



/ UGR / plataforma de  
apoyo a la docencia

Estudiante:  Valentín

0 notificaciones

enero

29  
10:07

  
Plataforma  
> España  
> ugr.es  
> ETSIIT  
> Db.Gr.Ing.Inf./Matem.  
>

EC



# Estructura de Computadores EC



Inicio



Asignatura



Evaluación



Archivos



Usuarios



Comunicación



Análisis



Perfil

## Frecuentes



Exámenes



Test



Timeline



Documentos



Calificaciones



Convocatorias

ENERO 2021

L M M J V S D

28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

2021-01-26  
17:09

EXAMEN 29  
ENERO 2021  
Plataforma  
PRADO...

...

Ignacio  
Rojas Ruiz

2021-01-26  
17:05

EXAMEN 29  
ENERO 2021.  
Temporización y...

...

Ignacio  
Rojas Ruiz

2021-01-26  
17:01

EXAMAN 29  
ENERO 2021.  
Plataforma  
PRA...

...

Ignacio  
Rojas Ruiz

2021-01-26  
16:58

EXAMEN 29  
ENERO 2021

El examen del  
2...

...

Ignacio  
Rojas Ruiz

2021-01-20  
15:37

Existian errores  
en el fichero de  
aye...

...

Ignacio  
Rojas Ruiz

2021-01-19  
11:09

Están publicadas  
las notas finales  
de...

...

Ignacio  
Rojas Ruiz

2021-01-14  
12:55

EXAMEN FINAL  
DE EC.

El examen será,  
...

...

Ignacio  
Rojas Ruiz

2021-01-11  
13:52

ACLARACION  
SOBRE  
CALIFICACION  
DE LA A...

...

Ignacio  
Rojas Ruiz

2021-01-08  
12:14

Curso 2020-  
2021.  
Practica Arduino.  
Fe...

...

Ignacio  
Rojas Ruiz

2020-12-17  
09:26

Curso 2020-  
2021.  
Examen tipo test.  
Fe...

...

Ignacio  
Rojas Ruiz

2020-12-10  
14:00

Curso 2020-  
2021.  
Examen tipo test.  
Fe...

...

Ignacio  
Rojas Ruiz

2020-11-24  
16:24

Ya se pueden ver  
los resultados del  
e...

...

Ignacio  
Rojas Ruiz

2020-11-12  
18:07

Curso 2020-  
2021.  
Examen tipo test.  
Re...

...

Ignacio  
Rojas Ruiz

2020-11-06  
13:03

Curso 2020-  
2021.  
Examen tipo test.  
F...

...

Ignacio  
Rojas Ruiz

2020-10-22  
12:46

CURSO 2020-  
2021. Practica 2.

La fech...

...

Ignacio  
Rojas Ruiz

2020-01-28  
11:12

NOTA  
IMPORTANTE:  
MODIFICACION  
HORA IN...

...

Ignacio  
Rojas Ruiz

2020-01-09  
13:15

Estimados  
estudiantes de  
EC,

Mañana,...

...

Ignacio  
Rojas Ruiz

2019-12-09  
23:13

Estimados  
estudiantes de  
EC,

Debido ...

...

Ignacio  
Rojas Ruiz

2019-09-16  
14:45

Ya estan abiertos  
los grupos de  
pract...

...

Ignacio  
Rojas Ruiz

2019-01-24  
14:18

Ya se han  
enviado las  
calificaciones ...

...

Ignacio  
Rojas Ruiz

2019-01-08  
11:06

Mediante este  
correo quiero

anunciar ...

...

Ignacio  
Rojas Ruiz

2018-01-17

11:48

Nanoprogramació  
Informacion  
actuali...

...

Ignacio  
Rojas Ruiz

2017-01-17

09:47

Desactivar el  
fichero  
Bomba\_NBA\_201

...

Ignacio  
Rojas Ruiz

2014-12-15

13:17

// Versión SSSE3  
(pshufb) web  
http://w...

...

Ignacio  
Rojas Ruiz

82 de EC

1 profesor

48'20"

81 estudiantes

Ignacio  
Rojas ...Alejandro  
Cárden...

3'21"

Jesús  
García ...

3'22"

Marta  
Molina ...

3'24"

David  
Muñoz ...

3'25"

Pablo  
Olivare...

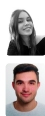
3'26"

Eva  
Rueda ...

