

Nombre:

DNI:

Grupo:

## Test de Teoría (3.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.1p si es correcta, 0p si está en blanco o claramente tachada, -0.03p si es errónea.

Anotar las respuestas (a, b, c ó d) en la siguiente tabla.

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

1. En la memoria de un procesador de la familia Intel 64 (x86-64) se almacena a partir de la dirección N los siguientes contenidos: 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x02. Posteriormente se lee (carga) %rax desde (a partir de) la dirección N. ¿Cómo es %rax entonces, considerándolo con signo?

- a. menor que -2000 millones
- b. entre -2000 millones y -1
- c. entre 0 y 2000 millones
- d. mayor que 2000 millones

2. ¿Cuál de las siguientes afirmaciones es **incorrecta**?

- a. El direccionamiento indexado es útil para manejo de estructuras
- b. El direccionamiento indirecto indica un puntero al operando
- c. En el direccionamiento implícito no se indica la ubicación del operando
- d. En el direccionamiento inmediato el dato se encuentra en la propia instrucción

3. Si **rcx** vale -1, tras ejecutar la instrucción **sal \$3,%ecx** el nuevo valor de RCX es

- a. 0xffff ffff ffff ffff
- b. 0xffff ffff ffff fff8
- c. 0x1fff ffff ffff ffff
- d. 0xffff fff8

4. ¿Cuál de las siguientes instrucciones **NO** es errónea?

- a. movb \$0xF, (%rax)
- b. movw (%rbx), 4(%rsp)
- c. movl %rcx, (%rsp)
- d. movq %rdx, \$0x123

5. ¿Cuál de las siguientes instrucciones es **errónea**?

- a. mov (%rax,%rbx,1), %rcx
- b. lea (%rdx,%rsi,2), %rdi
- c. lea (%rbp,%rsp,4), %r8
- d. mov (%r9,%r10,8), %r11

6. La instrucción **movzbl %eax, %rax**

- a. pone a 0 el registro %rax
- b. copia en %rax el valor sin signo almacenado en %eax, rellenando con ceros
- c. copia en %rax el valor de %eax si el indicador de cero está activado
- d. no existe, se debe usar **mov %eax, %eax**

7. Sabiendo que las instrucciones de salto condicional codifican la dirección de salto con direccionamiento relativo a contador de programa (de 8 o 32 bits con signo), indicar cuál es la dirección de la instrucción **pop** en el siguiente desensamblado, donde se ha tachado la parte de las direcciones.

```
xxxxxx: 77 02    ja 400547
xxxxxx: 5d      pop %rbp
```

- a. 400547
- b. 400549
- c. 400545
- d. 400543

8. La instrucción `seta %al` (seta significa set if above):

- a. pone AL a 1 si ZF=1 y CF=1
- b. pone AL a 1 si ZF=1 o CF=0
- c. pone AL a 1 si ZF=0 o CF=1
- d. pone AL a 1 si ZF=0 y CF=0

9. La instrucción `JNGE / JL` provoca un salto si...

- a. SF = 1
- b. CF = 1
- c. SF ≠ OF
- d. SF = CF

10. Para traducir una asignación condicional (`a = b ? c : d ;`) de lenguaje C a lenguaje ensamblador, gcc puede que utilice...

- a. Un salto condicional, según la condición expresada en el código C, y otro salto incondicional
- b. Un salto incondicional, según la condición opuesta a la del código C, y otro salto condicional
- c. Una instrucción de movimiento incondicional, pero sólo si el procesador es Pentium Pro/II o superior
- d. Una instrucción de movimiento condicional, pero sólo si el S.O. es de 32 bits

11. El rasgo distintivo de la traducción “salta-en-medio” que gcc hace de un bucle `while` de lenguaje C a lenguaje ensamblador es...

- a. el salto incondicional hacia adelante
- b. el salto incondicional hacia atrás
- c. el salto condicional hacia adelante
- d. el salto condicional hacia atrás

12. El rasgo distintivo de la traducción “copia-test” que gcc hace de un bucle `while` de lenguaje C a lenguaje ensamblador es...

- a. el salto condicional hacia adelante, según la misma condición expresada en lenguaje C
- b. el salto condicional hacia atrás, según la misma condición expresada en lenguaje C
- c. el salto condicional hacia adelante, según la condición opuesta a la expresada en leng. C
- d. el salto condicional hacia atrás, según la condición opuesta a la expresada en leng. C

13. En la convención de llamada SystemV AMD64 seguida por gcc Linux/x86-64...

- a. R12 es un registro salva-invocado, por eso en cualquier función hay que salvarlo antes de modificarlo
- b. R11 es un registro salva-invocante, por eso en cualquier función hay que salvarlo antes de modificarlo
- c. RBX es un registro salva-invocado, por eso si es necesario conservar su valor hay que salvarlo antes de llamar a función
- d. RBP es un registro salva-invocante, por eso si es necesario conservar su valor hay que salvarlo antes de llamar a función

14. En el fragmento de programa siguiente:

```
66b: e8 00 00 00 00 callq 670 <nxt>
670: 58                pop %rax
```

¿Qué valor termina almacenado en `%eax`?

- a. 0x670
- b. 0xe8000000
- c. 0x66b
- d. 0x58

15. ¿Cuál sería el “equivalente x86-64” del “pseudo-código C” `rcx=((long*)rax)[rbx]`?

