

1. ¿Cuál de las siguientes afirmaciones acerca del concepto de interrupción no es cierta?
 - a) Es una bifurcación normalmente externa al programa en ejecución
 - b) Su objetivo es reclamar la atención del procesador
 - c) Solicita que se ejecute un programa específico para tratarla
 - d) Ninguna de las anteriores

2. Un procesador con una unidad de control microprogramada tiene una memoria de control de 256 palabras de 16 bits, de las que 128 son diferentes. ¿Qué ahorro en número de bits obtendríamos si usáramos nanoprogramación?
 - a) No se produce ahorro
 - b) 3840 bits
 - c) 256 bits
 - d) 4096 bits

3. ¿Cuál de las siguientes listas está correctamente ordenada temporalmente?
 - a) 486, 8086, Core 2, Pentium III, Pentium 4, Pentium MMX
486 posterior a 8086
 - b) 486, 8086, Pentium MMX, Core 2, Pentium III, Pentium 4
486 posterior a 8086
 - c) 8086, 486, Pentium MMX, Pentium III, Pentium 4, Core 2
 - d) 8086, 486, Pentium III, Pentium MMX, Core 2, Pentium 4
Pentium III posterior a Pentium MMX

4. ¿Cuál es el contenido de la pila al terminar de ejecutarse la siguiente secuencia de instrucciones de una arquitectura de pila?:
push #4
push #7
push #8
add
push #10
sub
mul
 - a) 4, 7, 8, 10
 - b) 4, 7, 48
 - c) 20
 - d) 4

5. 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:

- a) $x > 9$
 - b) $x < 10$
 - c) $x \leq -1 \mid \mid x \geq 11$
 - d) $x \geq 0 \ \&\& \ x \leq 10$
6. ¿Cuál de los siguientes microprocesadores no es de 64 bits?
- a) Itanium
los 64bits "primeros" de Intel
 - b) Core i7
 - c) Core 2
posterior a los Pentium
 - d) **Pentium III**
anterior a Pentium 4e año 2004 primer EM64T
7. ¿Cuántos controladores de interrupciones 8259 hacen falta como mínimo para manejar 25 líneas de interrupción?
- a) **4**
 - b) 3
 - c) 2
 - d) 1
8. El Intel 8086:
- a) Fue el primer microprocesador.
de Intel de 16 bit
 - b) **Incluía instrucciones de multiplicación.**
mul e imul, en concreto
 - c) Contaba con 29 millones de transistores.
29K, no 29M
 - d) Podía direccionar 1 GB.
1MB, no 1GB

9. Un fragmento del desensamblado de una “bomba” similar a las estudiadas en prácticas es:

```
400634: lea 0x4(%rsp),%rsi
400639: lea 0x1a4(%rip),%rdi
          # 4007e4 <_stdin+0xe4>
400640: mov $0x0,%eax
400645: callq 4004c0 <scanf>
40064a: mov 0x4(%rsp),%eax
40064e: shr %eax
400650: jc 40066a <L6>
400652: shr %eax
400654: jnc 40066a <L6>
400656: shr %eax
400658: jc 40066a <L6>
40065a: shr %eax
40065c: jnc 40066a <L6>
40065e: shr %eax
400660: jc 40066a <L6>
400662: shr %eax
400664: jnc 40066a <L6>
400666: test %eax,%eax
400668: jz 40066f <L7>
40066a: callq 4005c7 <boom>
40066f: callq 4005eb <defused>
```

El código numérico correcto (pin) es...

- a) 42
- b) el entero 0x40066A
- c) el entero almacenado a partir de 0x4(%rsp)
- d) ninguno de los anteriores

10. Respecto a las unidades de control nanoprogramadas:

- a) El diseño de las unidades de control nanoprogramadas debe ser vertical.
- b) 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.
- c) La realización nanoprogramada de una unidad de control es más rápida que la microprogramada.
- d) 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 $(n \cdot m + 2^m \cdot w) - n \cdot w$.

11. Una máquina superescalar es aquella que:
- a) basa su funcionamiento en la segmentación software como forma de incrementar el paralelismo.
 - b) las instrucciones tienen un campo por cada unidad funcional al realizarse varias operaciones por instrucción.
 - c) emite simultáneamente múltiples instrucciones por ciclo de reloj, por ejemplo, una entera y otra de coma flotante.
 - d) ninguna respuesta de las anteriores es correcta.
12. ¿Cuál de las siguientes afirmaciones acerca del concepto de interrupción no es cierta?
- a) Es una bifurcación normalmente externa al programa en ejecución
 - b) Su objetivo es reclamar la atención del procesador
 - c) Solicita que se ejecute un programa específico para tratarla
 - d) Ninguna de las anteriores
13. ¿Cuál de los siguientes elementos no forma parte de un canal de un controlador de acceso directo a memoria?
- a) Registro de dirección.
 - b) Registro de prioridades.
- En tr.82 figura FUERA de los canales, porque la prioridad sería cuando hay VARIOS canales, no tiene sentido hablar de prioridad DENTRO de un canal
- c) Registro contador.
 - d) Todos los elementos anteriores forman parte de un canal de DMA.
14. En la práctica de la bomba, el segundo ejercicio consistía en crear un ejecutable sin “explosiones”, para lo cual se puede utilizar... (marcar la opción *falsa*)
- a) gdb
 - b) ddd
 - c) hexedit
 - d) objdump
15. El circuito que hay que montar en la práctica del zumbador pasivo con Arduino es un:
- a) Buffer triestado
 - b) Divisor de tensión
 - c) Seguidor de voltaje
 - d) Comparador

16. Si el acceso directo a memoria se realiza mediante robo de ciclo:
- a) es posible que la ejecución de una operación elemental de transferencia en el bus sea temporalmente detenida
 - b) es posible que la ejecución de una instrucción máquina sea temporalmente detenida
 - c) es necesario que termine de ejecutarse la instrucción máquina actual para comenzar una transferencia por DMA
 - d) ninguna de las anteriores es cierta
17. ¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC (modo real) es cierta?
- a) Existen 1024 vectores de interrupción.
 - b) vector de interrupción es una doble palabra de 32 bits formada en primer lugar (dirección menor) por el segmento y seguida por el desplazamiento (dirección mayor) de cada rutina de servicio de interrupción
 - c) No todas las interrupciones se pueden generar por software
 - d) Ninguna de las anteriores afirmaciones es cierta
18. En una unidad de control microprogramada se tienen dos campos de 5 señales de control mutuamente exclusivos, de manera que nunca se activarían señales de ambos campos en el mismo ciclo de reloj. Con esas 10 señales sería entonces posible...
- a) codificarlas con 4 bits, y sobraría un código que quedaría sin uso
 - b) codificarlas con 5 bits, y no sobraría ningún código sin uso
 - c) solaparlas en un solo campo de 5 bits, ahorrando por tanto 4 bits
 - d) solaparlas en un solo campo de 6 bits, ahorrando por tanto 5 bits
19. En los casos concretos indicados para las siguientes instrucciones x86-64, ¿cuál no funciona como instrucción de transferencia?
- a) `pushq %rax`
transfiere RAX a la pila
 - b) `movq $0x15, %rax`
Ajusta RAX a 15 (transfiere el valor inmediato)
 - c) `cmpq 0x6000f0, %rax`
No mueve datos, sólo ajusta flags, no es transferencia, es op.aritmética
 - d) `leaq variable, %rax`
Ajusta RAX a 0x6000... (transfiere la dirección de la variable)

20. Un computador con 15 líneas de direcciones tiene 3 módulos de memoria de 2^{13} palabras y utiliza E/S mapeada en memoria. ¿Cuál es el número máximo de periféricos que pueden conectarse, si cada uno de ellos utiliza 8 direcciones?

- a) 2^{10}
- b) 2^{13}
- c) 2^{11}
- d) 2^{12}

21. ¿Cuál de las siguientes instrucciones máquina copia en RAX la dirección efectiva resultante de la operación $RDX \cdot 8 + RBX$?

a) `leaq (%rbx, %rdx, 8), %rax`

lea calcula la dirección efectiva $[RBX + RDX \cdot 8]$ y la almacena en RAX
También podría hacerlo con direcciones 32b, pero no lo hemos visto en clase

Se ha preferido cambiar a registros 64bits, factor escala 8x

En la pregunta de examen original (Jul'17) se mencionaban regs.32b, factor 4x

b) `movq (%rbx, %rdx, 8), %rax`
mov mueve desde memoria

c) `leaq 8(%rdx, %rdx), %rax`
la dirección efectiva sería $8 + 2 \cdot RDX$

d) `movq 8(%rdx, %rdx), %rax`
mov mueve desde memoria, y además la dirección efectiva sería $8 + 2 \cdot RDX$

22. La extensión de signo a m bits de un número original N de n bits, con $m > n$, consiste en:

a) Realizar la operación $2^m - N$

b) Realizar la operación $2^m - N - 1$

c) Incrementar la cantidad de bits a m preservando el signo y el valor del número.

d) Incrementar la cantidad de bits a m rellenando con unos por la izquierda.

23. Al traducir de lenguaje C a ensamblador, gcc en máquinas Linux/x86-64 almacena (reserva espacio para) una variable “var” local a una función “fun” en una dirección de memoria referenciable (en lenguaje ensamblador) como:
- a) var (el nombre de la variable representa su posición de memoria)
no, eso sería para variables globales con gcc -m32 (ahora gcc viene con --enable-default-pie)
 - b) var(%rip)
eso sería una variable global (en Position-Independent-Executable)
 - c) k(%rsp), siendo k un número entero relativamente pequeño
sí, positivo (normal) o negativo (zona roja), aunque no hayamos visto zona roja en CS:APP3e
 - d) k(%rbp), siendo k un entero positivo relativamente pequeño
no, ni siquiera cuando gcc usara puntero marco (por tamaño variable del marco, ver CS:APP3e 3.10.5 p.326-329) ni si opcionalmente se compilara con -fno-omit-frame-pointer, porque en todo caso sería -k(%rbp)
24. En x86-64, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto al resto?
- a) `mov $-1, %edi`
mov no afecta a los flags, los demás sí: no hace falta comprobar más calculamos los flags por curiosidad, no porque haga falta para responder
 - b) `xor %edi, %edi`
`add $-1, %edi`
 - c) `sub %edi, %edi`
`adc $0xFFFFFFFF, %edi`
sub/adc afectan: CZSO=0100 -> 0010
 - d) `mov $-1, %edi`
`add $0, %edi`
add afecta: CZSO=0010
25. ¿Cuál de las siguientes afirmaciones es correcta?
- a) El lenguaje máquina es igual para todos los computadores.
 - b) Las instrucciones en lenguaje máquina se almacenan y tratan como cadenas de unos y ceros.
 - c) El lenguaje ensamblador es igual para todos los computadores.
 - d) Las instrucciones en lenguaje ensamblador se almacenan y tratan como cadenas de unos y ceros.

26. Si el contenido de RBP es 0x13000, ¿a qué dirección se hace referencia con la instrucción INC -0x80(%RBP)?
- a) 0x12F80
 - b) 0x80
 - c) 0x13080
 - d) 0x13000
27. La técnica de Consulta de Estado (polling) puede usarse para...
- a) identificar el origen de una interrupción
 - b) consultar si el dispositivo está dispuesto para entregar o recibir datos
 - c) consultar el sentido del DMA en curso (desde memoria / hacia memoria)
 - d) establecer un mecanismo software de asignación de prioridades a los dispositivos
28. Suponiendo que varios dispositivos comparten una única línea de solicitud de interrupción y que varios de ellos solicitan una interrupción al mismo tiempo, ¿qué dispositivo tendría mayor prioridad a la hora de ser atendidas sus peticiones?
- a) Si se emplea una técnica de sondeo ("polling"), el dispositivo cuyo estado se consulte primero
 - b) Si se emplea una técnica de encadenamiento ("daisy-chain"), el dispositivo al que primero se conecta la línea INTA# del procesador
 - c) Las respuestas a y b son ciertas
 - d) Las respuestas a y b son falsas
29. Al colocar un led en la placa de prototipado de Arduino, ¿cómo se sabe cuáles son el ánodo y el cátodo?
- a) Mirando las leyendas A+ o K- impresas en las patillas del led
 - b) No es necesario saberlo, la polaridad es indiferente a la hora de colocar un led en un circuito
 - c) Mirando los códigos de color dibujados sobre el encapsulado
 - d) Mirando la longitud de las patillas o qué lado del encapsulado es plano
30. Respecto a registros base e índice en x86-64, la excepción es que
- a) RBP no puede ser registro base
 - b) RBP no puede ser registro índice
 - c) RSP no puede ser registro base
 - d) RSP no puede ser registro índice
31. La microprogramación vertical se caracteriza por tener:
- a) microinstrucciones largas
 - b) escaso o ningún solapamiento entre campos
 - c) capacidad para expresar un alto grado de paralelismo en las microoperaciones a ejecutar
 - d) mucha codificación

32. ¿Qué modo de funcionamiento permite a la interfaz de periféricos programable 8255 utilizar un bus bidireccional?
- a) 2
 - b) 0
 - c) 3
 - d) 1
33. Se ha declarado en un programa C la variable `int val[5]={1,5,2,1,3};` ¿Cuál de las siguientes afirmaciones es cierta?
- a) `val[1] == 1`
 - b) `&val[3] == (char *)val + 12`
 - c) `sizeof(val) == 5`
 - d) `&val[2]` es de tipo `int *` y vale lo mismo que `val + 8`
34. Un computador usa el formato vertical de codificación de instrucciones para parte de las señales de control y el formato horizontal para k señales de control. El formato vertical posee n campos codificados de m bits cada uno. ¿Cuál es el máximo número de señales de control que pueden usarse en este computador?
- a) $k + n \cdot 2^m$
 - b) $k + n^m$
 - c) $k + n \cdot (2^m - 1)$
 - d) Ninguno de los anteriores
35. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de `popcount5`:

```
int popcount5(unsigned* array, size_t len)
{
    size_t i,j;
    int x;
    int val, result=0;

    for (i=0; i<len; i++)
    {
        x = array[i];
        val = 0;
        for (j=0; j<=8; j++)
        {
            val += x & 0x01010101;
            x >>= 1;
        }
        val += (val >> 16);
        val += (val >> 8);
    }
}
```

```

    result+= val & 0xFF;
}
return result;
}

```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en las condiciones del for interno y el tipo de x.

Esta función popcount5 fallaría:

- a) con array={1, 16, 256, 4096}
- b) con array={0x80000000, 0x00400000, 0x00000200, 0x00000001}
- c) con ambos ejemplos
- d) con ninguno de los dos ejemplos

36. Las instrucciones JB y JNAE provocan un salto si...

- a) CF == 1
- b) ZF == 0
- c) ZF != SF
- d) SF == 1

37. ¿Qué técnica de E/S se dice controlada por hardware?

- a) E/S programada
- b) E/S controlada por interrupciones
- c) Acceso directo a memoria
- d) Ninguna de las anteriores

38. ¿Cuál de las siguientes afirmaciones sobre el benchmark SPEC CPU es falsa?

- a) La última versión es SPEC CPU2017
- b) Se cronometran unos 10 tests de enteros y unos 14 tests de punto flotante
- c) Se usa como referencia un computador UltraSPARC-IV+ 2100MHz, y para cada test se calcula una velocidad cociente entre el tiempo de ejecución en el computador de referencia y en el computador a testear
- d) El resultado final es la media aritmética de las (10 ó 14) velocidades, bien sea de enteros ó de punto flotante (SPECspeedINT ó SPECspeedFP)

39. ¿Qué método de control de acceso directo a memoria es preferible por velocidad (más rápida), economía (coste no prohibitivo) y conveniencia de diseño (compatible con memorias y sistemas actuales)?

- a) Memoria multipuerto
- b) Transferencia de bloques o parada de CPU
- c) DMA intercalado o transparente
- d) Robo de ciclo

40. Un computador tiene una memoria de control de 640 palabras de 70 bits, de las que 280 son diferentes. ¿Qué ahorro en número de bits obtendríamos si usáramos nanoprogramación en lugar de microprogramación?
- a) 19440
 - b) 42280
 - c) 9840
 - d) ninguno de los anteriores resultados es exacto.
41. El sufijo q de la instrucción movq significa:
- a) Que la instrucción realiza un movimiento rápido, trabajando con la mitad menos significativa de los operandos (move quick).
 - b) Que la instrucción realiza un movimiento rápido, trabajando con la mitad más significativa de los operandos (move quick).
 - c) Que la instrucción trabaja con operandos de 64 bits (quad word).
 - d) Que la instrucción trabaja con operandos de 128 bits (quad word).
42. Un controlador de DMA suele ser programado con la siguiente información relativa a una operación de E/S:
- a) tipo de operación, tamaño de bloque a transferir, dirección final de memoria
 - b) tipo de operación, tamaño de bloque a transferir, dirección inicial de memoria
 - c) tipo de operación, dirección inicial de memoria, dirección final de memoria
 - d) Ninguna de las anteriores respuestas es cierta
43. ¿En qué tipos de técnicas de E/S la transferencia de información está bajo el control directo de la CPU?
- a) E/S programada y E/S controlada por interrupciones
 - b) E/S programada y acceso directo a memoria
 - c) E/S controlada por interrupciones y acceso directo a memoria
 - d) Ninguna de las anteriores respuestas es correcta
44. ¿En qué generación, dentro de la historia de los computadores digitales, aparece la memoria cache?
- a) Primera
 - b) Segunda
 - c) Tercera
 - d) Cuarta

45. Suponga la siguiente sentencia asm en un programa:

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

¿Cuál de las siguientes afirmaciones es correcta?

- a) a es una posición de memoria de entrada
 - b) i es un registro de entrada**
 - c) el código de retorno de la función asm se fuerza a que esté en la variable result
 - d) r es una posición de memoria de entrada/salida
46. Si AX = 0xFA50 y ejecutamos XOR \$0xFF, AX
- a) Se realiza el complemento a 1 de AH.
 - b) Se realiza el complemento a 1 de AL.**
 - c) El registro AL se pone a 0.
 - d) El registro AH se pone a 0
47. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros en doble precisión. El programa esqueleto ofrecido (suma.s) no es válido, en concreto...(marcar la opción FALSA)
- a) no está preparado para sumar más de 9 elementos**
 - b) no hace extensión con ceros de los elementos
 - c) no hace extensión de signo de los elementos
 - d) no consulta ni el flag de acarreo CF ni el de overflow OF
48. En 80x86, los parámetros a las subrutinas se pueden pasar:
- a) a través de variables globales
 - b) a través de los registros
 - c) a través de la pila
 - d) todas las anteriores son ciertas**
49. En la convención SysV AMD64 estándar para arquitecturas x86 de 64 bits, el resultado de una función se devuelve usualmente en el registro:
- a) RBX
 - b) RBP
 - c) RAX**
 - d) RSI

50. ¿Qué hace gcc -O?
- a) **Compilar con optimización suave**
 - b) Compilar .c→.o (fuente C a objeto)
 - c) Compilar .s→.o (fuente ASM a objeto)
 - d) Ambas (b) y (c), según la extensión de los ficheros que se usen como argumentos
51. Cuando se produce una interrupción hardware...
- a) **Se guarda el estado y se ejecuta la rutina de interrupción asociada**
 - b) Se salta a la dirección de memoria indicada en la instrucción actual
 - c) Se aborta la ejecución del programa actual generando un fallo de segmentación
 - d) Se salta al principio del programa actual
52. En un programa en ensamblador queremos crear espacio para una variable entera var inicializada a 1. La línea que hemos de escribir en la sección de datos es:
- a) .int var 1
 - b) **var: .int 1**
 - c) .int: var 1
 - d) int var 1
53. ¿Cuál de las siguientes afirmaciones es falsa?
- a) Las subrutinas necesitan ser capaces de reservar espacio en memoria para las variables locales sin sobrescribir ningún dato usado por el programa que hace la llamada
 - b) **Las subrutinas necesitan recibir parámetros desde el programa que hace la llamada que indiquen qué registros pueden alterar y cuáles no**
 - c) Los programas necesitan una forma de pasar parámetros a las subrutinas y de recibir las salidas de vuelta
 - d) Las subrutinas necesitan algún modo de saber desde dónde han sido llamadas para poder volver al programa que realizó la llamada cuando se completa la subrutina
54. alguna de las siguientes NO es una ventaja de la E/S independiente (separada, aislada)
- a) Protección de E/S más fácil (E/S mapeada añade dificultad a la protección de E/S)
 - b) **Diseño del procesador más sencillo (E/S mapeada añade complejidad al diseño)**
 - c) Mayor aprovechamiento del espacio de memoria (E/S mapeada resta espacio a la memoria)
 - d) Decodificación de memoria más elegante, limpia, sencilla (E/S mapeada añade complejidad a la decodificación)

55. ¿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) Es incompatible con la técnica de sondeo o polling
 - c) Los componentes se comportan de forma cooperativa: sólo al de mayor prioridad se le concede la interrupción o se apodera del bus de comunicaciones
 - d) Los componentes están conectados todos con todos y un gestor centralizado decide la prioridad
56. ¿Cuáles de las siguientes señales son entradas a la unidad de control?
- a) El contenido del contador de programa
 - b) Las señales de habilitación de buffers triestado entre registros y buses
 - c) El contenido del registro de instrucción
 - d) Las señales de control de la ALU
57. La predicción de saltos está relacionada con...
- a) Los riesgos de (dependencia de) datos (intenta que el dato esté disponible anticipadamente)
 - b) Los riesgos de transferencia (intenta agrupar las posibles transferencias de un conjunto de instrucciones)
 - c) Los riesgos estructurales (intenta evitar el efecto de un fallo de cache)
 - d) Los riesgos de control (intenta determinar de antemano el flujo de control)
58. En el programa "size" de la práctica de la cache, si el primer escalón pasa de tiempo = 1 para todos los tamaños de vector menores o iguales que 32 KB a tiempo = 3 para los tamaños 64 KB y 128 KB, podemos asegurar que:

(Nota: en las opciones de respuesta, "rapidez" se refiere a T_i , el tiempo de acceso efectivo al nivel i , no a t_i , el tiempo de acceso propio al nivel i . Si dice "La cache L1 es 3x más rápida que L2", se refiere a que $T_1 = T_2/3$, no a que $t_1 = t_2/3$. Ver T6 tr.26)

-
- a) la cache L2 es como mucho el doble de rápida que la memoria principal
 - b) la cache L1 es al menos tres veces más rápida que la cache L2
 - c) la cache L2 es al menos el doble de rápida que la memoria principal
 - d) la cache L1 es como mucho tres veces más rápida que la cache L2

59. Si un computador X ejecuta un programa de 450 millones de instrucciones en 26 segundos y un computador Y tarda 14 segundos en ejecutar ese mismo programa, ¿cuántas veces es más rápido el computador Y que el X?
- a) 1,538
 - b) 0,538
 - c) 1,857
 - d) 12
60. Para direccionar una memoria de 16K x 16 necesitamos un bus de direcciones de:
- a) 14 bits
 - b) 16 bits
 - c) 4 bits
 - d) Otro valor
61. Si queremos almacenar la palabra de 16 bits 0x8965 en una memoria de bytes según "big-endian", quedará almacenada a partir de la posición 0x1000 como:
- a) $M[0x1000]=0xA6$ y $M[0x1001]=0x91$
 - b) $M[0x1000]=0x65$ y $M[0x1001]=0x89$
 - c) $M[0x1000]=0x89$ y $M[0x1001]=0x65$
 - d) $M[0x1000]=0x91$ y $M[0x1001]=0xA6$
62. ¿Cuál de las siguientes afirmaciones sobre memorias cache es cierta? Recordar que llamamos "latencia" al tiempo transcurrido desde que se envía una dirección a la cache hasta que se obtiene el dato (suponiendo que se trata de un acierto), y depende por tanto del tiempo que emplee la circuitería hardware en obtenerlo. Suponer similares condiciones para ambas caches (tamaño de bloque, tamaño total, tecnología, frecuencia...)
- a) Las caches totalmente asociativas ofrecen mejor latencia, mientras que las de correspondencia directa tienen tasas de fallo más bajas
 - b) Las caches totalmente asociativas ofrecen tasas de fallo más bajas, mientras que las de correspondencia directa tienen mejor latencia
 - c) Ambas tienen en general similar latencia y tasa de fallos
 - d) Las caches de correspondencia directa tienen mejor latencia y tasa de fallos
63. En esta asignatura... (marcar la opción FALSA)
- a) se podrían acumular $NC=2p$ y $NL=2p$ respondiendo tests
 - b) Hay que obtener $N_{Teo} \geq 2.5p$ y $N_{Pra} \geq 2.5p$, para poder sumar teoría con prácticas y poder aprobar (obtener nota final no suspenso)
 - c) se pueden entregar ejercicios y prácticas aunque cuentan muy poca o ninguna nota
 - d) se podría obtener 10p en el examen final sin asistir un solo día ni a clase ni a prácticas

64. Respecto al sistema de Entrada / Salida, ¿cuál de las siguientes afirmaciones es incorrecta?
- a) Un protocolo sirve para “ponerse de acuerdo” en cosas como velocidad, paridad, nº de bits, etc.
 - b) La mayoría de los periféricos trabajan a velocidad muy superior a la CPU; por eso es necesario sincronizar.
 - c) Un controlador se encarga de la comunicación con la CPU.
 - d) La CPU se comunica con el periférico por medio del controlador y de software de E/S.
65. ¿Es cierto que la consulta de estado permite averiguar el estado de un periférico y alterar la prioridad de los periféricos?
- a) Sí, permite averiguar el estado de un periférico y también alterar la prioridad de los periféricos
 - b) No, porque no permite ni averiguar el estado de un periférico ni alterar la prioridad de los periféricos
 - c) No, porque no permite averiguar el estado de un periférico
 - d) No, porque no permite alterar la prioridad de los periféricos
66. El primer nivel de una jerarquía de memoria tiene una tasa de aciertos del 80% y las peticiones de memoria tardan 15 ns en completarse si dicha posición se encuentra en ese nivel y 100 ns si no es así. ¿Cuál es el tiempo medio de acceso de la jerarquía?
- a) 23 ns
 - b) 32 ns
 - c) 57.5 ns
 - d) 92 ns
67. Cuando dos o más instrucciones necesitan un recurso hardware en el mismo ciclo, se trata de un riesgo:
- a) estructural
 - b) de control
 - c) de salto
 - d) por dependencia de datos
68. En el arbitraje de un bus...
- a) los dispositivos pasivos pueden requerir el uso del bus para iniciar una transferencia
 - b) si hay un único dispositivo pasivo, siempre funciona como esclavo
 - c) si hay varios dispositivos activos, siempre funcionan como maestros
 - d) todas las respuestas anteriores son ciertas

69. Si un procesador no segmentado necesita 5 ns para leer una instrucción de memoria, 2 ns para decodificar la instrucción, 3 ns para leer del banco de registros, 3 ns para realizar el cálculo requerido por la instrucción, y 2 ns para escribir el resultado en el banco de registros, ¿cuál es la frecuencia de reloj máxima del procesador?
- a) 66,67 MHz
 - b) 500 MHz
 - c) 200 MHz
 - d) 40 MHz
70. ¿Cuál es el resultado de evaluar la expresión $0b1110 \wedge 0b1010$ en lenguaje C?
(0b indica que el número está expresado en binario)
- a) 0b1010
 - b) 0b1111
 - c) 0b0100
 - d) 0b0110
71. De los siguientes grupos de técnicas de E/S, ¿cuáles están controlados por programa?
- a) E/S programada y E/S mediante interrupciones.
 - b) E/S programada y E/S mediante acceso directo a memoria.
 - c) E/S mediante interrupciones y E/S mediante acceso directo a memoria.
 - d) Todas las respuestas anteriores son ciertas.
72. ¿Cuál de las siguientes listas menciona registros x86-64 del mismo tipo respecto a convenio de uso? (salva-invocante, invocado, etc)
- a) RBX, RSI, RDI
 - b) RAX, RBX, RCX, RDX
 - c) CL, DX, R8d, R9
 - d) RSP, RBP
73. Se ha declarado en un programa C la variable `int val[5]={1,5,2,1,3}`. ¿Cuál de las tres primeras opciones (a, b, o c) es FALSA?
- a) `val[1] == 1`
 - b) `&val[3] == val+3`
 - c) `sizeof(val) == 20`
 - d) No se puede marcar ninguna de ellas, todas (a, b y c) son ciertas

74. ¿Cuál de las siguientes es una idea fundamental de la jerarquía de memoria?
- a) Que dispositivos más pequeños y rápidos sirvan de cache para dispositivos más grandes y lentos
 - b) Que dispositivos más grandes y lentos sirvan de cache para dispositivos más pequeños y rápidos
 - c) Crear una gran cantidad de almacenamiento que sea caro y rápido
 - d) Crear una pequeña cantidad de almacenamiento que sea caro y lento
75. ¿Cuál de las siguientes afirmaciones es cierta?
- a) La E/S independiente facilita la protección
 - b) La E/S en memoria es mucho más rápida que la E/S independiente
 - c) La E/S en memoria emplea la patilla IO/M#
 - d) En E/S independiente, las instrucciones de acceso a memoria se emplean tanto para memoria como para E/S
76. Respecto a salvaguardar los registros de la CPU al inicio de una rutina de servicio de interrupción (ISR)
- a) se deben guardar los registros salva-invocado (p.ej. EBX, ESI, EDI en el caso de una CPU IA32), los registros salva-invocante ya los guarda el programa interrumpido
 - b) no es necesario salvar ninguno más, si el contador de programa y los flags de estado ya los salva la propia CPU como parte del mecanismo de interrupción
 - c) se deben guardar los registros que se modifiquen en la propia ISR. Eso es posible hacerlo porque el propio programador de la ISR conoce qué registros va a modificar
 - d) se deben guardar todos los registros, para restaurarlos a la salida y así garantizar que el programa interrumpido no sufre ninguna modificación (salvo el inevitable retraso temporal) debido a la interrupción
77. Según la clasificación m/n, las máquinas de acumulador son de tipo
- a) 0/0
 - b) 2/2 ó 2/3
 - c) 1/1
 - d) ½

78. 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 EEPROM (Erasable EPROM) se puede grabar (eléctricamente), y borrar (usando rayos ultravioleta)
 - d) Una EPROM (Electrically Progr. ROM) se puede grabar eléctricamente, sin fundir contactos, pero no se puede borrar
79. Ventajas de la E/S independiente (separada, aislada) (señalar la opción incorrecta)
- a) protección de E/S más fácil (E/S mapeada añade dificultad a la protección de E/S)
 - b) decodificación de memoria más elegante, limpia, sencilla (E/S mapeada añade complejidad a la decodificación)
 - c) mayor aprovechamiento del espacio de memoria (E/S mapeada resta espacio a la memoria)
 - d) diseño del procesador más sencillo (E/S mapeada añade complejidad al diseño)
80. En una jerarquía de memoria, a medida que nos alejamos del procesador:
- a) el coste por byte aumenta
 - b) el tamaño de la unidad de transferencia entre dos niveles aumenta
 - c) el tiempo de transferencia disminuye
 - d) el tamaño de la memoria disminuye
81. ¿Cuál es el tamaño de la marca de cache si el bus de direcciones es de 48 bits (256 TB de memoria principal) y hay 8MB de cache L3, con un tamaño de línea de 64 B y correspondencia asociativa por conjuntos con 16 vías?
- a) 29 bits
 - b) 48 bits
 - c) 13 bits
 - d) 6 bits

82. Un procesador con una unidad de control microprogramada tiene una memoria de control de 340 palabras de 16 bits, de las que 180 son diferentes. ¿Qué ahorro en número de bits obtendríamos si usáramos nanoprogramación?
- a) 19440 bits
 - b) 5260 bits
 - c) No se produce ahorro
 - d) 2560 bits
83. En la secuencia de programa siguiente:
- ```
400544: e8 07 00 00 00 callq 400550 <mult2>
400549: 48 89 03 mov %rax,%rbx
```
- ¿cuál es el valor que introduce en la pila la instrucción call?
- a) 0x400550
  - b) 0x400549
  - c) 0x40054b
  - d) 0x400544
84. En una arquitectura de registros de propósito general (a nivel de lenguaje máquina):
- a) operar usando registros es más rápido.
  - b) la generación de código resulta más simple que en arquitecturas de pila o acumulador.
  - c) se evita el cuello de botella (por ejemplo, pila, o acumulador) que otras arquitecturas presentan al evaluar expresiones aritméticas complejas
  - d) todas las respuestas anteriores son ciertas.
85. ¿Cuál de las siguientes afirmaciones es cierta?
- a) La memoria SRAM es más lenta que la DRAM
  - b) La lectura en la memoria SRAM es destructiva
  - c) La memoria DRAM es más cara que la SRAM
  - d) Ninguna de las anteriores
86. ¿Cuál de los siguientes microprocesadores NO es de 64 bits?
- a) Core i3
  - b) Core i7
  - c) AVR ATMega
  - d) Itanium
87. Suponga la expresión: `unsigned int val = x & 0x01010101;` donde x es un unsigned int. ¿Cuál de x da como resultado `val = 0x01010101`?

