

CSSE 132 – Introduction to Systems Programming
 Rose-Hulman Institute of Technology

Worksheet 04: ARM64 Assembly Basics

Name (Print): _____ RHIT Username: _____

In this worksheet, a C `int` is 4 bytes and an address is 8 bytes. Use a `wN` register for a 32-bit `int` and an `xN` register for an address. Offsets inside brackets are measured in bytes.

1. Complete the table for the steps used to build an executable program.

Step	Input	Output	Tool
Compile			
Assemble			
Link			

2. ARM64 gives two names to different-width views of the same register. Complete the table, then answer the question below it.

View	Width	Course use
w3		
x3		

What is the value of x3 after `mov w3, #1`?

3. Trace this code. Give the final value in each register, and identify where the destination operand appears in an arithmetic instruction.

```

1  mov w0, #11
2  add w1, w0, #14
3  sub w2, w1, #5
4

```

w0 =

w1 =

w2 =

The destination is the operand.

4. Assume $x2 = 0x1000$. Memory at $0x1000$ contains 9, and memory at $0x1004$ contains 12.

```

1  ldr w0, [x2]
2  ldr w1, [x2, #4]
3  add w0, w0, w1
4  str w0, [x2, #8]
5

```

- Give the final values of $w0$, $w1$, and $x2$.
 - What value is stored at $0x1008$?
 - For each `ldr` or `str`, draw an arrow showing the direction of data transfer. How many bytes move each time?
5. The first and last instructions reserve and release a 16-byte stack area. Label offsets 0, 4, and 8 as x , y , and $temp$, then write C-like pseudocode for the middle instructions.

```

1  sub sp, sp, #16
2  mov w3, #1
3  str w3, [sp]
4  mov w3, #16
5  str w3, [sp, #4]
6  ldr w3, [sp]
7  str w3, [sp, #8]
8  ldr w3, [sp, #4]
9  str w3, [sp]
10 ldr w3, [sp, #8]
11 str w3, [sp, #4]
12 add sp, sp, #16
13

```

6. Assume $x4$ is the base address of a writable, naturally aligned 24-byte region. Trace the code, draw the three objects in memory, and write C-like pseudocode. What is the final value of $w0$?

```

1  mov w0, #1
2  str w0, [x4]
3  mov x1, x4
4  str x1, [x4, #8]
5  mov w2, #2
6  str w2, [x4, #16]
7  ldr x1, [x4, #8]
8  ldr w2, [x4, #16]
9  str w2, [x1]
10 ldr w0, [x4]
11

```

7. **Reflection:** What is one thing from this topic that you found interesting, OR what is one question you are now curious to learn more about?