

1º de Grado en Ingeniería Informática (Grupos 1, 2, 3 y 4)
Examen final de Fundamentos de Computadores (convocatoria de enero)

29 de enero de 2015

EXAMEN COMPLETO (TEMAS 1, 2, 3, 4, 5 y 6)

Apellidos, Nombre: _____ **DNI:** _____ **Grupo(1/2/3/4):** __

Instrucciones para realizar el examen (tipo A)

- Utilice bolígrafo.
- Para las preguntas del test, seleccione una única respuesta en cada cuestión en el lugar habilitado para ello (señalando con una X en la tabla colocada al comienzo del test). Sólo se corregirán las respuestas pasadas a dicha tabla.
- Cada pregunta del test tiene una y sólo una opción de respuesta correcta.
- Cada dos respuestas incorrectas en el test anulan una correcta.
- Una pregunta sin contestar ni suma ni resta.
- Todos los alumnos deberán entregar esta hoja del test al acabarlo.

Parte I: tipo test (30.0%; 0.12 puntos por respuesta)

A	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25
a																									
b																									
c																									

T1. En una memoria RAM se accede a cada celda de 1 byte con una dirección que está conectada a un bus de direcciones de 32 bits. El tamaño máximo de la memoria es de:

- a) 4 MegaBytes.
- b) 4 GigaBytes.
- c) 32000 KiloBytes.

T2. Si una variable numérica de 4 bytes está almacenada en una memoria (direccionable byte a byte) con esquema de almacenamiento tipo *little-endian*:

- a) El byte menos significativo de la variable estará en la dirección más alta de los cuatro bytes.
- b) El byte menos significativo de la variable estará en la dirección más baja de los cuatro bytes.
- c) Cada dos bytes se le da la vuelta a éstos para recomponer ese valor de la variable.

T3. Para una escritura en memoria (por parte de la CPU):

- a) La CPU deja el dato en el bus de datos y la memoria le responde, por medio del bus de direcciones, en qué dirección lo ha almacenado.
- b) Según sea un esquema *little-endian* o *big-endian* la CPU deja el dato en el bus de direcciones o en el bus de datos.
- c) La CPU deja el dato a almacenar en el bus de datos y su dirección correspondiente en el bus de direcciones y luego le da una orden de escritura.

T4. El número 13_{16} (expresado en base 16 o hexadecimal) se corresponde con:

- a) $6,5_8$ (expresado en base 8 u octal).
- b) 23_8 (expresado en base 8 u octal).
- c) Dicho número es incorrecto (se tiene que expresar como D en hexadecimal).

T5. Sea el número de 8 bits AC_{16} , expresado en C2 de 8 bits. Si quisiéramos extenderlo a 32 bits, representando el mismo valor entero, tendríamos el siguiente valor de 32 bits (todas las secuencias de bits expresadas en hexadecimal):

- a) 000000AC.
- b) 111111AC.
- c) FFFFFFFAC.

T6. El resultado de hacer la operación lógica AND (bit a bit entre los bits de posiciones correspondientes), entre los bytes $F9_{16}$ y $A3_{16}$ será:

- a) $A1_{16}$
- b) FB_{16}
- c) $A3_{16}$

T7. En el formato IEEE-754 de 32 bits, un número no válido (NaN) tiene el siguiente valor en el campo del exponente (8 bits):

- a) 00000000
- b) 11111111
- c) 10000000

T8. Sea $F = \sum m(2,3,6,7)$. Dicha función lógica equivale a:

- a) $F(A,B,C) = A'B'C' + A'BC + A'BC' + ABC$
- b) $F(A,B,C) = A'B'C' + A'BC + AB'C' + ABC$
- c) $F(A,B,C) = A'BC' + A'BC + ABC' + ABC$

T9. Las salidas no determinadas (indeterminaciones) en el proceso de minimización:

- a) No se pueden usar, precisamente porque están indeterminadas.
- b) Se usan a voluntad como ceros o como unos en la minimización, tanto si minimizamos por suma de productos como por producto de sumas.
- c) Son minitérminos que han de minimizarse conjunta y separadamente del resto de salidas.

T10. En la implementación de circuitos combinacionales con sólo puertas NAND:

- a) Se minimiza similarmente a los circuitos por suma de productos, pero aplicando De Morgan se acaban usando dos niveles de puertas NAND.
- b) Se cubren por ceros en el mapa de Karnaugh y se llevan los implicantes a un nivel de puertas NAND y las salidas de este nivel a otra puerta NAND.
- c) No se puede minimizar con sólo puertas NAND pues nos haría falta usar inversores (puertas NOT).

T11. Una puerta XOR de dos entradas pondrá su salida a 1:

- a) Si sus dos entradas son iguales.
- b) Si sus dos entradas son distintas.
- c) Al revés que una OR: si alguna de sus entradas vale 1 su salida será 0.