- a) 0x10101010
- b) 0xaaaaaaaa
- c) 0xffffffff
- d) 0x13579bdf

88. ¿Qué arquitectura se caracteriza por presentar una gran variación en la longitud de las instrucciones?

- a) registro-registro
- b) registro-memoria
- c) memoria-memoria
- d) ninguna de las anteriores es cierta

89. En los modos de direccionamiento x86-64 del tipo Desplazamiento(Base,Índice,Factor Escala), puede usarse como

- a) base, cualquiera de los 8 registros enteros salvo %rsp
- b) índice, también cualquiera salvo %rbp
- c) factor de escala, cualquier constante de 1, 2, 4 u 8 bytes
- d) desplazamiento, cualquier constante de 1, 2 o 4 bytes (incluso el nombre de una variable, por su dirección)

90. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount6:

```
int popcount6(unsigned* array, size_t len)
{
```

```
 size_t i;
 unsigned x;
 int result=0;
```

```
 const unsigned m1 = 0x55555553;
 const unsigned m2 = 0x33333333;
 const unsigned m4 = 0x0f0f0f0f;
 const unsigned m8 = 0x00ff00ff;
 const unsigned m16 = 0x0000ffff;
```

```
 for (i=0; i<len; i++)
 {
```

```
 x = array[i];
```

```
 x = (x & m1) + ((x >> 1) & m1);
 x = (x & m2) + ((x >> 2) & m2);
 x = (x & m4) + ((x >> 4) & m4);
 x = (x & m8) + ((x >> 8) & m8);
 x = (x & m16) + ((x >> 16) & m16);
```

```

 result+= x;
}
return result;
}

```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en algún dígito de alguna máscara.

Esta función popcount6 fallaría:

- a) con array={0xA0000000, 0x00B00000, 0x00000C00, 0x0000000D}
- b) con array={0x10000000, 0x00200000, 0x00000400, 0x00000008}
- c) con ambos ejemplos
- d) con ninguno de los dos ejemplos

91. ¿Cuál de los siguientes modos de direccionamiento es \*menos\* preferible para un procesador con segmentación de cauce?
- a) Indirecto a través de memoria
  - b) Indexado (o relativo a base, o base+índice)
  - c) Indirecto a través de registro
  - d) Registro
92. El lenguaje máquina...
- a) es un conjunto de nombres simbólicos o nemotécnicos.
  - b) facilita la portabilidad de los programas.
  - c) es el mismo para todos los computadores.
  - d) Ninguna de las respuestas anteriores es correcta.
93. En una CPU de 32 bits con memoria de bytes, el problema es que...
- a) Hay que usar 4 instrucciones de lectura (o escritura) para leer (o escribir) un registro completo
  - b) No hay problema, cuando se salva un registro a memoria se escribe en la posición deseada
  - c) Hay que respetar el ordenamiento de bytes y reglas de alineamiento con que se diseñó la CPU
  - d) No tiene sentido, un registro no cabría en memoria
94. En la captación de un operando que reside en memoria:
- a) en MBR indicamos la dirección donde está y en MAR lo recogemos
  - b) en MAR indicamos la dirección donde está y en IR lo recogemos

- c) en MBR indicamos la dirección donde está y en IR lo recogemos
  - d) en MAR indicamos la dirección donde está y en MBR lo recogemos
95. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros CON signo produciendo un resultado .quad (doble precisión). El programa esqueleto ofrecido (suma.s) no es válido. Se podría comprobar imprimiendo el resultado SIN signo, y usando como contraejemplo las siguientes listas:
- A) 1 y -1 (el usuario piensa que sus datos tienen signo)
  - B) 0x4000 0000 y 0x4000 0000
  - C) ninguna de las dos
  - D) ambas
96. La instrucción movzbl %al, %eax
- a) Copia en %eax el valor del indicador de cero
  - b) Copia en %eax el valor de %al si el indicador de cero está activado
  - c) Pone a 0 el registro %eax
  - d) Copia en %eax el valor sin signo almacenado en %al, rellenando con ceros
97. ¿Qué modificador (switch) de ld hace falta para enlazar una aplicación de 32bits en un sistema de 64bits en el que se ha instalado también el compilador de 32bits?
- a) No hace falta modificador, ld lo deduce del tipo de objeto a enlazar
  - b) -32
  - c) -m elf\_i386
  - d) -m32
98. ¿Cuál de las siguientes secuencias de instrucciones calcula  $a=b-a$ , suponiendo que %eax es a y %ebx es b?
- a) subl %eax, %ebx
  - b) subl %ebx, %eax
  - c) notl %eax
  - d) addl %ebx, %eax
  - e) Varias o ninguna de las respuestas anteriores son correctas, no se puede marcar una y sólo una
99. ¿Qué salida produce el siguiente código? Asumir representación de datos de arquitectura x86-64.

```
unsigned int x = 0xDEADBEEF;
```

```
unsigned short y = 0xFFFF;
signed int z = -1;
if (x > (signed short) y)
 printf("Hello");
if (x > z)
 printf("World");
```

(Recordar que:

- 1.- las extensiones de tamaño se hacen según tenga o no signo el tipo fuente
- 2.- en comparaciones con un dato unsigned se pasa el otro dato a unsigned)

- a) No imprime nada
- b) Imprime "Hello"
- c) Imprime "HelloWorld"
- d) Imprime "World"

100. ¿Qué técnica de E/S consume menos tiempo del procesador?

- a) E/S programada
- b) E/S mediante interrupciones
- c) E/S mediante DMA
- d) Todas pueden consumir el mismo número de recursos en función de la velocidad del dispositivo de E/S al que se acceda

101. ¿Cuál de las siguientes características es posterior a la segunda generación de computadores?

- a) RISC.
- b) Transistor.
- c) Lenguaje ensamblador.
- d) Memoria de núcleos de ferrita.

102. En el siguiente código, ¿qué reordenamiento de los bucles muestra mejor localidad?

```
// X, Y, Z ctes #define previo
int a[X][Y][Z]
```



```
int i, j, k, sum = 0;
for (i = 0; i < Y; i++)
 for (j = 0; j < Z; j++)
 for (k = 0; k < X; k++)
 sum += a[k][i][j];
```

- a) k externo, i central, j interno
- b) j externo, k central, i interno
- c) i externo, j central, k interno (el orden en que están ahora)
- d) El orden de los bucles no afecta a la localidad

103. Sabiendo que la instrucción de llamada mostrada abajo codifica la dirección de la subrutina con direccionamiento relativo a contador de programa (de 32 bits con signo), indicar el valor de los bytes tachados en el siguiente desensamblado.

```
400544: e8 XX XX XX XX callq 400550 <f>
400549: 48 89 03 mov %rax,%rbx)
```

- a) 00 40 05 50
- b) 00 00 00 01
- c) 50 05 40 00
- d) 07 00 00 00

104. Una memoria que está organizada en palabras de 8 bits tiene una capacidad de 32 Kbits. ¿Cuántas líneas de dirección tiene dicha memoria?

- a) 32
- b) 8
- c) 4
- d) 12

105. ¿Qué arquitectura es típica en procesadores RISC?

- a) registro-registro
- b) registro-memoria
- c) memoria-registro
- d) memoria-memoria

106. Considerar los siguientes dos bloques de código almacenados en dos ficheros distintos:

```
/* main.c */
int i = 0;
int main() {
```

```

 func();
 return 0;
}

/* func.c */
int i = 1;
void func() {
 printf("%d", i);
}

```

¿Qué sucederá cuando se compile, enlace y ejecute este código?

- a) Escribe "1"
- b) Error al compilar o enlazar, no se obtiene ejecutable
- c) Escribe "0"
- d) A veces escribe "0" y a veces "1"

107. ¿Qué hace gcc -S?

- a) Compilar .s → .o (fuente ASM a objeto)
- b) Compilar .c → .s (fuente C a fuente ASM)
- c) Compilar optimizando tamaño (size), no tiempo
- d) Compilar borrando del ejecutable la tabla de símbolos (strip)

108. En un sistema con memoria de bytes y líneas de cache de 64 bytes, ¿dónde empieza el bloque de memoria que contiene la posición 0xBEE3DE72?

- a) 0xBEE3DE6E
- b) 0xBEE3DE70
- c) 0xBEE3DE40
- d) 0x0EE3DE72

109. En un sistema con memoria de bytes y líneas de cache de 32 bytes, ¿dónde empieza el bloque de memoria que contiene la posición 0xAC72?

- a) 0xAC60
- b) 0xAC6E
- c) 0xAC70
- d) 0xAC72

110. Un sistema de memoria tiene un tiempo medio de acceso de 10 ns por operación de lectura o escritura y un ancho de datos de 32 bits. ¿Cuál es el ancho de banda del sistema de memoria?
- a) 2,5 ns
  - b) 32 Mbytes por segundo
  - c) 400 millones de bytes por segundo
  - d) 32 bits
111. ¿Cuántos controladores de interrupciones 8259 hacen falta como mínimo para manejar 17 líneas de interrupción?
- a) 3
  - b) 4
  - c) 5
  - d) 2
112. ¿De qué tipo son los procesadores Intel que usamos en los laboratorios?
- a) el concepto de endian no es aplicable a estas máquinas, ya que un registro del procesador no cabe en una posición de memoria
  - b) puede ajustarse mediante un bit de control en el registro CRO que funcionen como little- o big-endian
  - c) big-endian
  - d) little-endian
113. En el programa "size" de la práctica de la cache, si el primer escalón pasa de tiempo=2 para un tamaño de vector menor que 32KB a tiempo=8 para un tamaño mayor que 32KB, podemos asegurar que:
- 
- (Nota: en las opciones de respuesta, "rapidez" se refiere a  $T_i$ , el tiempo de acceso efectivo al nivel  $i$ , no a  $t_i$ , el tiempo de acceso propio al nivel  $i$ . Si dice "La cache L1 es 4x más rápida que L2", se refiere a que  $T_1 = T_2/4$ , no a que  $t_1 = t_2/4$ . Ver T6 tr.26)
- 
- a) La cache L1 es seis veces más rápida que la cache L2
  - b) La cache L1 es cuatro veces más rápida que la cache L2
  - c) La cache L1 es como mucho cuatro veces más rápida que la cache L2
  - d) La cache L1 es al menos cuatro veces más de rápida que la cache L2

114. Dentro de una función declarada como `void swap(long *xp, long *yp)`, que intercambia los valores de los dos enteros (de tamaño `long`) cuyas direcciones de memoria (punteros) son pasadas como parámetros a la función, la instrucción `movq (%rsi),%rdx` copia en `%rdx`...
- A) el valor del entero apuntado por el puntero pasado como segundo parámetro
  - B) el valor del entero apuntado por el puntero pasado como primer parámetro
  - C) el valor del puntero pasado como segundo parámetro
  - D) el valor del puntero pasado como primer parámetro
115. Respecto a los registros enteros en arquitectura IA32
- a) Hay 8 de cada tamaño (32, 16, 8 bits), aunque no todos los registros tienen versión en 8 y 16 bits
  - b) No hay distintos tamaños, son sólo registros de 32 bits, como corresponde a dicha arquitectura
  - c) Hay 8, y en cada uno puede accederse a todos los 32 bits (p.ej. EAX), a los 16 bits menos significativos (p.ej. AX) ó a los 8 LSBs (p.ej. AL)
  - d) Son de 32bits en las aplicaciones de 32bit, y de 64bits en las aplicaciones de 64bit
116. Tras ejecutar las tres instrucciones que se muestran desensambladas a continuación, el registro RAX toma el valor
- 08048074 <\_start>:**  
**8048074: be 74 80 04 08 mov \$\_start, %rsi**  
**8048079: 46 inc %rsi**  
**804807a: 8b 06 mov (%rsi), %rax**
- a) 0x0804807a
  - b) 0x08048075
  - c) 0x08048079
  - d) 0x08048074
117. Usando el repertorio x86-64, para intercambiar el valor de 2 registros se pueden usar...
- a) 3 mov, no menos (se le llama "intercambio circular")
  - b) una instrucción mov y una instrucción lea
  - c) dos instrucciones mov
  - d) 4 mov, no menos (debido a la arquitectura R/M)

118. El fragmento de código:

```
poll: in a, 0x20
 cmp a, $0
 jnz poll
 load a, 0x11
 out 0x21, a
```

corresponde a:

- a) E/S programada con consulta de estado
- b) E/S programada sin consulta de estado
- c) E/S por DMA
- d) E/S por interrupciones

119. ¿Cuál de los siguientes es el ejemplo más acertado de localidad espacial?

- a) Iterar repetidamente el cuerpo de un bucle
- b) Reservar dinámicamente (malloc) espacio para una estructura o union
- c) Referenciar elementos de un array sucesivamente
- d) Referenciar continuamente la misma variable local

120. Las interrupciones generadas por el teclado interrumpirán al procesador:

- a) sólo si el procesador tiene activado el indicador de habilitación de interrupciones
- b) siempre que el usuario pulse una tecla en el teclado
- c) sólo si el procesador está chequeando el estado del teclado
- d) sólo cuando el procesador no esté realizando un trabajo útil

121. alguna de las siguientes líneas de código sirve para definir una variable entera llamada tam en GNU/as Linux x86. ¿Cuál?

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

122. Supongamos dos procesadores con bus de direcciones con idéntico número de líneas. Si uno de ellos emplea E/S mapeada en memoria y el otro E/S independiente, ¿cuál podrá acceder a una mayor cantidad de memoria?

- a) Depende del tamaño del bus de direcciones
- b) Ambos podrán acceder a la misma cantidad de memoria
- c) El que tiene E/S independiente
- d) El que tiene E/S mapeada en memoria

123. El ancho de palabra de una memoria corresponde a:
- a) El número que identifica unívocamente cada posición de la memoria.
  - b) La cantidad de bits que caben en una sola posición
  - c) El número de posiciones que la componen.
  - d) La longitud del registro de direcciones de la memoria.
124. Alguno de los siguientes NO es un motivo de que no se alcance la ganancia ideal en un cauce segmentado
- a) La duración del ciclo de reloj impuesta por la etapa más lenta
  - b) El propio coste de la segmentación (carga de los registros de acoplo, etc...)
  - c) Los riesgos (hazards)
  - d) La emisión múltiple (y posiblemente desordenada) de instrucciones
125. Si se dice que en un sistema computador cada dirección especifica uno o dos puertos de E/S, se refiere a que:
- a) La misma dirección puede usarse para transferir un byte o una palabra de mayor tamaño (ese byte y el siguiente)
  - b) La misma dirección (por ejemplo 0x0210) puede ser una posición de memoria o un puerto de E/S, según IO/M#
  - c) La pregunta es capciosa, una dirección puede especificar un puerto, no dos
  - d) Un puerto será de sólo lectura, otro de sólo escritura, y ambos se decodifican en la misma dirección
126. Si la dirección del primer elemento de un vector de enteros  $z$  está almacenada en el registro `%rdi` y el índice  $i$  está almacenado en el registro `%rsi`, la instrucción máquina que realiza la operación  $z[i]++$  es:
- a) `addl $4, (%rsi,%rdi)`
  - b) `addl $1, (%rsi,%rdi,4)`
  - c) `addl $1, (%rdi,%rsi)`
  - d) `addl $1, (%rdi,%rsi,4)`
127. Sea un formato de microinstrucción que incluye dos campos independientes de 8 bits cada uno. Si se rediseña de modo que se solapen los dos campos, ¿cuántos bits se ahorran en cada microinstrucción?
- a) 9
  - b) 7
  - c) 1
  - d) 8

128. Utilizando la sentencia `asm()`, las denominadas restricciones que se indican al final de dicha sentencia, involucran a:
- a) Solamente las entradas
  - b) Solamente las salidas
  - c) Solamente los sobrescritos
  - d) Ninguna de las anteriores es cierta
129. ¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar):
- a) `movb %sil, (%rax)`
  - b) `pushq $0xFF`
  - c) `movsbw (%rax), %dx`
  - d) `movzlw %edx, %rax`
130. La instrucción `test` es...
- a) Lo mismo que `sub`, pero no guarda el resultado, sólo ajusta los flags
  - b) Lo mismo que `and`, pero no guarda el resultado, sólo ajusta los flags
  - c) Lo mismo que `sub`
  - d) Lo mismo que `and`
131. La diferencia entre las instrucciones `test` y `cmp` consiste en que
- a) `test` realiza una operación `and` lógico, mientras que `cmp` realiza una resta
  - b) modifica sólo los flags lógicos (ZF,SF) mientras que `cmp` modifica los aritmético-lógicos (ZF,SF,CF,OF)
  - c) ambas respuestas son correctas
  - d) ambas respuestas son incorrectas
132. Un computador con 13 líneas de direcciones utiliza E/S mapeada a memoria. Si se supone que cada uno de los periféricos ocupa 4 direcciones y que el número de periféricos que se planea conectar es de  $2^{10}$ , ¿qué tamaño de memoria admite el computador?
- a)  $2^{10}$  palabras
  - b)  $2^{12}$  palabras
  - c)  $2^{13}$  palabras
  - d) Ninguna de las anteriores
133. De entre las siguientes construcciones de flujo de control en lenguaje C, la que se traduce más directamente a lenguaje ensamblador es...
- a) El bucle `for`
  - b) El bucle `do-while`
  - c) La selección `switch-case`
  - d) El bucle `while`

134. Respecto a registros salva-invocante y salva-invocado en GCC/Linux x86-64, ¿cuál de éstos es de distinto tipo que el resto?

- a) RAX
- b) RBX
- c) RSI
- d) RDI

135. En un procesador de la familia 80x86 las posiciones de memoria que representan una variable int (entero 4B compl.2) contiene los bytes: F0 FF FF FF. ¿Cuánto vale dicha variable?

- a) 4043309055
- b) 16
- c) 4294967280
- d) -16

136. Considerar la declaración C

**long array[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};**

Suponer que el compilador tiene la dirección de array en el registro %rcx.

¿Cómo se

movería el valor array[3] al registro %rax? Asumir que %rbx es 3.

- a) leaq (%rcx,%rbx,8),%rax
- b) leaq 12(%rcx),%rax
- c) movq 8(%rcx,%rbx,2),%rax
- d) movq (%rcx,%rbx,8),%rax

137. Por x86-64 se entiende la misma arquitectura de repertorio (ISA) que

- a) IA32
- b) AMD64
- c) x86
- d) IA64

138. El direccionamiento relativo a registro base utiliza...

- a) dos registros.
- b) un registro y un desplazamiento.
- c) un registro y un factor de escala
- d) dos desplazamientos contenidos en la propia instrucción.

139. Sobre el direccionamiento relativo a contador de programa:

- a) Favorece la implementación de código reubicable.
- b) Su uso en los saltos y llamadas a subrutinas reduce el tamaño de la instrucción.
- c) Es adecuado para alcanzar instrucciones próximas a la que se está ejecutando.



d) Todas las respuestas son ciertas.

140. Un sistema con direcciones de 8bits utiliza una puerta NAND conectada a las líneas A7...A5 para atacar la entrada CS# (activa baja) de un módulo de memoria. En el mapa de memoria las siguientes posiciones corresponderán a dicho módulo
- a) 0x70 a 0x7f y 0xf0 a 0xff
  - b) 0x00 a 0x0f y 0x80 a 0x8f
  - c) 0xe0 a 0xff
  - d) 0x00 a 0x1f
141. En EC podemos usar la palabra directo para referirnos a... (señalar la opción incorrecta)
- a) modo de direccionamiento directo
  - b) dispositivo de almacenamiento secuencial directo (DASD)
  - c) acceso directo a memoria
  - d) cache con correspondencia directa
142. ¿Cuál de las siguientes secuencias de tipos de memoria está ordenada de menores a mayores prestaciones?
- a) SDRAM, DDR, EDO
  - b) FPM, EDO, RDRAM
  - c) EDO, SRAM, FPM
  - d) DDR, SDRAM, FPM
143. ¿Cuántos bits hacen falta como mínimo para implementar tres niveles de inhibición de interrupciones (general, nivel y máscara) en un sistema con cuatro niveles de interrupción?
- a) 6
  - b) 5
  - c) 4
  - d) 7
144. Si %rdx contiene 0xf000 y %rcx contiene 0x0100, el direccionamiento 0x80(%rcx,%rdx,2) se refiere a la posición
- a) 0xf182
  - b) 0xf280
  - c) 0x1e180
  - d) Ninguna de las respuestas anteriores es correcta

145. La ganancia en velocidad ideal de un cauce de K etapas de igual duración T ejecutando un programa de N instrucciones es
- a)  $S = KN/(K+N-1)$
  - b)  $S = KN/(K-N+1)$
  - c)  $S = NT/(N+K-1)T$
  - d)  $S = NKT/(N-K+1)T$
146. ¿Por qué se impusieron las arquitecturas de registros de propósito general a las arquitecturas basadas en pila?
- a) Porque no se puede programar una arquitectura de pila en un lenguaje de alto nivel
  - b) Porque las basadas en registros permiten reducir el tamaño del programa
  - c) Porque la memoria es más cara que los registros
  - d) Porque las basadas en registros son capaces de lograr un mejor rendimiento cuando se asignan variables a registros
147. ¿Qué política de colocación en cache necesita más comparadores, la correspondencia asociativa por conjuntos o la correspondencia por sectores?
- a) Depende de si es mayor el número de conjuntos o el número de sectores
  - b) Depende de si es mayor el número de bloques por conjunto o el número de sectores
  - c) Correspondencia por sectores
  - d) Correspondencia asociativa por conjuntos
148. 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) [11...6]
  - b) [5...0]
  - c) [8...6]
  - d) [5...3]
149. ¿Cuál de las siguientes afirmaciones es cierta?
- a) En el direccionamiento a registro, el objeto direccionado es una constante contenida en la propia instrucción.
  - b) Los modos de direccionamiento de una instrucción se especifican en ensamblador mediante las directivas.
  - c) El modo de direccionamiento indirecto a través de registro es el modo más sencillo de direccionamiento a memoria.
  - d) En el direccionamiento inmediato, la dirección efectiva se encuentra en un registro de uso general del procesador.

150. ¿Cuál es el número máximo de niveles de interrupción que se pueden manejar utilizando 3 controladores de interrupciones programables 8259?
- a) 18
  - b) 20
  - c) 22
  - d) 24
151. Considere un sistema de memoria para un procesador de 32 bits con caches separadas para código y datos. Suponga que el procesador direcciona la memoria por bytes y realiza accesos a palabras de 32 bits y que el espacio de direcciones es de 232 bytes. La cache de datos tiene las siguientes características: 64 KB de capacidad, asociativa por conjuntos con 2 vías, y bloques de 2 palabras. ¿Cuántos bits tiene el campo etiqueta de una dirección de memoria?
- A) 17
  - B) 13
  - C) 15
  - D) 11
152. El lenguaje máquina es...
- a) portable entre arquitecturas.
  - b) fácilmente legible por el programador.
  - c) difícil de codificar manualmente.
  - d) una alternativa razonable al uso del lenguaje ensamblador.
153. En una máquina little-endian con memoria de bytes y representación en complemento a dos que permite accesos a memoria de tamaño byte (1B), media palabra (2B) y palabra (4B), 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) 0
  - B) -2
  - C) 1
  - D) -1
154. ¿Qué técnica de E/S requiere menos atención por parte del procesador?
- a) Todas requieren la misma atención
  - b) E/S mediante interrupciones
  - c) E/S mediante acceso directo a memoria
  - d) E/S programada
155. ¿Qué instrucciones son típicas del repertorio de un procesador que sólo disponga de E/S mapeada en memoria?
- a) IN, LOAD, OUT

- b) IN, LOAD, MOV
- c) LOAD, MOV, STORE
- d) Ninguno de los anteriores

156. Si el tiempo de acceso a la memoria cache es de 2 ns y el tiempo necesario para tratar un fallo de cache es de 80 ns, ¿cuál es la tasa de aciertos necesaria para que el tiempo medio de acceso al sistema de memoria sea de 10 ns?

- a) 0.9
- b) 0.95
- c) 0.75
- d) 0.8

157. ¿Qué forma tiene la gráfica que se debe obtener con el código size.cc?

- a) Forma de U, con un tramo descendente y otro ascendente.
- b) Forma de  $\cap$ , con un tramo ascendente y otro descendente.
- c) Una escalera decreciente con varios tramos horizontales.
- d) Una escalera creciente con varios tramos horizontales.

158. 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

159. Justo antes de que una instrucción máquina escriba un resultado en memoria:

- a) en MBR está el resultado y en MAR la dirección donde se almacenará
- b) en IR está el resultado y en MAR la dirección donde se almacenará
- c) en IR está el resultado y en MBR la dirección donde se almacenará
- d) en MAR está el resultado y en MBR la dirección donde se almacenará

160. ¿Cuál de las siguientes funciones no corresponde a un controlador (interfaz) de E/S?

- a) Almacenamiento de programas
- b) Almacenamiento temporal de datos
- c) Comunicación con el microprocesador
- d) Comunicación con el dispositivo

161. El bus del sistema es

- a) el que conecta las distintas partes del sistema: UC, ALU, E/S, M
  - b) en un sistema con buses separados, el que conecta el sistema E/S con el resto
  - c) el que conecta CPU-M, ya sea un sistema con bus único o con múltiples buses
  - d) en un sistema con bus único, todo el bus salvo la parte relacionada con E/S (SATA, GPU, USB, Ethernet, etc)
162. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros en doble precisión. El programa esqueleto ofrecido (suma.s) no es válido, en concreto...  
(marcar la opción FALSA)
- a) no está preparado para sumar más de 9 elementos
  - b) no hace extensión con ceros de los elementos
  - c) no hace extensión de signo de los elementos
  - d) no consulta ni el flag de acarreo CF ni el de overflow OF
163. En un sistema con un único bus...
- a) sólo un dispositivo puede escribir en un instante dado en el bus
  - b) se utilizan las mismas líneas de control para conectar todos los dispositivos
  - c) el procesador y los periféricos pueden funcionar a diferentes velocidades si el funcionamiento del bus es asíncrono
  - d) Todas las respuestas anteriores son ciertas
164. Las siguientes afirmaciones sugieren que el tamaño de varios tipos de datos en C (usando el compilador gcc) son iguales tanto en IA32 como en x86-64. Sólo una de ellas es FALSA. ¿Cuál?
- a) El tamaño de un double es 8 bytes
  - b) El tamaño de un puntero es 4 bytes
  - c) El tamaño de un short es 2 bytes
  - d) El tamaño de un int es 4 bytes
165. Respecto a la ecuación de rendimiento  $T=(N \cdot S)/R$ , el objetivo de un diseño CISC es:
- a) aumentar S (número medio de ciclos por instrucción)
  - b) disminuir R (frecuencia de reloj)
  - c) aumentar R (frecuencia de reloj)
  - d) disminuir N (número de instrucciones a ejecutar por el programa)

166. En un microprocesador de 4 bits, una operación en la que el bit 0 de un registro se copia en el acarreo, después el bit 1 se copia en el bit 0, después el bit 2 se copia en el bit 1, y por último el bit 3 se copia en el bit 2, es:
- a) Una rotación a la derecha.
  - b) Una rotación a la derecha a través de acarreo.
  - c) Un desplazamiento lógico a la derecha.
  - d) Un desplazamiento aritmético a la derecha.
167. Respecto a la E/S programada...
- a) No todos los pasos requieren la ejecución de instrucciones por parte de la CPU
  - b) La transferencia la realiza un procesador externo a la CPU
  - c) Las dos primeras afirmaciones son ciertas
  - d) Las dos primeras afirmaciones son falsas
168. ¿Cuál de las siguientes afirmaciones acerca de una jerarquía de memoria es cierta?
- a) Para aumentar la eficiencia se transfieren bloques completos.
  - b) Toda la información que el procesador necesita está en el nivel 1.
  - c) Si una palabra no se encuentra en el tercer nivel entonces se busca en el segundo nivel.
  - d) A medida que nos alejamos del procesador, el tamaño de memoria disminuye.
169. ¿Cuál de las siguientes instrucciones máquina copia en RAX el entero (long) almacenado en la posición de memoria cuya dirección efectiva es el resultado de la operación  $RDX * 8 + RBX$ ?
- a) `movq (%rbx, %rdx, 8), %rax`
  - b) `leaq 8(%rdx, %rdx), %rax`
  - c) `leaq (%rbx, %rdx, 8), %rax`
  - d) `movq 8(%rdx, %rdx), %rax`
170. La(s) instrucción(es) necesaria(s) para cargar el dividendo 0xa30bf18a en la pareja edx:eax como paso previo a una división sin signo son:
- a) `movl $0xf18a,%eax`  
`movl $0xa30b,%edx`
  - b) `movl $0xa30bf18a,%eax`  
`cld`
  - c) `movl $0xa30bf18a,%eax`  
`xorl %edx,%edx`
  - d) `movq $0xa30bf18a,%rax`

171. En una bomba como las estudiadas en prácticas, del tipo...

```
0x08048705 <main+149>: call 0x80484c4 <gettimeofday>
...
0x08048718 <main+168>: cmp $0x5,%eax
0x0804871b <main+171>: jle 0x8048722 <main+178>
0x0804871d <main+173>: call 0x8048604 <boom>
0x08048722 <main+178>: ...
```

ejecutada paso a paso con el depurador ddd, interesaría...

- a) ejecutar hasta jle, ajustar %eax a 6, y continuar ejecutando paso a paso
- b) ejecutar hasta jle, ajustar %eax a 4, y continuar ejecutando paso a paso
- c) cambiar jle por jmp usando ddd o un editor hex, salvar el programa y reiniciar la depuración con el nuevo ejecutable
- d) Ninguna de las opciones anteriores es de interés (bien porque no se pueda hacer eso o porque no sirva para evitar la bomba)

172. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount5:

```
int popcount5(unsigned* array, size_t len)
{
 size_t i,j;
 int x;
 int val, result=0;

 for (i=0; i<len; i++)
 {
 x = array[i];
 val = 0;
 for (j=0; j<=8; j++)
 {
 val += x & 0x01010101;
 x >>= 1;
 }
 val += (val >> 16);
 val += (val >> 8);
 result+= val & 0xFF;
 }
 return result;
}
```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en las condiciones del for interno y el tipo de x.

Esta función popcount5 fallaría:

- a) con array={1, 16, 256, 4096}
- b) con array={0x80000000, 0x00400000, 0x00000200, 0x00000001}
- c) con ambos ejemplos
- d) con ninguno de los dos ejemplos

173. Cuando un Intel Atom ejecuta una instrucción CALL

- a) el registro rip se guarda en la pila
- b) no se guarda ningún registro
- c) los registros rax, rcx, rdx y rip se guardan en la pila
- d) los registros rip, rbp y rsp se guardan en la pila

174. ¿Cuál de las siguientes direcciones está alineada a double (8-byte)?

