

1

[P2.2]

Si ECX vale 0, la instrucción `adc $0,%ecx`

Elección única

Usuario Profesores

a) Cambia CF

b) Pone CF=0

c) Pone CF=1

d) No cambia CF

Puntuación: 1,00

2

[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`?

Elección única

Usuario Profesores

a) `leal 4(%edx,%edx),%eax`

b) `movl (%ebx,%edx,4),%eax`

c) `leal (%ebx,%edx,4),%eax`

d) `movl 4(%edx,%edx),%eax`

Puntuación: 1,00

3

[P2T]

Dada la siguiente definición de datos:

Elección única

```
lista: .int 0x10000000, 0x50000000,
        0x10000000, 0x20000000
longlista: .int (.-lista)/4
resultado: .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n0"
```

y suponiendo que hemos llamado a una función `suma` que devuelve un número de 64 bits en la pareja `EDX:EAX`, las instrucciones que copian ese número en resultado son:

Usuario Profesores

a) `movl (%eax), resultado`

b) `movl (%edx), resultado+4`

c) `movl %eax, resultado`

d) `movl %eax, resultado+4`e) `movl %eax, resultado+4`f) `movl %edx, resultado+4`g) `movl %eax, resultado+4`h) `movl %edx, resultado`i) `movl (%eax), resultado+4`j) `movl (%edx), resultado+4`

Puntuación: -0,33

4

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

Elección única

```
bucle:
    mov (%ebx,%esi,4), %eax
    cld
    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...

Usuario Profesores

a) produce siempre el resultado correcto

b) fallaría con `lista: .int -1,-2,-4,-8`

c) fallaría con `lista: .int 0,1,2,3`

d) no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

Puntuación: 0,00

5

[P2T]

Dada la siguiente definición de datos:

Elección única

```
lista: .int 0x10000000, 0x50000000,
        0x10000000, 0x20000000
longlista: .int (.-lista)/4
resultado: .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n0"
```

La instrucción para copiar la dirección de memoria donde comienza `lista` en el registro `EBX` es:

Usuario Profesores

a) `movl lista,%ebx`

b) `movl (lista),%ebx`

c) `movl $lista,(%ebx)`

d) `movl $lista,%ebx`

Puntuación: 1,00

6

[T2.1.2]

En X86-64, el registro contador de programa se denomina:

Elección única

Usuario Profesores

a) RIP

b) EIP

c) IP

d) R15

Puntuación: 1,00

7

Elección única

[P2T]

Dada la siguiente definición de datos:

```

lista: .int 0x10000000, 0x50000000,
      0x10000000, 0x20000000
longlista: .int (-lista)/4
resultado: .quad 0x123456789ABCDEF
formato: .ascii "suma=%llu=%llx hex\n0"

```

la instrucción `movl longlista, %ecx` copia el siguiente valor:

Usuario Profesores

- ☐ a) 4
- ☐ b) 8
- ☐ c) 16
- ☐ d) 32

Puntuación: 1,00

8

Elección única

[P2T]

Tras ejecutar las tres instrucciones que se muestran desensambladas a continuación, el registro EAX toma el valor

```

08048074 <_start>:
8048074: be 74 80 04 08 mov $_start, %esi
8048079: 46             inc %esi
804807a: 8b 06          mov (%esi), %eax

```

Usuario Profesores

- ☐ a) 0x08048075
- ☐ b) 0x08048074
- ☐ c) 0x08048079
- ☐ d) 0x0804807a

Puntuación: -0,33

9

Elección única

[P2A2]

¿Cuál de los siguientes grupos de instrucciones IA32 sólo modifican los indicadores de estado sin almacenar el resultado de la operación?

Usuario Profesores

- ☐ a) CMP, TEST
- ☐ b) ADC, SBB
- ☐ c) IMUL, IDIV
- ☐ d) AND, OR, XOR

Puntuación: 1,00

10

Elección única

[T2.1.4]

Cuál de las instrucciones máquina siguientes es incorrecta en x86-64:

Usuario Profesores

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

Puntuación: -0,33

→ Porque r8 es de 64 bits y eax de 32

11

Elección única

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
cld
mov longlista,%ebx
idiv %ebx

```

El valor de %RDX después de la división es:

Usuario Profesores

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

Puntuación: -0,33

Handwritten notes and calculations:

- `edx = 0`
- Division:  $-17 \overline{) 16}$  with a remainder of  $-1$ .
- Result: `Resto` (Remainder) is `0xFFFFFFFF`.

## 12 Dado el siguiente fragmento de programa:

Elección única

```
.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
cld
mov longlista,%ebx
```

```
idiv %ebx
```

El valor de %RBX después de la división es:

Usuario Profesores

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

Puntuación: -0,33

## 13 Dado el siguiente fragmento de programa:

Elección única

```
.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 %RCX después de la operación LEA es:

Usuario Profesores

- ☒ a) Ninguna de las soluciones es correcta
- ☐ b) 0x00000004
- ☐ c) 0x00000002
- ☐ d) 0x00000008

Puntuación: -0,33

14 Dado el siguiente fragmento de programa:

Elección única

```
.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 %RDX después de la operación LEA es:

Usuario Profesores

- ☐ a) 0x00000010
- ☐ b) 0x00000004
- ☐ c) Ninguna de las soluciones es correcta
- ☐ d) 0x00000012

Puntuación: -0,33

?

15 Dado el siguiente fragmento de programa en ensamblador:

Elección única

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

```
.section .text
main: .global main
```

```
xor %edx,%edx
mov $-35,%eax
cld
mov $7,%ebx
```

```
idiv %ebx
```

```
...
```

Al finalizar la ejecución de la instrucción CLTD, los valores de los registros RDX y RAX son:

Usuario Profesores

- ☐ a) Ninguna de las soluciones es correcta
- ☐ b) RDX=0xFFFFFFFF RAX=0xFFFFFFFF23
- ☐ c) RDX=0xFFFFFFFF RAX=0x00000023
- ☒ d) RDX=0xFFFFFFFF RAX=0xFFFFFFFFDD

Puntuación: 1,00

$edx = 0$   $-35 \Rightarrow 10100011 \rightarrow 11011100 \rightarrow \underbrace{11011101}_0 \underbrace{11011101}_0$

16 Dado el siguiente fragmento de programa:

Elección única

```
.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 %RDX después de la operación LEA es:

Usuario Profesores

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

Puntuación: -0,33

$rcx = 0$

17 Dado el siguiente fragmento de programa:

Elección única

```
.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
cld
mov longlista,%ebx
idiv %ebx
```

El valor de %RDX después de la división es:

Usuario Profesores

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

Puntuación: -0,33

$16$   
 $edx = 0$   
 $eax = 15$   
 $15 \overline{) 16}$   
 $15 \ 0$   
 $\underline{15} \ 0$   
 $\rightarrow 0x0000000F$