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

do

a) rax

D

b) rbx

OB

c) rcx

D

d) rdx

2 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) movl (%ebx, %edx, 4), %eax



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



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



© withd) Pleafi 4 (%edx, %edx), %eax

[P2T] Dada la siguiente definición de datos: Unique choice 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: User Teachers 0 (a) movl lista, %ebx b) movl (lista), %ebx B c) movl \$lista, (%ebx) OD D d) movl \$lista, %ebx ¿Qué valor contendrá %edx tras ejecutar las siguientes instrucciones? Unique choice xor %eax, %eax \Rightarrow eax = 0 sub \$1, %eax => eax = - 1 cltd => Extensión de signa idiv %eax =D El caciente va a rax y el resto a rax. -2|-1=1=2=0 Resto=0 **User Teachers** a) 1 b) -1 B B c) no puede saberse Edit with WPS Office enunciado

d)

[P2A2]

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

User Teachers

do

a) AND, OR, XOR

de

b) ADC, SBB

D

C CMP, TEST

D

d) IMUL, IDIV

6 Jnique choice [T2.2.2]

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

User Teachers

9

a) or \$0b101, %edx

(B)

b or \$0x20, %edx

do

c) and \$32, %edx

d) and \$0x5, %edx

Porque la orden OR

cambia el bit a l

solo si una o los

das bits son l

& Bit O

00000000 è

que

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

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



nique noice

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

(a)) andq \$1, %rax

b) shrq %rax

c) orq \$0x1, %rax

D

d) sarq %rax

nique noice [P3T]

¿En qué registro se pasa el primer argumento a una función en Linux gcc x86-64?

User Teachers



a) ecx



(b)) edi



c) edx



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 ine bucle

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

User Teachers

D

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

0,1,2,3

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 rcx = 0
inc %cl
inc %cl cl = 00000010
shl %cl,%rcx => Mete el bit de más a
la izquierda en rcx, es decir, 0
mov lista,%ebx
lea (%rbx,%rcx,2),%rdx

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

User Teachers



a) 0x00000008



 Ninguna de las soluciones es correcta



@it wit (c)) v (0 x 0 0 0 0 0 0 0 0 4



d) 0x00000002

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 edx=0

mov \$-35,%eax

cltd => Extensión de signa a ROX:RAX

10.100011

MOV \$7,%ebx

4.1011101

###

idiv %ebx

...

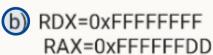
Al finalizar la ejecución de la instrucción CLTD, los valores de los registros RDX y RAX son:

User Teachers



 a) Ninguna de las soluciones es correcta







c) RDX=0xFFFFFFF RAX=0xFFFFFF23



Edit with WPS Office 0xFFFFFFFF RAX=0x000000023

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 همده ه mov \$-23,%eax cltd - عالة mov \$5,%ebx

RAX = cociente ROX = resto

6 2100 6 2101 => Byte menos signification

idiv %ebx

...

El valor de %RDX después de la división es:

User Teachers



a) 0x00000003



b) 0xFFFFFFC



 Ninguna de las soluciones es



Edit with WPSCOMFECTA



d) 0xFFFFFFF

Dado el siguiente fragmento de programa:

.section .data

lista: .int 1,2,0x10,3

longlista: .int .-lista 16 tamaño

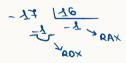
resultado: .quad 0

.section .text

main: .global main

xor %edx,%edx edx = 0 mov \$-17,%eax cltd mov longlista,%ebx

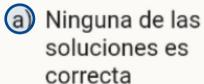
idiv %ebx



El valor de %RAX después de la división es:

User Teachers











Tit with (C) VFQXQQQQQQQQQQ



d) 0x00000004

Dado el siguiente fragmento de programa:

.section .data

lista: .int 1,2,0x10,3,-3

longlista: .int .-lista 46 tomaño

resultado: .quad 0

.section .text

main: .global main

xor %edx,%edx edx = 0 mov \$-12,%eax

cltd

mov longlista,%ebx

idiv %ebx

Resto = - 12

Q1100 Q10011 Q1010 DEQUETION

El valor de %RDX después de la división é es:

User Teachers



a) 0x00000004



b) 0x00000010



 Ninguna de las soluciones es



Edit with Weemecta





Dado el siguiente fragmento de programa:

.section .data

lista: .int 1,2,0x10,3

longlista: .int .-lista 16 tamaño

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) 0x00000004



(b) 0x00000010



c) 0x000000F



d) Ninguna de las Edit with ଏ**ଟିଆ ହିଆଁ ତ**nes es correcta

Dado el siguiente fragmento de programa:

.section .data

lista: .int 1,2,0x10,3

longlista: .int (.-lista)/4 4 tamaño

resultado: .quad 0

.section .text

main: .global main

xor %edx,%edx edx = 0 mov \$-35,%eax cltd mov \$7,%ebx idiv %ebx

... Cociente = -5 -> 1404 -> 1010 -> 1011 B

El valor de %RAX después de la divisiór es:

User Teachers

- D
- a) 0xFFFFFFC
- b) 0x00000005



d) 0xFFFFFFA