

Module: R2: Intro to RISC-V Assembly
Section: CALL Task: Spike Installation

Activity

Spike Installation

1: Running a Simple C Program:

■ Code Snippet:

```
#include<stdio.h>

int main()
{
    int x,y;
    printf("Enter two Numbers (x, y):\n");
    scanf("%d\n", &x);
    printf("\n");
    scanf("%d", &y);

    int sum = 0;
    sum = x + y;

    printf("Sum of x and y is: %d\n", sum);
    return 0;
}
```

Compiling the **main.c** file to create a relocatable object file which will then pass to the linker in the later stage.

```
riscv64-unknown-elf-gcc -c main.c -o main.o
```

Now, creating the final executable by linking this relocatable object file **main.o** and other necessary standard library files. (It produces a binary executable with **.out** extension).

```
riscv64-unknown-elf-gcc -c main.c -o main.o
```

Now, running the executable using spike

```
spike pk ./a.out
```

Terminal Output:

```

xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ ls
Assembly-Program  main.c
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ riscv64-unknown-elf-gcc -c main.c -o main.o
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ ls
Assembly-Program  main.c  main.o
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ riscv64-unknown-elf-gcc main.o
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ ls
a.out  Assembly-Program  main.c  main.o
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ spike pk ./a.out
Enter two Numbers (x, y):
12
13
Sum of x and y is: 25
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ spike pk ./a.out
Enter two Numbers (x, y):
3
6
Sum of x and y is: 9
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ spike pk ./a.out
Enter two Numbers (x, y):
22
45
Sum of x and y is: 67
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$ spike pk ./a.out
Enter two Numbers (x, y):
39
67
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program$

```

2: Running RISCV Assembly on Spike:

Downloaded Linked and other header files in the directory and run the assembly program:

```
riscv64-unknown-elf-gcc -march=rv64gc -mabi=lp64 -nostdlib -T link.ld
test.S -o test.elf
```

Run the following command to see the dis-assembly file:

```
riscv64-unknown-elf-objdump -d test.elf
```

```

xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ ls
encoding.h  env  env2  link.ld  riscv_test.h  spike-out  test.S
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ riscv64-unknown-elf-gcc -march=rv64gc -mabi=lp64 -nostdlib -T link.ld test.S -o test.elf
xe-user106@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ riscv64-unknown-elf-objdump -d test.elf
test.elf:      file format elf64-littlelscv

Disassembly of section .text.int:

0000000000000000 <_start>:
00000000: a111          j          80000040 <reset_vector>
00000002: 0001          nop

0000000000000004 <trap_vector>:
00000004: 34202f73      csrr     t5,_mcause
00000008: 4f31          li       t5,9
0000000a: 03fff063      beq     t5,t5,80000036 <write_tohost>
0000000c: 4f55          li       t5,9
0000000e: 03fff063      beq     t5,t5,80000036 <write_tohost>
00000010: 4f4d          li       t5,11
00000012: 03fff063      beq     t5,t5,80000036 <write_tohost>
00000014: 0000011f      nop
00000016: 140f0f11      addi    t5,t5,328 # 80000162 <ntvec_handler>
00000018: 00000136      beqz    t5,80000028 <trap_vector0x24>
0000001a: 0f02          jr       t5
0000001c: 34202f73      csrr     t5,_mcause
0000001e: 000f3063      bgez    t5,80000032 <handle_exception>
00000020: a009          j        80000032 <handle_exception>

0000000000000032 <handle_exception>:
00000032: 33160193      ori     gp,gp,1337

0000000000000036 <write_tohost>:
00000036: 0000f1f7      auipc   t5,0x1
00000038: f03f2523      sw      gp,14(t5) # 00001000 <tohost>
0000003a: 0f05          j        80000036 <write_tohost>

0000000000000040 <reset_vector>:
00000040: 4001          li       ra,0
00000042: 4101          li       t0,0x0
00000044: 4181          li       t0,0x0
00000046: 4201          li       t0,0x0
00000048: 4281          li       t0,0x0
0000004a: 4301          li       t0,0x0
0000004c: 4381          li       t0,0x0
0000004e: 4401          li       t0,0x0
00000050: 4481          li       t0,0x0
00000052: 4501          li       t0,0x0
00000054: 4581          li       t0,0x0
00000056: 4601          li       t0,0x0
00000058: 4681          li       t0,0x0

