

EJERCICIOS TEMA 9

1. Ejercicios en ensamblador IA-32

1) Indica qué registros cambian de contenido tras ejecutar la instrucción *push ebx*. ¿Cuántas posiciones de memoria cambian su valor tras ejecutar la instrucción anterior?

2) Indica qué registros cambian de contenido tras ejecutar la instrucción *pop ebx*. ¿Cuántas posiciones de memoria cambian su valor tras ejecutar la instrucción anterior?

3) Si sumamos o restamos 4 al registro de pila mediante instrucciones en ensamblador, ¿qué sucede?

4) Realiza las siguientes operaciones:

- 101010010110101010101010101010101011b a hexadecimal
- $0xAB \& 0x03$
- $0xAB | 0x03$
- $0xAB \wedge 0x03$
- $\sim 0xAB$
- $0xAB \gg 2$
- $0xAB \ll 2$

5) Escribir un programa en C que realice las operaciones del Ejercicio 4 (salvo la primera), mostrando el resultado en pantalla. Comprobar los resultados obtenidos con el ejercicio 4.

- 6) Escribir un programa en C que convierta un número binario, introducido por teclado y almacenado en una cadena de caracteres, a representación hexadecimal, decimal y octal. El resultado de dicha conversión se mostrará por pantalla.
- 7) Escribir un programa en ensamblador que sume los números almacenados en dos variables enteras declaradas en C, almacenando el resultado en el registro *eax*.
- 8) Escribir un programa en ensamblador que sume los números almacenados en dos variables enteras declaradas en C, almacenando el resultado en la pila.
- 9) Escribir un programa en ensamblador que sume los números 2 y -3 almacenados en dos variables enteras declaradas en C, dejando el resultado en el registro *eax*. ¿Qué sucede con los números negativos? ¿Cómo opera la máquina con ellos? Justifica la respuesta.
- 10) Escribir un programa en ensamblador que sume tres números almacenados en variables enteras declaradas en C, almacenando el resultado en la pila.
- 11) Escribir un programa en ensamblador que intercambie el valor de dos variables enteras declaradas en C.
- 12) Escribir un programa en ensamblador que fije a uno el tercer byte de una variable entera declarada en C.
- 13) Escribir un programa en ensamblador que fije a 0 los 4 bits de menor peso de una variable char declarada en C sin modificar los restantes bits.
- 14) Escribir una subrutina en ensamblador que permita realizar el complemento a 2 de un valor positivo almacenado en la cima de la pila utilizando operaciones lógicas. Dicha subrutina devolverá el resultado empleando también la pila de memoria.
- 15) Escribir una subrutina en ensamblador que fije a 1 el bit de mayor peso y a 0 el bit de menor peso de un valor almacenado en un registro. Dicha subrutina devolverá el resultado empleando el mismo registro empleado para suministrar el dato.
- 16) Escriba una subrutina en ensamblador que devuelva, en la cima de la pila, el mayor de dos números enteros. Los números serán proporcionados apilados en la pila de memoria.
- 17) Escriba una subrutina en ensamblador que devuelva, en la cima de la pila, la suma de dos números enteros. Los números serán proporcionados apilados en la pila de memoria.

- 18) Escribe una subrutina en ensamblador que devuelva, en la cima de la pila, la operación AND de dos números enteros. Los números serán proporcionados apilados en la pila de memoria.
- 19) Escribe una subrutina en ensamblador que devuelva, en la cima de la pila, todos los bits a uno de un valor apilado antes de entrar en dicha subrutina.
- 20) Escribe una subrutina en ensamblador que devuelva, en la cima de la pila, todos los bits a uno de un valor, si los valores almacenados en dos variables enteras de C son iguales.
- 21) Realizar la suma binaria de tres números de 8 bits cuyo resultado pueda superar el valor 0xFF. Esto significa que el resultado de la operación puede requerir más de 8 bits para su almacenamiento/representación. ¿Qué sucede si operamos con valores de 8 bits en ensamblador? Justifica la respuesta. Nota. Emplear el depurador del entorno de programación para comprobar la respuesta.
- 22) Realizar un programa en ensamblador que inicialice un contador a 0x06 y se vaya decrementando de uno en uno hasta llegar a 0.
- 23) Realizar un programa en ensamblador que inicialice el registro eax con el valor solicitado al usuario en C. Cuando dicho valor sea 0 se sumarán dos números almacenados en el registro ecx y el registro edx, y si no es 0 se sumará el contenido de los registros ebx y edx. Inicializar correctamente los registros para que la suma realizada sea válida.
- 24) Realizar un programa en ensamblador que sume ecx y ebx, si el contenido del registro ecx es 0. En caso contrario, el programa calculará la diferencia entre ecx y ebx.
- 25) Realizar un programa en ensamblador que inicialice un contador a 1. Dicho contador se irá incrementando de uno en uno hasta 0x08.
- 26) Realizar un programa en ensamblador que cree un bucle infinito si el valor del registro eax es igual a 0. En caso contrario, almacenará en la cima de la pila el valor de dicho registro.
- 27) Realizar un programa en ensamblador que calcule cuál es el mayor de tres números almacenados en registros.
- 28) Realizar un programa en ensamblador que calcule cuál es el menor de tres números almacenados en registros.

29) Escribe una subrutina en ensamblador que realice la suma de un determinado número de valores enteros de 32 bits. Dichos valores estarán almacenados en la pila de forma consecutiva, en el momento en el que se produce la llamada a la subrutina. Además, el número de valores a sumar será indicado por un valor también almacenado en la pila; concretamente, dicho valor se encontrará en la cima de la pila (inmediatamente a continuación del primer valor a sumar) antes de la invocación de la subrutina. El resultado de la operación deberá dejarse en la cima de la pila antes de finalizar la subrutina.

30) Escribe una rutina en ensamblador que tome de la cima de la pila dos números naturales, calcule el producto de ambos usando la operación suma, y deje el resultado en la cima de la pila.