Start: 2021-10-26, Tuesday, 12:45:35

End: 2021-10-26, Tuesday, 13:14:00

Questions: 17

Valid answers: correct($p_i$=1): 8; wrong(-1≤$p_i$<0): 7; wrong($p_i$=0): 0; wrong(0<$p_i$<1): 0; blank($p_i$=0): 2

Score: **5,67/17**

Grade: **0,67/2,00**

**1** Unique choice

[T2.2.1]
¿Cuál de las siguientes instrucciones máquina copia en EAX el entero almacenado en la posición de memoria cuya dirección efectiva es el resultado de la operación EDX*4 + EBX?

User Teachers

- a) leal (%ebx, %edx, 4), %eax
- b) movl 4(%edx, %edx), %eax
- c) leal 4(%edx, %edx), %eax
- • d) movl (%ebx, %edx, 4), %eax

Score: **1,00**

**2** Unique choice

Si RCX vale 0, la instrucción adc $-1,%rcx

User Teachers

- a) Pone CF=1 (independientemente de lo que valiera antes)
- b) Cambia CF (si valía 0 cambiará a 1, si valía 1 cambiará a 0)
- • c) Pone CF=0 (independientemente de lo que valiera antes)
- d) No cambia CF (si valía 0 permanecerá a 0, si valía 1 permanecerá a 1)

Score: **-0,33**

**3** Unique choice

Si el registro RAX contiene X, la sentencia en C
x &= 0x1;
se traducirá a ensamblador como:

User Teachers

- • a) andq $1, %rax
- b) orq $0x1, %rax
- c) sarq %rax
- d) shrq %rax

Score: **1,00**

**4** Unique choice

[T2.1.4]
Cuál de las instrucciones máquina siguientes es incorrecta en x86-64:

User Teachers

- a) addq $1, %rcx
- b) testl %edx, %edx
- c) movl %r8, %eax
- d) movl (%rdi,%rcx,4), %edx

Score: **0,00**

En la práctica "media" se pide sumar una lista de 16 enteros CON signo de 32

**5** Unique choice

En la práctica "media" se pide sumar una lista de 16 enteros CON signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el mayor valor negativo (menor en valor absoluto) que repetido en toda la lista de 16 enteros causaría overflow con 32bits?
PISTA: Sumar un número 16 veces == multiplicarlo por 16 == desplazarlo 4 bits a la izquierda

User Teachers

- a) 0xf000 0000
- b) 0xffff ffff
- c) 0xfc00 0000
- d) 0xf7ff ffff

Score: **-0,33**

**6** Unique choice

¿Cuál de los siguientes registros tiene que ser salvaguardado (si va a modificarse) dentro de una subrutina según la convención x86-64?

User Teachers

- a) rax
- b) rbx
- c) rcx
- d) rdx

Score: **1,00**

**7** Unique choice

¿Qué valor contendrá %edx tras ejecutar las siguientes instrucciones?

```
xor %eax, %eax
sub $1, %eax
cltd
idiv %eax
```

User Teachers

- a) no puede saberse con los datos del enunciado
- b) -1
- c) 1
- d) 0

Score: **1,00**

**8** Unique choice

[T2.1.2]
En X86-64, el registro contador de programa se denomina:

User Teachers

- a) RIP
- b) EIP
- c) IP
- d) R15

Score: **1,00**

**9** Unique choice

En la práctica "media" un estudiante usa el siguiente bucle para acumular la suma en EBP:EDI antes de calcular la media y el resto

```
bucle:
        mov (%ebx,%esi,4), %eax
        cltd
        add %eax, %edi
        adc %edx, %ebp
        jnc nocarry
        inc %edx
nocarry:
        inc %esi
```

```
cmp  %esi,%ecx
jne bucle
```

Estando bien programado todo lo demás, este código...

User Teachers

a) fallaría con lista: .int -1,-2,-4,-8
b) fallaría con lista: .int 0,1,2,3
c) no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos
d) produce siempre el resultado correcto

Score: **0,00**

**10**
Unique choice

En la práctica "media" se programa la suma de una lista de 16 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x0400 0000, ¿en qué se diferencian los resultados de ambos programas?

