0000000000000000 <_start+80>: ae 02 08 10 adr x14, 0x401010c
(gdb) c
Continuing.

Breakpoint 1, _start () at lab4.s:52
=> 0x0000000000000000 <_start+80>: 00 00 80 d2 mov x0, #0 // #0
(gdb)
Display all 197 possibilities? (y or n)
(gdb) s
=> 0x0000000000000000 <_start+84>: a8 0b 80 d2 mov x8, #0x5d // #93
(gdb)

```

► is generated by a compiler;
► cannot be explained clearly in person.

Earlybird Extra Credit: 2% of extra credit will be given if the lab is finished by Wednesday 11:59PM EST (1 day before the lab deadline). For specific policy, see syllabus.

Attendance: check off at the end of the lab to get attendance credit.

6. We continue execution from here. The program exits as expected.

```
Register group: general
x0 0x0 0 0 x1 0x0 0 0 x2 0x8c 140
x3 0x401013c 4260156 x4 0x0 0 0 x5 0x0 0 0
x6 0x0 0 0 x7 0x0 0 0 x8 0x0 0 0
x9 0x0 0 0 x10 0x1e 30 0 x11 0x3 3 0
x12 0x5a 90 x13 0x0 0 0 x14 0x401010c 4260188
x15 0x4010124 4260132 x16 0x0 0 0 x17 0x0 0 0
x18 0x0 0 0 x19 0x0 0 0 x20 0x0 0 0
x21 0x0 0 0 x22 0x0 0 0 x23 0x0 0 0
x24 0x0 0 0 x25 0x0 0 0 x26 0x0 0 0
x27 0x0 0 0 x28 0x0 0 0 x29 0x0 0 0
x30 0x0 0 0 sp 0x5500000120 0x5500000120
cpsr 0x40000000 1073741824 fpscr 0x0 0 0 fpcr 0x400104 0x400104 <- start+84>
vfpfr EL1_RESERVED 0x0 0 0 vfpfr EL1_RESERVED 0x0 0 0

lab4.s
53: mov x8, #3 // set up system call #93 for exit()
54: mov x0, 0 // call exit() service
55:
56:
57:
58:
59:
60:
61:
62:
63:
64:
65:

exec No process in:
=> 0x0000000000000000 <_start+40>: 62 00 40 f9 ldr x2, [x1]
(gdb) b 50
Breakpoint 1 at 0x0000000000000000: file lab4.s, line 52.
(gdb) s
=> 0x0000000000000000 <_start+80>: ae 02 08 10 adr x14, 0x401010c
(gdb) c
Continuing.

Breakpoint 1, _start () at lab4.s:52
=> 0x0000000000000000 <_start+80>: 00 00 80 d2 mov x0, #0 // #0
(gdb)
Display all 197 possibilities? (y or n)
(gdb) s
=> 0x0000000000000000 <_start+84>: a8 0b 80 d2 mov x8, #0x5d // #93
(gdb)
Continuing.
[Inferior 1 (process 1) exited normally]
(gdb)

```

► is generated by a compiler;
► cannot be explained clearly in person.

Earlybird Extra Credit: 2% of extra credit will be given if the lab is finished by Wednesday 11:59PM EST (1 day before the lab deadline). For specific policy, see syllabus.

Attendance: check off at the end of the lab to get attendance credit.