COMPUTER STRUCTURE

DEGREE IN COMPUTER ENGINEERING

DOUBLE DEGREE IN COMPUTER ENGINEERING AND BUSINESS ADMINISTRATION

uc3m | Universidad Carlos III de Madrid

ARCOS GROUP

Practice 1

Introduction to Assembly Language

Course 2019/2020

Content

Objective of the practice

The objective of the practice is to understand the concepts related to assembly programming. To do this, the MIPS32 assembler and the CREATOR simulator, which is available at https://creatorsim.github.io/creator/, will be used as a base.

To familiarize yourself with assembly programming and the CREATOR simulator, it is recommended to solve the exercises available in Aula Global in the section corresponding to the first lab.

This practice consists of 2 exercises.

Exercise 1

The objective of this exercise is to develop the following set of functions to work with matrices of integer numbers:

- Function **set**(int A[][], int M, int N). Initializes all values in matrix A to zero. M represents the number of rows and N the number of columns. The function returns 1 if M or N are negative or zero. Otherwise it returns 0.

- Function **add**(int A[][], int B[][], int C[][], int M, int N). It adds two matrices and leaves the result in another. This function accepts 5 arguments:
  - A: Matrix where the result of adding B and C will be stored.
  - B: represents the first matrix to add.
  - C: represents the second matrix to add.
  - M: number of rows of the three matrices.
  - N: number of columns of the three matrices.

  The function returns -1 if M or N are negative or zero; otherwise it returns 0.

- Function **extractRow**(int A[], Int B[][], int M, int N, int j). Given a matrix B of M rows and N columns, copy into vector A the row j of the matrix. The function returns -1 in the following cases:
  - M or N are negatives or zero.
  - j is less than 0 or j is greater than M; othrewise it returns 0.

- Function **moreZeros**(int A[][], int B[][], int M, int N). This function determines which of the matrices, A or B (both of M rows and N columns), has a greater number of elements equal to 0. The function returns -1 if M or N are negatives or zero; otherwise, the function returns 0 if A has more zeros than B or 1 if B has more zeros than A. If both have the same number of zeros, it returns 2. To develop this function, the following library function is provided as support material:
  - **calcular** (int A [] [], int M, int N, int value). This function returns the number of elements of the matrix A whose value coincides with the one in the fourth argument (value). The function returns -1 in case M or N are negative or zero. It is mandatory to invoke this function provided as support material for which the MIPS parameter passing agreement described in class must be followed. The library where this function is located is called **apoyo.o**.

When the value returned by any of these functions is -1, the function will not perform any other task, only the value -1 is returned.

Take into account that for all functions, the parameter passing agreement described in class must be followed.

Exercise 2

The objective of this exercise is to develop the following two functions that work with simple precision floating point number matrices (`float` type in C language):

- Function **extractValues**, which accepts the following arguments in the order indicated:

- Argument 1: Start address of a matrix (A) of floating point numbers.
- Argument 2: Number of rows in the matrix (M).
- Argument 3: Number of columns in the matrix (N)
- Argument 4: Start address of a vector V of 6 elements.

The function stores the following values in vector V:

- V [0] stores the number of elements in the A matrix whose value is 0.
- V [1] stores the number of elements in the A matrix whose value is plus infinity.
- V [2] stores the number of elements in the A matrix whose value is minus infinity.
- V [3] stores the number of elements in the A matrix with NaN value.
- V [4] stores the number of elements in the A matrix with non-normalized numbers.
- V [5] stores the number of elements in the A matrix with normalized numbers.

The function returns -1 if M or N are negative or zero. In this case it is not necessary to save something in V. Otherwise it returns 0 and leaves the corresponding values in V.

• Function **add**(float A [] [], float B [] [], float C [] [], int M, int N). This function adds two matrices of float-type numbers and leave the result in another. This function accepts 5 arguments:

- A: Matrix where the result of adding B and C will be stored.
- B: Represents the first matrix to add.
- C: Represents the second matrix to add.
- M: Number of rows of the three matrices.
- N: Number of columns of the three matrices.

The function returns -1 if M or N are negative or zero. Otherwise it returns 0.

