



Course Name: Computer Architecture Lab

Course Number and Section:
14:332:333:03

Experiment: Lab 5: RISC-V Functions and Pointers

Lab Instructor: Ali Haddad

Date Performed: 11/5/2018

Date Submitted: 11/27/2018

Submitted by: Abdullah Ghani 171009840

Exercise 1: Debugging megalistmanips.s

Fixed code in the file megalistmanips.s, uploaded in the folder

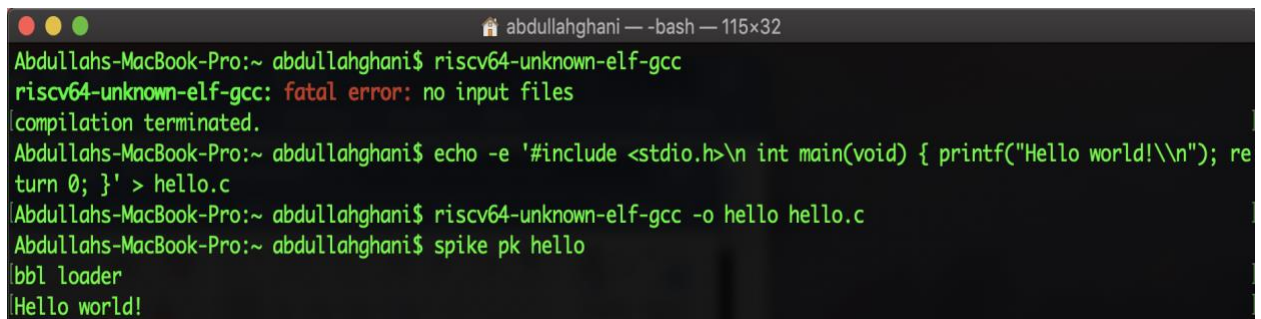
Exercise 2: Write a function without branches

Please see the submitted code file “discrete_fn.s” for my implementation of the function.

Exercise 3: RISC-V C Compiler and Spike

I have attached screenshots of the terminal for each section of this exercise.

Test the compiler:

A terminal window titled 'abdullahghani — -bash — 115x32' showing a series of commands and their outputs. The commands are: 1. 'riscv64-unknown-elf-gcc' which results in a 'fatal error: no input files' and 'compilation terminated.' 2. 'echo -e '#include <stdio.h>\n int main(void) { printf("Hello world!\\n"); return 0; }' > hello.c' which creates a file named 'hello.c'. 3. 'riscv64-unknown-elf-gcc -o hello hello.c' which compiles 'hello.c' into an executable named 'hello'. 4. 'spike pk hello' which runs the 'hello' program, outputting 'bbl loader' and 'Hello world!' on separate lines.

```
Abdullahs-MacBook-Pro:~ abdullahghani$ riscv64-unknown-elf-gcc
riscv64-unknown-elf-gcc: fatal error: no input files
compilation terminated.
Abdullahs-MacBook-Pro:~ abdullahghani$ echo -e '#include <stdio.h>\n int main(void) { printf("Hello world!\\n"); re
turn 0; }' > hello.c
Abdullahs-MacBook-Pro:~ abdullahghani$ riscv64-unknown-elf-gcc -o hello hello.c
Abdullahs-MacBook-Pro:~ abdullahghani$ spike pk hello
bbl loader
Hello world!
```

