



Práctica 4

Objetivo

Seleccionar el modo de direccionamiento adecuado, para manejo de memoria en aplicaciones de sistemas basados en microprocesador, mediante el análisis de los diversos modos de direccionamiento existentes, de forma responsable y eficiente.

Desarrollo

1. Capturar dos números y guardarlos en sus respectivas variables en memoria no inicializada.
2. Implementar un procedimiento que por medio de sumas consecutivas utilizando la instrucción loop explicada en clase genere el mismo funcionamiento que una multiplicación.
 - a. Ejemplo: $3 \times 4 = 12$
 - i. $4 + 4 + 4 = 12$.
 - ii. El primer número indica el número de veces a sumar el segundo número.
3. Implementar un procedimiento que por medio de restas consecutivas utilizando la instrucción jmp explicada en clase genere el mismo funcionamiento que una división de enteros.
 - a. Ejemplo: $12 / 3 = 4$
 - i. $12 - 3 = 9$; $9 - 3 = 6$; $6 - 3 = 3$; $3 - 3 = 0$.
 - ii. El primer número es el resultado de la anterior función, el segundo número es el que se le restara al primero hasta dar con el resultado de 0.
4. Implementar un procedimiento que muestre los números desde el 0 al 100 utilizando la subrutina printHex que se les entregó en moodle, únicamente utilizando las instrucciones vistas hasta el momento.
 - a. Ejemplo: 0, 1, 2, 3, ... , 100

5. Implementar un procedimiento que muestre los números pares desde el 0 al 100, utilizando la instrucción `cmp` y el salto condicional `je`.

Instrucciones vistas en clase y permitidas para esta práctica.

1. `add`
2. `sub`
3. `mov`
4. `jmp`
5. `cmp`
6. `je`
7. `loop`
8. `getch`
9. `getche`
10. `puts`
11. `putchar`

Obligatoriamente de cada paso debe tener las capturas del código y de la terminal.

Conclusiones

Dificultades