(Al no poder escribir el 2 como subíndice, aclaramos que ")2" indica binario)

- a) 1110110101110111)2
- b) 1110110101110100)2
- c) 1110110101110000)2
- d) Ninguna de ellas

175. Las técnicas write-through y write-back están relacionadas con

- a) métodos de E/S
- b) etapas de la unidad de control
- c) arbitraje de buses
- d) coherencia de cache

176. El resultado de desplazar 0xFF0F a la derecha aritméticamente 4 veces es:

- a) 0xF0F0
- b) 0xF0FF
- c) 0xFFFF0
- d) 0x0FF0



177. Un salto condicional de tipo "annulling branch", o salto anulante, ejecuta la(s) instrucción(es) siguiente(s)...
- a) sólo si el salto se produce (las ignora si NO se produce), de manera que instrucción(es) en el destino del salto podrían adelantarse tras la propia instrucción de salto
  - b) sólo si el salto NO se produce (las ignora si se produce), de manera que instrucción(es) en el destino del salto podrían adelantarse tras la propia instrucción de salto
  - c) siempre, de manera que instrucción(es) anterior(es) al salto podrían colocarse tras la propia instrucción de salto
  - d) nunca, de manera que instrucción(es) anterior(es) al salto podrían adelantarse tras la propia instrucción de salto
178. Si declaramos `int val[5]={1,5,2,1,3}`; entonces
- a) `val+4` es de tipo `int*` y se cumple que `*(val+4)==5`
  - b) `val[5]` es de tipo `int` y vale 3
  - c) `val+1` es de tipo `int` y vale 2
  - d) `&val[2]` es de tipo `int*` y vale lo mismo que `(void*)val+8`
179. ¿En qué generación, dentro de la historia de los computadores digitales, aparece la microprogramación?
- a) primera
  - b) segunda
  - c) tercera
  - d) cuarta
180. En una arquitectura RISC típica:
- a) suele usarse segmentación
  - b) la programación resulta mucho más simple que en una arquitectura CISC
  - c) la UC es más compleja que en una arquitectura CISC
  - d) se usan pocas instrucciones de las disponibles en el conjunto de instrucciones
181. ¿Qué modificador (switch) de gcc hace falta para compilar `.s` → `.o` sin llamar al enlazador?
- a) Eso no se puede hacer con gcc
  - b) `gcc -s`
  - c) `gcc -S`
  - d) `gcc -c`

182. 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 0x8048504 <scanf>
0x08048744 <main+212>: mov 0x24(%esp),%edx
0x08048748 <main+216>: mov 0x804a044,%eax
0x0804874d <main+221>: cmp %eax,%edx
0x0804874f <main+223>: je 0x8048756 <main+230>
0x08048751 <main+225>: call 0x8048604 <boom>
0x08048756 <main+230>:
```

...el código numérico correcto (pin) es...

- a) el entero almacenado a partir de la posición de memoria 0x24(%esp)
  - b) el entero 0x804a044
  - c) el entero almacenado a partir de la posición de memoria 0x804a044
  - d) el entero cuya dirección está almacenada en la posición de memoria 0x804a044
183. En la práctica de la cache, el código de size.cc accede al vector saltando de 64 en 64. ¿Por qué?
- a) Para anular los aciertos por localidad espacial, esto es, que sólo pueda haber aciertos por localidad temporal
  - b) Para recorrer el vector más rápidamente
  - c) Porque cada elemento del vector ocupa 64 bytes
  - d) Porque el tamaño de cache L1 de todos los procesadores actuales es de 64KB
184. Como parte del proceso de compilación de una aplicación en lenguaje C, enlazar .o → .exe (de objeto proveniente de fuente C a ejecutable) usando sólo as y ld, sin gcc...
- a) Ninguna de las anteriores respuestas es correcta
  - b) Se puede, repartiendo entre as y ld los modificadores (switches) que corresponda
  - c) Se puede, repartiendo modificadores entre as y ld, y añadiendo al comando ld el runtime de C
  - d) Basta usar ld, con los modificadores de gcc que corresponda, y añadiéndole el runtime de C

185. En una bomba como las estudiadas en prácticas, del tipo...

```
0x40079b <main+64> lea 0x30(%rsp),%rdi
0x4007a0 <main+69> mov 0x2008d9(%rip),
 %rdx # 0x601080
0x4007a7 <main+76> mov $0x64,%esi
0x4007ac <main+81> call 0x400600 <fgets>
0x4007b1 <main+86> test %rax,%rax
0x4007b4 <main+89> je 0x400785 <main+42>
0x4007b6 <main+91> lea 0x30(%rsp),%rdi
0x4007bb <main+96> mov $0xd,%edx
0x4007c0 <main+101> lea 0x2008a1(%rip),
 %rsi # 0x601068
0x4007c7 <main+108> call 0x4005d0<strncmp>
0x4007cc <main+113> test %eax,%eax
0x4007ce <main+115> je 0x4007d5<main+122>
0x4007d0 <main+117> call 0x400727 <boom>
0x4007d5 <main+122> lea 0x20(%rsp),%rdi
```

...la contraseña (alfanumérica) es...

- a) el string almacenado a partir de 0x10(%rsp)
- b) el string alm. a partir de 0x20096a+0x4006ff
- c) el string almacenado a partir de 0x601058**
- d) el string alm. a partir de 0x200939+0x400718

186. En la convención cdecl, es responsabilidad del procedimiento llamado salvaguardar los registros:

- a) %esi, %edi
- b) %eax, %ebx, %ecx, %edx
- c) %ebx, %esi, %edi**
- d) %eax, %edx, %ecx

187. ¿En qué técnica para determinar la dirección de comienzo de la rutina de servicio de interrupción se fija dicha dirección en los circuitos de la CPU?

- a) Envío de instrucción de bifurcación completa.
- b) Direcciones fijas.**
- c) Direccionamiento relativo.
- d) Direccionamiento absoluto.

188. Las instrucciones de salto...

- a) son las causantes de los riesgos de tipo RAW y WAW.
- b) complican el diseño eficiente de los procesadores segmentados.**
- c) siempre utilizan direccionamiento absoluto.
- d) Todas las afirmaciones anteriores son ciertas.

189. alguna de las siguientes señales no sirve de entrada a la unidad de control.  
¿Cuál?

- a) contador de programa (bits del registro PC)
- b) señal de reloj (CLK)
- c) estado de la unidad de proceso (flags Z, S, C, O...)
- d) instrucción actual (bits del registro IR)

190. Parecidos y diferencias entre los métodos de E/S (señalar la opción incorrecta)

- a) se suele avisar a la CPU (con una IRQ) de que debe realizar alguna tarea, tanto en E/S por IRQ como en E/S por DMA
- b) sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo
- c) la consulta del estado del dispositivo por parte de la CPU se suele/puede incluir en E/S programada y en E/S por IRQ
- d) sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S

191. El primer parámetro de printf:

- a) puede ser de cualquier tipo, incluso no existir
- b) es un puntero
- c) es un entero
- d) es un char

192. En una bomba como las estudiadas en prácticas, del tipo...

```
400746: lea 0xc(%rsp),%rsi
40074b: lea 0x1ab(%rip),%rdi
 # 4008fd <stdin_u+0x10d>
400752: mov $0x0,%eax
400757: callq 400590 <scanf@plt>
40075c: cmpl $0x400600,0xc(%rsp)
400764: je 40076b <main+0x8c>
400766: callq 400697 <boom>
40076b: callq 4006bb <defused>
```

...el código numérico (pin) es...

- a) el entero 0x400590
- b) el entero 0x400600
- c) el entero almacenado a partir de 0xc(%rsp)
- d) el entero almacenado a partir de 0x1ab+0x400752

193. ¿Qué modificador (switch) de gcc hace falta para compilar .c→.s (de fuente C a fuente ASM)?
- a) gcc -c
  - b) Eso no se puede hacer con gcc
  - c) gcc -S
  - d) gcc -s
194. ¿Cuál de los siguientes es el orden correcto en el ciclo de compilación de un programa en lenguaje C? (el fichero sin extensión es un ejecutable):
- a) fich.c → fich.s → fich → fich.o
  - b) fich.c → fich.o → fich.s → fich
  - c) fich → fich.s → fich.o → fich.c
  - d) fich.c → fich.s → fich.o → fich
195. Compilar .s → ejecutable, usando sólo as y ld, sin gcc...
- a) No se puede
  - b) Se puede, usando en as y ld los modificadores (switches) que corresponda
  - c) Basta usar as, con los modificadores que corresponda
  - d) Basta usar ld, con los modificadores que corresponda
196. Un computador tiene una memoria de control de 16000 palabras de 250 bits, de las que 447 son diferentes. ¿Cuántos bits ahorramos usando nanoprogramación en lugar de microprogramación?
- a) 3744250
  - b) 259206
  - c) 287935
  - d) ninguno de los resultados anteriores es exacto
197. ¿Cuál de las siguientes instrucciones es incorrecta?
- a) shr \$1,%rdx
  - b) shr %rdx
  - c) shr %cl,%rdx
  - d) shr %rcx,%rdx
198. En un sistema con dos buses separados, uno para el subsistema de memoria y otro para la E/S...
- a) el bus que une la memoria y el procesador suele funcionar a la velocidad de la memoria
  - b) el bus de E/S funciona a la velocidad del periférico más rápido
  - c) ambos buses tienen que tener el mismo ancho de banda
  - d) Ninguna de las respuesta anteriores es cierta

199. ¿Qué tipo de instrucciones se emplean más en una arquitectura de acumulador?
- a) de transferencia de datos entre registros
  - b) aritmético-lógicas
  - c) de transferencia de datos con memoria
  - d) de desplazamiento y rotación
200. ¿Cuál afirmación es FALSA al comparar las arquitecturas x86 y x86-64?
- a) El tamaño de un double es el mismo
  - b) El tamaño de un puntero es el mismo
  - c) El tamaño de un entero (int) es el mismo
  - d) El tamaño de las posiciones de memoria es el mismo
201. La arquitectura x86-64 tiene:
- a) 64 registros RPG de 64 bits
  - b) 8 registros de propósito general (RPG) de 64 bits (%rax, %rbx, ... %rsp, %rbp)
  - c) 32 registros RPG de 64 bits
  - d) 16 registros RPG de 64 bits
202. La técnica de sondeo, escrutinio o "polling"...
- a) Se utiliza para identificar el destino de una interrupción.
  - b) Permite establecer un mecanismo de asignación de prioridades a los distintos dispositivos.
  - c) En caso de utilizarse, es necesario emplear varias líneas para que los dispositivos soliciten una interrupción.
  - d) Es incompatible con la técnica de daisy-chain.
203. La "postescritura (write-back) marcada"
- a) es más eficiente que la "postescritura siempre"
  - b) provoca una menor tasa de faltas que la "postescritura siempre"
  - c) requiere más bits de modificación ("bits sucios") cuando aumenta el número de vías
  - d) requiere menos hardware que la "postescritura siempre"
204. Suponer un procesador tipo MIPS segmentado en etapas IF, ID, EX, MEM, WB, como el presentado de ejemplo en clase. ¿Cuál afirmación es correcta?
- a) La aceleración del procesador no segmentado es 5x
  - b) El periodo de reloj se escoge de acuerdo a la etapa más rápida del cauce
  - c) Conviene tener memorias separadas de código y datos, para evitar conflictos entre las etapas IF y WB
  - d) Conviene tener cache que permita acceso en un único ciclo de reloj

205. ¿Cuáles de las siguientes direcciones de memoria podrían estar simultáneamente en una memoria cache de 256 palabras con 16 palabras por bloque y con correspondencia directa ?
- a) 0x0000 y 0xFF0F
  - b) 0xABAB y 0xABAC
  - c) 0x08E3 y 0x74E1
  - d) Ninguna de las combinaciones anteriores
206. En las últimas generaciones de computadores la mejora de prestaciones viene dada por:
- a) avances en tecnología y avances en la estructura y arquitectura del computador.
  - b) avances en las tecnologías de fabricación.
  - c) avances en los sistemas operativos y aplicaciones.
  - d) avances en la estructura y arquitectura del computador.
207. Un computador tiene una memoria de control de 16000 palabras de 250 bits, de las que 447 son diferentes. ¿Cuántos bits ahorramos usando nanoprogramación en lugar de microprogramación?
- a) 3744250
  - b) 259206
  - c) 287935
  - d) ninguno de los resultados anteriores es exacto
208. ¿Qué dice la ley de Moore?
- a) Que el tamaño de los transistores se duplica cada 18 meses.
  - b) Que la memoria de los ordenadores se duplica cada 18 meses.
  - c) Que las prestaciones de los transistores en un chip se duplican cada 18 meses.
  - d) Todas las respuestas son ciertas.
209. Si almacenamos según el criterio little-endian la palabra de 64 bits 0xFACEB00C a partir de la dirección 0xCAFEBAFE, el byte 0xCE quedará almacenado en la dirección:
- a) 0xCAFEBAF1
  - b) 0xCAFEBAF0
  - c) 0xCAFEBAFF
  - d) 0xCAFEBAFE
210. ¿Qué novedad se desarrolló en la tercera generación de computadores?
- a) Los circuitos integrados
  - b) Los primeros lenguajes de programación de alto nivel
  - c) Los microprocesadores CISC
  - d) Los microprocesadores RISC

211. En X86-64, el registro contador de programa se denomina:
- a) RIP
  - b) EIP
  - c) IP
  - d) R15
212. Una CPU con bus de direcciones de 16 bits y bus de datos de 8 bits tiene un registro de 8 bits conectado al bus de datos y a la unidad de control. Puede tratarse del registro
- a) Puntero de pila
  - b) Contador de programa
  - c) De instrucción
  - d) De direcciones
213. La instrucción `movq %rsp,%rbp`
- a) Copia el contenido del registro RSP en el registro RBP.
  - b) Mueve el contenido del registro RSP al registro RBP, poniendo a 0 el registro RSP.
  - c) Introduce en la pila el contenido del registro RBP.
  - d) Intercambia los contenidos de los registros RSP y RBP.
214. Una instrucción de "salto si igual" tiene que comprobar el valor de:
- a) el bit de signo
  - b) el bit de acarreo
  - c) el bit de cero
  - d) los bits de signo y desbordamiento
215. Suponga el siguiente diseño cache/memoria: direcciones de 16 bits, direccionamiento por bytes, tamaño de cache 256 bytes, tamaño de bloque 8 bytes, tamaño de etiqueta 11 bits. ¿Cuál es la asociatividad (cuántas vías hay en la cache)?
- a) 2
  - b) 5
  - c) 4
  - d) 8



216. Sobre el programa ensamblador:

- a) La calidad de un programa ensamblador afectará menos al tiempo de ejecución de los programas generados por él que la calidad de un compilador.
- b) Las etiquetas permiten que el programador especifique el destino de un salto de forma que éste no tenga que modificarse manualmente cuando el programa varíe de tamaño.
- c) El lenguaje ensamblador elimina la posibilidad de errores en la generación de la representación en lenguaje máquina de cada instrucción.
- d) Todas las respuestas son ciertas.

217. ¿Cuál de las siguientes afirmaciones es \*falsa\*?

- a) Sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S
- b) La consulta del estado del dispositivo por parte de la CPU se suele hacer con E/S programada (salvo con dispositivos que siempre están listos para transferir) y con E/S por IRQ (cuando se usa polling para determinar el origen de la IRQ)
- c) Se suele avisar a la CPU (mediante una IRQ) de que debe realizar alguna tarea, tanto en E/S por IRQ (obligatoriamente, la tarea es la transferencia) como en E/S por DMA (optativamente, el controlador DMA puede avisar de que acabó)
- d) Sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo de E/S

218. La instrucción máquina DI (Disable Interrupts), conocida como CLI (Clear Interrupt Flag) en x86-64, se utiliza para desactivar:

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

219. 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

220. En la cache L1 de instrucciones, la tasa de fallos:
- a) Siempre tiende a crecer si el número de vías crece
  - b) Siempre tiende a crecer si el tamaño total de L1 crece
  - c) Siempre tiende a disminuir si el tamaño total de L1 crece
  - d) Siempre tiende a disminuir si el número de vías disminuye
221. Un computador emplea un sistema de memoria principal de 128 palabras y una memoria cache de 32 palabras. La organización de la memoria cache es totalmente asociativa y el tamaño de bloque es de 8 palabras. Se emplea el algoritmo de reemplazo FIFO. Si inicialmente la memoria cache está totalmente vacía, calcule el número de fallos cuando se lee la secuencia de direcciones de la memoria principal: 0000100, 1000001, 0000101, 0010011, 0100010, 1000100, 0000111.
- A) 4 fallos
  - B) 3 fallos
  - C) 5 fallos
  - D) 6 fallos
222. Los arrays bidimensionales en lenguaje C se almacenan en orden...
- a) "de mayor a menor" (major-to-minor)
  - b) "mayor-de-columna" (column-major)
  - c) "mayor-de-fila" (row-major)
  - d) "de fila a columna" (file-to-column)
223. ¿Qué tipo de sincronización es más conveniente en el caso de tener dispositivos con distintos requisitos de temporización?
- a) Síncrona
  - b) Asíncrona
  - c) No se pueden conectar dispositivos con distintos requisitos de temporización
  - d) Ninguna de las anteriores
224. En los casos concretos indicados para las siguientes instrucciones x86-64, ¿cuál no funciona como instrucción de transferencia?
- a) `cmpq 0x6000f0, %rax`
  - b) `movq $0x15, %rax`
  - c) `pushq %rax`
  - d) `leaq variable, %rax`

225. ¿Cuál de las siguientes afirmaciones es incorrecta?
- A) En el direccionamiento inmediato el dato se encuentra en la propia instrucción
  - B) El direccionamiento indexado es útil para manejo de vectores
  - C) En el direccionamiento implícito no se indica la ubicación del operando
  - D) El direccionamiento indirecto indica la dirección del operando
226. Un procesador x86 a 4 GHz dispone de 7 unidades de ejecución en paralelo, con 20 etapas de segmentación, y es capaz de emitir (comenzar a ejecutar) 4 instrucciones en cada ciclo de reloj. ¿Qué velocidad aproximada de ejecución de instrucciones será capaz de alcanzar (MIPS = millones de instrucciones por segundo)?
- A) 28000 MIPS
  - B) 16000 MIPS
  - C) 4000 MIPS
  - D) 80000 MIPS
227. ¿Cuál de las siguientes afirmaciones es incorrecta?
- a) En lenguajes de alto nivel no se tiene acceso a detalles del procesador como las tablas de páginas, el paso de modo usuario a modo supervisor, el bit de paridad, etc.
  - b) En lenguaje ensamblador cada instrucción se corresponde con una instrucción máquina.
  - c) El lenguaje de alto nivel es más portable que el lenguaje máquina.
  - d) En lenguaje ensamblador las instrucciones se escriben en binario.
228. Respecto a si un computador dispone de E/S independiente (separada) o usa E/S mapeada a memoria:
- a) Si el repertorio del procesador tiene instrucciones del tipo IN y OUT, es que el computador dispone de E/S separada
  - b) Si el encapsulado (chip) del procesador tiene patilla (pin) IO/M# (o patillas equivalentes), eso evidencia que el computador usa E/S mapeada a memoria
  - c) Si el repertorio del procesador tiene instrucciones del tipo LOAD y STORE, el computador sólo dispone de E/S mapeada a memoria
  - d) Si el encapsulado del procesador no dispone de patilla IO/M# (ni equivalentes), el computador sólo dispone de E/S separada
229. ¿Cuántos operandos acompañan a la instrucción PUSH en arquitecturas x86-64?
- a) 2
  - b) 3
  - c) 1
  - d) 0

230. ¿Cómo se devuelve en ensamblador x86 Linux gcc el valor de retorno de una función al terminar ésta?

- a) La instrucción RET lo almacena en un registro especial de retorno
- b) Se almacena en pila justo encima del (%rbp) del invocado
- c) Por convención se guarda en %rax
- d) Se almacena en pila justo encima de los argumentos de la función

231. ¿Cómo se devuelve en ensamblador x86 Linux gcc el valor de retorno de una función al terminar ésta?

- a) La instrucción RET lo almacena en un registro especial de retorno
- b) Se almacena en pila justo encima del (%rbp) del invocado
- c) Por convención se guarda en %rax
- d) Se almacena en pila justo encima de los argumentos de la función

232.

# WUOLAH



RullenCastro

[www.wuolah.com/student/RullenCastro](http://www.wuolah.com/student/RullenCastro)



73

## 750-preguntas-test-SWAD-EC-Resueltas.pdf

*750 preguntas test SWAD EC Resueltas*



2º Estructura de Computadores



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de  
Telecomunicación  
Universidad de Granada



## Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.





**KEEP  
CALM  
AND  
ESTUDIA  
UN POQUITO**

**1-La(s) instrucción(es) necesaria(s) para cargar el dividendo 0xa30bf18a en la pareja edx:eax como**

**paso previo a una división sin signo son:**

b)      `movl $0xa30bf18a,%eax`  
         `xorl %edx,%edx`

**2-¿Qué hace gcc -O?**

a) Compilar con optimización suave

**3-La diferencia entre el flag de acarreo y de overflow es que...**

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

**4-De las siguientes instrucciones, ¿cuál no utiliza un modo de direccionamiento implícito?**

Push

**5-El direccionamiento relativo a registro base utiliza...**

c) un registro y un desplazamiento.

**6-En un sistema de gestión de interrupciones mediante "polling" o sondeo, el dispositivo que**

**solicita la interrupción envía, junto con la señal de petición de interrupción, su correspondiente vector de interrupción.**

Falso

**7-Indique cuál de las siguientes afirmaciones sobre el ENIAC no es correcta:**

Como los computadores actuales, era una máquina binaria, es decir, los números estaban representados en

forma binaria y los cálculos aritméticos se realizaban también en el sistema binario.

**8-Se puede programar un controlador de interrupciones 8259 de manera que atienda equitativamente a 8 dispositivos de igual prioridad (cada vez que se atiende a un dispositivo, éste**

**pasa automáticamente a tener la prioridad más baja).**

Verdadero

**9-¿Qué modificador (switch) de ld hace falta para enlazar una aplicación de 32bits en un sistema de**

**64bits en el que se ha instalado también el compilador de 32bits?**

-m elf\_i386

**10-En una bomba como las estudiadas en prácticas, del tipo...**

**0x080486e8 <main+120>: call 0x8048524 <strncmp>**

0x080486ed <main+125>: test %eax,%eax  
0x080486ef <main+127>: je 0x080486f6 <main+134>  
0x080486f1 <main+129>: call 0x08048604 <boom>  
0x080486f6 <main+134>: ...

- a) el valor que tenga %eax
- b) el string almacenado a partir de donde apunta %eax
- c) el entero almacenado a partir de donde apunta %eax
- ✓ • d) ninguna de las anteriores

**11-¿Qué modo de funcionamiento permite a la interfaz de periféricos programable 8255 utilizar un bus bidireccional?**  
2

**12-En IA32, el registro contador de programa se denomina:**  
EIP

**13-La desventaja de las transferencias por bloques en un bus es que hay que transmitir todas las direcciones consecutivas de los datos del bloque.**  
Falso

**14-La consulta de estado que se puede llevar a cabo en la E/S programada sirve para...**  
Consultar si el dispositivo está dispuesto para recibir datos o tiene datos disponibles

**15-El controlador de interrupciones programable 8259 no permite enmascarar selectivamente líneas de interrupción.**  
Falso

**16-El concepto de "ventanas de registros solapadas" es un mecanismo eficiente de llamada/retorno de procedimientos, utilizado en el procesador RISC-I.**  
Verdadero

**17-Una posible codificación en microinstrucciones de la instrucción CALL X es:**  
SP=SP-1 ; m[SP]=PC ; PC=X

**18-La técnica de sondeo, escrutinio o "polling"...**  
Se utiliza para identificar la fuente de una interrupción

**19-El lenguaje máquina es...**  
Difícil de codificar manualmente.



**20-Un programa de ordenador que convierte un programa fuente de alto nivel completo en lenguaje máquina se llama un:**  
Compilador

**21-Un puerto de entrada de un bit puede estar constituido únicamente por un simple biestable tipo**  
D.  
Falso

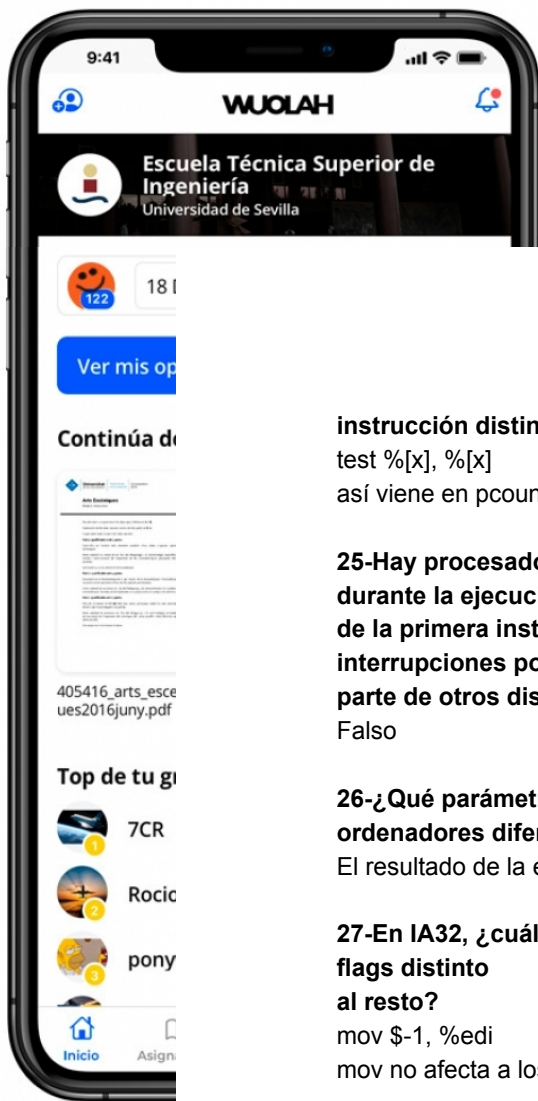
**22-Si el registro EAX contiene X, la sentencia en C**  
`x &= 0x1;`  
**se traducirá a ensamblador como:**  
`andl $1, %eax`

**23-En la práctica "media" se suma una lista de 32 enteros de 4 B con signo para producir una media y un resto usando la instrucción IDIV. ¿Cuál de las siguientes afirmaciones es falsa?**  
La media se redondea al entero más próximo  
división truncada IDIV - no se redondea, se trunca

**24-La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:**

```
int popcount3(unsigned* array,
int len){
int i, res = 0;
unsigned x;
for(i=0; i<len; i++) {
x = array[i];
asm("ini3: \n"
"shr %[x] \n"
"adc $0, %[r] \n"
"add $0, %[x] \n"
"jne ini3 \n"
: [r] "+r" (res)
: [x] "r" (x));
}
return res;
}
```

**Esta función produce siempre el resultado correcto, a pesar de que una instrucción máquina en la sección asm() es distinta a la que se esperaba después de haber estudiado pcount\_r en teoría. La**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



instrucción distinta también se podría haber cambiado por...

test %[x], %[x]

así viene en pcount\_r (Tema 2.3 tr.38)

**25-Hay procesadores que desactivan automáticamente las interrupciones sólo durante la ejecución**

**de la primera instrucción de la ISR, con lo cual es inevitable que se produzcan interrupciones por**

**parte de otros dispositivos de forma anidada.**

Falso

**26-¿Qué parámetro es más importante para comparar la velocidad de dos ordenadores diferentes?**

El resultado de la ejecución de un conjunto de programas de prueba.

**27-En IA32, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto**

**al resto?**

mov \$-1, %edi

mov no afecta a los flags

**28-El especificador de operando de una instrucción, cuando existe, es siempre una dirección de**

**memoria o de entrada/salida**

Falso

**29-Una de las diferencias entre el 8086 y el 8088 es que este último no dispone de BIU ("Bus Interface Unit").**

Falso

**30-Respecto a registros base e índice en IA32, la excepción es que**

ESP no puede ser registro índice

**31-La práctica "popcount" debía calcular la suma de bits de los elementos de un array. Un estudiante entrega la siguiente versión de popcount4:**

