

Inicio: 2021-10-26, martes, 12:45:16

Final: 2021-10-26, martes, 13:14:30

Preguntas: 17

Respuestas
válidas:

Puntuación:

Nota:

1
Elección única

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?

Usuario Profesores

- 64 bits de resultado en AM805 → c descartada*
- Se trata de un número positivo ⇒ Descartab*
↳ Su suma 16 veces es otro n° positivo → d descartada
- ☐ a) no se diferencian
- ☐ b) en uno se interpreta como negativo, en otro como positivo
- ☐ c) en uno ocupa 32 bits, en otro 64 bits
- ☐ d) en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no

Puntuación:

2
Elección única

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

Usuario Profesores

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

Puntuación:

3
Elección única

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
cld
add %eax, %edi
adc %edx, %ebp
jnc ncarry
inc %edx
```

nocarry:

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

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

Usuario Profesores



a) tallaria con lista: .int -1,-2,-4,-8

☒ b) produce siempre el resultado correcto

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: **4**

Elección única

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

Usuario Profesores



a) Pone CF=0 (independientemente de lo que valiera antes)

☒ b) No cambia CF (si valía 0 permanecerá a 0, si valía 1 permanecerá a 1)

c) Cambia CF (si valía 0 cambiará a 1, si valía 1 cambiará a 0)



d) Pone CF=1 (independientemente de lo que valiera antes)

Puntuación: **5**

Elección única

[T2.1.4]

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

Usuario Profesores

a) `addq $1, %rcx`b) `movl (%rdi,%rcx,4), %edx`c) `testl %edx, %edx`☒ d) `movl %r8, %eax`Puntuación: **6**

Elección única

[T2.1.2]

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

Usuario Profesores

☒ a) RIP

b) EIP



c) IP



d) R15

Puntuación: **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\n\0"
```

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

Usuario Profesores



a) 8

☒ b) 4

c) 16



d) 32

Puntuación:

8

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

Elección única

```
xor %eax, %eax    eax = 0
sub $1, %eax      eax = -1
cld
idiv %eax
```

Usuario Profesores

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

Puntuación: **9**

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

Elección única

Usuario Profesores

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

Puntuación: **10**

[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\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+4
movl (%edx), resultado
- ☐ b) movl (%eax), resultado
movl (%edx), resultado+4
- ☐ c) movl %eax, resultado+4
movl %edx, resultado
- ☒ d) movl %eax, resultado
movl %edx, resultado+4

Puntuación: **11**

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    edx = 0
mov $-23,%eax    eax = -23
```

cld

mov \$5,%ebx

idiv %ebx

...

El valor de %RDX después de la división es:

Usuario Profesores



a) 0x00000003



b) 0xFFFFFFFFC



c) 0xFFFFFFFF

☒ d) Ninguna de las soluciones es correcta

Puntuación:

$$\begin{array}{r} -23 \ 15 \\ -3 \ -4 \\ \hline \end{array}$$

$$-3 = 1011 \rightarrow 1100 \rightarrow \frac{1101}{0}$$

Resto

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) 0x00000004

☒ d) Ninguna de las soluciones es correcta

Puntuación:

13

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: **14**

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

Puntuación: **15**

Elección única

Dado el siguiente fragmento de programa en ensamblador:

```
.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) RDX=0xFFFFFFFF RAX=0x00000023
 b) Ninguna de las soluciones es correcta
 c) RDX=0xFFFFFFFF RAX=0xFFFFFFFF23
 ☒ d) RDX=0xFFFFFFFF RAX=0xFFFFFFFFDD

Puntuación: **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  rcx = 0
inc %cl        { cl = 2
inc %cl        } rcx también vale 2)
shl %cl,%rcx   rcx = 0010  Desplaz. 2 veces izq.  => rcx = 1000
mov lista,%ebx => ebx = 2
lea (%rbx,%rcx,2),%rdx
    ↳ Trabaja con las direcciones => 0x00000002 + 0x00000008 · 2 = 0x00000002
    + 0x00000010 =
    = 0x00000012
```

El valor de %RBX después de la operación LEA es:

Usuario Profesores

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

Puntuación:

17

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  rcx = 0
inc %cl        { cl = 2 => rcx = 2
inc %cl        }
shl %cl,%rcx   => rcx = 0010 => rcx = 1000 (8)
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) 0x00000008
☐ d) 0x00000002

Puntuación:

Información DocumentaUGR

CommunitySoftware liAndroid

iOS

[¿Qué es SWAD](#) [Manual breve](#) [Condiciones leg](#) [Twitter](#)
[What is SWAD?](#) [Brief manual](#) [\[E](#) [Protección de d](#) [Facebook](#)
[Publicaciones](#) [Guía usuario](#) [\[T](#) [Twitter SWAD](#) [L](#) [Wikipedia](#)
[Funcionalidad](#) [User guide](#) [\[E](#) [Estadísticas](#) [Google+](#)

[Source code](#) [SWADroid](#) [Googi](#) [SWAD App](#) [St](#)
[Download](#) [SWADroid](#) [Blog](#) [iSWAD](#) [Twitter](#)
[Install](#) [SWADroid](#) [Twittei](#) [SWAD](#) [GitHub](#)
[Database](#) [SWADroid](#) [Goog](#)