Important aspects to consider

General rules

1) The delivery of the practice will be made through the authorized delivery points in Aula Global. Delivery via email is not allowed.

2) The delivery will be made within the period given in delivery points in Aula Global. It is possible that for a delivery point in Aula Global the deadline for a delivery at 24:00 ends 10 minutes before. Please check the support of Aula Global

.

3) Special attention will be paid to detect features copied between two practices. In case of finding common implementations in two practices (or similar contents in memory), both will obtain a grade of 0 (zero).

4) The parameter passing agreement described in class must be followed. Those functions that do not correctly follow the parameter passing agreement will also be rated with a 0 (zero).

5) The two exercises requested must be submitted.

6) Exercises that do not compile or that do not conform to the functionality and requirements will obtain a grade of 0 (zero).

7) A program not commented will get a grade of 0 (zero).

Report of the practice

1) The report (a single document) must contain at least the following sections:
- Cover page, where the authors appear (including full name, NIA, class group to which they belong and email address).
- Contents index.
- Contents requested in the different exercises (one section per exercise).
- Conclusions and problems encountered.

2) **The report length should not exceed 10 pages** (cover and index included).

3) Regarding the description of the programs requested:

- The report should describe the behavior of the programs, as well as the main design decisions. **The pseudocode corresponding to each of the exercises in this practice should be included.**

- The battery of tests (as defined in the following section) used to validate the functionality of the requested functions and the results obtained must be included. Higher

scores will be given to advanced tests, extreme cases, and in general those tests that guarantee the correct functioning of the practice in all cases.

- Avoid duplicate tests that evaluate the same program flows. The score in this section is not measured based on the number of tests, but the degree of coverage of the tests. Few tests that evaluate different cases are better than many tests that always evaluate the same case.

NOTE: DO NOT NEGLECT QUALITY OF THE REPORT OF YOUR PRACTICE. Approving the report is as essential to approve the practice, as the proper functioning of it. If when evaluating the report of your practice, it is considered that it does not reach a minimum quality, your practice will be suspended.

Definition of tests

For the definition of tests for each of the exercises, a table with the following format should be included in the report:

Input Data:	Test description:	Expected output:	Obtained output:

Practice delivery procedure

The delivery of practice 1 will be done electronically through Aula Global.

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

The deadline for delivery is **November 3rd, 2019 at 11:55 p.m**.

It is possible to deliver as many times as you want within the given deadline, the final registered version of your practice is the last one delivered. The assessment of the practice will be the assessment of the content of the latest submitted. Always check what you are delivering.

Delivery format: A single compressed file (zip format) must be delivered. The name of this file must have the following format: ec_p1_AAAAAAAAA_BBBBBBBBB.zip where A… A and B… B are the NIAs of the group members.

The zip file must contain only the following files:

• **exercise1.s**.

This file will only contain the code of the functions. It will not have a main subroutine or data segment. Therefore, the functions you deliver must work without relying on a specific segment of data.

• **exercise2.s**.

This file will only contain the code of the functions. It will not have a main subroutine or data segment. Therefore, the functions you deliver must work without relying on a specific segment of data.

• **report.pdf**

This file will contain the contents of the report and will have a PDF format (it is not worth renaming a text file or similar with a .pdf extension)

• **authors.txt**

This file will contain a line for each member of the group with the NIA and the group to which he/she belongs.

Practice evaluation

The evaluation of the practice will be divided into two parts:

- **Code (7 points)**
- **Report (3 points)**

The score for each exercise will be:

- **Exercise 1 (6 points)**
- **Exercise 2 (4 points)**

If an exercise is not delivered your score will be 0. Keep in mind that, in order to continue the process of continuous evaluation, the minimum grade obtained in each practice must be 2 out of 10 and the average of the two practices must be 4 out of 10.

NOTES:

1. If a serious misconception is detected in practice (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, do not respect the name of the requested files, deliver a .rar file, deliver the files in a directory, etc.) the note will be significantly reduced.

3. In case of finding common implementations in two practices (or similar contents in the report) it will be understood that the practice has been copied and both will obtain a grade of 0.

4. If code fragments obtained directly from the Internet are found, it will be understood that the content has been copied and the practice will have a rating of 0.