User Teachers

a) en uno ocupa 32 bits, en otro 64 bits
b) en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no
• c) no se diferencian
d) en uno se interpreta como negativo, en otro como positivo

Score: **1,00**

**11**
Unique choice

Dado el siguiente fragmento de programa en ensamblador:

```
.section .data
lista: .int 1,2,0x10,3
longlista: .int (.-lista)/4
resultado: .quad 0

.section .text
main: .global main
xor %edx,%edx
mov $-23,%eax
cltd
mov $5,%ebx

idiv %ebx
...
```

El valor de %RAX después de la división es:

User Teachers

a) Ninguna de las soluciones es correcta
• b) 0xFFFFFFFC
c) 0xFFFFFFFF
d) 0x00000004

Score: **1,00**

**12**
Unique choice

Dado el siguiente fragmento de programa:

```
.section .data
lista: .int 1,2,0x10,3
longlista: .int .-lista
resultado: .quad 0

.section .text
```

main: .global main

xor %edx,%edx
mov $-17,%eax
cltd
mov longlista,%ebx

idiv %ebx

El valor de %RDX después de la división es:

User Teachers

- a) Ninguna de las soluciones es correcta
- b) 0x00000001
- c) 0xFFFFFFFF
- d) 0x0000000F

Score: **-0,33**

**13**

Unique choice

Dado el siguiente fragmento de programa:

.section .data
lista: .int 1,2,0x10,3
longlista: .int .-lista
resultado: .quad 0

.section .text
main: .global main

xor %edx,%edx
mov $-17,%eax
cltd
mov longlista,%ebx

idiv %ebx

El valor de %RAX después de la división es:

User Teachers

- a) 0xFFFFFFFC
- b) 0x0000000F
- c) 0x00000004
- d) Ninguna de las soluciones es correcta

Score: **1,00**

**14**

Unique choice

Dado el siguiente fragmento de programa:

.section .data
lista: .int 1,2,0x10,3,-3
longlista: .int .-lista
resultado: .quad 0

.section .text
main: .global main

xor %edx,%edx
mov $-12,%eax
cltd
mov longlista,%ebx

idiv %ebx

El valor de %RBX después de la división es:
User Teachers

- a) 0x00000005
- b) 0x00000014
- c) Ninguna de las soluciones es correcta
- d) 0xFFFFFFFF

Score: **-0,33**

**15**
Unique choice

Dado el siguiente fragmento de programa:

```
.section .data
lista: .int 2,-2,0x10,3,-3
resultado: .quad 0

.section .text
main: .global main

xor %rcx,%rcx
inc %cl
inc %cl
shl %cl,%rcx
mov lista,%ebx
lea (%rbx,%rcx,2),%rdx
```

El valor de %RBX despues de la operacion LEA es:
User Teachers

- a) 0x00000004
- b) 0x00000012
- c) 0x00000002
- d) Ninguna de las soluciones es correcta

Score: **-0,33**

**16**
Unique choice

Dado el siguiente fragmento de programa:

```
.section .data
lista: .int 1,2,0x10,3
longlista: .int .-lista
resultado: .quad 0

.section .text
main: .global main

xor %edx,%edx
mov $-17,%eax
cltd
mov longlista,%ebx

idiv %ebx
```

El valor de %RBX después de la división es:
User Teachers

- a) 0x0000000F
- b) 0x00000010
- c) 0x00000004
- d) Ninguna de las soluciones es correcta

Score: **-0,33**

**17**
Unique choice

Dado el siguiente fragmento de programa:

```
.section .data
lista: .int 1,2,0x10,3
longlista: .int .-lista
resultado: .quad 0

.section .text
main: .global main

xor %edx,%edx
mov $15,%eax
cltd
mov longlista,%ebx
idiv %ebx
```

El valor de %RDX despues de la division es:

User Teachers

- a) Ninguna de las soluciones es correcta
- b) 0x00000000
- c) 0xFFFFFFFF
- d) 0x0000000F

Score: **-0,33**