```

Run the following command to see the get the spike.log to see how the instructions were executed and learn about the contents of the registers:

```
spike --isa=RV64IMAFDC -l --log-commits test.elf 1>spike.out 2>spike.log
```

spike.log:

```

32 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
51 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
50 core 0: 3 0x000000000000003e (0xbfbef5)
49 core 0: >>>> write_tohost
48 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
47 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
46 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
45 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
44 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
43 core 0: 3 0x000000000000003e (0xbfbef5)
42 core 0: >>>> write_tohost
41 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
40 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
39 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
38 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
37 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
36 core 0: 3 0x000000000000003e (0xbfbef5)
35 core 0: >>>> write_tohost
34 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
33 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
32 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
31 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
30 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
29 core 0: 3 0x000000000000003e (0xbfbef5)
28 core 0: >>>> write_tohost
27 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
26 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
25 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
24 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
23 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
22 core 0: 3 0x000000000000003e (0xbfbef5)
21 core 0: >>>> write_tohost
20 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
19 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
18 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
17 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
16 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
15 core 0: 3 0x000000000000003e (0xbfbef5)
14 core 0: >>>> write_tohost
13 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
12 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
11 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
10 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
9 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
8 core 0: 3 0x000000000000003e (0xbfbef5)
7 core 0: >>>> write_tohost
6 core 0: 0x0000000000000036 (0x00001f17) auipc t5, 0x1
5 core 0: 3 0x0000000000000036 (0x00001f17) x30 0x0000000000001036
4 core 0: 0x000000000000003a (0xfc3f2523) sw gp, -54(t5)
3 core 0: 3 0x000000000000003a (0xfc3f2523) mem 0x0000000000001000 0x00000001
2 core 0: 0x000000000000003e (0xbfbef5) c.j pc - 8
1 core 0: 3 0x000000000000003e (0xbfbef5)
0 core 0: >>>> write_tohost

```

3 . Running Risc-V Assembly programs compatible to RiscOf on spike:

Downloaded the files from the given link and pasted in my **Assembly-Program/** folder.
Now, run this command:

```
riscv64-unknown-elf-gcc -march=rv64izicr -static -mcmodel=medany
-fvisibility=hidden -nostdlib -nostartfiles -T
~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2\ -\
Intro\ to\ RISCV/Spike-Installation/Simple-C-Program/Assembly-Program/link.ld
-I ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2\ -\
Intro\ to\ RISCV/Spike-Installation/Simple-C-Program/Assembly-Program/env/ -I
~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2\ -\
Intro\ to\ RISCV/Spike-Installation/Simple-C-Program/Assembly-Program/env2/
-mabi=lp64 test.S -o my.elf -Drvtest_mtrap_routine=True -DTEST_CASE_1=True
-DXLEN=64
```

Run the following command to see the dis-assembly file:

```
Riscv64-unknown-elf-objdump -d my.elf
```

```

x-user10@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program
$ ls
encoding.h  env2  riscv_test.h  'spike.log'  test.elf
$ gcc -march=rv64i2c -static -mcmodel=medany -fvlsiblltty=hidden -nostdlib -nostartfiles -T ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ clear
$ gcc -march=rv64i2c -static -mcmodel=medany -fvlsiblltty=hidden -nostdlib -nostartfiles -T ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ ls
encoding.h  env2  riscv_test.h  'spike.log'  test.elf
$ gcc -march=rv64i2c -static -mcmodel=medany -fvlsiblltty=hidden -nostdlib -nostartfiles -T ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ riscv64-unknown-elf-gcc -march=rv64i2c -static -mcmodel=medany -fvlsiblltty=hidden -nostdlib -nostartfiles -T ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program/env2/ -I ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program/env2/ -mabi=lp64 test.S -o my.elf -brvtest_ntrap_routine=true -DTEST_CA
SE_1=true -DLEN=64
$ gcc -march=rv64i2c -static -mcmodel=medany -fvlsiblltty=hidden -nostdlib -nostartfiles -T ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ ls
encoding.h  env2  riscv_test.h  'spike.log'  test.elf
$ gcc -march=rv64i2c -static -mcmodel=medany -fvlsiblltty=hidden -nostdlib -nostartfiles -T ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - \
Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program$ riscv64-unknown-elf-objdump -d
my.elf
my.elf:      file format elf64-littleliscv

Disassembly of section .text.int:

0000000000000000 <start>:
00000000: 04c0000f                j      8000004c <reset_vector>

0000000000000004 <trap_vector>:
00000004: 34202f73              csrr   t5, mcause
00000008: 00000f93              ll     t6, 0
0000000c: 03ff0663              beq    t5, t6, 80000040 <write_tohost>
00000010: 00000f93              ll     t6, 9
00000014: 03ff0663              beq    t5, t6, 80000040 <write_tohost>
00000018: 00000f93              ll     t6, 11
0000001c: 03ff0663              beq    t5, t6, 80000040 <write_tohost>
00000020: 00000f17              auipc  t5, 0x0
00000024: 1ac0f013              addi   t5, t5, 428 # 800001cc <ntvec_handler>
00000028: 00000463              beqz   t5, 80000030 <trap_vector+0x2c>
0000002c: 00000007              jr     t5
00000030: 34202f73              csrr   t5, mcause
00000034: 000f5463              bgez   t5, 8000003c <handle_exception>
00000038: 0000000f              j      8000003c <handle_exception>

000000000000003c <handle_exception>:
0000003c: 5391e193              ori    gp, gp, 1337

0000000000000040 <write_tohost>:
00000040: 00001f17              auipc  t5, 0x1
00000044: fc3f2023              sw     gp, -64(t5) # 80010000 <tohost>
00000048: f9ff060f              j      80000040 <write_tohost>

000000000000004c <reset_vector>:
0000004c: 00000093              ll     ra, 0
00000050: 00000113              li     sp, 0
00000054: 00000193              li     gp, 0
00000058: 00000213              li     tp, 0