```
int popcount4(unsigned* array, int len)
```

```
{
```

```
int i, j, res = 0;
```

```
for(i = 0; i < len; ++i) {
```

```
unsigned x = array[i];
```

```
int n = 0;
```

```
do {
```

```
n += x & 0x01010101L;
```

```
x >>= 1;
```

```
} while(x);
```

```

for(j = 16; j == 1; j /= 2){
n ^= (n >>= j);
}
res += n & 0xff;
}
return res;
}

```

**Esta función popcount4:**

produce el resultado correcto

Caso real, entregado en prácticas. La máscara está pensada para `for(j=0;j<8;j++)`. En lugar de eso,

se hace `do...while(x)`, con lo cual no se ahorran iteraciones y todo el resultado queda acumulado

en el LSB de `n`. El `for(j)` es absurdo, no itera ninguna vez. El resultado se extrae y acumula con

`n&0xFF`. Es correcto, pero no mejora la eficiencia. `popcount2` es igual de eficiente y más elegante, porque no tiene código superfluo.

**32-Una forma usual de realizar el arbitraje distribuido consiste en una competición por la concesión del bus realizada por medio del envío por cada maestro peticionario de un número de arbitraje que lo identifica, de manera que un solo número "gane", y se le conceda el bus al maestro con ese número ganador.**

Verdadero

**33-¿Qué modificador (switch) de gcc hace falta para compilar una aplicación de 32 bits en un sistema de 64 bits?**

-m32

**34-¿Cuál de las siguientes afirmaciones es incorrecta?**

- a) En el direccionamiento inmediato el dato se encuentra en la propia instrucción
- b) El direccionamiento indexado es útil para manejo de vectores.
- c) En el direccionamiento implícito no se indica la ubicación del operando
- \*\*\*d) El direccionamiento indirecto indica la dirección del operando.

**INDICA UN PUNTERO AL OPERANDO**

**35-¿Cuál de las siguientes afirmaciones sobre los lenguajes ensambladores es falsa?**

A cada sentencia le corresponde un conjunto preestablecido de instrucciones máquina.

**36-La conexión de un 8086 a un sistema de memoria y E/S requiere algún circuito externo más en modo máximo que en modo mínimo.**

Verdadero

**37-¿Cuál de las siguientes afirmaciones es incorrecta?**

- a) Las arquitecturas RISC simplifican la decodificación
- b) Las arquitecturas RISC son del tipo registro-registro.
- c) En las arquitecturas CISC hay más instrucciones que en las RISC.
- >d)El tamaño de una instrucción en lenguaje máquina siempre ocupa dos bytes en los procesadores RISC

**38-En la nomenclatura del ensamblador de IA32, una cantidad de 16 bits es designada como:**

Word

**39-Suponga un programa residente que utiliza DMA por robo de ciclo para reproducir a través de**

**una tarjeta de sonido una melodía de fondo, que se encontraba en memoria, de manera simultánea**

**a la ejecución de un programa de usuario. La velocidad de ejecución de dicho programa de usuario**

**no se verá afectada por el programa residente, ya que éste utiliza DMA.**

Falso

**40-Una instrucción típica de entrada / salida tiene**

tiene dos argumentos: un registro del procesador y una dirección de puerto de E/S claramente inspirado en el repertorio IA32 - no encuentro dónde aparece en las transparencias -

"Ida a Data" en transparencia 40 es mapeado a memoria - evitar volver a poner esta pregunta en exámenes

**41-Si D es un desplazamiento, RI un registro índice e I una constante apropiada, el modo de**

**direccionamiento indexado con postautodecremento realiza  $EA = RI + D$  y a continuación  $RI = RI - I$ .**

**(EA = Effective Address)**

Verdadero

**42-En un computador cuya pila "crece" hacia direcciones menores se puede simular la instrucción**

**PUSH con una instrucción MOV con direccionamiento indexado con preautodecremento a través del registro SP.**

Verdadero

**43-La instrucción  $ADD Rn, \#3$  suma el contenido del registro  $Rn$  con el de la posición de memoria 3.**

Falso

**44-El fragmento de código ensamblador de un microprocesador de 8 bits**

lds IOBuf ; Apuntar puntero pila a  
; ...área mem.intermedia  
ldx Count ; Inicializar X-contador  
poll Ida a Status; Leer estado en A  
bpl poll ; Signo(A)!=1 => repetir  
Ida a Data ; Leer dato en A  
psh a ; Transferir dato a pila  
dex ; Decrementar contador X  
bne poll ; Seguir leyendo si X!=0  
corresponde a:

Entrada programada con consulta de estado

Ida a Data -> lectura del puerto de Datos

Ida a Status -> poll (consulta) de estado

**45-Siendo EDX=0xf000 y ECX=0x0100, ¿cuál de las siguientes instrucciones tiene como dirección efectiva 0xf400?**

xorl (%edx, %ecx, 4), %eax

**46-Suponiendo que todos los registros inicialmente contienen el valor 1 y que el destino es el primer registro, ¿cuál es el valor de r1 tras la ejecución de la siguiente secuencia de instrucciones?**

mov r1, #4

mov r2, #3

add r3, r1, r1

sub r1, r3, r2

mul r3, r1, r1

5

**47-En la captación de la instrucción:**

en MAR indicamos la dirección donde está la instrucción y en MBR recogemos la instrucción.

**48-En el modo mínimo el 8086 genera menos señales que en el máximo, y por tanto depende del**

**controlador de bus 8288 para generar el conjunto completo de señales de control del bus.**

Falso

**49-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:**

$N \cdot C / F$

**50-En general, cualquier ordenador debe incluir instrucciones específicas para E/S, por ejemplo IN y OUT.**

Falso



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**51-¿Cuál de los siguientes elementos no forma parte de un canal de un controlador de acceso directo a memoria?**

Registro de prioridades

**52-Señale cuál de las siguientes opciones es una técnica habitual para llevar a cabo la transferencia de datos entre el computador y los dispositivos de E/S externos:**

Acceso directo a memoria (DMA)

**53-La instrucción xor \$3, %eax tiene como resultado:**

Cambiar 0<->1 (complemento a 1 de) los 2 bits menos significativos del registro eax

**54-¿Cuál es la diferencia entre los desplazamientos a la derecha lógico y aritmético?**

Insertan de forma distinta el bit más a la izquierda

**55-Cuando se ejecuta la instrucción ret al final de una subrutina:**

la dirección almacenada en la cima de la pila se transfiere al contador de programa

**56-Después de ejecutar el siguiente código, ¿qué variables serán igual a 0? (Suponer ints de 32bits y longs de 64bits)**

c y d

En el problema 3.4 sólo se explica que una extensión de tamaño se hace según el fuente sea signed (extensión sgn) o unsigned (ext. con ceros), pero no se explica la sección 2.2.6 (y 2.2.5) en donde se aclara que las operaciones que impliquen a algún unsigned se hacen en unsigned. Evitar esta pregunta en el futuro.

**57-En el modo mínimo el 8086 genera menos señales que en el máximo, y por tanto depende del controlador de bus 8288 para generar el conjunto completo de señales de control del bus.**

Falso

**58-¿Es posible utilizar 4 GB de memoria en un sistema cuya CPU emplea E/S mapeada en memoria, cuyo bus de direcciones es de 32 bits y que tiene al menos un puerto de E/S? Supondremos que no se puede emplear ninguna técnica de extensión del bus de direcciones.**

No

**59-En un sistema de gestión de interrupciones mediante "polling" o sondeo, el dispositivo que solicita la interrupción envía, junto con la señal de petición de interrupción, su correspondiente vector de interrupción.**

Falso

**60-Alguna de las siguientes líneas de código sirve para definir una variable entera llamada tam en GNU/as Linux x86. ¿Cuál?**

tam: .int .-msg

**61-En la práctica "media" se pide calcular la media y resto de una lista de 32 enteros CON signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando desbordamiento. ¿Qué (media : resto) se debe obtener para una lista rellena a -1 salvo el primer elemento, que valiera -31?**

(-1 :-30)

**62-Si declaramos `int val[5]={1,5,2,1,3};` entonces**

`&val[2]` es de tipo `int*` y vale lo mismo que `(void*)val+8`

En Sep15 faltaba `(void*)` y entonces sería falsa por aritmética de punteros

**63-Un procesador de 8 bits, ¿a cuántos puertos de E/S podrá acceder?**

Depende del método de selección de periféricos que emplee

**64-Las interrupciones generadas por el teclado interrumpirán al procesador: sólo si el procesador tiene activado el indicador de habilitación de interrupciones**

**65-La codificación Huffman no es la más utilizada debido a que el tamaño promedio del código de operación no es mínimo.**

Falso

**66-¿Qué hace `gcc -O1`?**

Compilar con optimización

**67-En un máquina con arquitectura de pila, todas las instrucciones aritméticas tienen dos operandos implícitos: la cima de la pila y el elemento siguiente de la cima de la pila.**

Falso

**68-Se suelen utilizar PLA en las unidades de control cableadas.**

Verdadero

**69-Suponga que la micropalabra de una máquina microprogramada tiene 8 bits de ancho y se usan 16 micropalabras diferentes en un microprograma de 256 micropalabras. Si se usa nanoprogramación...**

se ahorran bits pero el funcionamiento es más lento.

**70-De los siguientes grupos de técnicas de E/S, ¿cuáles están controlados por programa?**

E/S programada y E/S mediante interrupciones.

**71-¿Cuál de las siguientes instrucciones x86 se puede usar para sumar dos registros y guardar el resultado sin sobrescribir ninguno de los registros originales?**

lea

**72-La instrucción leave equivale a:**

movl %ebp, %esp; popl %ebp

**73-En el 8086, si tras una instrucción de comparación CMP A,B aparece una instrucción JC, ésta realiza un salto si  $A > B$ , siendo A y B números sin signo.**

Falso

**74-La instrucción necesaria para cargar 0x07 en %eax es:**

movl  
\$0x07,%eax

**75-Compilar .c→exe (de fuente C a ejecutable) usando sólo as y ld, sin gcc...**

No se puede

**76-Si varios dispositivos comparten una única línea de solicitud de interrupción:**

todas son ciertas

**77-¿Cuál es el contenido de una pila al terminar de ejecutarse la siguiente secuencia de operaciones push y pop:**

push #1  
push #2  
push #3  
pop a  
push #4  
pop a  
pop a



**78-¿Cuál de las siguientes secuencias de instrucciones calcula  $a=b-a$ , suponiendo que `%eax` es  $a$  y `%ebx` es  $b$ ?**

Varias o ninguna de las respuestas anteriores son correctas, no se puede marcar una y sólo una

**79-La práctica "parity" debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity6:**

```
int parity6(unsigned * array, int len)
{
 int i, result = 0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 asm("mov %[x], %%edx \n\t"
 "shr $16, %%edx \n\t"
 "shr $8, %%edx \n\t"
 "xor %%edx, %%edx \n\t"
 "setp %%dl \n\t"
 "movzx %%dl, %[x] \n\t"
 : [x] "+r" (x)
 :
 : "edx"
);
 result += x;
 }
 return result;
}
```

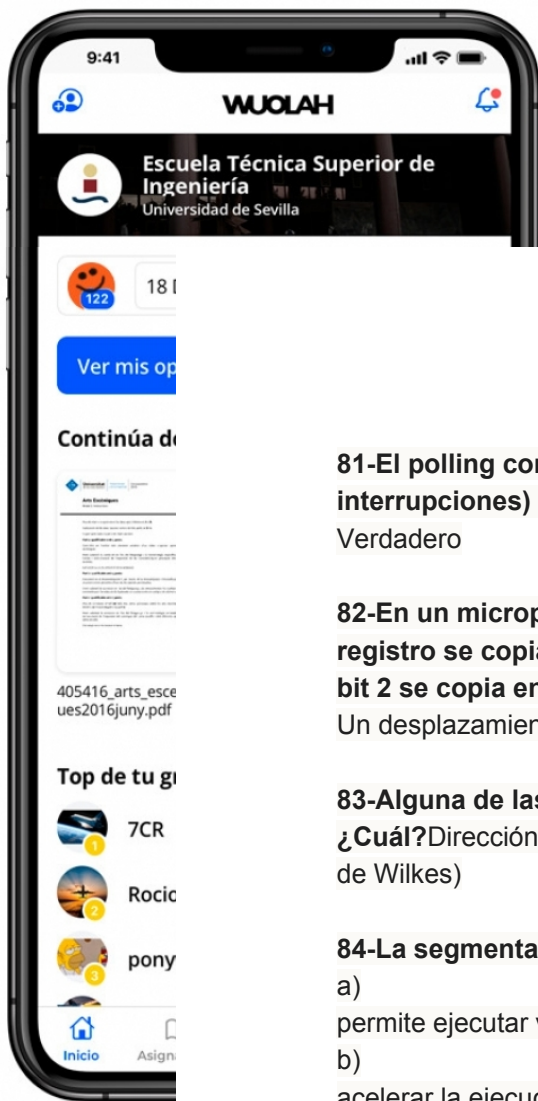
**Esta función parity6:**

no es correcta; fallaría por ejemplo con `array={0,1,2,3}`

Caso real, entregado en prácticas. Las tres primeras instrucciones asm se pierden al poner `edx` a 0 usando `xor`. Consecuentemente, se activa PF para ajustar a impar, y termina siendo `x=1`. Es decir, todos los elementos del array contabilizan paridad=1. El array `{1,2,4,8}` pasa desapercibido, pero `{0,1,2,3}` debería producir resultado=2<>4.

**80-En el contexto del lenguaje máquina, el acrónimo ISA suele referirse a:**

Instruction Set Architecture



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



**81-El polling consiste en chequear cada fuente de eventos (por ejemplo interrupciones) en algún orden predeterminado.**

Verdadero

**82-En un microprocesador de 4 bits, una operación en la que el bit 0 de un registro se copia en el acarreo, después el bit 1 se copia en el bit 0, después el bit 2 se copia en el bit 1, y por último el bit 3 se copia en el bit 2, es:**

Un desplazamiento aritmético a la derecha.

**83-Alguna de las siguientes señales NO es salida de la unidad de control.**

¿Cuál? Dirección de la siguiente microinstrucción (bits del campo DIR o Memoria B de Wilkes)

**84-La segmentación de cauce...**

- a) permite ejecutar varias instrucciones concurrentemente
- b) acelerar la ejecución de un programa
- c) provoca riesgos debido a datos
- ✓ d) todas las respuestas son ciertas

**85-El bus AGP permite que los aceleradores gráficos accedan directamente a texturas almacenadas en la memoria principal.**

Verdadero

**86-¿Qué método de control de acceso directo a memoria es preferible por velocidad (más rápida), economía (coste no prohibitivo) y conveniencia de diseño (compatible con memorias y sistemas actuales)?**

Transferencia de bloques o parada de CPU

**87-En la práctica "media" se pide sumar una lista de 32 enteros SIN signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando perder acarreo. De entre los siguientes, ¿cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32 bits (sin signo)? Se usa notación decimal y espacios como separadores de millares/millones/etc.**

1 000 000 000

Se pasa, 1000 millones >> 128M

**88-¿Qué método de identificación de la fuente de una interrupción suele ser más económico desde el punto de vista hardware?**

La identificación mediante la técnica de sondeo

**89-Un computador tiene una memoria de control de 16 Kpalabras de 250 bits, de las que 447 son diferentes. ¿Cuántos bits ahorramos usando nanoprogramación en lugar de microprogramación?**

a)

3928652

b)

259206

c)

287935

✓•d)

ninguno de los resultados anteriores es exacto

**90-Las instrucciones máquina que aparecen en segundo lugar en el análisis dinámico de uso de instrucciones son las de bifurcación.**

Verdadero

**91-Considerar las siguientes declaraciones de estructuras en una máquina Linux de 64-bit.**

```
struct RECORD {
 long value2;
 int ref_count;
 char tag[4];
};
```

```
struct NODE {
 double value;
 struct RECORD record;
 char string[8];
};
```

**También se declara una variable global "my\_node" como sigue:**

```
struct NODE my_node;
```

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

32 bytes

**92-La función gettimeofday() en la práctica de la "bomba digital" se utiliza para**  
Para cronometrar lo que tarda el usuario en introducir la contraseña

**93-En un sistema Linux x86-64, ¿cuál de las siguientes expresiones es equivalente a la expresión C  $(x[2] + 4)[3]$ ? Suponer que previamente se ha declarado `int **x`.**

`*((*(x + 2) + 4) + 3)`

sería `x[2][4+3]`, `+4+3 == +7` sí sería aritmética punteros

**94-En las arquitecturas RISC hay...**

muchos registros y pocos modos de direccionamiento.

**95-En la práctica "media" se pide sumar una lista de 32 enteros CON signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando desbordamiento. ¿Cuál es el valor negativo más pequeño (en valor absoluto) que repetido en toda la lista causaría desbordamiento con 32bits (en complemento a 2)?**

`0xfbff ffff`

**96-En el direccionamiento indirecto a través de registro, la dirección efectiva...**  
se encuentra en un registro general del procesador.

**97-En 80x86, los parámetros a las subrutinas se pueden pasar:**

a)

a través de variables globales

b)

a través de los registros

c)

a través de la pila

✓ • d)

todas las anteriores son ciertas

**98-La etiqueta del punto de entrada a un programa ensamblador en el entorno de las prácticas 1 a 4 (GNU/as Linux x86) es:**

`_start`

P3 y P4 se redactan en C.

P1 y P2 sí son en ensamblador GNU/as Linux x86.

Incluso en P2 se llega a usar main para ensamblar con gcc ya que usamos printf.

En cualquier caso, las otras opciones son descabelladas

**99-Sean un  $\text{int}^*a$  y un  $\text{int } n$ . Si el valor de `%ecx` es `a` y el valor de `%edx` es `n`, ¿cuál de los siguientes fragmentos de ensamblador se corresponde mejor con la sentencia C `return a[n]`?**

```
mov (%ecx,%edx,4),%eax
ret
```

**100-Durante un robo de ciclo DMA el procesador mantiene en alta impedancia el bus de direcciones.**

Verdadero.

**101-Una cola de precaptación sirve para:**

Reducir el efecto de los fallos de cache

**102-¿Cuál es el contenido de la pila al terminar de ejecutarse la siguiente secuencia de instrucciones de una arquitectura de pila: `push #4; push #7; push #8; add; push #10; sub; mul`?**

20

**103-En el RISC-I, una ventana de registros contiene:**

registros de propósito general.

**104-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:**

Habilitar la salida triestado del registro `r1` y activar la carga del registro `r2`

**105-Se llama bit slice a la operación de desplazamiento de bits que realiza una instrucción `SHL` o `SHR`.**

Falso

**106-De los siguientes grupos de técnicas de E/S, ¿cuáles están controlados por programa?**

E/S programada y E/S mediante interrupciones.



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



### 107-Respecto a MBR y MAR

AR requiere menos señales de control que MBR  
en Tema 3 tr.10, sólo una (LoadMAR), frente a 4 de MBR (Load/Enable Mem/Bus)

### 108-Las instrucciones máquina más usadas según el análisis dinámico son las de bifurcación

Falso

### 109-Si %edx contiene 0xf000 y %ecx contiene 0x0100, el direccionamiento 0x80(%ecx,%edx,2) se refiere a la posición

0x1e180

### 110-Señale cuál de las siguientes opciones no es un modo para llevar a cabo la transferencia de datos entre el computador y los dispositivos de E/S externos:

E/S por flanco

### 111-El hecho de utilizar "polling" no implica que la prioridad de los dispositivos interruptores quede fijada mediante encadenamiento ("daisy-chain").

Verdadero

### 112-¿En qué pareja de registros están el dato/instrucción que se leerá o escribirá en memoria, y la dirección de memoria?

MBR y MAR

### 113-Debido al pequeño número de operaciones simples, la sección de control de un procesador RISC puede ser cableada en lugar de microprogramada.

Verdadero

### 114-Se podría diseñar una CPU microprogramada de manera que la captación y la ejecución de microinstrucciones se solapasen en el tiempo.

Verdadero

### 115-De entre las siguientes construcciones de flujo de control en lenguaje C, la que se traduce más directamente a lenguaje ensamblador es...

El bucle do-while

### 116-¿Qué tipo de direccionamiento se usa para el registro destino en la instrucción IA32 add array(,%ebx,4), %edx?

Direccionamiento a registro

**117-La instrucción test es...**

Lo mismo que and, pero no guarda el resultado, sólo ajusta los flags

**118-El primer computador electrónico basaba su funcionamiento en:**

tubos de vacío

1ª generación

**119-¿Cuál de las siguientes afirmaciones es cierta respecto al lenguaje C?**

En lenguaje C, al llamar a una subrutina o función se introducen los parámetros en la pila y después se realiza una llamada a la subrutina

suponiendo convención cdecl x86, porque x86-64 usa regs.

**120-¿Cuál es la característica tecnológica principal de la segunda generación de computadores?**

Los transistores

**121-Un procesador cuya instrucción CALL guarda la dirección de retorno en un registro:**

No permite llamadas anidadas ni recursivas.

**122-Un procesador de 8 bits, ¿a cuántos puertos de E/S podrá acceder?**

Depende del método de selección de periféricos que emplee

**123-Las instrucciones de salto...**

complican el diseño eficiente de los procesadores segmentados.

**124-En la técnica de salto retardado:**

El compilador puede reorganizar el código para rellenar los huecos de retardo con instrucciones útiles

**125-En la actualidad todos los microprocesadores utilizan pipeline, tanto RISC como CISC.**

Verdadero

**126-En una máquina con 32 registros direccionables e instrucciones de 16 bits, es posible codificar 63 instrucciones de dos registros, 31 instrucciones de 1 registro y 32 instrucciones de 0 direcciones.**

Verdadero

**127-¿Qué nº de modo de funcionamiento permite a la interfaz de periféricos programable 8255 utilizar un bus bidireccional?**

2, el modo 2.

**128-En una arquitectura RISC típica:**

suele usarse segmentación

**129-El controlador de DMA programable 8237 puede realizar una operación de acceso directo a memoria en la que se transfiera un bloque de 512 KBytes sin intervención de la CPU una vez comenzada la transferencia.**

Verdadero

**130-¿Cuál de los siguientes elementos no forma parte del canal de un controlador de acceso directo a memoria?**

Registro de prioridades

**131-¿Cuál es el contenido de la pila al terminar de ejecutarse la siguiente secuencia de instrucciones de una arquitectura de pila?:**

push #4

push #7

add

push #10

sub

a) 1

**132-Un procesador está segmentado en las etapas F, D, E, M y W. Cada una de ellas consume un tiempo t. La aceleración ideal (si no hay riesgos) al ejecutar n instrucciones respecto a un procesador no segmentado será:**

a)  $5n / (4+n)$

$kn/(k+n-1)$ , con

$k=5$

**133-La idea de desarrollar máquinas CISC surgió para:**

tener instrucciones cercanas al lenguaje de alto nivel.

**134-¿En qué método para determinar la dirección de comienzo de una rutina de servicio de interrupción se envía parte de dicha dirección?**

Direccionamiento relativo



**135-Un procesador con unidad de control microprogramada tiene un generador de direcciones de memoria de microprograma en lugar de otros mecanismos de resolución de dirección efectiva.**

Falso

**136- Si un computador X ejecuta un programa de 450 millones de instrucciones en 26 segundos y un computador Y tarda 14 segundos en ejecutar ese mismo programa, ¿cuántas veces es más rápido el computador Y que el X?**

1.857

**137-Suponga la siguiente sentencia asm en un programa:**

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

**¿Cuál de las siguientes afirmaciones es correcta?**

a) r es una posición de memoria de entrada/salida

b) el código de retorno de la función asm se fuerza a que esté en la variable result



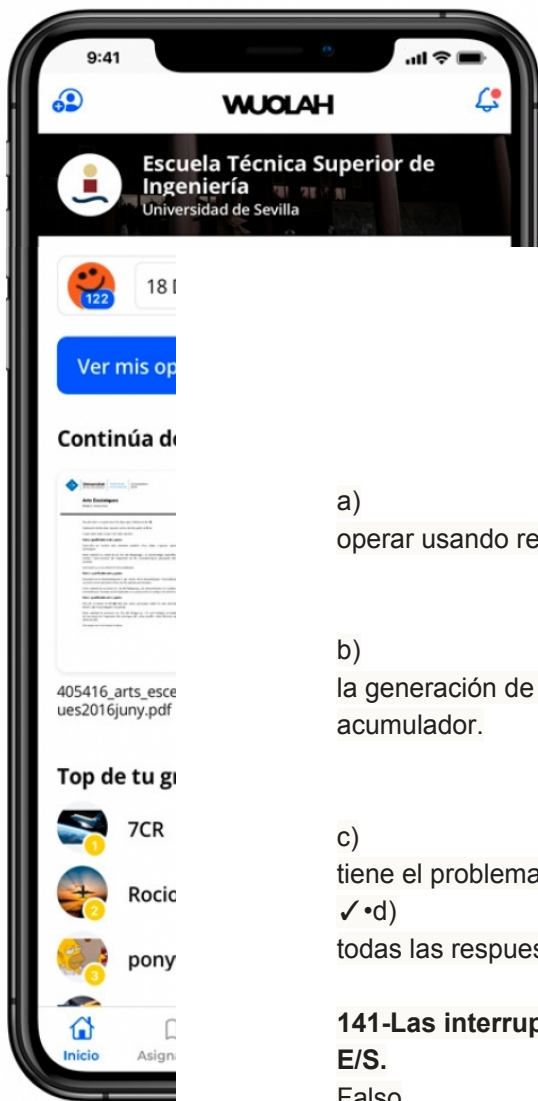
c) i es un registro de entrada

d) a es una posición de memoria de entrada

**138-Un computador que utilice únicamente controladores programables 8237 para realizar el DMA por robo de ciclo puede realizar:**  
más de cuatro transferencias por DMA concurrentes

**139-La instrucción máquina DI (Disable Interrupts), conocida como CLI (Clear Interrupt Flag) en x86, se utiliza para desactivar:**  
Todas las interrupciones enmascarables

**140-En una arquitectura de registros (a nivel de lenguaje máquina):**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



a)  
operar usando registros es más rápido.

b)  
la generación de código resulta más simple que en arquitecturas de pila o acumulador.

c)  
tiene el problema de que las instrucciones pueden ser largas.  
✓ •d)  
todas las respuestas anteriores son ciertas.

**141-Las interrupciones se utilizan únicamente para realizar operaciones de E/S.**  
Falso

**142-¿Cuál de las siguientes afirmaciones es falsa respecto al lenguaje C (en convención cdecl x86)?**

a)  
Al llamar a una subrutina o función se introducen los parámetros en la pila y después se realiza una llamada a la subrutina

b)  
Los parámetros se introducen en la pila en el orden inverso a como aparecen en la llamada de C, es decir, empezando por el último y acabando por el primero

•c)  
Antes de volver de la rutina llamada, el programa en C se encarga de quitar de la pila los parámetros de llamada realizando varios pop

x

d)  
Pasar a una función un puntero a una variable se traduce en introducir en la pila el valor de la dirección de memoria donde está almacenada la variable

**143-El objetivo del control residual es optimizar el tamaño del microprograma.**

WUOLAH

Verdadero

**144-Uno de los objetivos del uso de direccionamiento relativo al contador de programa en los saltos es posibilitar la escritura de programas relocizables.**

Verdadero

**145-En general, el acceso a una palabra no alineada es más lento que a una alineada.**

Verdadero

**146-El pipeline o segmentación consiste es una técnica para solapar la ejecución de varias instrucciones máquina.**

Verdadero

**147-El 8086 tiene:**

14 registros de 16 bits y un bus de direcciones de 20 bits

**148-En una memoria de bytes que contuviera a partir de la posición 0 los valores 1,0,0,0,0xFE,0xFF,0xFF,0xFF, se puede decir que...**

a)

Hay una palabra de 16bit big-endian con valor 1 en la posición 0

b)

Hay una palabra de 16bit little-endian con valor 254 en la posición 3

c)

Hay una palabra de 32bit little-endian con valor -1 en la posición 4

•d)

Todas las respuestas anteriores son incorrectas

**149-Los procesadores comerciales con unidad de control microprogramada suelen almacenar los microprogramas...**

en una ROM.

**150-El arbitraje estático de un bus es muy simple, pero es poco flexible, y además hay que calcular previamente qué ancho de banda va a requerir cada dispositivo para no bloquear al más rápido.**

Verdadero

**151-Si el registro %eax contiene el siguiente valor binario:**

**11111111 10101010 01010101 11110000**

**¿Cuál será el valor de %eax tras ejecutar la instrucción xorb %al, %al?**

**11111111 10101010 01010101 00000000**

**152-¿Cuál de las siguientes tareas no es responsabilidad de un circuito de interfaz o controlador de periféricos sencillo?**

Ejecutar el programa de transferencia de información entre el procesador y los dispositivos de E/S

**153-¿Cómo se almacenaría como palabra de 32 bits el número -128 en un sistema que utilice el criterio del extremo menor ("little endian")?**

a)

posición 0: FF pos.1:FF pos.2: FF pos.3: 00

b)

0:00 1:FF 2:FF 3:FF

c)

0:00 1:01 2: 00 3:80

✓•d)

Ninguna de las anteriores

**154-La E/S por DMA requiere la presencia de un circuito controlador de DMA.**

Verdadero

**155-¿Cuál de los siguientes registros tiene que ser salvaguardado (si va a modificarse) dentro de una subrutina según la convención cdecl para IA32?**  
ebx

**156-Una señal Bus Grant sirve para indicar que se cede el control del bus a un módulo que lo había necesitado.**

Verdadero

**157-La instrucción INTO del 8086 realiza la entrada de un dato desde un puerto de entrada al registro AL o AX.**

Falso

**158-Estudiando el listado de una función C presuntamente compilada con gcc en modo 64bit (x86-64), nos dicen que la primera instrucción, movl %eax, (%rdi), carga en EAX el valor adonde apunta el primer argumento.**

Está mal, porque EAX no se carga con ningún valor

**159-Alguna de las siguientes no es una operación básica de la unidad de control**

a)

(leer o escribir) un registro (de / a) memoria

✓•b)

(guardar o recuperar) un registro (en / de) la pila

c)

transferir un registro a otro

d)

realizar operación ALU y guardar resultado en registro

**160-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), se almacenan a partir de la posición 0xCAFEBA0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al consultar la media palabra de la posición 0xCAFEBA4?**

-1

los contenidos son

CAFEBA0: FF FF FF FF

CAFEBA4: FE FF FF FF

CAFEBA8: FD FF FF FF

CAFEBA0C: FC FF FF FF

las últimas dos posiciones, a partir de CAFEBA0C, contienen FF FF, que es -1



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



### 161-Si usamos una estructura de bus con DMA:

la CPU puede dejar las transferencias entre MP y periféricos en manos de este controlador (DMA) y seguir ejecutando otras instrucciones.

**162-La diferencia entre temporización de bus asíncrona y semisíncrona es que en la asíncrona las transferencias ocurren en algún múltiplo de ciclo de reloj, y en la semisíncrona no existe reloj del bus.**

Falso

### 163-Una sentencia en C del tipo "while (test) body;" puede transformarse en código "goto" como:

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

### 164-Alguna/s de las ventajas de la E/S mapeada en memoria frente a la E/S aislada o independiente es/son:

El diseño de la CPU es más sencillo.

### 165-Respecto a las unidades de control nanoprogramadas:

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.

### 166-¿Cuál de los siguientes fragmentos es correcto para comenzar un programa en ensamblador que conste de un solo archivo .s?

```
.section .text
.global _start
_start:
```

### 167-Las arquitecturas memoria-memoria tienen la ventaja de emplear instrucciones máquina muy cortas y así los accesos a memoria son mínimos.

Falso

### 168-¿Cuál de las siguientes parejas de buses contiene sólo buses de tipo serie?

**169-De las siguientes instrucciones, ¿cuál utiliza un modo de direccionamiento no implícito?**

push

**170-En una unidad de control microprogramada con formato de microinstrucciones vertical, un subcampo que deba especificar 16 señales de control codificadas de tal forma que pueda activarse sólo una o ninguna habrá de tener una anchura mínima de**

5 bits

**171-En un computador cuya pila "crece" hacia direcciones menores se puede simular la instrucción PUSH con una instrucción MOV con direccionamiento indexado con preautodecremento a través del registro SP.**

Verdadero

**172-El registro RDM (MAR en inglés) contiene la última instrucción o dato leído de memoria o el dato que se va a escribir en memoria.**

Falso

**173-¿Cuál de los siguientes es el orden correcto en el ciclo de compilación de un programa en lenguaje C? (el fichero sin extensión es un ejecutable):**

fich.c → fich.s → fich.o → fich

**174-Técnicas que se pueden usar para determinar la causa de una interrupción (señalar la opción incorrecta)**

a)

interrupciones vectorizadas

b)

múltiples líneas de interrupción INT1#, INT2#...

c)

consulta de estado, o polling

✓•d)

línea de reconocimiento INTA#

**175-En las instrucciones de salto condicional se suele usar direccionamiento relativo.**

Verdadero

**176-Para compilar un programa escrito en C en el entorno GNU/Linux se usa el programa:**

gcc

**177-La tendencia actual y futura en buses de periféricos es pasar de diseños serie a paralelo.**

Falso

**178-En el fragmento de código**

```
804854e:e8 3d 06 00 00 call 8048b90 <main>
8048553:50 pushl %eax
```

**la instrucción call suma al contador de programa la cantidad:**

0x0000063d

**179-En el pseudocódigo usado para representar las microinstrucciones, la expresión “goto f(IR)”:**

Salta a una dirección de memoria de control que depende de la instrucción máquina actual.

**180-¿Cuál fue el primer procesador de Intel de 64-bits en la familia x86(-64)?**

Pentium 4F

**181-Un procesador con una unidad de control microprogramada tiene una memoria de control de 256 palabras de 16 bits, de las que 128 son diferentes.**

**¿Qué ahorro en número de bits obtendríamos si usáramos nanoprogramación?**

256 bits



**182-En el secuenciamiento de microinstrucciones explícito cada microinstrucción incluye la dirección de la microinstrucción siguiente.**

Verdadero

**183-La instrucción JGE / JNL provoca un salto si...**

OF = SF

basta recordar que "Less" no era un flag solo (es  $OF \wedge SF$ )

recordar también que "Below" comprueba CF

**184-El direccionamiento directo a memoria utiliza...**

un desplazamiento.

**185-En un sistema con interrupciones vectorizadas, el dispositivo o interfaz siempre suministra la dirección de la rutina de servicio de interrupción, aunque a esa dirección le falten bits.**

Falso

**186-El direccionamiento relativo necesita hacer uso de una tabla de direcciones, cada una de las cuales apunta a la base del siguiente dato.**

Falso

**187-¿Cuál de los siguientes no es un tipo de bus?**

Secuencial

opuestos a buses paralelos son los buses serie

opuestos a programas secuenciales son los programas paralelos

**188-La operación aritmética calculada por el programa**

```
mov $5, %eax
```

```
mov $3, %ebx
```

```
mov $7, %ecx
```

```
mov $8, %edx
```

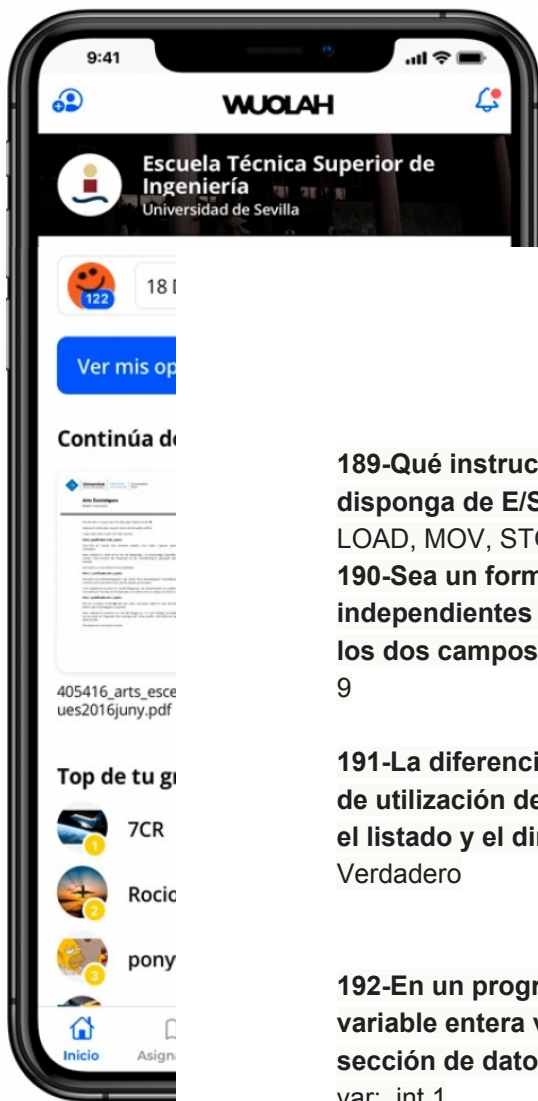
```
mul %ebx, %ecx
```