T12. Al utilizar un multiplexor del tamaño apropiado (n entradas de control en el multiplexor para 2^n entradas de la función) para implementar una función lógica:

- a) Conecto las variables de la función a sus entradas de control y por las entradas de datos meto ceros o unos según la función sea 0 ó 1.
- b) Conecto o no sus salidas a una puerta OR con tantas entradas como minitérminos hagan 1 la función.
- c) Conecto las variables a las entradas de datos y sus salidas las conecto o no a una puerta OR dependiendo del valor de los minitérminos.

T13. Los archivos cuyo nombre comienza por un carácter punto ('.'), en Linux:

- a) Se muestran sólo si al comando `ls` se le pasa la opción `-R`.
- b) No se muestran con un comando `ls` ejecutado sin opciones adicionales.
- c) Se muestran sólo si se usa un comando de listado especial de archivos, distinto al comando `ls`.

T14. El único nombre de archivo (de entre los mencionados) que se correspondería con el comodín `[!c-f]*.txt` sería:

- a) `antonio.txt`
- b) `daniel.txt`
- c) `fichero.TXT`

T15. El efecto de ejecutar el comando Linux `find /` sería (suponer que se ejecuta con permisos de superusuario):

- a) Mostrar los nombres de los archivos y subdirectorios al primer nivel del directorio raíz.
- b) Listar recursivamente todos los archivos y subdirectorios que cuelgan del directorio raíz.
- c) Buscar todos los subdirectorios que cuelgan al primer nivel del directorio raíz.

T16. El programa `bash` es:

- a) Un comando para desensamblar archivos binarios.
- b) Uno de los múltiples GUI disponibles para Linux.
- c) Un intérprete de comandos.

T17. Los registros `%rsp`, `%al`, `%cx` y `%edx` de la arquitectura x86-64 tienen las siguientes longitudes:

- a) 32 bits, 8 bits, 16 bits y 64 bits, respectivamente.
- b) 64 bits, 8 bits, 16 bits y 32 bits, respectivamente.
- c) Todos son de 64 bits.

T18. Di cuál de la siguiente secuencia de la jerarquía de traducciones es la correcta:

- a) Compilador-Enlazador-Ensamblador-Cargador
- b) Compilador-Ensamblador-Cargador-Enlazador
- c) Compilador-Ensamblador-Enlazador-Cargador

T19. El registro RFLAGS del procesador x86-64:

- a) Es un registro de uso general, y como tal puede ser usado en cualquier instrucción que use otro registro general como, por ejemplo, `%rax`.
- b) Es un registro especial que es afectado (entre otras posibilidades) por el resultado de las comparaciones previas que se suelen hacer para decidir si se toma o no un salto condicional posterior.
- c) Es un registro especial que se usa como puntero de instrucción.

T20. El cuarto elemento de un array de enteros de 32 bits que comienza en la etiqueta `array`: (e.d., `array[3]`, si dicho array empieza en `array[0]`) se almacenará exactamente en:

- a) La dirección `X+4`, siendo `X` la dirección correspondiente a la etiqueta `array`:
- b) La dirección `X+12`, siendo `X` la dirección correspondiente a la etiqueta `array`:
- c) La dirección `X+3`, siendo `X` la dirección correspondiente a la etiqueta `array`:

T21. La instrucción en ensamblador del Intel x86-64 `shr $0x3, %rax` es de tipo:

- a) Instrucción de control (salto condicional).
- b) Instrucción aritmético-lógica (desplazamiento de bits).
- c) Transferencia de datos.

T22. Ordenando de menor a mayor ancho de banda los siguientes enlaces en redes, tendríamos:

- a) Par trenzado, cable coaxial y fibra óptica.
- b) Fibra óptica, par trenzado y cable coaxial.
- c) Tanto a) como b) son falsas.

T23. El *router* por defecto de una subred debe tener, obligatoriamente, como dirección IP interna:

- a) La dirección de difusión (*broadcast*) menos 1.
- b) Cualquier dirección comprendida entre la dirección de red y la de broadcast.
- c) La dirección resultante de hacer un AND de la dirección IP con la máscara.

T24. Entre las ventajas de hacer uso de redes NAT se encuentra la de:

- a) Poder convertir nuestro *host* en un servidor de páginas web.
- b) Que podemos asignarnos una dirección IP pública cualquiera a nuestro anteojo.
- c) Que sólo hacemos uso de una dirección IP externa para toda la subred.

T25. Una URL es:

- a) Una manera de localizar un objeto accesible en la web (URL = *Uniform Resource Locator*).
- b) Un protocolo para configurar comunicaciones P2P (URL = *Unique Rapid Lookahead*).
- c) Es un servicio para conocer a quién pertenece una IP cualquiera (URL = *Universal Range Locator*).

Soluciones test tipo A (todo)

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25
a						X				X		X		X								X			X
b	X	X		X			X		X		X		X		X		X		X	X	X		X		
c			X		X			X								X		X						X	