

ESTRUCTURA DE COMPUTADORES

GRADO EN INGENIERÍA INFORMÁTICA
DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y
ADMINISTRACIÓN DE EMPRESAS

uc3m | Universidad **Carlos III** de Madrid

Grupo de Arquitectura de Computadores

Práctica 1

Introducción al lenguaje ensamblador

Curso 2019/2020

Contenido

| | |
|----------------------------------------------|---|
| Objetivos de la práctica | 3 |
| Ejercicio 1 | 3 |
| Ejercicio 2 | 4 |
| Aspectos importantes a tener en cuenta | 5 |
| Normas generales | 5 |
| Memoria de la práctica | 5 |
| Definición de pruebas..... | 6 |
| Procedimiento de entrega de la práctica..... | 7 |
| Evaluación de la práctica | 8 |

Objetivos de la práctica

El objetivo de la práctica consiste en entender los conceptos relacionados con la programación en ensamblador. Para ello, se utilizará como base el ensamblador del MIPS32 y el simulador CREATOR, que está disponible en <https://creatorsim.github.io/creator/>.

Para familiarizarse con la programación en ensamblador y el simulador anterior, se recomienda resolver los ejercicios disponibles en Aula Global en el apartado correspondiente a la primera práctica.

La práctica consta de 2 ejercicios.

Ejercicio 1

El objetivo de este ejercicio es desarrollar el siguiente conjunto de funciones para trabajar con matrices de números enteros:

- Función **inicializar**(int A[], int M, int N). Inicializa todos los valores de la matriz A a cero. M representa el número de filas y N el número de columnas. La función devuelve -1 si M o N son negativos o cero. En caso contrario devuelve 0.
- Función **sumar**(int A[], int B[], int C[], int M, int N). Suma dos matrices y dejar el resultado en otra. Esta función acepta 5 argumentos:
 - A: Matriz donde se almacenará el resultado de sumar B y C.
 - B: representa la primera matriz a sumar.
 - C: representa la segunda matriz a sumar.
 - M: número de filas de las tres matrices.
 - N: número de columna de las tres matrices.

La función devuelve -1 si M o N son negativos o cero. En caso contrario devuelve 0.

- Función **extraerFila**(int A[], int B[], int M, int N, int j). Dada la matriz B de M filas y N columnas, copia en el vector A la fila j de la matriz. La función devuelve -1 en los siguientes casos:
 - M o N son negativos o alguno de ellos es cero.
 - j es menor que 0 o bien j es mayor que >M

En caso contrario devuelve 0.

- Función **masCeros**(int A[], int B[], int M, int N). Esta función determina cuál de las matrices, A o B (ambas de M filas y N columnas) tiene un mayor número de elementos igual a 0. La función devuelve -1 si M o N son negativos o alguno de ellos es cero, en caso contrario devuelve 0 si A tiene más ceros que B o 1 si B tiene más ceros que A. Si ambas tienen el mismo número de ceros devuelve 2. Para desarrollar esta función se proporciona como **material de apoyo** la siguiente función de biblioteca:
 - **calcular** (int A[], int M, int N, int valor). Esta función devuelve el número de elementos de la matriz A cuyo valor coincide con el pasado en el cuarto argumento (valor). La función devuelve -1 en caso de que M o N sean negativos o alguno de ellos sea cero. Es **obligatorio** invocar a esta función proporcionada como material de apoyo para lo cual habrá de seguirse el convenio de parámetros del MIPS. La biblioteca donde se encuentra esta función se denomina **apoyo.o**.

Para todas las funciones ha de seguirse correctamente el convenio de paso de parámetros del MIPS32. Cuando el valor que devuelve alguna de estas funciones es -1, la función no realizará ninguna otra tarea, solo se devuelve el valor -1.