```
add %ecx, %eax
```

```
sub %edx, %eax
```

es:

$5 + (3 \times 7) - 8$



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



**189-Qué instrucciones son típicas del repertorio de un procesador que sólo disponga de E/S mapeada en memoria?**

LOAD, MOV, STORE

**190-Sea un formato de microinstrucción que incluye dos campos independientes de 10 bits cada uno. Si se rediseña de modo que se solapen los dos campos, ¿cuántos bits se ahorran en cada microinstrucción?**

9

**191-La diferencia entre un análisis estático y uno dinámico sobre la frecuencia de utilización de las instrucciones máquina es que el estático recuenta sobre el listado y el dinámico contabiliza las repeticiones debidas a bucles, etc.**

Verdadero

**192-En un programa en ensamblador queremos crear espacio para una variable entera var inicializada a 1. La línea que hemos de escribir en la sección de datos es:**

var: .int 1

**193-El espacio direccionable de memoria de un computador depende del diseño del:**

Bus de direcciones

**194-Si almacenamos según el criterio little-endian la palabra de 64 bits 0xFACEB00C a partir de la dirección 0xCAFEBAE, el byte 0xCE quedará almacenado en la dirección:**

0xCAFEBA0

**195-Un fragmento de una “bomba” desensamblada es:**

```
0x0804873f: call 0x8048504 <scanf>
0x08048744: mov 0x24(%esp),%edx
0x08048748: mov 0x804a044,%eax
0x0804874d: cmp %eax,%edx
0x0804874f: je 0x8048756 <main+230>
0x08048751: call 0x8048604 <boom>
0x08048756: ...
```

**La contraseña/clave en este caso es...**

el entero almacenado a partir de la posición de memoria 0x804a044  
se hace `mov 0x0804a044, %eax` justo antes de `cmp %eax, %edx`, en donde `%edx` está relacionado con `scanf`

**196-En el 8086, la secuencia de instrucciones `ADD SP,10` seguida de `RET` es equivalente a `RET 10`.**

Falso

**197-Si la variable `val` está almacenada en `ebx` y la variable `x` está almacenada en `eax`, la sentencia `val ^= x`; se puede traducir a ensamblador como:  
gunas de las ventajas de la E/S mapeada en memoria frente a la E/S aislada o independiente son:  
`xorl %eax,%ebx`**

**198-La primera generación de computadores se caracteriza por el uso de:**  
Tubos de vacío.

**199-La práctica “parity” debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de `parity6`:**

```
int parity6(unsigned* array, int len){
 int i,j,res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 asm("\n"
 "mov %[x],%%edx \n\t"
 "shr $16, %%edx \n\t"
 "xor %%edx,%[x] \n\t"
 "mov %[x],%%edx \n\t"
 "mov %%dh, %%dl \n\t"
 "xor %%edx, %[x]\n\t"
 "setpo %%cl \n\t"
 "movzx %%cl, %[x]"
 :[x] "+r" (x)
```

```

:
:"edx","ecx"
);
res+=x;
}
return res;
}

```

La sentencia asm() del listado anterior tiene las siguientes restricciones

a)

ninguna

b)

arquitectura de 32 bits

☒ X

c)

dos entradas y una salida

•d)

un registro y dos sobrescritos (clobber)

el registro es [x] "+r" y los sobrescritos son "edx","ecx"

**200-En la E/S controlada por interrupciones la CPU no tiene que ejecutar un programa para realizar la transferencia de datos.**

Falso

**201-Hay procesadores que desactivan automáticamente las interrupciones sólo durante la ejecución de la primera instrucción de la ISR, con lo cual es inevitable que se produzcan interrupciones por parte de otros dispositivos de forma anidada**

Falso

**202-En los procesadores CISC gran parte del área del chip se consume en la unidad de control.**

Verdadero

**203-¿Cuál es la diferencia entre las instrucciones mov y lea?**

mov referencia (accede) la posición indicada, mientras que lea no lo hace

**204-Un sistema no segmentado tarda 10 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce (pipeline) de 5 segmentos con un ciclo de reloj de 4 ns. Cuando se procesan muchas tareas, la ganancia máxima de velocidad que se obtiene se acerca a:**

2,5

**205-Considerar las siguientes declaraciones de estructuras en una máquina Linux de 64-bit.**

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

```
struct NODE {
 long value;
 struct RECORD record;
 char string[8];
};
```

**También se declara una variable global "my\_node" como sigue:**

**struct NODE my\_node;**

**Si la dirección de my node es 0x600940, ¿cuál es el valor de &my\_node.record.tag[1]?**

0x60094f

**206-Un computador tiene una memoria de control de 640 palabras de 70 bits, de las que 280 son diferentes. ¿Qué ahorro en número de bits obtendríamos si usáramos nanoprogramación en lugar de microprogramación?**

19440



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**207-El arbitraje dinámico de un bus es muy simple, pero es poco flexible, y además hay que calcular previamente qué ancho de banda va a requerir cada dispositivo para no bloquear al más rápido**

Verdadero

**208-La E/S programada con consulta de estado ha de ser mapeada en memoria.**

Falso

**209-Si el contenido de ESP es 0xAC00 antes de ejecutar la instrucción push %ebx. ¿Cuál será su contenido tras ejecutarla?**

0xABFC

**210-¿En qué modo de funcionamiento la interfaz de periféricos programable 8255 permite la E/S con bus bidireccional?**

modo 2

**211-Utilizando E/S programada y como modo de direccionamiento selección lineal, ¿cuántos periféricos podrían conectarse a un 8086? Recordar que el 8086 disponía de E/S separada, con bus de direcciones de 20 bits para memoria y de 16 bits para E/S.**

16 periféricos

**212-Un sistema no segmentado tarda 50 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce ("pipeline") de 5 segmentos con un ciclo de reloj de 50 ns. Cuando se procesan muchas tareas, la ganancia de velocidad que se obtiene se acerca a 5.**

Falso

**213- ¿Es cierto que la consulta de estado permite averiguar el estado de un periférico y alterar la prioridad de los periféricos?**

Sí, permite averiguar el estado de un periférico y también alterar la prioridad de los periféricos

**214-Las siguientes afirmaciones sugieren que el tamaño de varios tipos de datos en C (usando el compilador gcc) son iguales tanto en IA32 como en x86-64. Sólo una de ellas es FALSA. ¿Cuál?**

•a)

El tamaño de un puntero es 4 bytes

X

b)

El tamaño de un int es 4 bytes

c)

El tamaño de un double es 8 bytes

d)

El tamaño de un short es 2 bytes

**215-Un computador que utilice únicamente controladores programables 8237 para realizar el DMA puede realizar más de cuatro transferencias por DMA concurrentes.**

Verdadero

**216-La conexión de las salidas de tres registros hacia un bus común en el camino de datos puede realizarse usando...**

dos multiplexores de 2 a 1

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

CMP, TEST

**218-¿Cuál de los siguientes grupos de instrucciones podrá pertenecer a un procesador con E/S mapeada en memoria?**

LOAD, MOV, STORE

**219-Al traducir de lenguaje C a ensamblador, gcc en máquinas Linux/IA-32 almacena (reserva espacio para) una variable “var” local a una función “fun” en una dirección de memoria referenciable (en lenguaje ensamblador) como: -k(%ebp), siendo k un número constante positivo relativamente pequeño**

**220-¿En qué técnica para determinar la dirección de comienzo de la rutina de servicio de interrupción se fija dicha dirección en los circuitos de la CPU?**  
Direcciones fijas.

**221-Si  $R0=2$ ,  $R1=5$  y  $M[3]=3$  ¿Qué valor toman  $R0$ ,  $R1$  y  $M[3]$  tras ejecutarse la instrucción  $XOR\ 1h[R0],R1$ ?**  
 $R0=2$  ,  $R1=5$  ,  $M[3]=6$

**222-El sufijo l de la instrucción movl significa:**  
Que la instrucción trabaja con operandos de 32 bits (long word).

**223-Las ventanas de registros solapadas se utilizan para ahorrar registros en el diseño del procesador.**  
Falso

**224-¿Cuál de los siguientes lenguajes no permite el paso de parámetros por referencia?**  
C

**225-Teniendo en cuenta que el tiempo de ejecución de un programa es proporcional al producto número de instrucciones del programa \* número medio de ciclos de reloj por instrucción, la filosofía CISC pretende reducir el tiempo de ejecución reduciendo el número de instrucciones del programa y la RISC reduciendo el número medio de ciclos de reloj por instrucción.**  
Verdadero

**226-En el 80x86, la instrucción "JNE etiqueta" es equivalente a "JNZ etiqueta".**  
Verdadero

**227-Tras comparar números con signo, para saltar si menor se utilizan:**  
El overflow y el signo

**228-Dada la siguiente declaración en lenguaje C, una estructura de este tipo podría ocupar, bien sea en un sistema Linux IA32 o bien en uno x86-64, un total de...**



```
struct a{
 int i;
 double d;
 char c;
 short s; };
```

24 B

4 B relleno entre i y d, 1B entre c y s, 4B relleno tras s.

**229-Cuál de las siguientes características es posterior a la segunda generación de computadores?**

RISC.

**230-Si usamos una estructura de bus con DMA:**

la CPU puede dejar las transferencias entre MP y periféricos en manos de este controlador (DMA) y seguir ejecutando otras instrucciones.

**231-La especificación de un bus a nivel eléctrico debe incluir las siguientes partes:**

Alimentación, impedancia, nivel de señal,...

**232-¿Cuál de las siguientes afirmaciones es cierta?**

En E/S independiente, las instrucciones de acceso a memoria suelen ser más largas que las de E/S

**233-Un conjunto de microoperaciones compatibles constituye una microinstrucción, y una secuencia de microinstrucciones es un microprograma.**

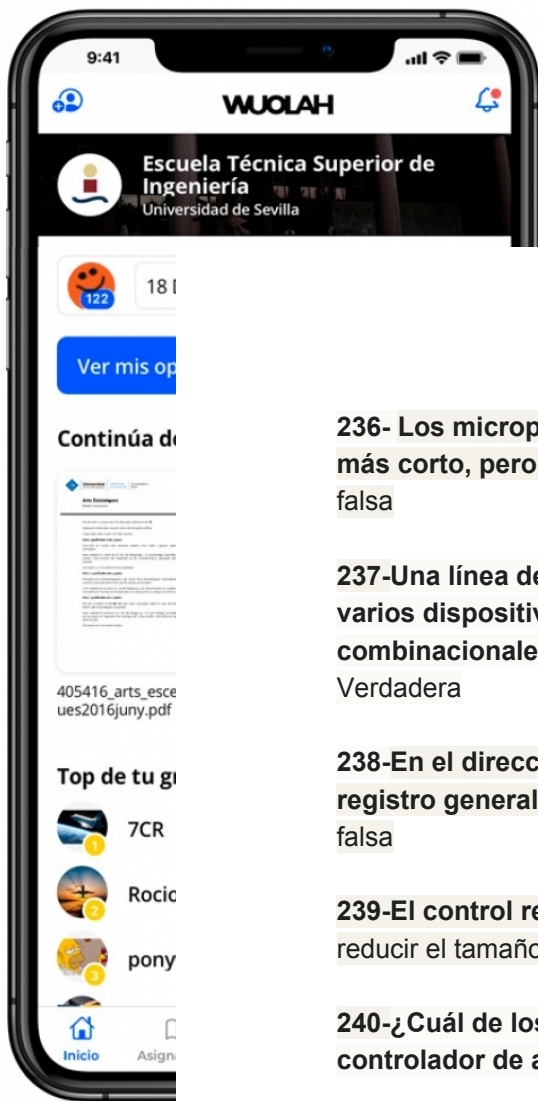
Verdadero

**234-¿Cuáles de las siguientes instrucciones utilizan sólo direccionamiento implícito?**

lahf, movs

**235-La instrucción cmovb %edx, %eax**

copia en %eax el contenido de %edx si el indicador de acarreo es 1  
"below" equivale a CF



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



**236- Los microprocesadores RISC puede ser implementados en un tiempo más corto, pero requieren mayor área de silicio que los CISC.**  
falsa

**237-Una línea de interrupción organizada en colector abierto permite conectar varios dispositivos de manera sencilla sin necesidad de circuitos combinacionales (puertas OR).**  
Verdadera

**238-En el direccionamiento directo, la dirección efectiva se encuentra en un registro general del procesador.**  
falsa

**239-El control residual se utiliza para:**  
reducir el tamaño de la memoria de control

**240-¿Cuál de los siguientes elementos no forma parte de un canal de un controlador de acceso directo a memoria?**

- a)Registro contador
- b)Registro de órdenes
- c)Registro de dirección
- \*\*d)Registro de prioridades**

**241-En la E/S mapeada en memoria, el procesador (CPU) no distingue entre accesos a memoria y accesos a los dispositivos de E/S.**  
verdadero

**242-Si varios dispositivos comparten una única línea de solicitud de interrupción se puede usar "polling" para identificar el origen de una interrupción.**  
verdadero

**243-Una operación de transferencia READ-MODIFY-WRITE es ideal para implementar la instrucción TEST AND SET.**  
verdadero

**244-La convención de llamada Linux/GCC x86-32 considera, respecto a convenios de uso de registros:**

3 registros salva-invocante, 3 registros salva-invocado, y 2 especiales

**245-¿Qué valor contendrá el registro edx tras ejecutar las dos instrucciones siguientes?**

`movl $-1, %edx`

`movb $1, %dl`

11111111 11111111 11111111 00000001

**246-En la realización de la práctica de la bomba digital, una parte del código máquina es el siguiente:**

`0x080486e8 <main+120>: call 0x8048524 <strcmp>`

`0x080486ed <main+125>: test %eax,%eax`

`0x080486ef <main+127>: je 0x80486f6 <main+134>`

`0x080486f1 <main+129>: call 0x8048604 <boom>`

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

`set *(char*)0x080486ef=0xeb`

**247-El direccionamiento inmediato no es más lento que el direccionamiento directo**

verdadero

**248-Supongamos dos CPU con bus de direcciones de ancho idéntico. Si una de ellas emplea E/S independiente y la otra mapeada en memoria, ¿cuál podrá acceder a una mayor cantidad de memoria?**

La CPU con E/S independiente.

**249-Sea un formato de microinstrucción que incluye dos campos independientes de 9 bits cada uno. Si se rediseña de modo que se solapen los dos campos, ¿cuántos bits se ahorran en cada microinstrucción?**

8

**250- ¿Cuál de las siguientes afirmaciones es incorrecta?**

a)El repertorio de instrucciones es el conjunto de operaciones que es capaz de interpretar la unidad de control.

b)El repertorio de instrucciones debe ser capaz de realizar una tarea en un tiempo finito.

c)El modo de direccionamiento permite determinar un operando o la ubicación del operando.

\*\*\*d)Los operandos siempre están almacenados en memoria.

**251- ¿De qué depende el tamaño del contador de programa?**

a) de la longitud del código de operación

b) del ancho del bus de datos

c) el tamaño no importa

\*\*\*d) ninguna de las anteriores es cierta

**252- El número -12 se almacenará en complemento a 2 en el registro %eax como:**

0xFFFFFFFF4

**253- En el controlador de DMA 8237 los registros de dirección de memoria y contador de bytes están duplicados para cada canal para que en el modo de autoinicio el circuito sea capaz de recordar los valores originales de esos registros.**

Verdadero

**254- Un sistema no segmentado tarda 200 ns en procesar una instrucción. Las instrucciones pueden ser procesadas en un cauce segmentado de 20 etapas con un ciclo de reloj de 12 ns. Cuando se procesan muchas instrucciones, la máxima ganancia de velocidad que podría obtenerse se acerca a:**

16,67

**255- El punto de entrada de un programa ensamblador en GNU/as Linux x86 se llama**

\_start

**256- Como parte del proceso de compilación de una aplicación en lenguaje C, enlazar .o → .exe (de objeto proveniente de fuente C a ejecutable) usando sólo as y ld, sin gcc...**

Basta usar ld, con los modificadores de gcc que corresponda, y añadiéndole el runtime de C

**257- En los modos de direccionamiento del tipo Desplazamiento (Base, Índice, Factor Escala), puede usarse como**

desplazamiento, cualquier constante de 1, 2 o 4 bytes (incluso el nombre de una

variable, por su dirección)

**258- Un sistema no segmentado tarda 10 ns en procesar una instrucción. Las instrucciones pueden ser procesadas en un cauce (pipeline) de 5 segmentos con un ciclo de reloj de 4 ns. Cuando se procesan muchas instrucciones, la ganancia máxima de velocidad que se obtiene se acerca a:**

2,5

**259-Indique cuál de las siguientes características no es cierta en el direccionamiento indirecto a memoria a través de memoria:**

La instrucción contiene la dirección de memoria exacta en que se encuentra el objeto.

**260- En la práctica “media” se pide sumar una lista de enteros \*con\* signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el menor valor positivo que repetido en los \*dos\* primeros elementos de la lista causaría overflow con 32 bits al realizar la suma de \*esos dos\* primeros elementos de la lista?**

0x4000 0000

**261- Las interrupciones no se utilizan únicamente para realizar operaciones de E/S.**

Verdadero

**262- Si una pila crece hacia direcciones de memoria decrecientes, una instrucción PUSH utiliza direccionamiento indexado con postautodecremento, siendo SP el registro índice.**

falsa

**263- La técnica de sondeo, escrutinio o "polling"...**

a)Se utiliza para identificar el destino de una interrupción

\*\*\*b)Permite establecer un mecanismo de asignación de prioridades a los distintos dispositivos

c)En caso de utilizarse, es necesario emplear varias líneas para que los dispositivos soliciten una interrupción

d)Todas las respuestas anteriores son falsas

**264-Cuál de las siguientes características es típica de la microprogramación horizontal?**

Ninguna o escasa codificación.



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



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

-1

**266- En una estructura de computador de bus único (bus del sistema): sólo una unidad funcional puede tener el control del bus en cada momento**

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

0xe0 en 0x0804913c y 0x1f en 0x08049140

**268-Un computador con 13 líneas de direcciones utiliza E/S mapeada a memoria. Si se supone que cada uno de los periféricos ocupa 4 direcciones y que el número de periféricos que se planea conectar es de  $2^{10}$ , ¿qué tamaño de memoria admite el computador?**

$2^{12}$  palabras

**269-La segunda generación de computadores se caracteriza tecnológicamente por la utilización del transistor.**

V

**270- Son funciones de la unidad de control:**

el secuenciamiento de las instrucciones máquina

**271- La ecuación básica de rendimiento calcula**

cuánto tiempo tarda en ejecutarse un programa concreto conociendo su número de instrucciones y el número de etapas (promedio) y la frecuencia del procesador

**272- ¿Cuál de las siguientes afirmaciones es cierta?**

a) la unidad de control necesita como entrada el registro contador de programa para saber cuál es la instrucción que debe ejecutar a continuación realmente la UC copia PC en MAR, y lo que lee en MDR (captación) lo lleva a IR que sí es entrada a la UC (decodificación)

b) la arquitectura von Neumann de los computadores tradicionales consiste en tener almacenados los datos separados de las instrucciones en memorias distintas vimos en [T4.2EjmSeg] tr.12 que tener memoria de instrucciones separada de la de datos es una variante llamada "arquitectura Harvard", llamándose "arquitectura Princeton" la original  
[https://en.wikipedia.org/wiki/Von\\_Neumann\\_architecture](https://en.wikipedia.org/wiki/Von_Neumann_architecture)  
[https://en.wikipedia.org/wiki/Modified\\_Harvard\\_architecture](https://en.wikipedia.org/wiki/Modified_Harvard_architecture)



c) el registro de estado (flags) es un registro de propósito específico cuyo contenido puede ser visto directa o indirectamente por el usuario mediante el uso de ciertas instrucciones específicas aunque no hemos visto pushf y popf, por eliminación ésta es la respuesta

d) el registro de direcciones de memoria es un registro de propósito general que puede contener tanto direcciones como datos  
MAR no está disponible al programador, lo usa la UC para indicar la dirección de memoria a leer/escribir

**273- La 5ª generación de computadores comenzó aproximadamente en 1971 con el primer microprocesador.**

**F**

**274-¿Cuántos bits hacen falta como mínimo para implementar tres niveles de inhibición de interrupciones (general, nivel y máscara) en un sistema con cuatro niveles de interrupción?**

**7**

**275-Un circuito controlador de interrupciones permite resolver prioridades entre las fuentes de interrupción conectadas a él.**

**Verdadero**

**276-En un sistema de 32bits, ¿cuál de las siguientes expresiones C es equivalente a la expresión  $(x[2] + 4)[3]$ ?**

**(Asumir que x se ha declarado como `int **x`. Recordar que C usa aritmética de punteros. Notar que muchos de los paréntesis no son necesarios, sólo se han añadido para evitar confusiones por precedencia de operadores)**

**`*((*(x + 2)) + 7)`**

**277-Alguna de las siguientes técnicas NO es de utilidad para determinar la causa de una interrupción**

**•a)**

**Línea de reconocimiento INTA#**

**b)**

**Interrupciones vectorizadas**

**x**

**c)**

**Múltiples líneas de interrupción INT1#, INT2#...**

**f**

**d)**

**Consulta de estado, o polling**



**278-¿A qué tipo de interrupciones corresponde la forma de determinar la dirección de comienzo de una rutina de interrupción en la que se envía una instrucción de bifurcación completa?**

Interrupciones vectorizadas

**279-El programa ensamblador siempre resuelve todas las referencias usadas en el fichero fuente.**

Falso

**280-La instrucción TEST\_AND\_SET comprueba si la posición de memoria especificada como operando vale 0 y a continuación la pone a 1, todo ello sin posibilidad de interrupción.**

Verdadero

**281-En una arquitectura de registros de propósito general (a nivel de lenguaje máquina):**

a)

operar usando registros es más rápido.

b)

la generación de código resulta más simple que en arquitecturas de pila o acumulador.

c)

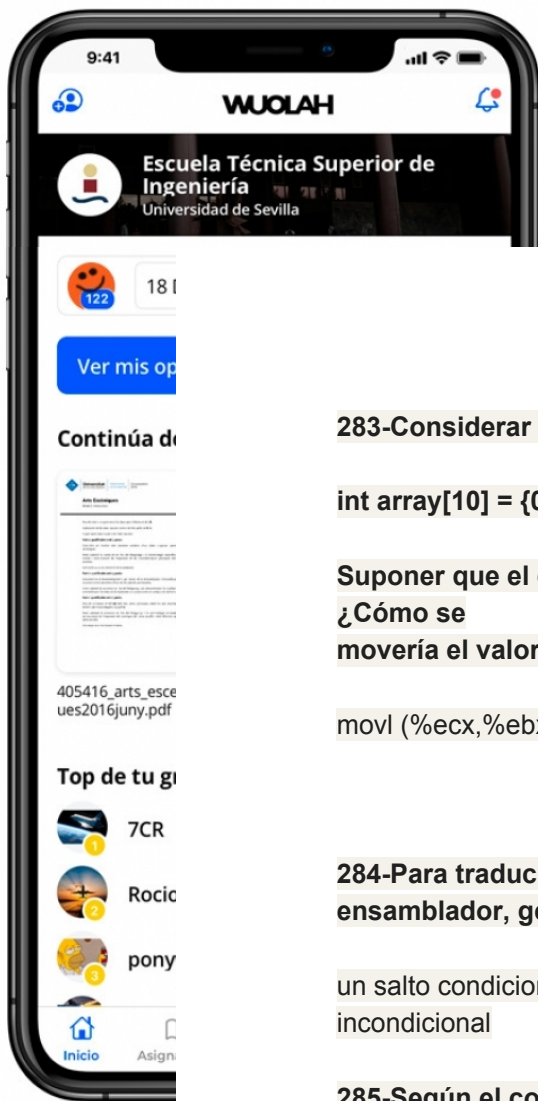
se evita el cuello de botella (por ejemplo, pila, o acumulador) que otras arquitecturas presentan al evaluar expresiones aritméticas complejas

✓•d)

todas las respuestas anteriores son ciertas.

**282-¿Cuál de las siguientes direcciones NO está alineada a double (8-byte)? (Al no poder escribir el 2 como subíndice, aclaramos que ")2" indica binario)**

1110110101110100)2



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



**283-Considerar la declaración C**

```
int array[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

Suponer que el compilador tiene la dirección de array en el registro %ecx.  
¿Cómo se movería el valor array[3] al registro %eax? Asumir que %ebx es 3.

```
movl (%ecx,%ebx,4),%eax
```

**284-Para traducir una construcción if-then-else de lenguaje C a lenguaje ensamblador, gcc utiliza generalmente**

un salto condicional, según la condición opuesta a la del código C, y otro salto incondicional

**285-Según el convenio little-endian, encontraremos el byte más significativo en una dirección menor.**

Falso

**286-Para conectar las salidas de dos registros hacia un bus común en el datapath...**

se pueden usar dos buffers triestado.

**287-Las operaciones elementales de transferencia de datos en E/S por interrupciones deben ser realizadas por programa.**

Verdadero

**288-Cuanto más horizontal es la microprogramación más largas son las microinstrucciones.**

Verdadero

**289-¿Qué técnica de E/S requiere menos atención por parte del procesador?**

E/S mediante acceso directo a memoria

**290-¿Qué técnica de E/S consume menos tiempo del procesador?**

E/S mediante DMA

**291-Una instrucción máquina puede desglosarse en las siguientes operaciones elementales:**

$sp := sp - 1; m[sp] := pc; pc := x$

llamada a subrutina

**292-El 8255 es un interfase paralelo de periféricos que puede utilizarse para E/S programada, pero también puede generar interrupciones.**

Verdadero

**293-En 8086, los parámetros a las subrutinas se pueden pasar:**

a)

a través de variables globales

b)

a través de los registros

c)

a través de la pila

✓•d)

todas las anteriores son ciertas

**294-En arbitraje por "daisy-chain" la señal BUS\_REQUEST es compartida por los maestros potenciales.**

Verdadero

**295-Las instrucciones JB y JNAE del Pentium provocan un salto si...**

CF == 1

**296-Una instrucción de llamada a subrutina es más compleja en general que una de salto.**

Verdadero

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

- a) Es incompatible con la técnica de sondeo o polling
- b) Todos los componentes se conectan directamente y con igual prioridad al procesador o gestor de interrupciones
- ✓ • c) Los componentes se comportan de forma cooperativa: sólo al de mayor prioridad se le concede la interrupción o se apodera del bus de comunicaciones  
lo del "bus de comunicaciones" no se explica en las transparencias actuales - el daisy-chain se puede usar también como método de arbitraje para conceder el bus - las señales suelen llamarse BR/BG (Bus request/grant) en lugar de INTR/INTA (IRQ request/acknowledge)
- d) Los componentes están conectados todos con todos y un gestor centralizado decide la prioridad

**298-¿En qué modo de transferencia el controlador de DMA programable 8237 no tiene en cuenta el registro contador?**

Transferencia por demanda

**299-¿Cuál afirmación es FALSA en arquitecturas x86-64?**

- a) El tamaño de las posiciones de memoria es 64 bits ←Esta es la falsa
- b) El tamaño de un double es 64 bits
- c) El tamaño de los registros es 64 bits
- d) El tamaño de un puntero es 64 bits

**300-En la práctica de la bomba, el tercer ejercicio consistía en usar un editor hexadecimal para crear un ejecutable sin “explosiones”. Para saber qué contenidos del fichero hay que modificar, se puede utilizar... (marcar la opción \*falsa\*)**

hexedit

el enunciado implícitamente indica que no basta con hexedit. Aún así hubo quien la dejó en blanco (pocos) y quien falló (50%)

**301-En los computadores con arquitectura Von Neumann los programas están en memoria principal (o caché) cuando se ejecutan.**

Verdadero

**302-La instrucción seta %al (seta significa "set if above"):**

Pone AL a 1 si CF=0 y ZF=0

**303-¿Cuál de las siguientes afirmaciones acerca del concepto de interrupción no es cierta?**

- a) Es una bifurcación normalmente externa al programa en ejecución
- b) Su objetivo es reclamar la atención del procesador
- c) Solicita que se ejecute un programa específico para tratarla
- d) Ninguna de las anteriores <-Esta

**304-En las instrucciones aritméticas con dos operandos de un procesador con arquitectura de pila, los dos operandos, implícitos, son la cima de la pila y el elemento siguiente de la cima de la pila.**

Verdadero

**305-La principal desventaja de la nanoprogramación es que requiere un programa ensamblador muy complejo.**

Falso



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**306-Respecto a las bombas estudiadas en la práctica "bomba digital", ¿en cuál de los siguientes tipos de bomba sería más difícil descubrir la contraseña? Se distingue entre enteros definidos en el código fuente de la bomba, y enteros solicitados al usuario mediante scanf(). Por "procesar" se entiende calcular el n-ésimo elemento de la serie de Fibonacci.**

1 entero del usuario se procesa, y se compara con el entero del fuente

En los otros dos casos usando el debugger se puede ver el resultado literal que hay que introducir como contraseña, parando justo antes de comparar

**307-¿Cuál de las siguientes afirmaciones acerca del concepto de interrupción es falsa?**

- a) Ninguna de las afirmaciones es falsa **<-ESTA**
- b) Solicita que se ejecute un programa específico para tratarla
- c) Su objetivo es reclamar la atención del procesador
- d) Es una bifurcación normalmente externa al programa en ejecución

**308-Respecto a la convención de llamada usada en Linux/gcc:**

Hay registros que pueden ser modificados libremente por las subrutinas, y otros que, si se modifican, se deben restaurar posteriormente. Y también hay registros especiales

**309-¿Cuál de las siguientes afirmaciones es cierta?**

- a) Ninguna de las otras respuestas es cierta
- b) Los estándares ATA/IDE, SCSI y Firewire (IEEE 1394) definen buses de funcionamiento paralelo a diferencia del estándar USB que define un bus de funcionamiento serie
- c) El bus AGP se utiliza para conectar tarjetas gráficas y controladoras de disco
- d) Los buses PC XT, AT/ISA, MCA, EISA, VLB, PCI y AGP son buses de placa madre de PC **<-ESTA**

**310-Un 8086 necesita al menos 1 KB de memoria RAM para funcionar correctamente.**

Falso

**311-¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC es cierta?**

- a) Ninguna de las otras respuestas es cierta **<-ESTA**
- b) No todas las interrupciones se pueden generar por software
- c) Cada vector de interrupción es una palabra de 16 bits
- d) Existen 1024 vectores de interrupción

**312-Respecto a la diferencia entre dispositivos activos y pasivos en un bus podemos decir:**

a) Los dispositivos pasivos siempre son esclavos

b) Los dispositivos activos pueden actuar como maestros



•

c) Las respuestas a y b son ciertas

d) Las respuestas a y b son falsas

**313-Respecto a registros de propósito general (RPG), el 80386 tiene:**

8 registros de 32 bits

**314-Suponga una memoria de microprograma de n microinstrucciones de w bits cada una. Suponiendo que de esas n microinstrucciones, hay  $2^m$  distintas, el ahorro en bits si se utiliza nanoprogramación es  $n \cdot w - (n \cdot m + 2^m \cdot w)$ .**

V

**315- Sobre las técnicas de transferencia en operaciones de E/S:**

a) Pueden ser controladas por programa o por hardware

b) Si se emplea E/S programada puede hacerse con consulta de estado o sin consulta de estado

c) En el caso de utilizar E/S mediante DMA hace falta emplear un controlador de DMA



•

d) Todas las respuestas anteriores son ciertas

**316-Un 8086 puede funcionar correctamente con 1 MB de memoria ROM.**

V

**317-En IA32, tras dividir 0x640000 (64 bits) entre 0x8000 (32 bits), el resultado será:**

0xC8 en EAX y 0x0 en EDX

**318-La práctica “parity” debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity5:**

```
int parity5(unsigned* array, int len){
 int i,j,res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 for (j=sizeof(unsigned)*4;
 j>0; j=j/2){
 x^=x>>j;
 }
 x = x & 0x1;
 res+=x;
 }
 return res;
}
```

**Esta función parity5:**

produce siempre el resultado correcto

solemos escribir `res+=x&0x1`, en lugar de ponerlo en 2 sentencias C

**319-El registro MBR...**



contiene el valor que va a ser almacenado en la memoria, o bien se usa para recibir un valor procedente de la memoria

**320-Un sistema no segmentado tarda 20 ns en procesar una instrucción. Las instrucciones pueden ser procesadas en un cauce (pipeline) de 4 segmentos con un ciclo de reloj de 5 ns. Cuando se procesan muchas instrucciones, la ganancia máxima de velocidad que se obtiene se acerca a:**

4

**321-Si queremos almacenar la palabra de 16 bits 8965h en memoria según "little-endian", quedará almacenada a partir de la posición 8600h como:**  
en el byte 8600h se guarda 65h y en el byte 8601h se guarda 89h

**322-En las últimas generaciones de computadores la mejora de prestaciones viene dada por:**

avances en tecnología y avances en la estructura y arquitectura del computador.