3'27"



3'29"



Mario  
Rodríguez...  
Rocío  
Barragán...  
José A...  
Villarejo...

3'30"

3'30"



Sistema Actividades Proyectos Convocatorias Test **Exámenes** Juegos Encuestas

## Examen\_29\_ENERO\_2021



Universidad de Granada - Doble Grado en Ingeniería Informática y  
Matemáticas  
Estructura de Computadores



Estudiante: 45338112 Guerrero Cano, Valentín



### PRACTICAS

20 preguntas

**1**Elección  
única

En la secuencia de programa siguiente:

5f5: e8 cd ff ff ff callq 5c7 <suma>

5fa: 48 83 c4 20 add \$0x20,%rsp

¿cuál es el valor que introduce en la pila la instrucción callq?

- ☐ a) 0x5f5
- ☐ b) 0x5c7
- ☐ c) 0xfffffffcd
- ☒ d) 0x5fa

**2**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:

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

**3**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

- ☒ a) 0xf7ff ffff

- ☐ b) 0xffff ffff
- ☐ c) 0xfc00 0000
- ☐ d) 0xf000 0000

**4**

Suponga la siguiente sentencia asm en un programa:

Elección  
única

```
asm(" add (%[a],[i],4),%[r]"
    :[r] "+r" (result)
    :[i] "r" (i),
      [a] "r" (array)
    );
```

¿Cuál de las siguientes afirmaciones es incorrecta?

- ☐ a) r es un registro de entrada/salida
- ☒ b) i es un registro de salida
- ☐ c) se desea que el valor calculado por la instrucción ensamblador quede almacenado en la variable result
- ☐ d) a es un registro de entrada

**5**Elección  
única

[P5.2]

¿Cuál de las siguientes afirmaciones sobre el programa size.cc de la práctica 5 es cierta?

- ☐ a) Si continuáramos multiplicando por 2 el tamaño del vector en el eje X obteniendo más puntos de la gráfica, esta continuaría horizontal para cualquier valor más allá de 64 MB.
- ☒ b) La gráfica tiene escalones hacia arriba porque en cada punto del eje X accedemos al mismo número de elementos del vector y el número de aciertos por localidad temporal disminuye bruscamente en ciertos puntos al aumentar el tamaño del vector.
- ☐ c) La gráfica tiene tramos horizontales porque el hecho de realizar la mitad de accesos al vector en cada punto de un tramo horizontal respecto al anterior punto de ese mismo tramo horizontal es compensado por el número de fallos creciente en ese mismo tramo horizontal.
- ☐ d) La diferencia de velocidades entre L2 y L3 es mayor que la diferencia de velocidades entre L1 y L2.

**6**Elección  
única

Si queremos almacenar la palabra de 16 bits 0x8965 en memoria según little-endian, quedará almacenada a partir de la posición 0x1000 como:

- ☐ a) en el byte 0x1000 se guarda 0x89 y en el 0x1001 0x65
- ☒ b) en el byte 0x1000 se guarda 0x65 y en el 0x1001 0x89
- ☐ c) en el byte 0x1000 se guarda 0xA6 y en el 0x1001 0x91
- ☐ d) en el byte 0x1000 se guarda 0x91 y en el 0x1001 0xA6

**7**Elección  
única

En x86-64, es responsabilidad del procedimiento llamado (callee) guardar, entre otros, los registros:

- ☐ a) %rbx, %rsi, %rdi
- ☒ b) %r12, %r13
- ☐ c) %rax, %r10, %r11
- ☐ d) %rax, %rbx, %rcx, %rdx



**8**Elección  
única

¿Cuál de las siguientes líneas de código sirve para definir una variable entera de 64 bits llamada tam en GNU/as Linux x86-64?

- ☒ a) tam: .quad .-msg
- ☐ b) int tam;
- ☐ c) \_int tam = 0
- ☐ d) var tam : integer;

**9**Elección  
única

El servidor de SWAD tiene dos procesadores Intel Xeon E5-2620 v3@ 2.40GHz (2,4 Ghz, 6 núcleos, 12 hebras, 15MiB de caché L3, reloj DDR4 a 1866 MHz).

