

COMPUTER STRUCTURE

DEGREE IN COMPUTER ENGINEERING

DOUBLE DEGREE IN COMPUTER ENGINEERING AND
BUSINESS ADMINISTRATION

DEGREE IN APPLIED MATHEMATICS AND COMPUTING

Assignment 1

Introduction to Assembly Language

Course 2023/2024

Content

Objectives of the assignment.....	3
Exercise 1	3
Exercise 2	4
Recommendations for carrying out the exercises	4
Optional exercise	5
Important things to keep in mind	5
General rules	5
Report of the assignment.....	6
Definition of tests.....	7
Procedure for delivery of the assignment. Mandatory part.....	7
Procedure for delivery of the assignment. Optional part	8
Evaluation of assignment	9

Objectives of the assignment

The objective of the assignment is to understand the concepts related to assembly programming. For this, the RISC-V assembler (RV32IMF) and the CREATOR simulator, which is available at <https://creatorsim.github.io/creator/>, will be used as a basis.

To become familiar with assembly programming and the previous simulator, it is recommended to solve the **exercises available in Aula Global** in the section corresponding to the first practice.

The assignment consists of 2 mandatory exercises (Exercise 1 and Exercise 2) and an optional exercise, which are described below.

Exercise 1

The objective of this exercise is to:

- **sin(x)**: The sine function will receive as an argument a floating-point number in simple precision and returns the result as a floating number in simple precision.
- **cos(x)**: The cosine function will receive as an argument a floating-point number in simple precision and returns the result as a floating number in simple precision.
- **tg(x)**: The tangent function will receive as an argument a single-precision floating-point number and returns the result as a single-precision float number.
- **E()**: The E function takes no argument and returns the value of the number e as a single-precision floating-point value.

The following power series can be used to calculate these functions:

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

All functions will return the result with an error less than 0.001. The sine and cosine functions will be tested for x values between -10 and 10.

Exercise 2

In this exercise, the sine function developed above will be applied to calculate the sine of all the elements of a matrix. To do this, the **SinMatrix** function will be implemented. This function accepts the following arguments in the indicated order:

- Address of a matrix (**A**) of numbers represented as floating-point single precision, where the values to be calculated are stored.
- Address of a matrix (**B**) of floating-point numbers where results will be stored.
- The number (**N**) of rows in the matrix.
- Number (**M**) of columns in the matrix.

The function does not return any results.

This function performs: $\mathbf{B[i,j] = \sin(A[i,j])}$;

Recommendations for carrying out the exercises

It is recommended first of all to implement the above programs using some high-level programming language, like, for example, Python. The presentation of these programs in the report of the assignment delivered will be valued.

Optional exercise

The objective of this exercise is to use Creator to run code implemented on the [ESP32-C3](#) microcontroller.

This microcontroller can be purchased on different platforms by consulting:

<https://www.espressif.com/en/products/devkits>

The test programs used shall be tested on the microcontroller. The execution time of the code in the microcontroller must be measured, invoking the developed functions several times, analyzing the effect that the cache has on the execution time.

Important things to keep in mind

General rules

- (1) The names of the functions indicated in the statement must be written **in the form indicated, as written**.
- 2) The practices will be carried out in **groups of two**.
- 3) The delivery of the practice will be made through the authorized deliverers. **Delivery** via email is not permitted.
- 4) Delivery shall be made within the period given by the deliverers.
- 5) Special attention will be paid to detecting functionalities copied between two practices. In case of finding common implementations in two practices (or similar contents in memory), both will obtain a rating of 0 (zero).
- 6) The parameter passing convention described in class must be followed. **Those functions that do not correctly follow the parameter passing convention will also be rated with a 0 (zero)**.
- 7) All requested exercises must be delivered. If an exercise is not delivered, the practice will be considered **as not delivered**.
- 8) The exercises that do not compile or that do not conform to the functionality and requirements raised, will obtain a grade of 0 (zero).
- 9) An uncommented program will get a rating of 0 (zero).

Report of the assignment

1) The report (a single document) must contain at least the following sections:

- Cover page where the authors appear (including full name, NIA, email address, class group to which they belong, and degree taken).
- Table of Contents.
- Content requested in the different exercises (one section per exercise).
- Conclusions, problems encountered, and estimation of the time spent in the assignment.

2) **The length of the report must not exceed 10 pages** (cover and table of contents included).