**323-Los arrays bidimensionales en lenguaje C se almacenan en orden...**  
mayor-de-fila" (row-major)

**324-La secuencialidad es la forma de direccionamiento implícita en el flujo de ejecución de un programa.**

V

**325-Un controlador de DMA suele ser programado con la siguiente información relativa a una operación de E/S:**

tipo de operación, tamaño de bloque a transferir, dirección inicial de memoria

**326-Una implementación diferencial del bus SCSI permite mayor distancia o velocidad que una "single-ended".**

V

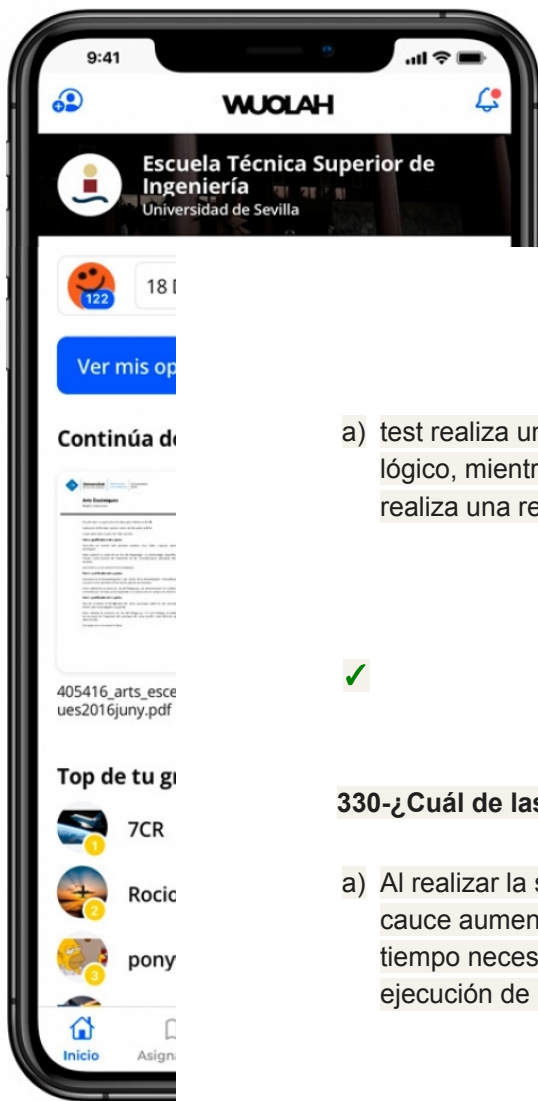
**327-Más de la cuarta parte de las instrucciones máquina que se ejecutan en un programa típico son de movimiento o transferencia de datos.**

V

**328-A qué tipo de interrupciones pertenecen las condiciones de tiempo real y los fallos hardware?**

No enmascarables

**329-La diferencia entre las instrucciones test y cmp consiste en que**



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



a) test realiza una operación and lógico, mientras que cmp realiza una resta



b) test modifica sólo los flags lógicos (ZF,SF) mientras que cmp modifica los aritmético-lógicos (ZF,SF,CF,OF)

c) ambas respuestas son correctas

d) ambas respuestas son incorrectas

## 330-¿Cuál de las siguientes afirmaciones es cierta?

a) Al realizar la segmentación de cauce aumenta en general el tiempo necesario para la ejecución de un programa



b) Debido a que pueden existir dependencia de datos, los resultados de un programa pueden ser diferentes a si el programa se ejecutara sin segmentación

c) La segmentación de cauce disminuye el número de instrucciones necesarias para la ejecución de un programa

d) Ninguna de las afirmaciones anteriores

331-Una instrucción de salto si menor, para números positivos sin signo, tiene que comprobar el valor de:  
el bit de acarreo

332-¿Cuál es el contenido de la pila al terminar de ejecutarse la siguiente secuencia de instrucciones de una arquitectura de pila?:

push #4

push #7

add

push #10

sub

1

333-Para traducir una construcción if-then-else de lenguaje C a lenguaje ensamblador, gcc utiliza generalmente

un salto condicional, según la condición opuesta a la del código C, y otro salto incondicional

### **334-No en todas las instrucciones máquina**

hay una fase de captura de operandos

### **335-El programa RISC**

**ld r4,(r2)**

**ld r5,(r3)**

**add r6,r4,r5**

**st (r1),r6**

Almacena la suma de los contenidos de las posiciones de memoria direccionadas por los registros r2 y r3, y almacena el resultado en la posición de memoria direccionada por el registro r1.

### **336-¿Cuál de las siguientes características es posterior a la segunda generación de computadores?**

RISC

### **337-¿En qué pareja de registros están el dato/instrucción que se leerá o escribirá en memoria, y la dirección de memoria?**

MBR y MAR

### **338-En un procesador con segmentación de cauce, aumentar el número de etapas (p.ej. de 2 a 4, o de 4 a 8), tiene en general como consecuencia:**

Un incremento de las prestaciones

### **339-¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC es cierta?**

Todas las interrupciones se pueden generar por software

### **340-¿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)

CL, DX, R8d, R9

c)  
RBX, RSI, RDI

☒

d)  
RSP, RBP

**341-Con tres controladores de interrupciones 8259 se pueden manejar exactamente:**

a)  
8 niveles de prioridad

b)  
16 niveles de prioridad

☒

c)  
24 niveles de prioridad

•d)  
Ninguna de las anteriores es cierta

**342-La instrucción SKIP pone a 0 un registro o posición de memoria si se cumple una condición.**

Falso

**343-Si AX = FA50h y ejecutamos AND AX, 00FFh**

El registro AH se pone a 0

**344-El cuerpo del siguiente código C: unsigned copy(unsigned u) {return u;} puede traducirse a ensamblador como:**

`movl 8(%ebp), %eax`

**345-¿Qué modificador (switch) de as hace falta para ensamblar una aplicación de 32bits en un sistema de 64bits en el que se ha instalado también el compilador de 32bits?**

-32

**346-En el direccionamiento inmediato, tras captarse completamente la instrucción:**

se accede al operando, que es una constante contenida en la propia instrucción.

**347-¿Cuál de los siguientes contenidos no está incluido en un fichero ELF ejecutable?**

a)

código máquina  
sección .text

b)

tabla de símbolos  
incluso símbolos de depuración con -g

c)

variables globales  
secciones .data/.bss

✓•d)

pila del usuario  
por exclusión... y por lógica, para qué almacenarla si está vacía

**348-La precaptación (cola de instrucciones) está relacionada con...**

Los riesgos estructurales (intenta evitar el efecto de un fallo de cache)

**349-¿Cuántos niveles de interrupción podremos gestionar si disponemos de 7 controladores de interrupciones programables 8259?**

50



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**350-No hay diferencia entre un desplazamiento lógico a la derecha y un desplazamiento aritmético a la derecha.**

Falso

**351-Si se almacena el número de 16 bits 1234h en la palabra (16 bits) 0 de memoria en una arquitectura "Big Endian", el byte 0 contendrá 12h y el byte 1 contendrá 34h.**

Verdadero

**352-En las arquitecturas RISC:**

se usan instrucciones muy simples que se pueden segmentar.

**353-¿Cuál de las siguientes parejas de mnemotécnicos de ensamblador IA32 corresponden a la misma instrucción máquina?**

JZ (saltar si cero), JE (saltar si igual)

**354-El primer parámetro de printf:**

es un puntero

**355-En un 8086, la ejecución de una instrucción LODSB seguida inmediatamente de una STOSB nunca modifica el contenido de ninguna posición de memoria**

Falso

**356-En la arquitectura Von Neumann...**

el programa se encuentra residente en memoria.

**357-En una microinstrucción que hace uso del registro de control residual, el contenido de éste normalmente se interpreta como señales de control.**

V

**358-Para direccionar una memoria de bytes en la que quepan 1G palabras de 32 bits se necesitarán:**

32 bits

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

**360-La instrucción INC [3] no tiene sentido.**

F

**361-Cuando una CPU dispone de muchas líneas de interrupción con un dispositivo en cada una, será necesario utilizar "polling" para detectar la fuente de la interrupción.**

F

**362-¿Cuál de las siguientes afirmaciones sobre el direccionamiento absoluto es falsa?**

El tamaño del operando direccionado queda limitado por el nº de bits del campo dirección de memoria.

**363-Un programa con muchas bifurcaciones hace que no se aprovechen al máximo las prestaciones del pipeline.**

V

**364-Si el sistema operativo lo permite, la tabla de vectores de interrupción puede ser modificada por el usuario.**

V

**365-Si el registro EAX contiene X, la sentencia en C**

**x &= 0x1;**

**se traducirá a ensamblador como:**

**andl \$1, %eax**

**366-Al diseñar el formato de instrucción:**

se suele omitir el campo que indica la siguiente instrucción (la siguiente a ejecutar es la siguiente en memoria, salvo en caso de salto).

**367-Se llama "Broadcall" a la escritura simultánea en varios esclavos.**

F

**368-La transmisión isócrona garantiza un ancho de banda fijo sin que se produzcan interrupciones en el flujo de datos.**

V

**369-Sobre la E/S mapeada en memoria podemos decir que:**

Usa direcciones de memoria para acceder a puertos de E/S

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

long int pasa de 4 B a 8 B

**371-En los modos de direccionamiento del tipo**

**Desplazamiento(Base,Indice,Factor Escala), puede usarse como**

desplazamiento, cualquier constante de 1, 2 o 4 bytes (incluso el nombre de una variable, por su dirección)

**372-Una computadora puede funcionar prescindiendo de:**

un acumulador

**373-Los compiladores pueden generar código más fácilmente, y también código más eficiente, para conjuntos de instrucciones tipo RISC.**

V

**374-Alguna de las siguientes afirmaciones sobre sistemas Linux x86-64 no es cierta**

Todos los argumentos de función se pasan a través de la pila

**375-Los microprocesadores RISC puede ser implementados en un tiempo más corto, y requieren menos área de silicio que los CISC.**

V

**376-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"
```

```
movl $lista, %ebx
```

**377-La instrucción xor \$3, %eax tiene como resultado:**

Cambiar 0<->1 (complemento a 1 de) los últimos 2 bits del registro EAX

**378- El direccionamiento indirecto a través de registro es más rápido que el indirecto a través de memoria.**

V



**379-** Una de las diferencias entre el 8086 y el 8088 es que este último no dispone de BIU ("Bus Interface Unit").

F

**380-** Las arquitecturas memoria-memoria tienen la ventaja de emplear instrucciones máquina muy cortas y así los accesos a memoria son mínimos.

F

**381-** Todas las instrucciones de salto/bifurcación realizan una operación del tipo  $PC = X$ , si bien X puede ser un dato inmediato, el contenido de un registro, o el contenido de una posición de memoria.

F

**382-** Un procesador emplea codificación en bloque del código de operación. Existen 123 instrucciones que tienen una longitud de 16 bits. ¿A cuántas direcciones de memoria pueden acceder como máximo si todas emplean una estructura "código de operación + dirección de memoria"?

512

**383-** Si una pila crece hacia direcciones de memoria decrecientes, una instrucción PUSH utiliza direccionamiento indexado con postautodecremento, siendo SP el registro índice.

F

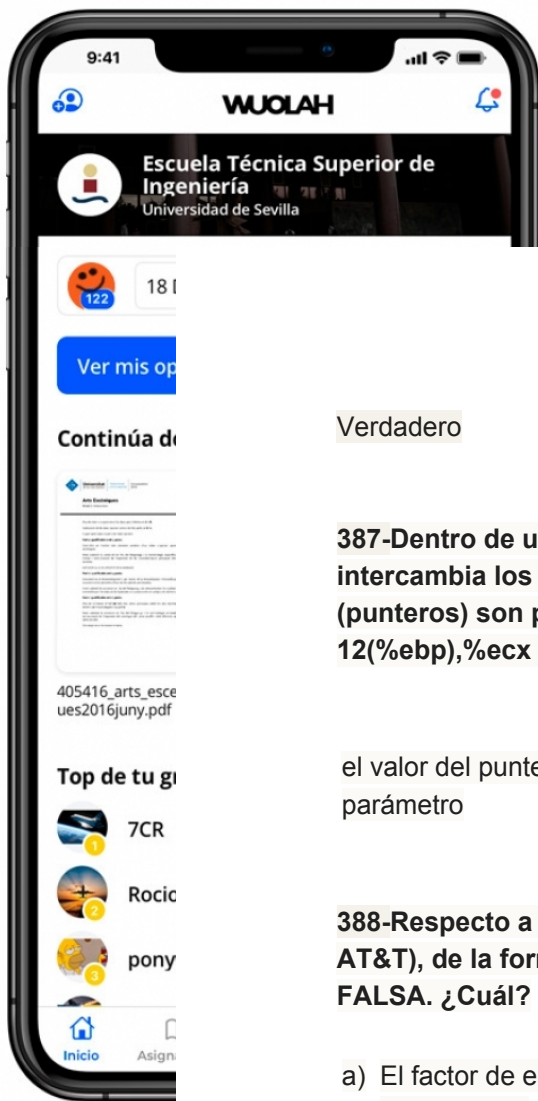
**384-** ¿Cuántos controladores de interrupciones 8259 hacen falta como mínimo para manejar 25 líneas de interrupción?

✓ • d) 4

**385-** Una instrucción de "salto si igual" tiene que comprobar el valor de:

- c) el bit de  
cero

**386-** Un diseño CISC pretende disminuir el número de instrucciones a ejecutar por un programa, mientras que uno RISC pretende disminuir el número medio de ciclos por instrucción.



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Verdadero

**387-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...**

el valor del puntero pasado como segundo parámetro

**388-Respecto a direccionamiento a memoria en ensamblador IA32 (sintaxis AT&T), de la forma D(Rb, Ri, S), sólo una de las siguientes afirmaciones es FALSA. ¿Cuál?**

- a) El factor de escala S puede ser 1, 2, 4, 8
- b) ESP no se puede usar como registro índice
- c) EBP no se puede usar como registro base
- d) El desplazamiento D puede ser una constante literal (1, 2 ó 4 bytes)

**389-Para obtener una única velocidad comparativa final, el benchmark SPEC CPU combina las velocidades de ejecución de una serie de tests, respecto a un ordenador de referencia, usando la media...**

- c) geométrica

**390-Por x86-64 se entiende la misma arquitectura de repertorio (ISA) que AMD64**

**391-Las rutinas de servicio de interrupción son microprogramas que pueden ser escritos por el usuario.**

Falso

**392-En un PC, un dispositivo que pida interrupciones a través del controlador de interrupciones 8259 no podrá interrumpir por segunda vez a la CPU hasta que no mandemos el comando EOI (20h) al 8259 (puerto 20h).**

VERDADERO

**393-En las instrucciones aritméticas con dos operandos de un procesador con arquitectura de pila, los dos operandos...**

son la cima de la pila y el elemento siguiente de la cima de la pila.

**394-Si queremos almacenar la palabra de 16 bits 8965h en memoria según "big-endian", quedará almacenada a partir de la posición 1000h como en el byte 1000h se guarda 89h y en el 1001h 65h**

**395-¿Qué novedad se desarrolló en la tercera generación de computadores?**

Los circuitos integrados

**396-En el direccionamiento inmediato, la dirección efectiva se encuentra en un registro general del procesador**

Falso

**397-¿Cuál de las siguientes afirmaciones sobre la segmentación de cauce es cierta?**

En general, un operación segmentada ("pipelined") requiere el mismo tiempo o más, desde el principio hasta el fin, que la misma operación en una implementación no segmentada

**398-¿Cuál de las siguientes afirmaciones es correcta?**

- ✓ • a) Las instrucciones en lenguaje máquina se almacenan y tratan como cadenas de unos y ceros.
- b) Las instrucciones en lenguaje ensamblador se almacenan y tratan como cadenas de unos y ceros.
- c) El lenguaje máquina es igual para todos los computadores.
- d) El lenguaje ensamblador es igual para todos los computadores.

**399-Cuando se usan interrupciones vectorizadas, el periférico puede suministrar al procesador un índice referente a una tabla de vectores de interrupción.**

Verdadero

**400-Los modos de direccionamiento de una instrucción se especifican en ensamblador mediante las directivas**

Falso

**401-El programador de lenguaje ensamblador necesita conocer:**

la arquitectura del ordenador.

**402-¿Qué forma de realizar acceso directo a memoria es más rápida?**

Transferencia de bloques o parada de CPU

**403-En un sistema IA32 Linux, ¿cuál es el tamaño de un long?**

4 bytes

**404-¿Qué hace gcc -S?**

Compilar .c → .s (fuente C a fuente ASM)

**405-Un sistema no segmentado tarda 20 ns en procesar una instrucción. Las instrucciones pueden ser procesadas en un cauce (pipeline) de 4 segmentos con un ciclo de reloj de 6 ns. Cuando se procesan muchas instrucciones, la ganancia máxima de velocidad que se obtiene se aproxima a:**

3,33

**406-Las unidades de control microprogramadas utilizan habitualmente memorias RAM para almacenar los microprogramas.**

Falso

**407-Justo antes de que una instrucción máquina escriba un resultado en memoria:**

en MBR está el resultado y en MAR la dirección donde se almacenará

**408-En la práctica "media" se suma una lista de 32 enteros de 4 B con signo para producir una media y un resto usando la instrucción IDIV. ¿Cuál de las siguientes afirmaciones es falsa?**

a) El resto siempre tiene el mismo signo que la suma  
división truncada IDIV - resto del mismo signo que dividendo  
En matemáticas se usa división euclídea - resto modular  $0 \leq r < \text{divisor}$   
[https://es.wikipedia.org/wiki/Divisi%C3%B3n\\_eucl%C3%ADdea](https://es.wikipedia.org/wiki/Divisi%C3%B3n_eucl%C3%ADdea)  
a

•

b) La media se redondea al entero más próximo división truncada IDIV - no se redondea, se trunca



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



x

c) IDIV  
produce el  
mismo  
resto que el  
operador %  
en lenguaje

C  
división  
truncada  
ambos -  
resto del  
mismo  
signo que  
dividendo

d) IDIV  
produce el  
mismo  
cociente  
que el  
operador /  
en lenguaje

C  
división  
truncada  
ambos -  
resto del  
mismo  
signo que  
dividendo

**409-¿Qué modificador (switch) de ld hace falta para enlazar una aplicación de 32bits en un sistema de 64bits en el que se ha instalado también el compilador de 32bits?**

-m elf\_i386

**410-¿Cuál de las siguientes afirmaciones es correcta?**

Las instrucciones en lenguaje máquina se almacenan y tratan como cadenas de unos y ceros.

**411-Un sistema no segmentado tarda 50 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce ("pipeline") de 5 segmentos con un ciclo de reloj de 10 ns. Cuando se procesan muchas tareas, la ganancia de velocidad que se obtiene se acerca a 5.**

Verdadero

**412-Si queremos almacenar la palabra de 16 bits 0x8965 en una memoria de bytes según "little-endian", quedará almacenada a partir de la posición 0x8600 como**

**M[0x8600]=0x65 y M[0x8601]=0x89**

**413-Considerar las siguientes declaraciones de estructuras en una máquina Linux de 64-bit.**

```
struct RECORD {
 long value2;
 int ref_count;
 char tag[4];
};
```

```
struct NODE {
 double value;
 struct RECORD record;
 char string[8];
};
```

**También se declara una variable global "my\_node" como sigue:**  
**struct NODE my\_node;**

**Si la dirección de my\_node es 0x601040, ¿cuál es el valor de &my\_node.record.tag[1]?**

**0x601055**

**414-¿Qué hace gcc -O?**

**Compilar con optimización, igual que -O1**

**415-En una bomba como las estudiadas en prácticas, del tipo...**

**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>**

**0x080486f6 <main+134>: ...**

**la contraseña es...**

a) el string almacenado a partir de  
0x8048524

b) el string almacenado a partir de  
0x80486f6

c) el string almacenado a partir de  
0x8048604

d) ninguna de las anteriores

**416-Las técnicas principales de E/S son (señalar la respuesta falsa)**

a) IRQ (por  
interrupciones)



•

b) E/S cableada  
(hardwired)

c) E/S programada

d) DMA (por acceso  
directo)

**417-Una forma usual de realizar el arbitraje distribuido consiste en una competición por la concesión del bus realizada por medio del envío por cada maestro peticionario de un número de arbitraje que lo identifica, de manera que un solo número "gane", y se le conceda el bus al maestro con ese número ganador.**

**VERDADERO**

**418-Si AX = FA50h y ejecutamos XOR AX, 00FFh**

**Se realiza el complemento a 1 de AL**



**419-La instrucción "desplazar a la derecha aritméticamente" suele ser idéntica a la "desplazar a la derecha lógicamente".**

FALSO

**420-Suele existir una instrucción en el repertorio de la mayoría de los microprocesadores para inhibir la petición HOLD del controlador de DMA.**

FALSO

**421-Respecto a los conceptos de procesamiento segmentado y superescalar, una de las siguientes afirmaciones es falsa**

En cualquier procesador resulta ventajoso usar una cola de instrucciones, pero es más importante para uno segmentado (fundamental) que para uno superescalar (conveniente)

**423-La CPU no guarda automáticamente el contexto del programa que ejecuta cuando le llega y atiende una petición de DMA.**

Verdadero

**424-La diferencia entre temporización de bus semisíncrona y asíncrona es que en la semisíncrona las transferencias ocurren en algún múltiplo de ciclo de reloj, y en la asíncrona no existe reloj del bus**

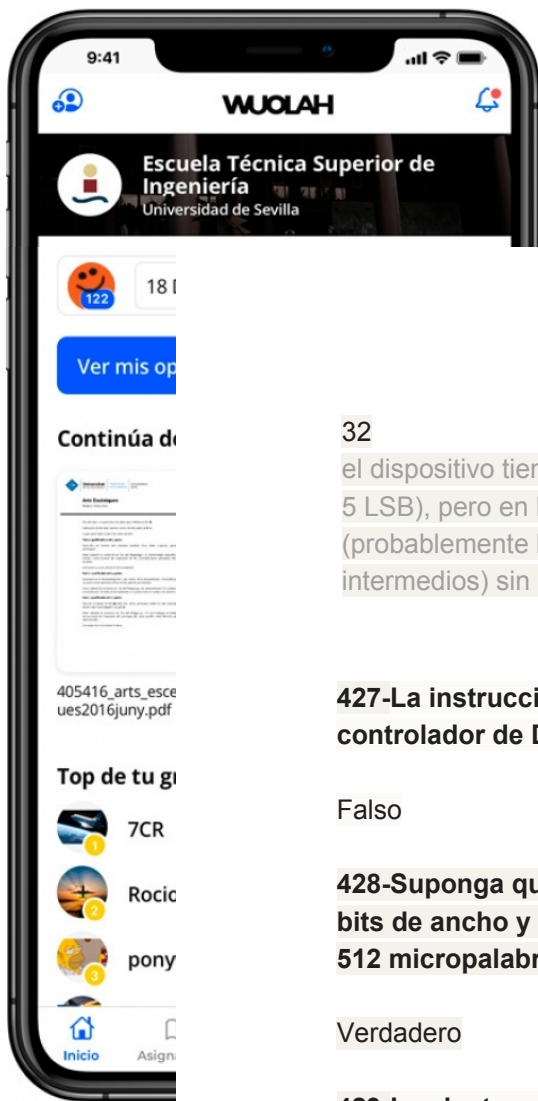
VERDADERO

**425-Una unidad de control microprogramada con secuenciamiento explícito con dos direcciones por microinstrucción, tiene una memoria de control con 35 bits de longitud de palabra. Si las microinstrucciones emplean 15 bits en total para los campos de control y de tipo y condición de salto, el número máximo de palabras de la memoria de control de esta unidad de control microprogramada es de:**

$2^{10}$

**426-Si en un bus de direcciones de 32 bits se decodifica parcialmente la dirección de un dispositivo de 32 posiciones usando 22 bits, ¿cuántas veces aparecerá repetido en el mapa de memoria?**

32



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



32

el dispositivo tiene 32 puertos, usa 5 bits de direccionamiento (probablemente los 5 LSB), pero en lugar de decodificarse con 27, se usan sólo 22 bits (probablemente los 22 MSB) de manera que quedan 5 bits (probablemente intermedios) sin usar, que pueden tomar 32 combinaciones posibles

**427-La instrucción DISABLE (CLI en el 8086) inhibe la petición HOLD del controlador de DMA.**

Falso

**428-Suponga que la micropalabra de una máquina microprogramada tiene 8 bits de ancho y se usan 256 micropalabras diferentes en un microprograma de 512 micropalabras. No se ahorran bits usando nanoprogramación.**

Verdadero

**429-Las instrucciones máquina más usadas según el análisis dinámico son las de bifurcación.**

Falso

**430-El ajuste de marco de pila que gcc (Linux/IA-32) prepara para todas las funciones consiste en las instrucciones**

```
pushl %ebp
movl %esp,
%ebp
```

**431-Algunas de las ventajas de la E/S mapeada en memoria frente a la E/S aislada o independiente son:**

El diseño de la CPU es más sencillo.

**432-En la práctica “media” se pide sumar una lista de 32 enteros \*con\* signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el mayor valor negativo (menor en valor absoluto) que repetido en toda la lista causaría overflow con 32bits?**

0xfbff

ffff

**433-La transferencia de datos en un computador y los dispositivos de E/S puede manejarse de diversos modos. Uno de los siguientes es falso; indíquelo:**

Manejo de todas las líneas del bus de control, paralizando la CPU

**434-Las señales de carga/incremento/desplazamiento de registros, las de selección de entradas de multiplexores, y las de selección de función de la ALU son entradas a la unidad de control**

FALSO

**435-¿Cuál es el contenido de la pila al terminar de ejecutarse la siguiente secuencia de instrucciones de una arquitectura de pila: push #4; push #7; add; push #10; sub?**

1

**436-Un controlador de DMA de un sistema de que emplee buses separados avanzados suele ser programado con la siguiente información relativa a una operación de E/S**

tipo de operación, tamaño de bloque a transferir, dirección inicial de memoria

**437-¿Por qué se impusieron las arquitecturas de registros de propósito general a las arquitecturas basadas en pila?**

Porque las basadas en registros son capaces de lograr un mejor rendimiento cuando se asignan variables a registros

**438-¿Cuál de los siguientes microprocesadores no es de 64 bits?**

Pentium III

**439-¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC (modo real) es cierta?**

a) Existen

1024

vectores de  
interrupción



b) Cada vector de interrupción es una doble palabra de 32 bits formada en primer lugar (dirección menor) por el segmento y seguida por el desplazamiento (dirección mayor) de cada rutina de servicio de interrupción

c) No todas las interrupciones se pueden generar por software

d) Ninguna de las anteriores afirmaciones es cierta

**440-Un procesador emplea codificación en bloque del código de operación. Existen 120 instrucciones que tienen una longitud de 16 bits. ¿A cuántas direcciones de memoria pueden acceder como máximo si todas emplean una estructura "código de operación + dirección de memoria"?**

512

**441-El direccionamiento a registro es similar al directo. La diferencia es que el campo de direcciones referencia un registro en lugar de una dirección de memoria**

VERDADERO

**442-La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:**

```
int popcount3(unsigned* array,
 int len){
 int i, res = 0;
 unsigned x;
 for(i=0; i<len; i++) {
 x = array[i];
 asm("ini3: \n"
 "shr %[x] \n"
 "adc $0, %[r] \n"
 "add $0, %[x] \n")
```

```

 "jne ini3 \n"
 : [r] "+r" (res)
 : [x] "r" (x));
}
return res;
}

```

Esta función produce siempre el resultado correcto, a pesar de que una instrucción máquina en la sección asm() es distinta a la que se esperaba después de haber estudiado pcount\_r en teoría. La instrucción distinta también se podría haber cambiado por

```
test %[x], %[x]
```

**443-¿Cuál es el contenido de una pila al terminar de ejecutarse la siguiente secuencia de operaciones push y pop:**

```

push #1
push #2
push #3
pop a
push #4
pop a
pop a

```

1

**444-En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x0400 0000, ¿en qué se diferencian los resultados de ambos programas?**

no se diferencian

**445-El campo tipo de secuenciamiento indica al generador de direcciones de una unidad de control microprogramada el mecanismo de cómputo de la EA del operando de la instrucción.**

FALSO

**446-La conexión de un 8086 a un sistema de memoria y E/S requiere algún circuito externo más en modo máximo que en modo mínimo.**

Verdadero



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**447-Para desplazar %eax a la derecha un número variable de posiciones  $\leq 32$ , indicado en %ebx, se puede hacer**

`mov %ebx, %ecx`

`sar %cl, %eax`

ver problema Hallaron 3.8 y sección  
3.5.3

**448-Un microprocesador es superencauzado ("superpipelined") si permite la emisión de dos o más instrucciones en un mismo ciclo de reloj**  
FALSO

**449-Un procesador con una unidad de control microprogramada tiene una memoria de control de 300 palabras de 100 bits, de las que 200 son diferentes. Si se rediseñara como unidad de control nanoprogramada, ¿qué tamaño ocuparía la nanomemoria que contiene las microinstrucciones completas sin repeticiones?**

20000 bits

200 uinstr. x 100 bits

**450-Una máquina que almacene el número 4321h a partir de la dirección 0 como 21h, 43h utiliza el sistema little-endian.**  
VERDADERO

**451-En las arquitecturas de registros de propósito general suele ser menor el tráfico entre procesador y memoria que en las de acumulador.**

Verdadero

**452-En la codificación por extensión de campo, el campo de código de operación aumenta su tamaño a medida que disminuye el número o la longitud de campos de dirección de operandos.**  
VERDADERO

**453-**

**Si un procesador no segmentado necesita 5 ns para leer una instrucción de memoria, 2 ns para decodificar la instrucción, 3 ns para leer del banco de registros, 3 ns para realizar el cálculo requerido por la instrucción, y 2 ns para**

**escribir el resultado en el banco de registros, ¿cuál es la frecuencia de reloj máxima del procesador?**

66,67 MHz

**454-El especificador de operando de una instrucción, cuando existe, es siempre una dirección de memoria o de entrada/salida.**

F

**455-¿Cuál de las siguientes instrucciones de IA32 (en sintaxis Intel) no es una instrucción de transferencia?**

cmp eax,15h

**456-El análisis dinámico de la frecuencia de utilización de instrucciones se realiza mediante el estudio de un gran número de listados de programas.**

F

**457-¿Cuál de las siguientes instrucciones convierte  $\%eax = 5 * \%eax$ ?**

1) mov (%eax, %eax, 4), %eax

2) lea (%eax, %eax, 4), %eax

Sólo 2

**458-La codificación Huffman es la más utilizada debido a que consigue resultados óptimos y su proceso de decodificación es sencillo.**

F

**459-Un computador que almacene el número 2143h como 43h en el byte de la dirección 0 y 21h en el byte de la dirección 1, utiliza el sistema big-endian.**

F

**460-¿Cuál de las siguientes afirmaciones es incorrecta?**

a) En lenguajes de alto nivel no se tiene acceso a detalles del procesador como las tablas de páginas, el paso de modo usuario a modo supervisor, el bit de paridad, etc.

b) El lenguaje de alto nivel es más portable que el lenguaje máquina.



- c) En lenguaje ensamblador cada instrucción se corresponde con una instrucción máquina.
- d) En lenguaje ensamblador las instrucciones se escriben en binario.

**461-El direccionamiento relativo a contador de programa favorece la implementación de código reubicable.**

V

**462-Un computador usa el formato vertical de codificación de instrucciones para parte de las señales de control y el formato horizontal para k señales de control. El formato vertical posee n campos codificados de m bits cada uno. ¿Cuál es el máximo número de señales de control que pueden usarse en este computador?**

$k + n \cdot (2^m - 1)$

**463-¿De qué depende el tamaño del contador de programa?**

Del número de direcciones de memoria.

**464-En X86-64, el registro contador de programa se denomina:**

RIP

**465-En una CPU de 32 bits con memoria de bytes, el problema es que...**

Hay que respetar el ordenamiento de bytes y reglas de alineamiento con que se diseñó la CPU

**466-En el direccionamiento inmediato:**

el tamaño (rango de valores) de la constante está limitado.



**467-Un procesador emplea codificación en bloque del código de operación. Existen 130 instrucciones que tienen una longitud de 16 bits. ¿A cuántas direcciones de memoria pueden acceder como máximo si todas emplean una estructura "código de operación + dirección de memoria"?**

256

**468-Se puede programar un controlador de interrupciones 8259 de manera que atienda equitativamente a 8 dispositivos de igual prioridad (cada vez que se atiende a un dispositivo, éste pasa automáticamente a tener la prioridad más baja)**

VERDADERO

**469-¿Cuál es el resultado de evaluar la expresión  $0b1110 \wedge 0b1010$  en lenguaje C?**

0b0100 (hace el AND y pone 1 cuando no es AND, 1 y 1 = 0, 0 y 1 = 1)

**470-La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:**