```

Run the following command to see the get the spike.log to see how the instructions were executed and learn about the contents of the registers:

```
spike --isa=RV64IMAFDC -l --log-commits my.elf 1>spike.out 2>spike.log
```

```

x-user10@noman-10xengineers: ~/10x-Engineers/Remedial-Training/R2-Intro-to-RISCV/fa21-lab-starter/R2 - Intro to RISCV/Spike-Installation/Simple-C-Program/Assembly-Program
$ spike --isa=RV64IMAFDC -l --log-commits my.elf 1>spike.out 2>spike.log
1 core 0: 0x0000000000001000 (0x00000297) auipc  t0, 0x0
1 core 0: 0x0000000000001000 (0x00000297) x5  0x0000000000001000
2 core 0: 0x0000000000001004 (0x02028593) addi   a1, t0, 32
3 core 0: 0x0000000000001004 (0x02028593) x11 0x0000000000001020
4 core 0: 0x0000000000001008 (0xf1402573) csrr   a0, mhartid
5 core 0: 0x0000000000001008 (0xf1402573) x10 0x0000000000000000
6 core 0: 0x000000000000100c (0x182b283) ld     t0, 24(t0)
7 core 0: 0x000000000000100c (0x182b283) x5  0x0000000000000000 men 0x0000000000001018
8 core 0: 0x0000000000001010 (0x00028067) jr     t0
9 core 0: 0x0000000000001010 (0x00028067) jr     t0
10 core 0: >>> $rsv64i2pi_2lcar2p0
11 core 0: 0x0000000000000000 (0x4c00000f) j      pc + 0x4c
12 core 0: 0x0000000000000000 (0x4c00000f) j      pc + 0x4c
13 core 0: >>> reset_vector
14 core 0: 0x000000000000004c (0x00000093) ll     ra, 0
15 core 0: 0x000000000000004c (0x00000093) x1  0x0000000000000000
16 core 0: 0x0000000000000050 (0x00000113) li     sp, 0
17 core 0: 0x0000000000000050 (0x00000113) x2  0x0000000000000000
18 core 0: 0x0000000000000054 (0x00000193) li     gp, 0
19 core 0: 0x0000000000000054 (0x00000193) x3  0x0000000000000000
20 core 0: 0x0000000000000058 (0x00000213) li     tp, 0
21 core 0: 0x0000000000000058 (0x00000213) x4  0x0000000000000000
22 core 0: 0x000000000000005c (0x00000293) ll     t0, 0
23 core 0: 0x000000000000005c (0x00000293) x5  0x0000000000000000
24 core 0: 0x0000000000000060 (0x00000313) li     t1, 0
25 core 0: 0x0000000000000060 (0x00000313) x6  0x0000000000000000
26 core 0: 0x0000000000000064 (0x00000393) li     t2, 0
27 core 0: 0x0000000000000064 (0x00000393) x7  0x0000000000000000
28 core 0: 0x0000000000000068 (0x00000413) li     s0, 0
29 core 0: 0x0000000000000068 (0x00000413) x8  0x0000000000000000
30 core 0: 0x000000000000006c (0x00000493) li     s1, 0
31 core 0: 0x000000000000006c (0x00000493) x9  0x0000000000000000
32 core 0: 0x0000000000000070 (0x00000513) li     a0, 0
33 core 0: 0x0000000000000070 (0x00000513) x10 0x0000000000000000
34 core 0: 0x0000000000000074 (0x00000593) li     a1, 0
35 core 0: 0x0000000000000074 (0x00000593) x11 0x0000000000000000
36 core 0: 0x0000000000000078 (0x00000613) li     a2, 0
37 core 0: 0x0000000000000078 (0x00000613) x12 0x0000000000000000
38 core 0: 0x000000000000007c (0x00000693) li     a3, 0
39 core 0: 0x000000000000007c (0x00000693) x13 0x0000000000000000
40 core 0: 0x0000000000000080 (0x00000713) li     a4, 0
41 core 0: 0x0000000000000080 (0x00000713) x14 0x0000000000000000
42 core 0: 0x0000000000000084 (0x00000793) li     a5, 0
43 core 0: 0x0000000000000084 (0x00000793) x15 0x0000000000000000
44 core 0: 0x0000000000000088 (0x00000813) li     a6, 0
45 core 0: 0x0000000000000088 (0x00000813) x16 0x0000000000000000
46 core 0: 0x000000000000008c (0x00000893) li     a7, 0
47 core 0: 0x000000000000008c (0x00000893) x17 0x0000000000000000
48 core 0: 0x0000000000000090 (0x00000913) li     s2, 0
49 core 0: 0x0000000000000090 (0x00000913) x18 0x0000000000000000
50 core 0: 0x0000000000000094 (0x00000993) li     s3, 0
51 core 0: 0x0000000000000094 (0x00000993) x19 0x0000000000000000
52 core 0: 0x0000000000000098 (0x00000a13) li     s4, 0

```