

## Pauta Control 4

Profesor: Yadran Eterovic

### 1. Direccionamiento (3 puntos)

Con respecto a cada uno de los siguientes modos de direccionamiento: **inmediato**, **directo**, **por registro**, **indirecto por registro**, e **indexado**, responde lo siguiente: Suponiendo que hay un arreglo de  $n$  números enteros de 4 *bytes* en memoria, ¿qué dificultades y/o qué facilidades entrega cada modo de direccionamiento para poder sumar los  $n$  números? Sé preciso en tus respuestas. Considera que cuentas con las instrucciones del lenguaje similares a la del IA-32 visto en clases.

#### Solución:

**D. inmediato:** El argumento de la instrucción no es la dirección del dato sino el dato propiamente tal. No hay cómo hacer un código general que sume  $n$  números: es necesario escribir  $n$  instrucciones SUM, cada una acompañada por uno de los  $n$  números. Dos ejecuciones del programa suman los mismos  $n$  números.

**D. directo:** El argumento de la instrucción es la dirección que el dato ocupa en memoria; pero esta dirección es fija, por lo que hay que conocerla de antemano y especificarla explícitamente para cada dato. También son necesarias  $n$  instrucciones SUM, cada una acompañada por la dirección en memoria de uno de los números. Dos ejecuciones del programa pueden sumar dos conjuntos distintos de  $n$  números.

**D. por registro:** El argumento de la instrucción es el registro en que está el dato. Es necesario ir cargando en el registro cada uno de los  $n$  números, para esto se necesita otro tipo de direccionamiento para instrucciones LOAD, en cuyo caso (tal vez) mejor usamos este otro direccionamiento para la suma.

**D. indirecto por registro:** El argumento de la instrucción es el registro en el que está la dirección de memoria del dato; puede hacer referencia a diferentes direcciones de memoria en cada ejecución de la instrucción (guardando una dirección diferente en el registro cada vez), lo que permite recorrer fácilmente el arreglo de números.

**D. indexado:** El argumento de la instrucción está compuesto de dos partes: una dirección de memoria constante (p.ej., la del primer dato del arreglo) y un registro; la dirección especificada por el argumento es la suma de la dirección constante más el valor almacenado en el registro, que puede variar en cada ejecución de la instrucción, permitiendo recorrer fácilmente el arreglo de números.

#### Asignacion de puntaje:

- 0,6 puntos por cada argumento correcto.

## 2. IA-32 / MOVs

Se ejecuto el siguiente código (la sección .data se ejecutó antes) en un computador Von Neumann con un procesador Core i7 (**NOTA:** las palabras en esta arquitectura ocupan 2 bytes en memoria). Calcule el estado final de las primeras 17 direcciones de la memoria del computador. **NO** puede asumir el contenido de una posición de memoria desconocida. Luego de la ejecución de la sección data, las primeras 17 direcciones de la memoria quedaron así:

a: 00000101
00000000
b: 00000100
00000000
res: 00000000
00000000
size: 00000101
arr: 00000010
00000011
00000100
00000101
00000110
are: 00000000
00000000
00000000
00000000
00000000

**Solución:**

Data
a: 00000101
00000000
b: 00000100
00000000
res: 01110001
00000010
size: 00000101
arr: 00000010
00000011
00000100
00000101
00000110
are: 00011011
00011100
00011101
00011110
00011111

**Asignacion de puntaje:**

- 1 pto por **res.**
- 2 ptos por todo **are.**
- -0,5 ptos por no expresar o expresar de forma incorrecta el binario.