

1 [P2A2]
Elección única ¿Cuál de los siguientes grupos de instrucciones IA32 sólo modifican los indicadores de estado sin almacenar el resultado de la operación?

Usuario Profesores

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

Puntuación: 1,00

2 [P2T]
Elección única 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\n\0"
```

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: -0,33

3 [T2.2.2]
Elección única Si la variable val está almacenada en ebx y la variable x está almacenada en eax, la sentencia val ^= x; se puede traducir a ensamblador como:

Usuario Profesores

- ☐ a) andl %ebx,%eax
- ☒ b) xorl %ebx,%eax
- ☐ c) testl %eax,%ebx
- ☐ d) xorl %eax,%ebx

Puntuación: -0,33

4 [T2.2.1]
Elección única ¿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?

Usuario Profesores

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

Puntuación: 1,00

5 [P3T]
Elección única ¿En qué registro se pasa el primer argumento a una función en Linux gcc x86-64?

Usuario Profesores

- ☐ a) edx
- ☐ b) ecx
- ☐ c) esi
- ☒ d) edi

Puntuación: 1,00

6 Elección única 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

Usuario Profesores

- ☐ a) 0xffff ffff
- ☐ b) 0xf7ff ffff
- ☐ c) 0xf000 0000
- ☐ d) 0xc000 0000

Puntuación: 1,00

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

Usuario Profesores

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

Puntuación: 0,00

8 Elección única [T2.4.1] Si `%rsp` vale `0xdeadbeefdead0d0`, ¿cuál será su nuevo valor después de que se ejecute `pushq %rbx`?

Usuario Profesores

- ☐ a) `0xdeadbeefdead0c8`
- ☐ b) `0xdeadbeefdead0d8`
- ☐ c) `0xdeadbeefdead0cc`
- ☐ d) `0xdeadbeefdead0d4`

Puntuación: 1,00

9 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\n\0"
```

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`
`movl (%edx), resultado+4`
- ☐ b) `movl (%eax), resultado+4`
`movl (%edx), resultado`
- ☐ c) `movl %eax, resultado+4`
`movl %edx, resultado`
- ☐ d) `movl %eax, resultado`
`movl %edx, resultado+4`

Puntuación: 1,00

10

Elección única

[T2.2.2]

Para poner a 1 el bit 5 del registro %edx sin cambiar el resto de bits podemos usar la instrucción máquina:

Usuario Profesores

- ☐ a) and \$32, %edx
- ☐ b) and \$0x5, %edx
- ☐ c) or \$0b101, %edx
- ☐ d) or \$0x20, %edx

Puntuación: 0,00

11

Elección única

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
cld
mov $5,%ebx
```

idiv %ebx

El valor de %RAX después de la división es:

Usuario Profesores

- ☐ a) Ninguna de las soluciones es correcta
- ☐ b) 0xFFFFFFFF
- ☐ c) 0x00000004
- ☐ d) 0xFFFFFFFFC

Puntuación: 1,00

12

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 %RAX después de la división es:

Usuario Profesores

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

13 Dado el siguiente fragmento de programa:

Elección única

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

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

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

El valor de %RDX después de la división es:

Usuario Profesores

- ☐ a) 0xFFFFFFFFB
- ☐ b) 0xFFFFFFFFF
- ☐ c) 0xF0000000
- ☐ d) 0x00000000

Puntuación: 1,00

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

Puntuación: -0,33

15 Dado el siguiente fragmento de programa:

Elección única

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

```
section .text
```

Puntuación: -0,33

15

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
```

```
div %ebx
```

El valor de %RDX después de la división es:

Usuario Profesores



a) 0x0000000F



b) 0x00000001

•



c) 0xFFFFFFFF



d) Ninguna de las soluciones es correcta

Puntuación: 1,00

16

Elección única

Dado el siguiente fragmento de programa:

```
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
cld
mov $5,%ebx
```

```
div %ebx
```

El valor de %RDX después de la división es:

Usuario Profesores



a) 0x00000003



b) 0xFFFFFFFFC

•



c) Ninguna de las soluciones es correcta



d) 0xFFFFFFFF

Puntuación: 1,00

17

Elección única

Dado el siguiente fragmento de programa:

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

```
mov $-23,%eax
cld
mov $5,%ebx

idiv %ebx
```

El valor de %RDX después de la división es:

Usuario Profesores

- ☐ a) 0x00000003
- ☐ b) 0xFFFFFFFFC
- ☒ c) Ninguna de las soluciones es correcta
- ☐ d) 0xFFFFFFFF

Puntuación: 1,00

17

Elección única

Dado el siguiente fragmento de programa:

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

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

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

El valor de %RAX después de la división es:

Usuario Profesores

- ☐ a) 0xFFFFFFFFFA
- ☐ b) 0xFFFFFFFFFB
- ☐ c) 0xFFFFFFFFFC
- ☐ d) 0x00000005

Puntuación: 1,00

Información

[¿Qué es SWAD? \[ES\]](#)
[What is SWAD? \[EN\]](#)
[Publicaciones](#)
[Funcionalidad](#)
[Difusión](#)
[Prensa](#)

Documentación

[Manual breve \[ES\]](#)
[Brief manual \[EN\]](#)
[Guía usuario \[ES\]](#)
[User guide \[EN\]](#)
[Presentaciones](#)
[Videotutoriales](#)
[Logos](#)

UGR

[Condiciones legales](#)
[Protección de datos](#)
[Twitter SWAD UGR](#)
[Estadísticas](#)
[Póster](#)
[Servidor](#)
[Encuentro](#)

Community

[Twitter](#)
[Facebook](#)
[Wikipedia](#)
[Google+](#)
[YouTube](#)
[alternativeTo](#)
[startupRANKING](#)
[Captorra](#)
[SourceForge](#)
[GitHub](#)
[Open HUB](#)

Software libre

[Source code](#)
[Download](#)
[Install](#)
[Database](#)
[Translation](#)
[API](#)
[Changelog](#)
[Roadmap](#)
[Authors](#)
[Implementación](#)

Android

[SWADroid Google Play](#)
[SWADroid Blog](#)
[SWADroid Twitter](#)
[SWADroid Google+](#)
[SWADroid GitHub](#)
[SWADroid Open HUB](#)

IOS

[iSWAD App Store](#)
[iSWAD Twitter](#)
[iSWAD GitHub](#)