```
int popcount3(unsigned* array,
 int len){
 int i, res = 0;
 unsigned x;
 for(i = 0; i < len; i++) {
 x = array[i];
 asm("ini3: \n"
 "shr %[x] \n"
 "adc $0, %[r] \n"
 "add $0, %[x] \n"
 "jne ini3 \n"
 : [r] "+r" (res)
 : [x] "r" (x));
 }
 return res;
}
```

**Esta función sólo tiene una diferencia con la versión "oficial" recomendada en clase. En concreto, una instrucción máquina en la sección asm() es distinta.**

**Esta función popcount3:**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



produce siempre el resultado correcto

Si sumar  $x+0$  activa ZF sólo puede ser porque ya era  $x==0$ , así que la lógica es equivalente a la deseada.

**471-En E/S por DMA, las operaciones elementales de transferencia de datos deben ser realizadas por medio de la ejecución de instrucciones del procesador**

FALSO

**472-Si A y B son dos enteros almacenados respectivamente en %eax y %ebx, ¿cuál de las siguientes implementaciones de if (!A && !B) {...then part...} es incorrecta?**

```
test %ebx, %eax
jne not_true
...then part...
not_true:
...
```

test hace AND, no OR

equivale a if (A&&B) {... then\_part ...}

**473-Si el contenido de r4 es 0x13000, ¿a qué dirección se hace referencia con la instrucción LD -0x80(r4)?**

(Nota: 0x indica notación hexadecimal en C y C++)

0x12F80

**474-¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar)**

a)  
movw %dx, (%eax)

b)  
movzbl %dl, %eax

c)

movswl (%eax), %edx

•d)

pushb \$0xFF

**475-El "handshaking" se utiliza en transferencias de datos "síncronas" entre CPU y periféricos.**

Falso

**476-El primer microprocesador de 32 bits de la familia x86 fue 80386**

**477-Si el acceso directo a memoria se realiza mediante robo de ciclo:**  
es posible que la ejecución de una instrucción máquina sea temporalmente detenida

**478-Un cauce ("pipeline") de instrucciones inicialmente vacío y con 3 etapas tardará siempre 5 ciclos de reloj en ejecutar 3 instrucciones si cada una de ellas utiliza las 3 etapas.**

Falso

**479-Para direccionar una memoria de 2 G palabras de 32 bits cada una, que se direcciona byte a byte, se necesitarán:**  
33 bits como mínimo

**480-En los RISC la sección de control del procesador debe ser microprogramada en lugar de cableada debido al uso de un pequeño número de operaciones simples.**

**481-¿En qué generación, dentro de la historia de los computadores digitales, aparecen los sistemas operativos multiusuario?**  
tercera

**482-¿Qué dice la ley de Moore?**

Que el número de transistores de un chip se duplica cada 18 meses.

**483-En el acceso directo a memoria la CPU pone en el bus de direcciones del sistema las direcciones de memoria correspondientes a cada dato que se transfiere por DMA.**

FALSO

**484-¿Qué circuito suele utilizarse para traducir el código de operación de una instrucción máquina a dirección de comienzo en la memoria de control del microprograma correspondiente?**

Una memoria.

**485-¿Con cuál de los siguientes dispositivos tendría sentido utilizar E/S programada sin consulta de estado?**

Salida a un display de 7 segmentos

**486-Después de ejecutar el siguiente código, ¿qué variables serán igual a 0? (Suponer ints de 32bits y longs de 64bits)**

```
unsigned int a = 0xffffffff;
unsigned int b = 1;
unsigned int c = a + b;
unsigned long d = a + b;
unsigned long e = (unsigned long)a + b;
```

c y d

En el problema 3.4 sólo se explica que una extensión de tamaño se hace según el fuente sea signed (extensión sgn) o unsigned (ext. con ceros), pero no se explica la sección 2.2.6 (y 2.2.5) en donde se aclara que las operaciones que impliquen a algún unsigned se hacen en unsigned. Evitar esta pregunta en el futuro.

**487-Un microprocesador es superencauzado ("superpipelined") si permite la emisión de dos o más instrucciones en un mismo ciclo de reloj.**

Falso

**488-Una arquitectura de registros de propósito general puede tener instrucciones máquina de la ALU con un único operando explícito.**

Verdadero

**489-¿Qué tipo de instrucciones se emplean más en una arquitectura de acumulador?**

de transferencia de datos con memoria

**490-¿Qué modificador (switch) de gcc hace falta para compilar .s → .o sin llamar al enlazador?**

gcc -c

**491-Si queremos almacenar la palabra de 16 bits 9660h en memoria según "little-endian", quedará almacenada a partir de la posición 1000h como:**

en el byte 1000h se guarda 60h y en el 1001h se guarda 96h

**492-En una resta de dos números en complemento a dos, se produce desbordamiento cuando...**

a)

los dos operandos son negativos y el resultado es positivo.

b)

los dos operandos son positivos y el resultado es negativo.

☒ x

c)

Las dos respuestas a y b son correctas.

•d)

Ninguna de las anteriores es correcta.

**493-Si el registro %edx contiene la variable definida con la sentencia en C "int n", y queremos dividirla por 2, usaremos la instrucción:**

sarl %edx,1

**494-Una de las funciones de una interfaz de E/S es realizar la conversión de cualquier formato que pueda ser necesaria para transferir datos entre el bus de E/S y el dispositivo de E/S (por ejemplo, de paralelo a serie).**



# ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



Verdadero

**495-Indicar cuál es la dirección de salto (en qué dirección empieza la subrutina <main>) para esta instrucción call**

**0804854e: e8 3d 06 00 00 call <main>**

**08048553: 50                      pushl %eax**

08048b90

**496-Si tenemos un número n, de 64 bits, almacenado en la pareja de registros EDX:EAX (EDX contiene los 32 bits más significativos y EAX los 32 bits menos significativos) y queremos realizar la división  $n / (2^{32})$  entonces:**

Podemos quedarnos con EDX tanto si n es un número con signo como sin signo.

Cocientes positivos o sin signo redondeados hacia cero, cocientes negativos redondeados hacia más negativo: es decir, redondeado siempre hacia inferior.

No es la división entera estándar de C (que redondea hacia cero) pero el enunciado tampoco lo exige.

**497-¿Cuántos puertos de E/S permite manejar la interfaz de periféricos programable 8255?**

- c) 4 puertos de 32 bits

**498-Con respecto a los buses de la placa base:**

El bus ISA es un bus

- a) local.

x

•

b) El bus EISA es un bus local.

c) El bus MCA es un bus local.

d) Ninguna de las anteriores es cierta.

**499-**Una función C llamada `get_el()` genera el siguiente código ensamblador. Se puede adivinar que:

```
movl 8(%ebp), %eax
leal (%eax,%eax,4), %eax
addl 12(%ebp), %eax
movl var(,%eax,4), %eax
```

`var` es un array bidimensional de enteros, con cinco columnas

se puede adivinar incluso el código:

```
int get_el(int fila, int columna){
 return var[fila][columna];
}
```

porque se calcula `%eax=fila*5+columna` para indexar en `var(,%eax,4)`

**500-**La práctica “parity” debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de `parity4`:

```
int parity4(unsigned* array, int len){
 int val,i,res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 val=0;
 asm("\n"
 "ini3: \n\t"
 "xor %[x],%[v] \n\t"
 "shr %[x] \n\t"
 "test %[x], %[x]\n\t"
 "jne ini3 \n\t"
 :[v]" +r" (val)
 :[x] "r" (x)
);
 val = val & 0x1;
 res+=val;
 }
 return res;
}
```

La sentencia `asm()` del listado anterior tiene las siguientes restricciones

dos entradas y una salida

- c) [v] cuenta como salida y entrada, [x] como entrada

**501-¿Cuántas patillas de dirección tiene una memoria DRAM de 1G palabra, siendo la longitud de palabra de 16 bits?**

15

**502-Una memoria que está estructurada en palabras de 8 bits tiene una capacidad de 64 Kbits. ¿Cuántas líneas de dirección tiene dicha memoria?**

13

**503-Las celdas de memoria estática...**

mantienen la información almacenada por tiempo indefinido mientras se mantenga la alimentación

**504-Toda celda de memoria DRAM tiene al menos cuatro transistores.**  
FALSO

**505-Una organización de interrupciones de niveles múltiples con prioridad significa que, durante la ejecución de una rutina de servicio de interrupción, se aceptarán solicitudes de interrupción de algunos dispositivos, pero no de otros, según sea su prioridad**

**506-¿Cuál de los siguientes grupos de señales son entradas a la unidad de control?**

- a) Las señales de carga/incremento/desplazamiento de registros

•

- b) Las señales de selección de entradas de multiplexores del datapath  
c) Los bits del registro de indicadores (flags)  
d) Los bits de las opciones b y c

x



**507-Para direccionar una memoria de bytes en la que quepan 2G palabras de 32 bits se necesitarán:**

33 bits como mínimo

**508-Los riesgos de datos consisten en que...**

una instrucción necesita un dato calculado por otra anterior

**509-Las unidades de control cableadas memorizan los pasos de ejecución de una instrucción máquina en una memoria de control, y las microprogramadas en una PLA.**

Falso

**510-La instrucción IA32 test sirve para...**

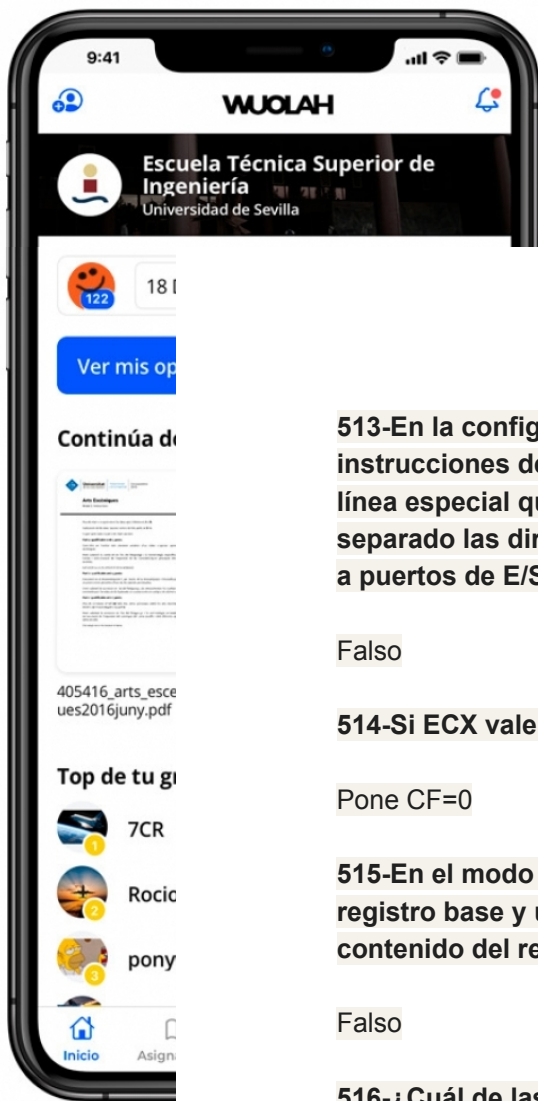
Realizar la operación and lógico bit-a-bit (a&b) pero no guardar el resultado, sino simplemente ajustar los flags

**511-Respecto a las técnicas de direccionamiento por selección lineal, decodificación centralizada y distribuida**

la selección lineal permitiría escribir un mismo dato a varios puertos E/S

**512-¿Cuál sería el "equivalente x86-64" del "pseudo-código C" `rcx = ((int*)rax)[rcx]`?**

`mov (%rax,%rcx,4),%rcx`



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



**513-En la configuración de E/S mapeada en memoria, la CPU tiene instrucciones de E/S, y cuando se ejecuta una de ellas, la CPU habilita alguna línea especial que sirve para que la circuitería externa decodifique por separado las direcciones correspondientes a memoria y las correspondientes a puertos de E/S.**

Falso

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

Pone CF=0

**515-En el modo de direccionamiento relativo a registro base se especifica un registro base y una dirección de memoria cuyo contenido se suma al contenido del registro base para obtener la dirección efectiva.**

Falso

**516-¿Cuál de las siguientes afirmaciones sobre el benchmark SPEC CPU es falsa?**

El resultado final es la media aritmética de las (12 ó 17) velocidades, bien sea de enteros ó de punto flotante (SPECint2006 ó SPECfp2006)

**517-¿Qué valor contendrá `%edx` tras ejecutar las siguientes instrucciones?**

`xor %eax, %eax`

`sub $1, %eax`

`cld`

`idiv %eax`

0

**518-¿Cuál de las siguientes no es una sección de un fichero ELF?**

a)

`.text`

para el código

•b)

`.static`

inventada

c)

.data

para datos inicializados como en la Práctica "media"

~~x~~ d)

.bss

para datos sin inicialización, mencionada en P2 Apéndice 2 Tabla 9

**519-La evolución de la arquitectura de computadores a lo largo de la historia de éstos se ha producido de forma desligada a la evolución de la tecnología de fabricación de dispositivos electrónicos.**

Falso

**520-El incremento de velocidad de los RISC se consigue a costa de un aumento del área de chip dedicado a la unidad de control.**

Falso

**521-El objetivo del control residual es aumentar la velocidad de ejecución de microinstrucciones, aunque esto tiene el inconveniente de aumentar el tamaño del microprograma.**

FALSO

**522-¿De qué depende el tamaño del contador de programa?**

Del número de direcciones de memoria.

**523-En un sistema con un único bus...**

- a) sólo un dispositivo puede escribir en un instante dado en el bus
- b) se utilizan las mismas líneas de control para conectar todos los dispositivos
- c) el procesador y los periféricos pueden funcionar a diferentes velocidades si el funcionamiento del bus es asíncrono



d) Todas las respuestas anteriores son ciertas

**524-Escribiendo en el registro de máscara de interrupciones es posible inhabilitar todas las interrupciones enmascarables.**

Verdadero

**525-¿Cuál de las siguientes no es una característica de los computadores RISC?**

La decodificación de las instrucciones

- a) debe ser simple: un computador RISC debería emplear un único formato de instrucción
- b) Para acelerar el computador RISC se emplean técnicas de pipelining.
- c) Las funciones que realizan los computadores RISC deben ser lo más complejas y potentes que sea posible.
- d) Un computador RISC no debe emplear microprogramación.

**526-Los procesadores RISC no tiene instrucciones para sumar un registro con una posición de memoria.**

Verdadero

**527-En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor que se indica a continuación, ¿en qué caso ambos programas producen el mismo resultado?**

0x1111 1111

resultado 0x0000 0002 2222 2220

porque es positivo incluso en complemento a 2

todos los demás valores se interpretan como negativos, lo primero que hace la suma con signo es extenderlos a 64bit de manera que se activan los 32 bits superiores... resultado radicalmente distinto

**528-El entrelazado de orden superior se utiliza más que el de orden inferior en procesadores vectoriales debido a que su expansión es más fácil.**

FALSO

**529-La expresión  $d(t+k) = d(t)$ , donde  $d(x)$  es la dirección de memoria referenciada en el instante de tiempo  $x$  y  $k$  es un pequeño incremento de tiempo, corresponde al principio de localidad espacial.**

FALSO

**530-La penalización por una falta de página suele ser de unas pocas decenas de ciclos de reloj de la CPU.**

Compilar sin optimización

**531-De las siguientes parejas de instrucciones, ¿cuál utiliza únicamente direccionamiento implícito?**

- a) mul,  
div
- b) push,  
call
- c) jnz, cmp
- d) sti, popf

**532-Suponga un ordenador que usa un tamaño de palabra de 32 bits y un espacio de direccionamiento de 20 bits. El valor hexadecimal de la dirección más alta es FFFFFFFFh.**

FALSO

**533-¿Qué hace gcc -O0?**

**534-El 8255 es un interfase paralelo de periféricos que suele utilizarse para E/S programada, pero también puede generar interrupciones.**

VERDADERO

**535-¿Cuál de las siguientes secuencias de tipos de memoria está ordenada de menores a mayores prestaciones?**



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



FPM, EDO, RDRAM

**536-Un sistema no segmentado tarda 20 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce (pipeline) de 4 segmentos con un ciclo de reloj de 5 ns. Cuando se procesan muchas tareas, la ganancia máxima de velocidad que se obtiene se acerca a:**

4

**537-La codificación Huffman...**

utiliza menos bits para las instrucciones más frecuentes y más bits para las menos utilizadas.

**538-La penalización por una falta de página suele ser de unas pocas decenas de ciclos de reloj de la CPU.**

FALSO

**539-En un procesador capaz de direccionar la memoria a nivel de bytes, con una longitud de palabra y tamaño del bus de datos de 32 bits, ¿cuántos accesos a memoria hacen falta para llevar de la memoria al procesador un objeto de tamaño doble palabra situado a partir del byte de dirección 4 (inclusive)?**

2

**540-El primer microprocesador de 32 bits de la familia x86 fue el:**

80386

**541-En los procesadores RISC todas las instrucciones son del tipo registro-registro, exceptuando las de movimiento de datos entre registros.**

Falso

**542-El acceso a memoria en modo página no se utiliza en segmentación no paginada.**

Falso

**543-¿Qué técnica de E/S se dice controlada por hardware?**

Acceso directo a memoria

**544-GCC/Linux IA32 resuelve el ajuste de marco de pila mediante las instrucciones:**

`pushl %ebp; movl %esp, %ebp`

**545-En una memoria virtual con segmentos paginados, cada entrada de la tabla de segmentos contiene un puntero a la primera dirección física del segmento correspondiente.**

Falso

**546-Respecto a los términos microinstrucción y microcódigo:**

Microcódigo es el contenido de la memoria de control, y una microinstrucción es una palabra de dicha memoria

tr.42 define literalmente ucód. como conjunto de microprogs. que a su vez son el conjunto de uinstr. de una instr.

**547-Sólo los microprocesadores RISC utilizan segmentación en la ejecución de sus instrucciones.**

FALSO

**548-SP y PC no son registros de uso general (GPR).**

VERDADERO

**549-La propiedad de localidad espacial se puede enunciar como: "la información que se usará en un futuro próximo es la misma que se está usando actualmente".**

Falso

**550-Un procesador de 1GHz tarda 4ns en realizar 4 instrucciones sin realizar segmentación de cauce. ¿Cuanto tardaría en realizar 9 instrucciones un procesador con segmentación de cauce de 4 etapas si no existiera ningún retraso en ninguna de las instrucciones?**

3 ns

**551-Cuando se usan interrupciones vectorizadas, el periférico siempre suministra a la CPU la dirección de memoria (completa o incompleta) donde comienza la rutina de servicio de interrupción.**

FALSO

**552-RAS significa impulso de acceso a filas o selección de acceso a filas.**

VERDADERO

**553-¿Cuál de las siguientes afirmaciones es falsa?**

a)  
el bus de control transporta señales de estado

x

b)  
el bus de direcciones es unidireccional

c)  
el bus de datos es bidireccional

•d)  
la anchura del bus de datos es de 16 bits



**554-No existen instrucciones máquina que únicamente contengan campo de operación.**

FALSO

**555-La eficiencia de un sistema que emplea memoria caché (definida como la razón entre el tiempo de acceso a caché y el tiempo medio de acceso al sistema caché-M.P.) vale 0 cuando la razón de aciertos de la caché vale también 0.**

VERDADERO

**556-En una arquitectura RISC típica:**

se usan muchas instrucciones de las disponibles en el conjunto de instrucciones.

**557-Si un byte puede agrupar píxeles consecutivos, un modo de vídeo de 512 x 256 píxeles y 16 colores por píxel ocupa una memoria de:**

64 KB

**558-La instrucción `movl %esp,%ebp`**

Copia el contenido del registro ESP en el registro EBP.

**559-Es más caro emplear buses dedicados que no dedicados, pero ello permite establecer más comunicaciones simultáneamente, aumentando las prestaciones de un computador.**

VERDADERO

**560-En una bomba como las estudiadas en prácticas, del tipo...**

```
0x08048705 <main+149>: call 0x80484c4 <gettimeofday>
```

```
...
```

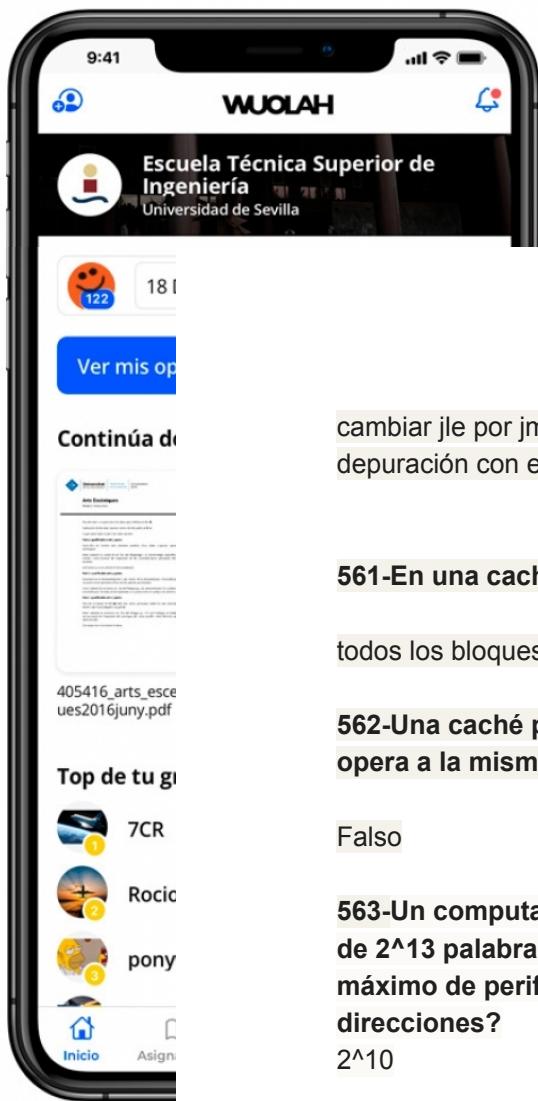
```
0x08048718 <main+168>: cmp $0x5,%eax
```

```
0x0804871b <main+171>: jle 0x8048722 <main+178>
```

```
0x0804871d <main+173>: call 0x8048604 <boom>
```

```
0x08048722 <main+178>: ...
```

ejecutada paso a paso con el depurador ddd, interesaría...



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



cambiar jle por jmp usando ddd o un editor hex, salvar el programa y reiniciar la depuración con el nuevo ejecutable

**561-En una caché asociativa por conjuntos, la vía i está constituida por:**

todos los bloques i-ésimos de cada conjunto

**562-Una caché puede incrementar las prestaciones de la memoria incluso si opera a la misma velocidad que la memoria principal.**

Falso

**563-Un computador con 15 líneas de direcciones tiene 3 módulos de memoria de  $2^{13}$  palabras y utiliza E/S mapeada en memoria. ¿Cuál es el número máximo de periféricos que pueden conectarse, si cada uno de ellos utiliza 8 direcciones?**

$2^{10}$

**564-Para un tamaño de línea fijo, llega un momento en que la tasa de faltas aumenta al aumentar el tamaño de la memoria caché.**

Falso

**565-En las transferencias asíncronas, es común acompañar a cada dato transferido de una señal de control que indica la presencia del dato en el bus.**  
VERDADERO

**566-¿En qué tipo de traducción de memoria virtual se utilizan los campos base y límite?**

Segmentación

**567-En el contexto de una llamada a función cdecl: si los contenidos de ESP y EBP son, respectivamente, 0x0008d040 y 0x00000000 antes de ejecutar una instrucción call, tras ejecutar el ret correspondiente, los contenidos serán:**  
0x0008d040 y 0x00000000

**568-La directiva de ensamblador "DW 100 DUP(0)" reserva 100 palabras inicializadas a cero.**

Verdadero

**569-En un sistema con dos buses separados, uno para el subsistema de memoria y otro para la E/S...**

el bus que une la memoria y el procesador suele funcionar a la velocidad de la memoria

**570-El marco de pila en x86-64 Linux...**

se crea para funciones en las que GCC no puede evitar que RSP baje más, como por ejemplo: que haya que calcular la dirección de una variable local, o pasar más de 6 argumentos a otra función

**571-El objetivo de un diseño CISC es...**

disminuir el número de instrucciones a ejecutar por un programa.

**572-¿Cuál de las siguientes afirmaciones es falsa?**

Las memorias DRAM

a) presentan generalmente una capacidad de almacenamiento mucho mayor que las SRAM.

b) La lectura de un bit de la matriz de almacenamiento de una memoria DRAM proporciona una señal mucho más débil que la suministrada por los inversores de una celda de memoria SRAM.

•

c) Las memorias DRAM son en general mucho más rápidas que las SRAM

d) Una celda DRAM sólo necesita un transistor y un condensador.

**573-¿Cómo actúa el indicador Z del registro de indicadores de estado?**

Se pone a 1 cuando el resultado de una operación es 0

**574-Los buses de placa madre ("board level") son relativamente rápidos y no requieren terminadores.**

VERDADERO

**575-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"
```

y suponiendo que hemos llamado a una función suma que devuelve un número de 64 bits en la pareja EDX:EAX, las instrucciones que copian ese número en resultado son:

```
movl %eax, resultado
movl %edx, resultado+4
little-endian => primero el menos significativo
```

**576-Para construir una DRAM de 4GB con pastillas de 512Mx4bit hacen falta 16 pastillas**

**577-En una jerarquía de memoria, a medida que nos alejamos del procesador...**

El tamaño de la unidad de transferencia entre dos niveles aumenta.


**578-¿Cuál de las siguientes afirmaciones es falsa respecto al lenguaje C?**

- En lenguaje C, al llamar a una
- a) subrutina o función se introducen los parámetros en la pila y después se realiza una llamada a la subrutina
  - b) Pasar a una función un puntero a una variable se traduce en introducir en la pila el valor de la dirección de memoria donde está almacenada la variable

•  


- c) Antes de volver de la rutina llamada, el programa en C se encarga de quitar de la pila los parámetros de llamada realizando varios pop
- d) Los parámetros se introducen en la pila en el orden inverso a como aparecen en la llamada de C, es decir, empezando por el último y acabando por el primero

**579-**Un computador emplea un sistema de memoria principal de 128 palabras y una memoria caché de 32 palabras. La organización de la memoria caché es totalmente asociativa y el tamaño de bloque es de 8 palabras. Se emplea el algoritmo de reemplazo FIFO. Si inicialmente la memoria caché está totalmente vacía, calcule el número de fallos cuando se lee la secuencia de direcciones de la memoria principal: 0000100, 1000001, 0000101, 0010011, 0100010, 1000100, 0000111.

- 6
- a) fallos
  - b) 3 fallos
  -  • c) 4 fallos
  - d) 5 fallos

**580-**Respecto a direccionamiento a memoria en ensamblador IA32 (sintaxis AT&T), de la forma D(Rb, Ri, S), sólo una de las siguientes afirmaciones es FALSA. ¿Cuál?

Los registros base e índice (Rb y Ri) pueden ser cualesquiera de los 8 registros enteros (EAX...ESP)

**581-**La política de correspondencia de una memoria cache con la mitad de conjuntos que líneas es:

Asociativa por conjuntos de 2 vías



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



"vía" se refiere a cuántas "alternativas" (líneas) hay en un conjunto.  
Además, todas las demás opciones tienen defectos

**582-¿Cuál es el tamaño de la marca de caché en un ordenador capaz de direccionar 1 MB de memoria principal y 32 KB de memoria caché, que emplea un tamaño de palabra de 32 bits y correspondencia asociativa por conjuntos con 16 palabras por bloque y 8 bloques por conjunto, suponiendo que las direcciones de memoria utilizan 20 bits?**  
8 BITS

**583-La ventaja de las transferencias de bloques en un bus es que sólo hay que transmitir la primera dirección y se ahorran las siguientes direcciones consecutivas.**

Verdadero

**584-Los modos de direccionamiento de una instrucción se especifican en ensamblador mediante las directivas.**

Falso

**585-En las arquitecturas del tipo registro-registro se usan instrucciones LOAD y STORE para transferir información entre registros y memoria principal.**

Verdadero

**586-El bus de direcciones contiene líneas para indicar el sentido de la transferencia de datos, por ejemplo una línea para distinguir entre lectura y escritura.**

Falso

**587-Si se necesitan 60 ns para escribir una palabra de datos de cache en memoria principal y cada bloque de cache tiene 8 palabras, ¿cuántas veces "seguidas" (sin que haya reemplazo) se tiene que escribir en un mismo bloque para que una cache de postescritura sea más eficiente que una de escritura inmediata?**

No se puede responder con los datos proporcionados

**588-Si el proceso de empaquetado de un producto (50 segundos de duración) puede segmentarse en 5 etapas, cada una de 10 segundos, de modo que 5 operarios puedan trabajar cada uno en una etapa, ¿cuál de las siguientes afirmaciones es falsa al aplicar la segmentación?**

Cada 50 s saldrá un nuevo producto empaquetado, el mismo tiempo que cuando no había cadena de empaquetamiento

**589-Son funciones de la unidad de control la lectura de memoria principal de la instrucción apuntada por el uPC y la codificación de esa instrucción.**

Falso

**590-Lo que se lee de una dirección de E/S siempre es lo mismo que lo último que se ha escrito en ella.**

Falso

**591-¿Cuál de las siguientes listas está correctamente ordenada temporalmente?**

8088, 80386, Pentium, Pentium Pro,  
a) Pentium 4, Pentium II.

**X**

•

b) 8088, 80386, Pentium, Pentium II,  
80486, Pentium 4.

c) 8088, 80486, Pentium, Pentium Pro,  
Pentium II Pentium 4.

d) 80286, 8088, Pentium, Pentium Pro,  
Pentium III, Pentium 4.

**592-En IA32, ¿cuál de las siguientes afirmaciones es incorrecta?**

CALL ETIQUETA guarda en la pila la dirección de retorno y el registro de indicadores

**593-Un ciclo de refresco en una memoria DRAM requiere un impulso RAS y un CAS.**

FALSO

**594-Un computador con 8 bits en el bus de direcciones puede direccionar como máximo:**

256 palabras

**595-¿Cuál de las siguientes afirmaciones es cierta?**

Los modos de

a) direccionamiento de una instrucción se especifican en ensamblador mediante las directivas.



•

- b) El direccionamiento a registro es similar al directo. La diferencia es que el campo de direcciones referencia un registro en lugar de una dirección de memoria principal.
- c) En el direccionamiento directo, la dirección efectiva se encuentra en un registro de uso general del procesador.
- d) En el modo de direccionamiento implícito, el objeto direccionado es una constante contenida en la propia instrucción.

**596-Un microprocesador es superescalar si tiene un número mayor de etapas y éstas son más cortas que las de un cauce ("pipeline") normal, permitiendo una velocidad de reloj mayor.**

FALSO

**597-Un mapa de direcciones de memoria es una representación pictórica del espacio de direcciones asignado a cada chip o tipo de memoria en el sistema.**  
VERDADERO

**598-Se dice que las máquinas con arquitectura Von Neumann siguen un modelo de programa...**

Almacenado.