- a. `mov (%rax,%rbx,4), %rcx`
- b. `lea (%rax,%rbx,4), %rcx`
- c. `lea (%rax,%rbx,8), %rcx`
- d. `mov (%rax,%rbx,8), %rcx`

16. Habiendo declarado `int array={-4,-3,-2,-1};` y `char *ptr=array;` ¿cuánto vale `ptr[1]`?

- a. -1
- b. -2
- c. -3
- d. -4

17. En una unidad de control microprogramada, bits del registro IR dirigen una ROM cuya salida puede cargarse en el registro  $\mu$ PC (micro-PC). Esa unidad de control...

- a. es una UC con secuenciamiento explícito
- b. puede realizar la micro-operación `IR =  $\mu$ PC`
- c. puede realizar la  $\mu$ -op. `IR = ROM[MAR]`
- d. dispone de la funcionalidad “goto f(IR)”

18. Para el procesador con unidad de control microprogramada estudiado en clase, Tanenbaum propone codificar los N registros y añadir una señal “PERC” para habilitar la carga desde el bus C (recordar que era un diseño típico con 3 buses) y así

no perder expresividad/paralelismo. Si fuera  $N=8$ , el ahorro de bits en cada microinstrucción debido a esta técnica es de

- a. 35 bits
  - b. 34 bits
  - c. 25 bits
  - d. 14 bits
- 

19. Para el procesador con unidad de control microprogramada estudiado en clase, Tanenbaum propone codificar 16 microoperaciones sobre 2 de los 16 registros en lugar de especificar directamente las 24 señales de control y los 8 bits de dirección, aunque así se pierda paralelismo. El ahorro de bits en cada microinstrucción debido a esta técnica es de

- a. 24 bits
  - b. 20 bits
  - c. 12 bits
  - d. 8 bits
- 

20. Respecto a saltos retardados y anulantes, condicionales o no, **NO** sería apropiado intentar reordenar instrucciones...

- a. anteriores al condicional retardado para ponerlas después (en memoria)
  - b. en el destino del salto incondicional retardado para ponerlas después (en mem.) del salto
  - c. anteriores al condicional anulante para ponerlas después (en memoria)
  - d. en el destino del salto condicional anulante para ponerlas después (en mem.) del salto
- 

21. Una SRAM de 128Kx8bit (1Mbit) puede venir organizada en 1024 filas, dedicando por tanto al decodificador de columnas...

- a. 7 bits
  - b. 8 bits
  - c. 9 bits
  - d. 10 bits
- 

22. Respecto a las técnicas de direccionamiento por selección lineal, decodificación centralizada y distribuida

- a. todas ellas impiden que haya cortocircuito en el bus de datos
- b. todas ellas impiden que haya cortocircuito en el bus de direcciones

- c. la selección lineal permitiría leer una palabra simultáneamente desde varios puertos de E/S
  - d. usando decodificación distribuida es más fácil realizar expansiones al sistema de E/S
- 

23. La instrucción máquina DI (Disable Interrupts), conocida como CLI (Clear Interrupt Flag) en x86, se utiliza para desactivar:

- a. algunas interrupciones no enmascarables
  - b. algunas interrupciones externas
  - c. algunas excepciones o *traps*
  - d. algunos niveles de interrupción selectivamente
- 

24. ¿Cuál de las siguientes características es menos probable que pueda programarse en un canal DMA?

- a. dirección de memoria y dirección de E/S
  - b. si la dirección de la transferencia es hacia memoria, o hacia E/S
  - c. contador de memoria y contador de E/S
  - d. si la dirección de memoria se incrementa o decrementa, y si la dirección de E/S se incrementa o decrementa
- 

25. El ancho de banda de memoria es:

- a. el número de bits del bus de datos en el bus del sistema
  - b. la reunión (en paralelo) de los buses de datos de los módulos que forman la memoria
  - c. la velocidad a la que se pueden leer o escribir los datos en memoria
  - d. el intervalo de frecuencias de reloj permitidas para la CPU que vaya a conectarse a dicha memoria
- 

26. Una memoria estática tiene un bus de datos de 64 bits y su bus de direcciones es de 30 bits, ¿cuál es su capacidad?

- a. 80 MBytes
  - b. 1 GByte
  - c. 8 GBytes
  - d. 64 GBytes
- 

27. ¿Cuántas líneas de dirección (patillas) son necesarias para direccionar un chip de memoria DRAM de 16M x 16?

- a. 10
- b. 12

- c. 24
  - d. 28
- 

28. En un computador con una jerarquía de memoria de dos niveles se observa para una carga concreta que el tiempo medio de acceso a la memoria es de 900 ns cuando en realidad el tiempo de acceso al primer nivel es de 9 ns. Sabiendo que el tiempo de acceso al segundo nivel es de 3 microsegundos, ¿cuál sería aproximadamente el porcentaje de fallos en los accesos al primer nivel?

- a. 90%
  - b. 20%
  - c. 10%
  - d. 30%
- 

29. En una cache con correspondencia directa de  $2^p$  palabras y líneas de  $2^w$  palabras, el gestor de memoria **no** considera como campo (conjunto de bits contiguos con significado o relevancia) los siguientes bits:

- a. últimos  $w$  bits ( $0..w-1$ ) (los menos significativos)
  - b. bits  $w \dots w+p-1$
  - c. bits  $w \dots p-1$
  - d. primeros bits, desde el más significativo hasta el bit  $p$
- 

30. Un sistema monoprocesador con memoria de bytes y direcciones de 32 bits dispone de un único nivel de cache L1 compartida (instrucciones y datos) asociativa por conjuntos de 256KB. El tamaño del campo de etiqueta es

- a. 14 bits
  - b. depende del tamaño de línea
  - c. depende del número de vías
  - d. depende del número de vías y del tamaño de línea
-