Suponga que un proceso swad, que se ejecuta en un núcleo, tiene que ordenar un vector de estudiantes accediendo repetidamente a sus elementos. Cada elemento es una estructura de datos de un estudiante y tiene un tamaño de 4KB. Si representamos en una gráfica las prestaciones en función del número de estudiantes a ordenar, ¿para qué límites teóricos en el número de estudiantes se observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria? Busque los tamaños de los distintos niveles de cache en <http://www.cpu-world.com/>

- ☐ a) 32 / 256 / 8192 estudiantes
- ☐ b) 48 / 384 / 23040 estudiantes
- ☐ c) 4 / 32 / 512 estudiantes
- ☐ d) 8 / 64 / 3840 estudiantes

**10**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 para copiar la dirección de memoria donde comienza lista en el registro EBX es:

- ☐ a) movl lista, %ebx
- ☐ b) movl (lista), %ebx
- ☒ c) movl \$lista, %ebx
- ☐ d) movl \$lista, (%ebx)

**11**Elección  
única

Suponga la expresión:

unsigned int val = x & 0x01010101;

donde x es un unsigned int.

¿Cuál de los siguientes valores de x da como resultado val = 0x01010101?

- ☐ a) 0x10101010
- ☒ b) 0x13579bdf
- ☐ c) 0xeeeeeeee
- ☐ d) 0xaaaaaaaa

**12**Elección  
única

Si el registro RAX contiene un long (64 bits CON signo) x, la secuencia de instrucciones siguiente:

```
cmpq $10, %rax
```

```
ja     dest
```

saltará a la etiqueta dest si:

salta a la etiqueta dest si:

- ☒ a)  $x \leq -1 \parallel x \geq 11$
- ☐ b)  $x \geq 0 \ \&\& \ x \leq 10$
- ☐ c)  $x < 10$
- ☐ d)  $x > 9$

**13**

Elección  
única

[T2.4.1]

Si `%rsp` vale `0xdeadbeefdeadd0d0`, ¿cuál será su nuevo valor después de que se ejecute `pushq %rbx`?

- ☐ a) `0xdeadbeefdeadd0d8`
- ☐ b) `0xdeadbeefdeadd0d4`
- ☒ c) `0xdeadbeefdeadd0c8`
- ☐ d) `0xdeadbeefdeadd0cc`

**14**

Elección  
única

Si el registro `RAX` contiene un long (64 bits con signo) `x`, la secuencia de instrucciones siguiente:

`cmpq $6, %rax`

`ja dest`

salta a la etiqueta `dest` si:

- ☐ a)  $x \leq 6$
- ☐ b)  $x \geq 0 \ \&\& \ x \leq 6$
- ☐ c)  $x \geq 6$
- ☒ d)  $x < 0 \parallel x > 6$

**15**

Elección  
única

[P2.2]

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

- ☐ a) Pone `CF=1`
- ☐ b) Cambia `CF`
- ☐ c) Pone `CF=0`
- ☐ d) No cambia `CF`

**16**

Elección  
única

Dada la siguiente declaración:

`longsaludo: .quad . - saludo`

¿Cuál de las siguientes órdenes de `gdb` NO muestra el contenido de `longsaludo`?

- ☒ a) `x/8db &longsaludo`
- ☐ b) `p *((long *) &longsaludo)`
- ☐ c) `x/1dg &longsaludo`
- ☐ d) `p (long) &longsaludo`

**17**

Elección  
única

En una bomba que siga la misma estructura que la estudiada en el ejemplo dado en el guión de prácticas, y compilada en 32 bits:

```
0x0804873f <main+207>: call 0x08048504 <scanf>
0x08048744 <main+212>: mov 0x24(%esp),%edx
0x08048748 <main+216>: mov 0x804a044,%eax
0x0804874d <main+221>: cmp %eax,%edx
0x0804874f <main+223>: je 0x08048756 <main+230>
0x08048751 <main+225>: call 0x08048604 <boom>
0x08048756 <main+230>:
```

...el código numérico (pin) es...

- ☐ a) el entero almacenado a partir de la posición de memoria 0x24(%esp)
- ☒ b) el entero almacenado a partir de la posición de memoria 0x804a044
- ☐ c) el entero 0x804a044
- ☐ d) el entero cuya dirección está almacenada en la posición de memoria 0x804a044

**18**

Elección  
única

En la realización de la práctica de la bomba digital compilada en 32 bits, una parte del código máquina es el siguiente:

```
0x080486e8 <main+120>:  call 0x8048524 <strncmp>
0x080486ed <main+125>:  test %eax,%eax
0x080486ef <main+127>:  je 0x80486f6 <main+134>
0x080486f1 <main+129>:  call 0x8048604 <boom>
```

¿Cuál de los siguientes comandos de gdb cambiaría el salto condicional por un salto incondicional?