Ejercicio 2

El objetivo de este ejercicio es desarrollar las dos siguientes funciones que trabajan con matrices de números de coma flotante de simple precisión (tipo `float` en C):

- Función **extraerValores** que acepta los siguientes argumentos en el orden indicado:
 - Argumento 1: Dirección de inicio de la matriz A de números en coma flotante (A).
 - Argumento 2: Número de filas de la matriz (M).
 - Argumento 3: Número de columnas de la matriz (N)
 - Argumento 4: Dirección de inicio de un vector V de 6 posiciones.

La función almacena en el vector V los siguientes valores:

- V[0] almacena el número de elementos de la matriz cuyo valor es 0.
- V[1] almacena el número de elementos de la matriz cuyo valor es más infinito.
- V[2] almacena el número de elementos de la matriz cuyo valor es menos infinito.
- V[3] almacena el número de elementos de la matriz con valor NaN.
- V[4] almacena el número de elementos de la matriz que se codifican como no normalizados.
- V[5] almacena el número de elementos de la matriz normalizados.

La función devuelve -1 si M o N son negativos o cero. En este caso no es necesario almacenar nada en V. En caso contrario devuelve -1 y deja en V los valores correspondientes indicados.

- Función **sumar**(float A[], float B[], float C[], int M, int N). Suma dos matrices de números de tipo `float` y dejar el resultado en otra. Esta función acepta 5 argumentos:
 - A: Matriz donde se almacenará el resultado de sumar B y C.
 - B: representa la primera matriz a sumar.
 - C: representa la segunda matriz a sumar.
 - M: número de filas de las tres matrices.
 - N: número de columna de las tres matrices.

La función devuelve -1 si M o N son negativos o cero. En caso contrario devuelve 0.

Aspectos importantes a tener en cuenta

Normas generales

- 1) La entrega de la práctica se realizará a través de los entregadores habilitados. No se permite la entrega a través de correo electrónico.
- 2) La entrega se realizará en el plazo dado por los entregadores. Es posible que para un entregador de Aula Global el fin del plazo para una entrega a las 24:00 termine 10 minutos antes. Revise el soporte de Aula Global.
- 3) Se prestará especial atención a detectar funcionalidades copiadas entre dos prácticas. En caso de encontrar implementaciones comunes en dos prácticas (o contenidos similares en la memoria), ambas obtendrán una calificación de 0 (cero).
- 4) Debe seguirse el convenio de paso de parámetros descrito en clase. Aquellas funciones que no sigan correctamente el convenio de paso de parámetros también serán calificadas con un 0 (cero).
- 5) Han de entregarse los dos ejercicios solicitados.
- 6) Los ejercicios que no compilen o que no se ajusten a la funcionalidad y requisitos planteados, obtendrán una calificación de 0 (cero).
- 7) Un programa no comentado obtendrá una calificación de 0 (cero).

Memoria de la práctica

- 1) La memoria (un único documento) tendrá que contener al menos los siguientes apartados:
 - Portada donde figuren los autores (incluyendo nombre completo, NIA, grupo de clase al que pertenecen y dirección de correo electrónico).
 - Índice de contenidos.
 - Contenidos pedidos en los distintos ejercicios (una sección por ejercicio).
 - Conclusiones y problemas encontrados.
- 2) **La longitud de la memoria no deberá superar las 10 páginas** (portada e índice incluidos).
- 3) Al respecto de la posible descripción de los programas pedidos:
 - La memoria debe describir el comportamiento de los programas, así como las principales decisiones de diseño. **Se deberá incluir el pseudocódigo correspondiente a cada uno de los ejercicios de esta práctica.**
 - Se ha de incluir la **batería de pruebas (como la definida en la siguiente sección)** utilizadas para validar la funcionalidad de las funciones solicitadas y los **resultados** obtenidos. Se dará mayor puntuación a pruebas avanzadas, casos extremos, y en general a aquellas pruebas que garanticen el correcto funcionamiento de la práctica en todos los casos.

