

Nombre: Mónica Elizabeth Alba Gonzalez Clave única: 160502 Calif: 8.20

Cada una de las cinco preguntas vale 2.0 pts.

CONTESTAR CON TINTA NEGRA.

Duración 1:50 hs.

No usará ni celular, ni calculadora, ni tablet, así como tampoco computadora, como apoyo del examen.

1.- Microcomputadora IA-32.

A) ¿Qué situación, dentro del hardware de la computadora, originó la agregación de la memoria caché y cuáles beneficios trajo?

B) La memoria caché, desde el punto de vista de la velocidad de acceso, ¿cómo es con respecto a los registros del CPU por un lado, y por otro lado con respecto a la memoria central (xRAM)?

C) Ya respondió en A) los beneficios de la memoria caché, ahora ¿Qué problemática puede presentarse entre el contenido de la memoria caché y el contenido de la memoria central, cuando el CPU ejecuta un programa?

2.- Capacidades de un entero.

A) Suponga que tienen varios registros de 13 bits cada uno, en un CPU marca MegaQuack. Escriba dos ejemplos en hexadecimal de valores que puedan ser asignados en estos registros.

B) ¿Cuál es el rango de enteros sin signo a 13 bits? Muestre los cálculos de cómo llegó a estos valores extremos del rango. Recuerde que $2^{11} = 2048_{10}$.

C) ¿Cuál es el rango de enteros con signo a 13 bits? Muestre los cálculos de cómo llegó a estos valores extremos del rango.

3.- Pedazo de programa en Lenguaje ensamblador. Ensamble.

A) Elabore tanto la Tabla de Etiquetas como el correspondiente Mapa de Memoria Central del siguiente segmento .DATA que empieza en la dirección offset 0000h (cero).

.DATA

one BYTE 1, 2, 3, 4

two WORD 2 DUP(5), 3

WORD 67h, 78h

three

four DWORD 12345678h, 34567890h

five DWORD three

six BYTE "CDEFG", 0

seven

byt

8 bits

word

16 bits

d word

32 bits

B) Escriba, para cada instrucción, primero si es válida y en caso de ser válida muestre el resultado en el registro. En caso de ser inválida explique porqué.

.CODE

MOVSX EAX, two+4

MOV CX, four+2

MOV EBX, OFFSET three

Todos los valores resultantes que usted responda deben estar justificados.

>>> VUELTA

4.- Pedazo de programa en Lenguaje Ensamblador. Tipos y formatos.

La siguiente asignación con expresión aritmética en algún lenguaje de alto nivel tradúzcala a lenguaje ensamblador: $R = -(C + 8) + F * 6 - 4$;

Con el siguiente

.DATA

R SDWORD ?
C SDWORD 0Ch
F SDWORD 0Ah

.CODE

Agregue en .CODE las instrucciones en ensamblador para implementar la asignación expresión aritmética.

Al final, también escriba lo que quedaría en el lugar de memoria R, en hexadecimal, tal y como luciría en la memoria en formato Little-endian.

32 bits → 4 bytes

5.- Pedazo de programa en lenguaje ensamblador.

.DATA

mmsg BYTE "OSAC",0

.CODE

mov EDX, OFFSET mmsg

call WriteString ; imprime el contenido de mmsg

A) Agregue, a lo anterior, las instrucciones en lenguaje ensamblador para que dentro de la cadena mmsg en lugar de "OSAC" diga "COSA". Ya que usted no puede asignar directamente caracteres (por esta vez), p.e. mov mmsg, "C", deberá hacer las modificaciones moviendo lo caracteres que ya tiene almacenados en memoria. Realice esta parte de manera óptima, esto quiere decir con la menor cantidad de instrucciones y operando bien.

B) También agregue las instrucciones pertinentes para mandar imprimir nuevamente mmsg, claro ahora deberá aparecer en desplegado en la pantalla "COSA".

>>> VUELTA

