Tecnológico
de Monterrey

SCHOOL OF ENGINEERING AND SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Laboratory of Embedded Systems

Prof. Matías Vázquez

February—June 2021

PR4

ASSEMBLER PROGRAMMING IN RASPBIAN

Objectives

- Become familiar with Assembler instructions to manipulate registers at low level programming
- Become familiar with development tools for building executable images from Assembler coding

Activities (part I)

The following activities provide testing ASM codes for you to assemble, compile and run on the Raspberry Pi. Try out each of the sample programs for each activity on your assigned RPi and take screenshots of your outputs. All ASM codes are available on GitHub on the following link: https://github.com/matias-vazquez/SistemasEmbebidos/tree/main/Lab04.

To assemble the program test.s:

```
as -o test.o test.s
```

Then, to compile it:

```
gcc -o test test.o
```

And finally, to run it:

```
./test
```

- Activity 1. Program that returns an Exit code: `first.s`
  - For this case, run the program as follows to see the Exit code in the Terminal:

```
./first | echo $?
```

- Activity 2. Hello, World!: `hello.s`
- Activity 3. Arithmetic with integer variables: `sum1.s`
- Activity 4. Arithmetic with integer variables, version 2: `sum2.s`
- Activity 5. Passing parameters by value: `sum3.s`
- Activity 6. Passing parameters by reference: `sum4.s`
- Activity 7. Using the C-function *scanf()* for User input: `scanfExample.s`
- Activity 8. Pausing the program using the *sleep()/usleep()* C functions: `delayExample.s`
- Activity 9. Recursive Towers of Hanoi: hanoi.s and `hanoi2.s`
- Activity 10. Blinking LED: `blink.s`