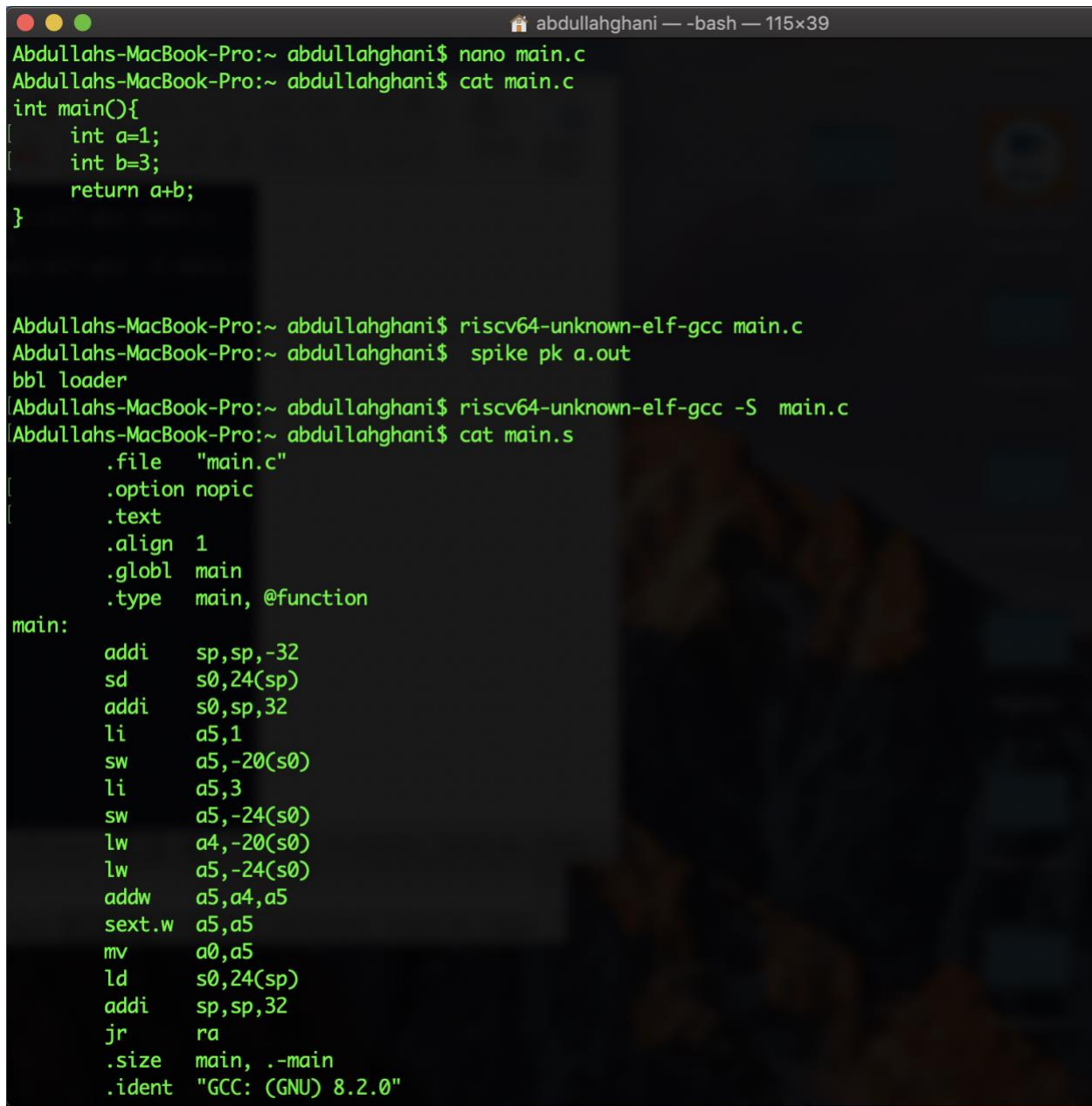
Exploring Compiled C Code:

```
Abdullahs-MacBook-Pro:~ abduallahghani$ nano foo.c
Abdullahs-MacBook-Pro:~ abduallahghani$ cat foo.c
int foo(int a, int b){
    return a+b;
}
Abdullahs-MacBook-Pro:~ abduallahghani$ riscv64-unknown-elf-gcc -S foo.c -o foo.s
Abdullahs-MacBook-Pro:~ abduallahghani$ cat foo.s
        .file     "foo.c"
        .option   nopie
        .text
        .align    1
        .globl    foo
        .type      foo, @function
foo:
        addi      sp,sp,-32
        sd        s0,24(sp)
        addi      s0,sp,32
        mv        a5,a0
        mv        a4,a1
        sw        a5,-20(s0)
        mv        a5,a4
        sw        a5,-24(s0)
        lw        a4,-20(s0)
        lw        a5,-24(s0)
        addw      a5,a4,a5
        sext.w    a5,a5
        mv        a0,a5
        ld        s0,24(sp)
        addi      sp,sp,32
        jr        ra
        .size     foo, .-foo
        .ident    "GCC: (GNU) 8.2.0"
```

After Optimization:

```
Abdullahs-MacBook-Pro:~ abduallahghani$ riscv64-unknown-elf-gcc -S foo.c -o foo.s -O2
Abdullahs-MacBook-Pro:~ abduallahghani$ cat foo.s
        .file     "foo.c"
        .option   nopie
        .text
        .align    1
        .globl    foo
        .type      foo, @function
foo:
        addw      a0,a0,a1
        ret
        .size     foo, .-foo
        .ident    "GCC: (GNU) 8.2.0"
```

Compiling to an executable:



```
Abdullahs-MacBook-Pro:~ abduallahghani$ nano main.c
Abdullahs-MacBook-Pro:~ abduallahghani$ cat main.c
int main(){
[   int a=1;
[   int b=3;
    return a+b;
}

Abdullahs-MacBook-Pro:~ abduallahghani$ riscv64-unknown-elf-gcc main.c
Abdullahs-MacBook-Pro:~ abduallahghani$ spike pk a.out
bbl loader
Abdullahs-MacBook-Pro:~ abduallahghani$ riscv64-unknown-elf-gcc -S main.c
Abdullahs-MacBook-Pro:~ abduallahghani$ cat main.s
    .file "main.c"
[    .option nopic
[    .text
    .align 1
    .globl main
    .type main, @function
main:
    addi    sp,sp,-32
    sd      s0,24(sp)
    addi    s0,sp,32
    li      a5,1
    sw      a5,-20(s0)
    li      a5,3
    sw      a5,-24(s0)
    lw      a4,-20(s0)
    lw      a5,-24(s0)
    addw    a5,a4,a5
    sext.w  a5,a5
    mv      a0,a5
    ld      s0,24(sp)
    addi    sp,sp,32
    jr      ra
    .size   main, .-main
    .ident  "GCC: (GNU) 8.2.0"
```

Then we execute the following command, which performs an object dump and redirects the output into a text file named “dump.txt”:

```
$ riscv64-unknown-elf-objdump -d a.out > dump.txt
```

Below is a screenshot of the **main** function found in the file “dump.txt”:

00000000001019c <main>:

1019c:	1101	addi	sp,sp,-32
1019e:	ec22	sd	s0,24(sp)
101a0:	1000	addi	s0,sp,32
101a2:	4785	li	a5,1
101a4:	fef42623	sw	a5,-20(s0)
101a8:	478d	li	a5,3
101aa:	fef42423	sw	a5,-24(s0)
101ae:	fec42703	lw	a4,-20(s0)
101b2:	fe842783	lw	a5,-24(s0)
101b6:	9fb9	addw	a5,a5,a4
101b8:	2781	sxt.w	a5,a5
101ba:	853e	mv	a0,a5
101bc:	6462	ld	s0,24(sp)
101be:	6105	addi	sp,sp,32
101c0:	8082	ret	