- ☐ a) set \*(char\*)0x080486ef=0xeb
- ☐ b) set \*(char\*)0x080486f6=jump
- ☐ c) set %0x080486ef=0xeb
- ☐ d) set \$0x080486ef=0xeb

**19**

Elección  
única

[T2.1.4]

Respecto a tamaños de tipos integrales en x86 y x86-64, la excepción es que

- ☐ a) int pasa de 4 B (x86) a 8 B (x86-64)
- ☒ b) long int pasa de 4 B a 8 B
- ☐ c) long long int pasa de 4 B a 8 B
- ☐ d) ninguna de las anteriores

**20**

Elección  
única

Tras dividir 0x00000000 00040000 (en EDX:EAX) entre 0x00008000 (ECX), el resultado será:

- ☒ a) 0x8 en EAX y 0x0 en EDX
- ☐ b) 0x0 en EAX y 0x8 en EDX
- ☐ c) 0x0 en AX y 0x8 en DX
- ☐ d) 0x8 en AX y 0x8 en DX

## TEORIA

30 preguntas

**21**

Elección  
única

En la secuencia de programa siguiente:

```
628: e8 cd ff ff      callq 5fa <suma>
62d: 48 83 c4 20      add $0x20,%rsp
```

¿cuál es el valor que introduce en la pila la instrucción callq?

- ☐ a) 0xfffffcd
- ☐ b) 0x5fa
- ☐ c) 0x628
- ☒ d) 0x62d

**22**

¿Cuál de las siguientes afirmaciones acerca de la memoria es cierta?

**22**Elección  
única

- ☐ a) La memoria dinámica se emplea en las caches L1 y L2.
- ☒ b) La memoria dinámica usa señales de control RAS# y CAS#.
- ☐ c) Las celdas de memoria estática están constituidas por un transistor y un condensador.
- ☐ d) Las celdas de memoria estática tienen que ser constantemente refrescadas.

**23**Elección  
única

¿Cuántos conjuntos tendría una cache de 256 B asociativa por conjuntos de 4 vías con líneas de 16 B?

- ☐ a) Ninguno
- ☐ b) 64
- ☒ c) 4
- ☐ d) 16

**24**Elección  
única

Una función C devuelve el valor de un elemento de un array mediante el siguiente código ensamblador:

```
leaq  (%rdi,%rdi,4), %rax
addq  %rsi, %rax
movl  var(%rax,4), %eax
retq
```

Se puede adivinar que:

- ☒ a) var es un array bidimensional de enteros, con cinco columnas
- ☐ b) var es un array multi-nivel pero no se pueden adivinar las dimensiones
- ☐ c) var es un array multi-nivel (punteros a enteros) de cuatro filas
- ☐ d) var es un array bidimensional de enteros, no se pueden adivinar dimensiones

**25**Elección  
única

¿Cuál de las siguientes funciones NO calcula el factorial de un unsigned long?

- ☐ a) fact:

```
cmpq  $1, %rdi
ja     .L8
movl  $1, %eax
ret
```

.L8:

```
pushq %rbx
movq  %rdi, %rbx
leaq  -1(%rdi), %rdi
call  fact
imulq %rbx, %rax
popq  %rbx
ret
```

- ☐ b) fact:

```
movl  $1, %eax
cmpq  $1, %rdi
ja     .L8
ret
```

.L8:

```
pushq %rbx
movq  %rdi, %rbx
leaq  -1(%rdi), %rdi
call  fact
imulq %rbx, %rax
popq  %rbx
```

```
---
```

- ret
- ☐ c) fact:
- ```
movl    $1, %eax
.L3:
cmpq    $1, %rdi
jbe     .L1
imulq   %rdi, %rax
decq    %rdi
jmp     .L3
.L1:
ret
```
- ☐ d) fact:
- ```
movl    $1, %eax
cmpq    $1, %rdi
ja      .L8
ret
.L8:
pushq   %rbx
movq    %rdi, %rbx
call    fact
subq    $1, %rbx
imulq   %rbx, %rax
popq    %rbx
ret
```

**26**Elección  
única

El primer nivel de una jerarquía de memoria de dos niveles tiene una tasa de aciertos del 75%. Las peticiones de memoria tardan 12 ns en completarse si dicha posición se encuentra en ese nivel; si hay que acceder al segundo nivel se añaden otros 100 ns. ¿Cuál es el tiempo medio de acceso de la jerarquía?

