

**Nombre:****DNI:****Grupo:****Examen Test (3.0p)****Todas las preguntas son de elección simple sobre 4 alternativas.****Cada respuesta vale 3/30 si es correcta, 0 si está en blanco o claramente tachada, -1/30 si es errónea.****Anotar las respuestas (a, b, c o 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. Si queremos almacenar la palabra de 64bits 0x0000001f ffffffe0 en una memoria de bytes según la convención little-endian a partir de la posición 0x0804913c, quedará
- a. 0x00 en 0x0804913c y 0xe0 en 0x08049143
  - b. 0x1f en 0x0804913c y 0xe0 en 0x08049140
  - c. 0xe0 en 0x0804913c y 0x1f en 0x08049140
  - d. Todas las respuestas anteriores son incorrectas

2. La diferencia entre el flag de acarreo y de overflow es que...
- a. uno se activa cuando se opera con números con signo y otro cuando son sin signo
  - b. el flag de acarreo indica que ha habido acarreo en una operación con números enteros (ints), el de overflow indica que ha habido desbordamiento en una operación con números en punto flotante (p.f.)
  - c. ambos se recalculan tras cada operación aritmético-lógica con ints, correspondiendo al programador consultar uno u otro según piense que sus datos son con o sin signo
  - d. el de acarreo indica que el resultado es demasiado grande (para p.f.) o positivo (si se trata de ints) para poder almacenarse, el de overflow indica que es demasiado pequeño (p.f.) o negativo (ints)

3. ¿Cuál de las siguientes secuencias de instrucciones multiplica el (contenido del) registro EAX por 18?
- a. imull \$0x18, %eax
  - b. leal (%eax,%eax,8), %eax  
leal (%eax,%eax), %eax
  - c. shll \$18, %eax
  - d. sarl \$1, %eax  
imul \$9, %eax

4. La instrucción IA32 test sirve para...
- a. Testear el código de condición indicado, y poner un byte a 1 si se cumple
  - b. Mover el operando fuente al destino, pero sólo si se cumple la condición indicada
  - c. Realizar la operación resta (a-b) pero no guardar el resultado, sino simplemente ajustar los flags
  - d. Realizar la operación and lógico bit-a-bit (a&b) pero no guardar el resultado, sino simplemente ajustar los flags

5. La instrucción leave equivale a:
- a. movl %esp, %ebp; popl %esp
  - b. movl %ebp, %esp; popl %ebp
  - c. pushl %esp; movl %ebp, %esp
  - d. pushl %ebp; movl %esp, %ebp

6. alguna de estas opciones contiene algún elemento que no corresponde a los contenidos de un marco de pila GCC/Linux IA32.

- a. Argumentos de llamada a la función y Dirección de retorno
- b. Antiguo marco de pila y Registros salva-invocante
- c. Variables locales y Registros salva-invocado
- d. Variables globales y Valor de retorno de la función

7. Respecto a registros salva-invocante y salva-invocado en GCC/Linux IA32, ¿cuál de éstos es de distinto tipo que el resto?

- a. EAX
- b. EBX
- c. ECX
- d. EDX

8. La arquitectura x86-64 tiene:

- a. 8 registros de propósito general (RPG) de 64 bits (%rax, %rbx, ... %rsp, %rbp)
- b. 16 registros RPG de 64 bits
- c. 32 registros RPG de 64 bits
- d. 64 registros RPG de 64 bits

9. ¿Cuál de las siguientes listas menciona registros x86-64 del mismo tipo respecto a convenio de uso? (salva-invocante, invocado, etc)

- a. RAX, RBX, RCX, RDX
- b. RBX, RSI, RDI
- c. RSP, RBP
- d. CL, DX, R8d, R9

10. La zona roja en x86-64 Linux es...

- a. una zona de pila en donde no deben escribir las funciones invocadas (porque si se produjera una interrupción, el manejador de interrupción sobrescribiría los valores escritos en pila)
- b. una zona de pila en donde pueden escribir las funciones invocadas, pero teniendo en

cuenta que los valores escritos pueden verse alterados si se produce una interrupción

- c. una zona bajo (RSP) (adonde apunta RSP) que una función puede usar sin reservarla, pero sólo si no llama a ninguna otra función mientras la usa
- d. una zona bajo (RBP) (adonde apunta RBP) que una función puede usar sin reservarla, pero sólo si no llama a ninguna otra función mientras la usa

11. alguna de las siguientes \*no\* es una operación básica de la unidad de control:

- a. Transferir un registro a otro
- b. (Leer o escribir) un registro (de / a) memoria
- c. (Guardar o recuperar) un registro (en / de) la pila
- d. Realizar operación ALU y guardar resultado en registro

12. Una cola de precaptación sirve para:

- a. Reducir el efecto de los fallos de cache
- b. Disminuir el periodo de ciclo del cauce
- c. Aumentar el número de etapas del cauce
- d. Resolver ciertos problemas de dependencia de datos

13. Sólo una de las siguientes afirmaciones sobre memorias ROM es correcta. ¿Cuál?

- a. Para fabricar una ROM se deben conocer los datos que se desea que almacene
- b. Una PROM (Programmable ROM) se puede grabar usando un dispositivo programador que selectivamente funde contactos aplicándoles altas temperaturas mediante diminutas cabezas soldadoras ("equipo de puntas")
- c. Una EPROM (Electrically Progr. ROM) se puede grabar eléctricamente, sin fundir contactos, pero no se puede borrar
- d. Una EEPROM (Erasable EPROM) se puede grabar (eléctricamente), y borrar (usando rayos ultravioleta)

14. Una SRAM de 32Kx8bit (256Kbit) puede venir organizada en 512 filas, dedicando por tanto al decodificador de columnas...