(3) With regard to the possible description of the programs ordered:

- The report should describe the behavior of the programs, as well as the main design decisions. **The pseudocode corresponding to each of the exercises of this assignment must be included.**
- The **battery tests (as defined in the next section)** used to validate the functionality of the requested functions. **Results** obtained must be included. Higher scores will be given to advanced tests, extreme cases, and in general to those tests that guarantee the correct functioning of the practice in all cases.
 - Avoid duplicate tests that evaluate the same program flows. The score of this section is not measured according to the number of tests, but the degree of coverage of the same. It is better few tests that evaluate different cases than many tests that always evaluate the same case.

NOTE: DO NOT NEGLECT THE QUALITY OF THE REPORT OF YOUR ASSIGNMENT.

Approving the report is as essential to approve the assignment, as the correct functioning of [it](#). If, when the report is evaluated, it is considered that it does not reach the minimum admissible, your assignment will be suspended.

Definition of tests

For the definition section and battery of tests of each of the exercises that must be included in the memory a table that will follow the following format:

Data to enter:	Description of the test:	Result expected:	Result obtained:

Procedure for delivery of the assignment. Mandatory part

The delivery of Assignment 1 (including Exercise 1 and Exercise 2) will be done electronically through Aula Global.

The assignment will be done in groups of **two students**.

The deadline for both is **October 22th, 2023, at 11:55 p.m.**

It is possible to deliver as many times as you want within the given timeframe, the only recorded version of your assignment is the last one delivered. The assessment of the practice is the assessment of the content of this last installment. Always check what you deliver.

Deliverer: A single compressed file must be delivered in **zip** format with the name `ec_p1_AAAAAAAAAA_BBBBBBBBBB.zip` where A... A and B... B are the NIAs of the group members.

The **zip** file should contain only the following files:

- **Exercise1.S**
This file will only contain the code of the functions. It will have neither *main* [function](#) nor data segment. Therefore, the functions you deliver must work without depending on a specific data segment.
- **Exercise2.S**
This file will only contain the code of the requested function. It will have neither *main* [function](#) nor data segment. Therefore, the functions you deliver must work without depending on a specific data segment.
- **report.pdf**
This file will contain the contents of the memory, and will have PDF format (it is not worth renaming a text file or similar with .pdf extension).

- **Authors.txt**

This file will contain a line for each member of the group with the NIA and the group to which it belongs (example: 1000001;81).

Procedure for delivery of the assignment. Optional part

The delivery of the optional part will be made electronically through Aula Global.

The practice will be done in groups of **two students**.

The deadline for both is December **3rd, 2023, at 11:55 p.m.**

It is possible to deliver as many times as you want within the given timeframe, the only recorded version of your practice is the last one delivered. The assessment of the practice is the assessment of the content of this last installment. Always check what you deliver.

Deliverer: A single compressed file must be delivered in **zip** format with the name `ec_p1_AAAAAAAAAA_BBBBBBBBBB.zip` where A... A and B... B are the NIAs of the group members.

The **zip** file should contain only the following files:

- **memory.pdf**

This file will contain the contents of the memory, and will have PDF format (it is not worth renaming a text file or similar with .pdf extension). The file should include the *screenshots* obtained from the execution of the test programs on the Esp32-C3 microcontroller and an analysis of the execution times obtained.

- **Authors.txt**

This file will contain a line for each member of the group with the NIA and the group to which it belongs (example: 1000001;81).

Evaluation of assignment

- The compulsory part will allow to obtain 10 points and the following scale will be used:
 - Memory: 10%.
 - Test plan: 15%
 - Code delivered: 75%
- The optional part will allow you to add 0.5 points to the final grade of the subject.

Please note that, **to follow the continuous evaluation process**, the **minimum grade obtained in each practice** (compulsory part) must be 2 out of 10 **and the average of the two practices must be 4 out of 10. All the exercises of the compulsory part (Exercise 1 and 2) must be submitted** to be considered that the practice has been delivered.

NOTES:

1. **If a serious misconception is detected in assignment (in any section of any exercise), the overall assessment of the entire practice will be zero points (0 points).**
2. **If you do not respect the delivery format (for example, not respecting the name of the requested files, delivering a .rar file, delivering the files to a directory, etc.) the grade will be significantly reduced.**
3. **In case of finding common implementations in two assignment (or similar contents in the memory) it will be understood that the practice has been copied and both will obtain a grade of 0.**
4. **If fragments of code obtained directly from the Internet are found, it will be understood that said content has been copied and the practice will have a rating of 0.**