- Evite pruebas duplicadas que evalúan los mismos flujos de programa. La puntuación de este apartado no se mide en función del número de pruebas, sino del grado de cobertura de las mismas. Es mejor pocas pruebas que evalúan diferentes casos a muchas pruebas que evalúan siempre el mismo caso.

NOTA: NO DESCUIDE LA CALIDAD DE LA MEMORIA DE SU PRÁCTICA.

Aprobar la memoria es tan imprescindible para aprobar la práctica, como el correcto funcionamiento de la misma. Si al evaluarse la memoria de su práctica, se considera que no alcanza el mínimo admisible, su práctica estará suspensa.

Definición de pruebas

Para la sección de definición y batería de pruebas de cada uno de los ejercicios que se debe incluir en la memoria una tabla que seguirá el siguiente formato:

| Datos a introducir: | Descripción de la prueba: | Resultado esperado: | Resultado obtenido: |
|--------------------------------|--------------------------------------|--------------------------------|--------------------------------|
| | | | |
| | | | |

Procedimiento de entrega de la práctica

La entrega de la práctica 1 se realizará de forma electrónica a través de Aula Global,

La práctica se podrá hacer en grupos de **dos alumnos**.

La fecha límite de entrega para ambos es el día **3 de noviembre de 2019 a las 23:55 horas**.

Es posible entregar tantas veces como quiera dentro del plazo dado, la única versión registrada de su práctica es la última entregada. La valoración de la práctica es la valoración del contenido de esta última entrega. Revise siempre lo que entregue.

Entregador: Se deberá entregar un único archivo comprimido en formato **zip** con el nombre `ec_p1_AAAAAAAAAA_BBBBBBBBBB.zip` donde A...A y B...B son los NIA de los integrantes del grupo.

El archivo **zip** debe contener solo los siguientes archivos:

- **ejercicio1.s.**
Este archivo solo contendrá el código de las funciones. No tendrá ni subrutina main ni segmento de datos. Por tanto, las funciones que entregue han de funcionar sin depender de un segmento de datos específico.
- **ejercicio2.s.**
Este archivo solo contendrá el código de las funciones. No tendrá ni subrutina main ni segmento de datos. Por tanto, las funciones que entregue han de funcionar sin depender de un segmento de datos específico.
- **memoria.pdf**
Este archivo contendrá el contenido de la memoria, y tendrá formato PDF (no vale renombrar un fichero de texto o similar con extensión .pdf)
- **autores.txt**
Este archivo contendrá una línea para cada integrante del grupo con el NIA y el grupo al que pertenece.

Evaluación de la práctica

La evaluación de la práctica se va a dividir en dos partes:

- **Código (7 puntos)**
- **Memoria (3 puntos)**

La puntuación de cada ejercicio será:

- Ejercicio 1 (6 *puntos*)
- Ejercicio 2 (4 *puntos*)

Si un ejercicio no se entrega su puntuación será 0. Tenga en cuenta que, para seguir el proceso de evaluación continua, la nota mínima obtenida en cada práctica debe ser de 2 sobre 10 y la media de las dos prácticas ha de ser 4 sobre 10.

NOTAS:

1. **Si se detecta un error de concepto grave en la práctica (en cualquier apartado de cualquier ejercicio), la valoración global de toda la práctica será de cero puntos (0 puntos).**
2. **Si no respeta el formato de entrega (por ejemplo no respetar el nombre de los ficheros pedidos, entregar un archivo .rar, entregar los ficheros en un directorio, etc.) la nota quedará reducida significativamente.**
3. **En caso de encontrar implementaciones comunes en dos prácticas (o contenidos similares en la memoria) se entenderá que se ha copiado la práctica y ambas obtendrán una calificación de 0.**
4. **En caso de encontrarse fragmentos de código obtenidos directamente de Internet se entenderá que se ha copiado dicho contenido y la práctica tendrá una calificación de 0.**