- ☐ a) 37 ns
- ☐ b) 112 ns
- ☐ c) 25 ns
- ☐ d) 88 ns

**27**Elección  
única

En x86-64, una función con 10 parámetros de tipo long que devuelve el valor del 8º parámetro y no modifica el puntero de pila puede traducirse a ensamblador como:

- ☐ a) movq 8(%rsp), %rax  
ret
- ☐ b) movq %r10, %rax  
ret
- ☐ c) movq %r8, %rax  
ret
- ☒ d) movq 16(%rsp), %rax  
ret

**28**Elección  
única

¿Cuál de las siguientes sentencias sobre la unidad de control es FALSA?

- ☐ a) Cuanto más horizontal es la microprogramación, más largas son las microinstrucciones
- ☐ b) Debido al pequeño número de operaciones simples, la sección de control de un procesador RISC puede ser cableada en lugar de microprogramada
- ☒ c) El programador de lenguaje ensamblador necesita conocer la microarquitectura del ordenador
- ☐ d) Es posible realizar el diseño físico de una unidad de control cableada de

- ☐ a) Es posible realizar el diseño físico de una unidad de control cableada de manera semiautomática

**29**Elección  
única

En x86-64, una función con 10 parámetros de tipo long que devuelve el valor del 10º parámetro y no modifica el puntero de pila puede traducirse a ensamblador como:

- ☐ a) `movq %r10, %rax`  
`ret`
- ☐ b) `movq %r13, %rax`  
`ret`
- ☒ c) `movq 0x20(%rsp), %rax`  
`ret`
- ☐ d) `movq 0x10(%rsp), %rax`  
`ret`

**30**Elección  
única

En una cache con 64 bytes de longitud de línea, ¿qué bits de una dirección de memoria de 64 bits se utilizan para determinar a qué byte dentro de la línea se refiere dicha dirección? (Memoria direccionable por bytes)

- ☐ a) [5...0]
- ☐ b) [8...6]
- ☐ c) [5...3]
- ☐ d) [11...6]

**31**Elección  
única

¿Cuál de las siguientes características es típica de la microprogramación horizontal?

- ☐ a) Muchos campos solapados.
- ☐ b) Poca capacidad para expresar paralelismo entre microoperaciones.
- ☒ c) Poca codificación.
- ☐ d) Microinstrucciones cortas.

**32**Elección  
única

Si queremos almacenar la palabra de 64 bits 0x0000001f fffffe0 en una memoria de bytes según la convención little-endian a partir de la posición 0x0804913c, quedará:

- ☒ a) 0xe0 en 0x0804913c y 0x00 en 0x08049140
- ☐ b) 0x1f en 0x0804913c y 0xe0 en 0x08049140
- ☐ c) 0xe0 en 0x0804913c y 0x1f en 0x08049140
- ☐ d) 0x00 en 0x0804913c y 0xe0 en 0x08049143

**33**Elección  
única

La ganancia en velocidad ideal de un cauce de  $K$  etapas de igual duración  $T$  ejecutando un programa de  $N$  instrucciones es

- ☐ a)  $S = \frac{K \cdot N}{K - N + 1}$
- ☐ b)  $S = \frac{N \cdot T}{(N + K - 1) \cdot T}$
- ☒ c)  $S = \frac{K \cdot N}{K + N - 1}$
- ☐ d)  $S = \frac{N \cdot K \cdot T}{(N - K + 1) \cdot T}$

**34**

Elección

Al traducir la sentencia C  
`r->i = val;`

Elección  
única

gcc genera el código ASM:

```
movl %edx, 12(%rax)
```

Se puede deducir que:

- ☐ a) i es un entero que vale 12
- ☐ b) r es un puntero que apunta a la posición de memoria 12
- ☐ c) val es un entero que vale 12
- ☒ d) el desplazamiento de i en \*r es 12

**35**

Elección  
única

Considere las siguientes declaraciones en una máquina Linux de 64 bits:

```
struct R {
    int value2;
    short ref_count;
    char tag[10];
};

struct N {
    long value;
    struct R record;
    char string[8];
};
```

También se declara la siguiente variable global:

```
struct N my_node;
```

¿Cuál es el tamaño de my\_node en bytes?

