

Organización del Procesador

año 2016

Guía de ejercicios prácticos 3

Assembler

Requerimientos: Esta práctica está basada en la utilización de lenguaje Assembly, en particular NASM. Para obtenerlo puede descargarlo de su página oficial para diferentes arquitecturas: <http://www.nasm.us>. Además, para simplificar la generación de archivos ejecutables y la Entrada/Salida, deberá descargar los archivos *driver.c*, *asm_io.asm* y *asm_io.inc* de la página de su autor <http://pacman128.github.io/pcasm/> o del aula virtual.

1. Escribir las instrucciones que representa cada inciso en el lenguaje NASM e indicar el modo de direccionamiento de cada una de ellas:
 - a Incrementar el valor del registro EAX.
 - b Cargar el registro EBX con el decimal 18 utilizando una etiqueta *label* binaria.
 - c Sumar al registro EAX el valor 200 (expresado en decimal).
 - d Cargar el registro AX (parte baja del registro EAX) con el contenido de la celda de memoria cuya dirección está almacenada en el registro EBX.
 - e Multiplicar el valor 52 almacenado en una etiqueta en formato hexadecimal, con el valor cuya dirección de memoria está almacenada en el registro EBX.
 - f Sumar al registro EAX el contenido de la celda de memoria cuya dirección es calculada incrementando el registro EBX en 4 unidades.
2. Indicar con qué valores queda representado el registro extendido EAX de la arquitectura 80386, según las operaciones aplicadas:

(a)

```
1 L1 db 0
2 mov eax, L1
```

(b)

```
1 L6 db 18h
2 mov eax, [L6]
3 add eax, [L6]
```

(c)




```
1 L1 dd 1Ah
2 mov AL, [L1]
```

(d)

```
1 mov ax, 5h
2 shl ax, 1
3 shr ax, 1
```

(e)

```
1 mov ax, 0110b
2 rol ax, 1
3 ror ax, 2
```

3. ★ Determine cómo simularía los esquemas de las estructuras de control (comparaciones, saltos condicionales y ciclos) en assembler. 
4. ¿Qué diferencia existe entre el segmento de programa *.data* y el *.bss*? Dé un ejemplo que muestre la utilidad de cada uno. 
5. Escriba un programa en assembler que utilice tres arreglos definidos en el segmento de datos, pero cada uno con un tamaño diferente de elementos (byte, word, dword). Determine cuántos bytes se necesitan para obtener cada elemento de estos arreglos. Realice un programa en assembler que muestre los elementos de los tres arreglos. 

6. Defina cómo se puede determinar si un número es par utilizando operadores a nivel de bits. Luego, construya un programa assembler que dado un arreglo de 10 enteros de 16 bits (words), definido en el segmento *.data*, sume los números pares y muestre el resultado por pantalla. **Nota:** revise el capítulo 5 del libro de Paul A. Carter para conocer en detalle el tratamiento de arreglos en NASM.
7. Construya un programa assembler que calcule el factorial de un número dado. (resuelva de manera iterativa).
8. Construya un programa assembler que dado un número entero positivo N , calcule la serie de fibonacci hasta N y la muestre por pantalla (resuelva de manera iterativa). Ejemplo: La serie de fibonacci para $n = 6$ es 11235.