- a. 6 bits
  - b. 7 bits
  - c. 8 bits
  - d. 9 bits
- 

15. Para construir una DRAM de 4GB con pastillas de 512Mx4bit hacen falta

- a. 64 pastillas
  - b. 32 pastillas
  - c. 16 pastillas
  - d. 8 pastillas
- 

16. Una sentencia en C del tipo `"while (test) body;"` puede transformarse en código "goto" como:

- a. 

```
if (!test) goto done;
loop: body;
if (test) goto loop;
done:
```
  - b. 

```
loop: body;
if (test) goto loop;
```
  - c. 

```
if (test) goto true;
goto done;
true: body;
done:
```
  - d. 

```
loop: if (test) goto done;
body;
goto loop;
done:
```
- 

17. El cuerpo del siguiente código C:

```
unsigned copy(unsigned u) {return u;}
puede traducirse a ensamblador como:
```

- a. `movl 8(%ebp), %eax`
  - b. `movl %ebp, 8(%eax)`
  - c. `movl 8(%ebp), (%eax)`
  - d. `movl 8(%esp), %ebp`
- 

18. Si representamos la fase Decode con una D, Execute con una E, Fetch con una F y Writeback con una W, el orden correcto de

las distintas fases de una instrucción máquina es:

- a. D E F W
  - b. F D E W
  - c. F W D E
  - d. D F E W
- 

19. Si N es el número de instrucciones máquina de un programa, F es la frecuencia de reloj, y C el número promedio de ciclos por instrucción, el tiempo de ejecución del programa será:

- a.  $N \cdot F / C$
  - b.  $N \cdot F \cdot C$
  - c.  $N \cdot C / F$
  - d.  $N / (F \cdot C)$
- 

20. ¿Cuál de las siguientes expresiones representa un direccionamiento inmediato?

- a. `%eax`
  - b. `$0x400`
  - c. `(%eax)`
  - d. `8(%ebp)`
- 

21. La instrucción `movzbl %al, %eax`

- a. Pone a 0 el registro `%eax`
  - b. Copia en `%eax` el valor sin signo almacenado en `%al`, rellenando con ceros
  - c. Copia en `%eax` el valor del indicador de cero
  - d. Copia en `%eax` el valor de `%al` si el indicador de cero está activado
- 

22. Dentro de una función declarada como `void swap(int *xp, int *yp)`, que intercambia los valores de los dos enteros cuyas direcciones de memoria (punteros) son pasadas como parámetros a la función, la instrucción `movl 12(%ebp), %ecx` copia en `%ecx`...

- a. el valor del entero apuntado por el puntero pasado como primer parámetro
- b. el valor del entero apuntado por el puntero pasado como segundo parámetro

- c. el valor del puntero pasado como primer parámetro
  - d. el valor del puntero pasado como segundo parámetro
- 

**23.** Si la dirección del primer elemento de un vector de enteros  $z$  está almacenada en el registro `%edx` y la variable entera  $i$  está almacenada en el registro `%eax`, la instrucción máquina que realiza la operación  $z[i]++$  es:

- a. `addl $1, (%edx,%eax,4)`
  - b. `addl $1, (%eax,%edx,4)`
  - c. `addl $1, (%edx,%eax)`
  - d. `addl $4, (%eax,%edx)`
- 

**24.** En un camino de datos con un solo bus, para realizar la operación de copia de un registro  $r1$  en un registro  $r2$ , es decir  $r2 \leftarrow r1$ , es necesario:

- a. Activar la carga del registro  $r1$  y habilitar la salida triestado del registro  $r2$
  - b. Habilitar la salida triestado del registro  $r1$  y activar la carga del registro  $r2$
  - c. Habilitar las salidas triestado de los registros  $r1$  y  $r2$  y activar la carga del registro  $r2$
  - d. Habilitar la salida triestado del registro  $r2$  y activar la carga de los registros  $r1$  y  $r2$
- 

**25.** La salida de un campo del registro de microinstrucción que solapa dirección de salto y algunas señales de control han de conectarse a:

- a. una ROM o PLA
  - b. la memoria de control
  - c. el registro de instrucción
  - d. un demultiplexor controlado por el tipo de salto
- 

**26.** Cuando dos o más instrucciones necesitan un recurso hardware en el mismo ciclo, se trata de un riesgo:

- a. estructural
- b. por dependencia de datos
- c. de control

- d. de salto
- 

**27.** Una instrucción típica de entrada / salida tiene

- a. no tiene ningún parámetro
  - b. tiene un parámetro: un registro del procesador
  - c. tiene dos parámetros: un registro del procesador y una dirección de puerto de E/S
  - d. tiene tres parámetros: un registro del procesador, una dirección de puerto de E/S y una dirección de memoria
- 

**28.** ¿Cuál de las siguientes afirmaciones acerca del daisy-chain es cierta?

- a. Todos los componentes se conectan directamente y con igual prioridad al procesador o gestor de interrupciones
  - b. Los componentes se comportan de forma cooperativa: sólo al de mayor prioridad se le concede la interrupción o se apodera del bus de comunicaciones
  - c. Los componentes están conectados todos con todos y un gestor centralizado decide la prioridad
  - d. Es incompatible con la técnica de sondeo o polling
- 

**29.** La memoria DRAM:

- a. Se denomina dinámica porque para mantener almacenado un dato hay que recargarlo cada cierto tiempo en un ciclo de refresco
  - b. Es menos densa que la memoria SRAM
  - c. Se inventó en la década de los 90
  - d. Necesita 6 transistores por cada celda
- 

**30.** Los módulos de memoria dinámica compactos que suelen usarse en los portátiles se denominan:

- a. SIMM
  - b. SODIMM
  - c. SLIM
  - d. MIN
-