- ☐ a) 28
- ☐ b) 40
- ☐ c) 32
- ☐ d) 48

**36**

Elección  
única

¿Cuántas líneas de dirección (patillas) son necesarias para direccionar un chip de memoria DRAM de 4096 x 4?

- ☐ a) 12
- ☐ b) 10
- ☒ c) 6
- ☐ d) 11

**37**

Elección  
única

Considere la siguiente declaración:

```
struct rec {
    int i;
    int j;
    int a[10];
    int *p;
};
```

y una función void f (struct rec \*r); cuyo código en ensamblador es:

```
mov 0x4(%rdi),%eax
add (%rdi),%eax
cltq # RAX <- (long) EAX
lea 0x8(%rdi,%rax,4),%rax
mov %rax,0x30(%rdi)
retq
```

¿Cuál es el código C de la función f?

- ☐ a) r->p = (int \*) (long) (r->a[r->i] + r->a[r->j]);
- ☐ b) r->a[r->i] = r->a[r->j];
- ☐ c) r->p = &(r->a[r->i + r->j]);
- ☐ d) r->a[r->i] = r->j;

**38**

Respecto a las unidades de control nanoprogramadas:

- ☐ a) La realización nanoprogramada de una unidad de control es más rápida que la

Elección  
única

- ☐ a) La realización nanoprogramada de una unidad de control es más rápida que la microprogramada.
- ☐ b) El diseño de las unidades de control nanoprogramadas siempre tiene que ser vertical.
- ☐ c) Suponiendo una memoria de microprograma con  $n$  microinstrucciones de  $w$  bits cada una, de las cuales  $2^m$  son distintas, el ahorro en bits si se utiliza nanoprogramación es  $(n \cdot m + 2^m \cdot w) - n \cdot w$ .
- ☒ d) La anchura de la memoria de nanoprograma es la misma que la de memoria de microprograma en un diseño de la misma unidad de control que no usara nanoprogramación.

**39**Elección  
única

Dada una matriz de enteros de dimensiones 5x3, una posible traducción a ensamblador de una función que devuelve el elemento  $i, j$ :

int elem (int A[5][3], size\_t i, size\_t j);  
podría ser:

- ☐ a) `leaq (%rsi, %rsi, 4), %rax`  
`leaq (%rdx, %rdx, 2), %rdx`  
`addq %rdx, %rax`  
`movl (%rax, %rdi), %eax`  
`ret`
- ☐ b) `leaq (%rdx, %rsi, 4), %rax`  
`movl (%rdi, %rax, 4), %eax`  
`ret`
- ☐ c) `movq (%rdi, %rsi, 4), %rax`  
`movl (%rax, %rdx, 4), %eax`  
`ret`
- ☐ d) `leaq (%rsi, %rsi, 2), %rax`  
`leaq (%rdi, %rax, 4), %rax`  
`movl (%rax, %rdx, 4), %eax`  
`ret`

**40**Elección  
única

¿Cuántos conjuntos tendría una cache de 256 KB asociativa por conjuntos de 16 vías con líneas de 64 B?

- ☐ a) 4
- ☐ b) 64
- ☒ c) 256
- ☐ d) 16

**41**Elección  
única

En una cache con 32 bytes de longitud de línea, ¿qué bits de una dirección de memoria de 64 bits se utilizan para determinar a qué byte dentro de la línea se refiere dicha dirección? (Memoria direccionable por bytes)

- ☐ a) [9...5]
- ☐ b) [7...3]
- ☐ c) [4...0]
- ☐ d) [8...4]

**42**Elección  
única

Al traducir la sentencia C  
`r->i = val;`  
gcc genera el código ASM:  
`movl %edx, 8(%rax)`  
Se puede deducir que:

- ☐ a)  $i$  es un entero que vale 8



☐ a) r es un entero que vale 8

☒ b) r es un puntero que apunta a la posición de memoria 8

☐ c) el desplazamiento de i en \*r es 8

☐ d) val es un entero que vale 8

**43**

Elección  
única

El primer nivel de una jerarquía de memoria de dos niveles tiene una tasa de aciertos del 80%. Las peticiones de memoria tardan 12 ns en completarse si dicha posición se encuentra en ese nivel; si hay que acceder al segundo nivel se añaden otros 100 ns. ¿Cuál es el tiempo medio de acceso de la jerarquía?

