Práctica 1 sesión 1 Arquitecturas avanzadas de procesadores

Resumen

En esta sesión de prácticas queremos convertir un código en C++ a ensamblador de MIPS.

Se dan varias opciones:

- Un almacenaje de una matriz de 16 x 16 en memoria de forma lineal, siguiendo un algoritmo de 'Column-Major'.
- Un almacenaje de los 16 primeros elementos de la serie de Fibonacci.

1. Presentación de los algoritmos

1.1. Column-Major

el algoritmo **Column-Major** sirve para seguir un criterio de almacenaje de una matriz multi-dimensional en memoria continua. Es decir, almacenamos varias dimensiones en una sola en memoria.

Podéis ver una descripción detallada de lo que hace en la misma wikipedia.

El código fuente en C++ a convertir en MPIS sería el siguiente:

```
int size = 16;
int[size][size] data;
int value = 0;
for (int col = 0; col < size; col++)
{
    for (int row = 0; row < size; row++)
    {
        data[row][col] = value;
        value++;
    }
}
```

1.2. Fibonacci

La sucesión de **Fibonacci** es una sucesión de números donde un valor es igual a la suma de los dos anteriores, siendo los dos primeros valores de la serie iguales a 1.

Podéis saber más de la sucesión en la wikipedia.

El código fuente en C++ a convertir en MPIS sería el siguiente:

```
int size = 16;
int data[size];
data[0] = 1;
data[1] = 1;
for (int i = 2; i < size; ++i)
{
    data[i] = data[i-1] + data[i-2];
}
```

2. ¿Qué tenéis que desarrollar?

Tenéis que desarrollar el código correspondiente en ensamblador de MIPS, que sea ejecutable en el simulador MARS, y **que imprima al final el resultado de los datos de la memoria**.

Como realizar el código no tiene mérito :-D , pues con una consulta rápida en internet o ChatGPT se puede conseguir, **dispondréis en la sesión 2 de las prácticas, de entre 5 y 7 minutos para realizar una defensa verbal del ejercicio**, donde se valorará:

- Control de como se estructura un código ensamblador en MIPS, especial atención a las directivas! *directives*
- Control de las instrucciones de ensamblador, por qué se usan y qué hacen.
- Control de la zona de memoria, donde está alojada, qué dirección y valores tienen.
- Control de las llamadas de *syscall* para imprimir valores.