☐ a) 37 ns

☐ b) 112 ns

☐ c) 32 ns

☐ d) 88 ns

**44**

Elección  
única

¿A qué tipo de localidad de memoria hace referencia la siguiente afirmación: "si se referencia un elemento, los elementos cercanos a él serán referenciados pronto"?

☒ a) Localidad espacial

☐ b) Localidad temporal

☐ c) Localidad secuencial

☐ d) Localidad asociativa

**45**

Elección  
única

Una posible codificación en microinstrucciones de la instrucción call X es:

☐ a)  $SP := PC - 1$  ;  $m[SP] := PC$  ;  $PC := X$

☒ b)  $SP := SP - 1$  ;  $m[SP] := PC$  ;  $PC := X$

☐ c)  $SP := SP - 1$  ;  $m[SP] := PC$  ;  $PC := PC + 1$

☐ d)  $PC := X$  ;  $SP := SP - 1$  ;  $m[SP] := PC$

**46**

Elección  
única

En una máquina little-endian con memoria de bytes y representación en complemento a dos que permite accesos a memoria de tamaño byte (1 B), media palabra (2 B) y palabra (4 B), si se almacena en la posición 0xbabc una palabra de valor -2, ¿qué se obtendría al consultar la media palabra en la posición 0xbabe?

☐ a) -2

☐ b) -1

☐ c) 1

☐ d) 0

**47**

Elección  
única

Se pretende almacenar una palabra de 4 B en una memoria de bytes a partir de una dirección determinada. ¿Cuál de las siguientes es válida, si la palabra debe quedar alineada?

☒ a) 0xFACEB00C

☐ b) 0xDEADBEEF

☐ c) 0xABADF00D

☐ d) 0xCAFEBAFE

**48**

Elección  
única

¿Qué componentes necesitamos para construir una memoria de 2K x 8 bits?

☒ a) 16 memorias de 512 x 2 bits y un decodificador de 2 a 4

☐ b) 64 memorias de 128 x 1 bits y un decodificador de 4 a 16

☐ c) 8 memorias de 256 x 4 bits y un decodificador de 2 a 4

- ☐ c) 8 memorias de 256 x 4 bits y un decodificador de 2 a 4
- ☐ d) 8 memorias de 512 x 2 bits y un decodificador de 1 a 2

**49**Elección  
única

La técnica de "adelanto de registros" (register forwarding) en un cauce segmentado se usa para limitar el impacto de los riesgos...

- ☒ a) (por dependencias) de datos
- ☐ b) organizativos
- ☐ c) estructurales
- ☐ d) de control

**50**Elección  
única

¿Qué necesitamos para construir una memoria de 1K x 8 bits?

- ☐ a) 64 memorias de 128 x 1 bits
- ☒ b) 8 memorias de 256 x 4 bits y un decodificador de 2 a 4
- ☐ c) 8 memorias de 512 x 2 bits
- ☐ d) 4 memorias de 512 x 2 bits y un decodificador de 2 a 4

He terminado

**Información DocumentalUGR****CommunitySoftware lilAndroid****iOS**

¿Qué es SWADManual breve [Condiciones legTwitter

What is SWAD?Brief manual [EProtección de dFacebook

Publicaciones Guía usuario [ETwitter SWAD UWikipedia

Funcionalidad User guide [ENEstadísticas

Difusión PresentacionesPóster

Prensa VideotutorialesServidor

Logos Encuentro

Source code

Download

Install

Database

Translation

API

Changelog

Roadmap

Authors

Implementación

Open HUB

SWADroid GoogliSWAD App Str

SWADroid Blog iSWAD Twitter

SWADroid TwitteiSWAD GitHub

SWADroid Googl

SWADroid GitHub

SWADroid Open HUB

UNIVERSIDAD  
DE GRANADA

Universidad de Granada

Consultas y problemas: swad@ugr.es

Acerca de SWAD 20.10.1 (2021-01-28)

Página generada en 87 ms y enviada en 2 ms