**599-¿Cuál de las siguientes afirmaciones respecto a la memoria RDRAM no es cierta?**

El bus de datos suele ser muy ancho.



**600-¿Cuál de las siguientes es una idea fundamental de la jerarquía de memoria?**

Que dispositivos más pequeños y rápidos sirvan de cache para dispositivos más grandes y lentos

**601-Sea un computador de 32 bits que dispone de una memoria cache de 512 KB y líneas de 64 bytes. ¿Cuántas líneas tiene la cache?**

8192

**602-Una placa madre de un 486 con un único SIMM de 30 contactos con 8 chips de 1M x 1, tiene 2 M palabras de memoria principal.**

Falso

**603-Con el repertorio IA32, para sumar %eax y %ebx dejando el resultado en %ecx se podría hacer lo siguiente:**

lea (%eax, %ebx, 1), %ecx

**604-El registro MDR/MBR...**

contiene el valor que va a ser almacenado en la memoria, o bien se usa para recibir un valor procedente de la memoria

**606-Respecto a requisitos de alineamiento de structs en gcc/IA32 x86 y x86-64, alguna de las siguientes afirmaciones es falsa**

en x86-64 Linux alinea float a 8x (Windows también)

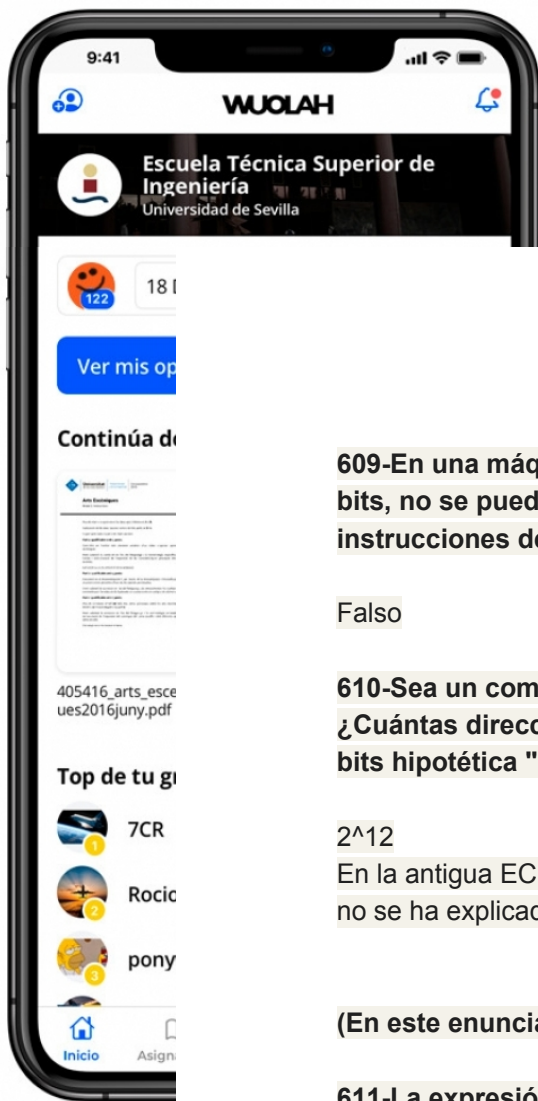
redacción original era "x86-64 Linux long double 8x, Windows no", motivo por el cual notamos errata transparencias (no dicen nada de x86-64 Windows, y dicen que x86-64 Linux long double 8x, cuando libro pág.325 dice que 16x). Posible corrección: "alinea long double a 12x (Windows no)". Redacción final es aún más claramente errónea.

**607-¿En qué generación, dentro de la historia de los computadores digitales, aparece la microprogramación?**

tercera

**608-Respecto a los conceptos de interfaz de dispositivo, controlador(a), puerto de E/S:**

La controladora o interfaz contiene los puertos necesarios para utilizar el dispositivo



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



**609-En una máquina con 32 registros direccionables e instrucciones de 16 bits, no se pueden codificar 32 instrucciones de dos registros, 32 instrucciones de 1 registro y 32 instrucciones de 0 direcciones.**

Falso

**610-Sea un computador con 48 registros y 200 instrucciones máquina. ¿Cuántas direcciones de memoria permite el formato de la instrucción de 32 bits hipotética "beqz r1, r2, dir"?**

$2^{12}$

En la antigua ECI se hacían estos cálculos, códigos Hamming, etc. En la nueva EC no se ha explicado con tanto detalle. Evitar esta pregunta.

(En este enunciado el símbolo  $\wedge$  representa potenciación)

**611-La expresión que cumplen las direcciones de una memoria de bytes en las que queda mal alineada una palabra de 32 bits es Dirección mod 4  $\neq$  0.**  
VERDADERO

**612-La instrucción NOP no se usa nunca, ya que no hace nada.**  
FALSO

**613-Para que el modo de direccionamiento indirecto a través de registro coincida con el modo de direccionamiento indexado, el campo de desplazamiento en el direccionamiento indexado ha de ser igual al contenido del registro, pero cambiado de signo.**

Falso

**614-Respecto al sistema de Entrada / Salida, ¿cuál de las siguientes afirmaciones es incorrecta?**

- a) La CPU se comunica con el periférico por medio del controlador y de software de E/S.
- b) Un protocolo sirve para “ponerse de acuerdo” en cosas como velocidad, paridad, nº de bits, etc.

- - c) Un controlador se encarga de la comunicación con la CPU.
  - d) La mayoría de los periféricos trabajan a velocidad muy superior a la CPU; por eso es necesario sincronizar.

**615-En las transacciones de bloques del bus PCI se envía una dirección por cada dato, ya que direcciones y datos no están multiplexados en el tiempo.**

Falso

**616-La codificación Huffman utiliza menos bits para las instrucciones menos frecuentes y más bits para las instrucciones más utilizadas.**

FALSO

**617-La gestión distribuida del arbitraje consiste en un árbitro central del bus al que llegan múltiples líneas de petición de varios maestros potenciales y del que salen las correspondientes señales de concesión.**

FALSO

**618-Una memoria SRAM tiene una capacidad de 64 Kbits y utiliza 12 líneas para direccionamiento. Indique cuál es el tamaño de palabra de dicha memoria:**

16 bits

**619-Una dirección de memoria se refiere siempre a:**

a)  
una palabra

b)  
16 bits

c)  
un byte

✓ • d)  
ninguna de las anteriores

**620-La idea fundamental en la que se basan los computadores RISC es minimizar el número de instrucciones necesarias para realizar una tarea determinada.**

F

**621-En la ejecución de una instrucción...**

la ALU realiza las operaciones aritméticas y lógicas

**622-¿Para qué se utiliza la función gettimeofday en la práctica de la "bomba digital"?**

Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la contraseña o el PIN

**623-Respecto a si un computador dispone de E/S independiente (separada) o usa E/S mapeada a memoria:**

Si el repertorio del procesador tiene instrucciones del tipo IN y OUT, es que el computador dispone de E/S separada

**624- Compilar .s → ejecutable, usando sólo as y ld, sin gcc...**

Se puede, usando en as y ld los modificadores (switches) que corresponda

**625-Si AX = 0xFA50 y ejecutamos XOR \$0xFF, AX**

Se realiza el complemento a 1 de AL.

**626-Dado un camino de datos concreto, un posible formato de microprogramación se caracteriza como horizontal o vertical según tenga más o menos (señalar la respuesta falsa)**

microbifurcaciones

**627-En una arquitectura Von Neumann cada palabra de memoria contiene un campo que permite diferenciar entre instrucción o dato.**

F

**628-Se puede construir procesadores de varios anchos de palabra usando varios chips de sección de 4 bits.**

V

**629-El 8086 sólo permite direccionar directamente 640 KB de memoria. Por eso los PCs basados en este chip no disponían de más memoria RAM.**

F

**630-Al llamar a una función de 2 argumentos foo(arg1, arg2), ¿cuál es el orden correcto de las operaciones? (suponiendo convención de llamada x86 cdecl, y que foo requiere ajustar marco de pila, esto es, salvar %ebp)**  
push arg2, push arg1, call foo, push %ebp

**631-En la E/S controlada por interrupciones:**

La CPU transfiere el control a una rutina de servicio cuando recibe una interrupción.

**632-Los computadores con estructura Von Neumann utilizan el modelo de programa externo, frente a programa cableado y almacenado.**

F

**633-Un computador que utilice el sistema big-endian, almacena el número 2143h a partir de la dirección 0 como:**

21h en el byte de la dirección 0 y 43h en el byte de la dirección

1

**634-La microprogramación vertical se caracteriza por tener:**

mucha codificación

**635-Se dispone de un computador cuyo tiempo medio de acceso al sistema de memoria caché y memoria principal es de 18 ns. Si la tasa de fallo de la caché es de 0,2 y el tiempo de acceso a la memoria principal es 50 ns. ¿Cuál es el tiempo de acceso a la caché?**

8 ns

**636-Siempre que se utiliza "polling" la prioridad de los dispositivos interruptores queda fijada mediante encadenamiento ("daisy-chain").**

Falso

**637-¿Cuál de las siguientes afirmaciones es cierta?**

El modo de direccionamiento indirecto a través de registro es el modo más sencillo de direccionamiento a memoria.



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**638-Con 8 circuitos de memoria RAM de 1K x 8 se puede crear un memoria de:**

a)  
1K x 64

b)  
8K x 8

c)  
2K x 32

✓•d)

Todas las combinaciones anteriores son posibles

**639-Al ejecutar el fragmento de código:**

```
leal -48(%eax), %edx
cmpl $9, %edx
ja .L2
```

**se salta a .L2 si el contenido del registro %eax:**

está fuera del intervalo [48,57]

**640-Cada bit de una RAM dinámica ha de ser refrescado cada pocos nanosegundos.**

Falso

**641-Cuál de los siguientes no es un modo de direccionamiento IA32?**

•a)  
Cache  
no existe, ver [T2.1.3ConASM] tr.27

b)

Registro

c)

Inmediato

d)

Memoria

**642-¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC es cierta?**

Ninguna de las otras afirmaciones es cierta

**643-Si se dice que en un sistema computador cada dirección especifica uno o dos puertos de E/S, se refiere a que:**

Un puerto será de sólo lectura, otro de sólo escritura, y ambos se decodifican en la misma dirección

**644-Un bus se compone de:**

líneas de control/estado, líneas de dirección y líneas de datos

**645-El Intel 8086:**

Incluía instrucciones de multiplicación.

**646-El fragmento de código:**

```
poll: in a, 0x20
 cmp a, $0
 jnz poll
 load a, 0x11
 out 0x21, a
```

**corresponde a:**

E/S programada con consulta de estado

**647-La idea de la arquitectura RISC se debe, entre otros, a:**

John Cocke

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

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

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

CMP, TEST

**650-Un TLB (buffer de traducción anticipada) tiene un número de entradas o elementos mucho menor que el número de páginas de la memoria virtual.**

V

**651-En una arquitectura little-endian el código de operación aparece al final de la instrucción.**

F

**652-Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señalar la respuesta falsa)**

cola de instrucciones (precaptación)

**653-Nunca puede utilizarse "polling" para identificar la fuente de una interrupción en un sistema con "daisy-chain"**

F

**654-SCSI son las siglas de Small Computer System Interface (interfaz del sistema para computadores pequeños).**

Verdadero

**655-Un procesador cuya instrucción CALL guardara la**



**dirección de retorno en un registro RL (llamado "de enlace"):**

No permitiría llamadas anidadas ni recursivas.

**656-En un procesador RISC, las bifurcaciones no degradan las prestaciones del "pipeline".**

F

**657-Una de las funciones de una interfaz de E/S es generar señales de temporización cuando se necesiten, según el esquema que se utilice el subbus de control.**

**658-¿Cuál afirmación es FALSA al comparar las arquitecturas x86 y x86-64?**

**659-A medida que aumenta el tamaño de página en un sistema de memoria virtual, ¿qué ocurre con el tamaño de las tablas de páginas?**

Disminuye

**660-¿En qué tipo de memoria virtual es un problema la fragmentación externa?**

**661-Se tiene una memoria que emplea entrelazado. Si fallan varias celdas contiguas de uno de sus chips de memoria, ¿con qué tipo de entrelazado de memoria sería más fácil poder utilizarla?**

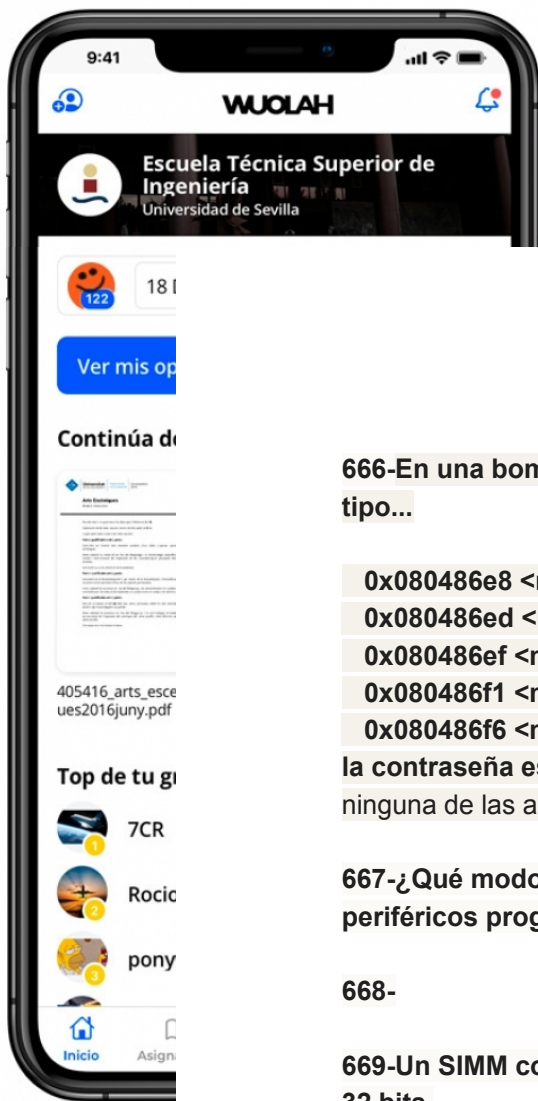
**662-Una cache de 256 B asociativa por conjuntos de 4-vías con líneas de 16 B tendría**

4 conjuntos

**663-Para diseñar una memoria con ancho de palabra  $k \cdot m$  (y mismo nº palabras que los módulos) a partir de módulos con ancho de palabra  $m$ , se utilizan  $k$  módulos (mediante  $\_$  se representa subíndice)**

**664-El Pentium III es superescalar porque permite "emitir" varias instrucciones por ciclo de reloj.**

**665-¿Cómo actúa el indicador N del registro de indicadores de estado?**



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



666-En una bomba como las estudiadas en prácticas, del tipo...

```
0x080486e8 <main+120>: call 0x08048524 <strcmp>
0x080486ed <main+125>: test %eax,%eax
0x080486ef <main+127>: je 0x080486f6 <main+134>
0x080486f1 <main+129>: call 0x08048604 <boom>
0x080486f6 <main+134>: ...
```

la contraseña es...  
ninguna de las anteriores

667-¿Qué modo de funcionamiento permite a la interfaz de periféricos programable 8255 utilizar un bus bidireccional?

668-

669-Un SIMM con 8 chips de 4Mx1 contiene 1 M palabras de 32 bits.

F

670-En el contexto de las DRAM, la activación de la señal CAS se emplea para realizar ciclos de refresco.

671-La "postescritura ("write-back") marcada" es más eficiente que la "postescritura siempre".

V

672-La cache es gestionada por:  
unidades de "manejo" (gestión) hardware

673-Un modo de direccionamiento en el que se especifica un registro y una dirección de memoria cuyo contenido se suma al contenido del registro base para obtener la dirección efectiva, se conoce como:

- a)base con desplazamiento
- b)absoluto o directo
- c)indirecto a registro a través de memoria
- d)ninguna de las anteriores\*\*\*\*\*

674-Si la estructura struct a ocupa un espacio de 28 bytes

en memoria, ¿cuántos bytes ocupa la siguiente estructura  
struct b cuando se compila en 64 bits?

```
struct b {
 struct a a1;
 int i;
 struct a a2;
}
```

60 bytes

675-En un sistema de memoria con entrelazado de orden inferior y M módulos de memoria la dirección de memoria a está en el módulo a mod M.

V

676-Alguna de las siguientes señales no sirve de entrada a la unidad de control. ¿Cuál?

contador de programa (bits del registro PC)

677-¿Cuál es la característica tecnológica principal de la tercera generación de computadores?

Los circuitos integrados

678-¿Cuál de las siguientes secuencias de instrucciones multiplica el (contenido del) registro EAX por 18?

leal (%eax,%eax,8), %eax

leal (%eax,%eax), %eax

679-Un sistema no segmentado tarda 20 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce (pipeline) de 4 segmentos con un ciclo de reloj de 6 ns. Cuando se procesan muchas tareas, la ganancia máxima de velocidad que se obtiene se aproxima a:

3,33

680-Al traducir la sentencia C `r->i = val;` gcc genera el código ASM `movl %edx, 12(%eax)`. Se deduce que

el desplazamiento de i en \*r es 12  
y en concreto r está en %eax

**681-El computador EDVAC, propuesto por John Von Neumann, presentaba dos importantes diferencias respecto al ENIAC:**

Empleaba aritmética binaria y permitía trabajar con un programa almacenado.

**682-Cada celda de un chip de memoria DRAM de 1M x 1, organizada en una matriz de 512 filas x 2048 columnas, necesita ser refrescada cada 16 ms. ¿Cada cuánto tiempo ha de realizarse una operación de refresco en el chip?**

31,25 microsegundos

**683-El lenguaje máquina...**

a)  
es un conjunto de nombres simbólicos o nemotécnicos.

b)  
facilita la portabilidad de los programas.

c)  
es el mismo para todos los computadores.

✓•d)  
Ninguna de las respuestas anteriores es correcta.

**684-La instrucción negl:**

realiza el complemento a dos

**685-Una caché totalmente asociativa es equivalente a una asociativa por conjuntos de una vía.**

Falso

**686-En la secuencia de programa siguiente:**

**804854e:e8 3d 06 00 00 call 8048b90 <main>**

**8048553:50 pushl %eax**

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

**0x8048553**

**687-Una máquina que almacene el número 4321h a partir de la dirección 0 como 21h, 43h utiliza el sistema big-endian.**

Falso

**688-Las instrucciones que asignan la dirección 0x78e00 a un puntero situado en la posición 0xff00 son:**

**mov \$0x78e00,%eax**

**mov %eax,0xff00**

**689-En general, un operación segmentada ("pipelined") requiere el mismo tiempo o más, desde el principio hasta el fin y para un único par de operandos, que la misma operación en una implementación no segmentada.**

**V**

**690-La penalización por una falta de página suele ser de cientos de miles de ciclos de reloj de la CPU.**

Verdadero

**691-En el direccionamiento inmediato el operando reside en:**

**en la instrucción tras el código de operación**

**692-El bus del sistema es**

**el que conecta CPU-M, ya sea un sistema con bus único o con múltiples buses**

**693-La CPU guarda automáticamente el contexto del programa que ejecuta cuando le llega y atiende una petición de DMA.**



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**694-Según la clasificación m/n, las máquinas de acumulador son de tipo**

1/1

**695-En la realización de la práctica de la bomba digital, 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 cambiaría el salto condicional por un salto incondicional?**

```
set *(char*)0x080486ef=0xeb
```

**696-Respecto a las bombas estudiadas en la práctica "bomba digital", ¿en cuál de los siguientes tipos de bomba sería más fácil descubrir la(s) contraseña(s)? Se distingue entre strings definidos en el código fuente de la bomba, y strings solicitados al usuario mediante scanf(). Por "cifrar" se entiende aplicar la cifra del César (sumar o restar una constante fija a los códigos ASCII).**

1 string del fuente se cifra, y se compara con el string del usuario  
aunque hubieran sido 2, aunque su hubieran cifrado y concatenado o al revés, sigue siendo igual de fácil, porque con el debugger se puede ver qué hay que poner de contraseña literalmente

**697-Un archivo .o que contiene código objeto:**  
Contiene instrucciones máquina binarias.

**698-¿En qué tipo de transferencias es necesario establecer un periodo de tiempo máximo después del cual se considera que ha fallado?**

En las transferencias asíncronas

**699-La instrucción MOVSB del 8086 transfiere el byte apuntado por ES:DI al acumulador AL.**

FALSO

**700-En el bus PCI las direcciones y los datos están multiplexados en el tiempo, y existen transacciones de bloques de datos en las que se envía una única dirección y se envían (o reciben) muchos datos.**

Verdadero

**701-¿Qué tipo de localidad de las referencias a memoria se define como: "si se referencia un elemento, volverá a ser referenciado pronto"?**

Localidad temporal

**702-Después de ejecutar una instrucción de suma sobre dos números con signo de la que sabemos que no provocará overflow (los dos números son pequeños en valor absoluto), queremos comprobar si el resultado de la suma es menor que 0. ¿Qué flag necesita comprobar la instrucción de salto condicional equivalente a "if (resultado<0) then goto label"?**

SF

**703-Un bus de ciclo partido está disponible para otro maestro en las ranuras entre la petición de una lectura y la contestación con el dato.**

VERDADERO

**704-En una caché con 64 bytes de longitud de línea, ¿qué bits de una dirección de memoria se utilizan para determinar a qué byte dentro de la línea se refiere dicha dirección?**

Los 6 bits menos significativos

**705-La memoria caché del computador es:**

a)

Más rápida que la memoria principal

b)

De menor capacidad que la memoria principal

✓•c)

a) y b) son correctas

d)

Ninguna de las anteriores es correcta

**706-Una jerarquía de memoria consta de una cache de con una tasa de aciertos del 92% y 4 ns de tiempo de acceso y una memoria principal con una tasa de aciertos del 100% y 100 ns de tiempo de acceso. ¿Cuál es el tiempo promedio estimado de acceso a memoria?**

12 ns

$0.92 \cdot 4\text{ns} + 0.08 \cdot (100+4)\text{ns}$

**707-EI bus GPIB utiliza handshake de 6 flancos. Esto es necesario para que un emisor pueda enviar datos a varios receptores distinguiéndose el hecho de que el dato haya sido leído por todos del hecho de que todos estén listos para recibir un nuevo dato.**

VERDADERO

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

EAX

**709-Para la memoria virtual se suele utilizar correspondencia directa**

FALSO



**710-Un programa de ordenador que convierte un programa fuente de alto nivel completo en lenguaje máquina se llama un:**  
compilador

**711-¿Cómo se devuelve en ensamblador x86-64 Linux gcc el valor de retorno de una función long int al terminar ésta?**

- a) La instrucción RET lo almacena en un registro especial de retorno.
- b) Por convención se guarda en %eax.
- c) Se almacena en pila justo encima de los argumentos de la función
- d) **CORRECTA**->Ninguna de esas formas es la correcta

**712-¿Qué política de memoria virtual para colocar nuevos segmentos en los huecos libres de la memoria principal evita el que se generen huecos pequeños?**

Peor ajuste

**713-¿Cuántas señales de control se necesitan como mínimo para implementar un sistema de gestión de interrupciones?**

2

**714-El tiempo de acceso de un SIMM es mayor que el de un único chip, debido al retraso de los decodificadores que contiene para generar los distintos "Chip Select".**

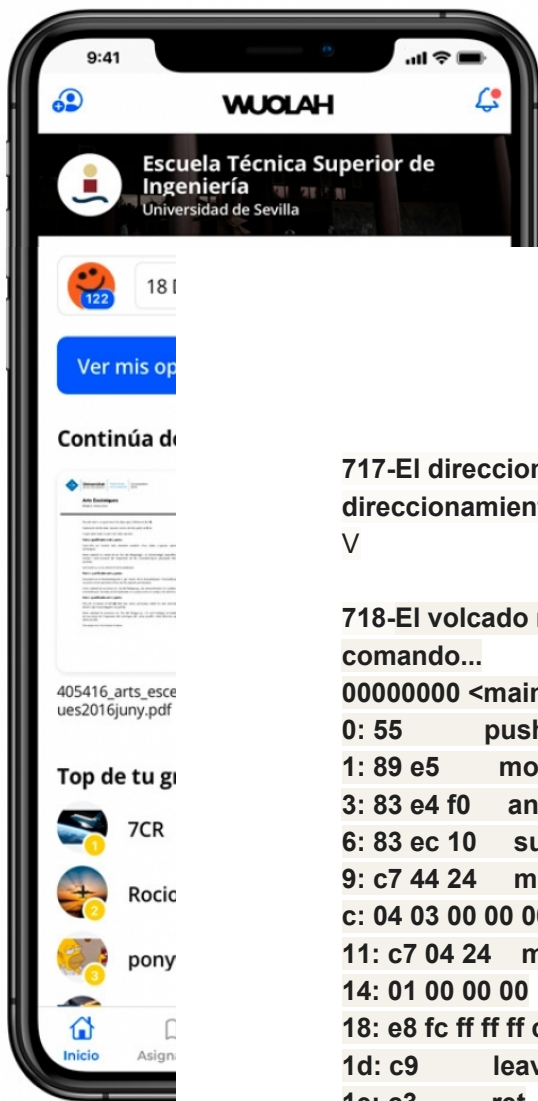
FALSO

**715-Usar un procesador más rápido siempre implica un incremento proporcional en las prestaciones de un computador, incluso si la velocidad de la memoria principal permanece inalterada.**

FALSO

**716-El resultado de desplazar aritméticamente dos posiciones hacia la derecha el número de 8 bits en complemento a dos -32 es:**

-8



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



**717-El direccionamiento inmediato no es más lento que el direccionamiento directo.**

V

**718-El volcado mostrado abajo se ha obtenido con el comando...**

**00000000 <main>:**

**0: 55      push %ebp**

**1: 89 e5      mov %esp,%ebp**

**3: 83 e4 f0    and \$-16,%esp**

**6: 83 ec 10    sub \$0x10,%esp**

**9: c7 44 24    movl \$3, 4(%esp)**

**c: 04 03 00 00 00**

**11: c7 04 24    movl \$0x1, (%esp)**

**14: 01 00 00 00**

**18: e8 fc ff ff ff call <main+0x19>**

**1d: c9      leave**

**1e: c3      ret**

**objdump -d p2.o**

**-d porque es un desensamblado (no cabeceras -h ni tablas -t)**

**719-La conexión entre un dispositivo de E/S y el procesador mediante bus:**

Permite conectar en paralelo varios dispositivos

**720-Según el concepto de máquina virtual de Tanenbaum, el nivel de máquina convencional consiste en una computadora hipotética con un determinado lenguaje máquina, cuyos programas son interpretados por la máquina virtual del nivel de microprogramación, supuesto que éste existe.**

Verdadero

**721-Si desplazamos a la izquierda un registro 3 posiciones:**

Lo multiplicamos por 8

**722-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"
```

```
push resultado+4
push resultado
push resultado+4
push resultado
push $formato
call printf
add $20, %esp
Sí. $20 = 8 (\%llx) + 8 (\%llu) + 4 (\$formato)$
```

**723-Alguno de los siguientes no es un nombre de registro en una máquina IA32 en modo 32 bits**

sil  
Sí lo sería en modo 64 bits

**724-Las instrucciones máquina contienen toda la información necesaria para ejecutarse, y además su interpretación es independiente de la posición que ocupan en el programa.**

Verdadero

**725-Para realizar la microoperación  $MAR \leftarrow PC$ , habrá que activar:**

EnPC y LdMAR

**726-El primer nivel de una jerarquía de memoria tiene una tasa de aciertos del 75% y las peticiones de memoria tardan 12 ns en completarse si dicha posición se encuentra en ese nivel y 100 ns si no es así. ¿Cuál es el tiempo medio de acceso de la jerarquía?**

34 ns

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

edi

**728-¿Cuál de las siguientes secuencias de instrucciones multiplica %eax por 10?**

a)

leal(%eax,%eax,4), %eax  
sall \$2, %eax  
sería  $x5 \times 4 = x20$

b)

imull \$0x10, %eax  
sería  $x16$

c)

addl %eax, %eax  
shll \$5, %eax  
sería  $x2 \times 32 = x64$

✓•d)

Varias o ninguna de las respuestas anteriores son correctas, no se puede marcar una y sólo una

**729-¿Qué política de colocación en caché necesita más comparadores, la correspondencia asociativa por conjuntos o la correspondencia por sectores?**

Depende de si es mayor el número de bloques por conjunto o el número de sectores

**730-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 movl longlista, %ecx copia el siguiente valor:**

4

Práct.2, Tut, pág.7

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

Ninguna o escasa codificación.

**732-HIJO DE PUTA**

**733-En una memoria con entrelazado de orden inferior, M módulos y acceso simultáneo, el tiempo de acceso a las 2M palabras cuyas direcciones son 0, 1, ..., 2M-1 es menor o igual que 3Ta, siendo Ta el tiempo de acceso a cada módulo de memoria.**

Verdadero

**734-Los buses síncronos y semisíncronos disponen de una línea de reloj "maestro", y todas las actividades del bus tienen una duración igual a un número entero de ciclos de ese reloj.**

Verdadero

**735-Suponiendo que todos los registros inicialmente contienen el valor 1, ¿cuál es el valor de ECX tras la ejecución de la siguiente secuencia de instrucciones?**

```
mov $4, %eax
mov $3, %ebx
lea (%eax, %eax), %ecx
sub %ebx, %ecx
imul %ecx, %eax
```

5



## ¿Clases de Inglés telefónicas con Profesores Nativos dónde y cuándo quieras?

25 minutos de conversación en inglés por teléfono desde 3,25 €/sesión.  
Conoce más en [www.ringteacher.com](http://www.ringteacher.com)

ringteacher



**736-¿Cuántas líneas de dirección son necesarias en un memoria RAM dinámica de 64 K palabras? ¿Y en una estática?**

8 / 16

**737-En los sistemas con una jerarquía de memoria dividida en varios niveles se da el problema de la consistencia o coherencia de los datos entre los distintos niveles. Si una palabra se modifica en un nivel, en algún momento habrá que traspasar ese cambio a los niveles inferiores. Para ello hay varias políticas:**

Post-escritura: se retrasa la actualización en los niveles inferiores hasta que el bloque modificado tenga que ser reemplazado

**738-En contraposición a un ejecutable Linux ELF, un fichero objeto (obtenido con gcc -c)**

ubica el código (y los datos) a partir de la posición 0x0, las direcciones definitivas sólo se calculan tras enlazar

**739-Un Pentium 4 a 3,2 GHz dispone de 7 unidades de ejecución en paralelo, con 20 etapas de segmentación, y es capaz de emitir (comenzar a ejecutar) 3 instrucciones en cada ciclo de reloj. ¿Qué velocidad aproximada de ejecución de instrucciones será capaz de alcanzar (MIPS = millones de instrucciones por segundo)?**

9000 MIPS

**740-En IA32, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto al resto?**

mov \$-1, %edi  
mov no afecta a los flags

**741-Un sistema no segmentado tarda 200 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce segmentado de 20 etapas con un ciclo de reloj de 12 ns. Cuando se procesan muchas tareas, la máxima ganancia de velocidad que podría obtenerse se acerca a:**

16,67

**742-Las memorias caché con política de extracción anticipativa (prebúsqueda o preextracción) no soportan búsqueda por demanda.**

Falso

**743-Si el tiempo de acceso a la memoria cache es de 2 ns y el tiempo necesario para tratar un fallo de cache es de 80 ns, ¿cuál es la tasa de aciertos necesaria para que el tiempo medio de acceso al sistema de memoria sea de 10 ns?**

0.9

**744-La cache con correspondencia directa se puede considerar como un caso límite de la asociativa por conjuntos, en donde...**

**745-Un overflow nunca puede ocurrir cuando:**

•d)

se suma un número positivo a un número negativo  
resultado queda entre ambos => se puede representar

**746-El algoritmo LRU ("Menos Recientemente Usado/a") se puede utilizar tanto en memoria caché como en sistemas de memoria virtual con paginación.**

Verdadero

**747-En una CPU de 32 bits con memoria de bytes, el problema es que.**

Hay que respetar el ordenamiento de bytes y reglas de alineamiento con que se diseñó la CPU

**748-La primera instrucción ensamblador de una subrutina compilada con gcc en Linux/x86 cdecl suele ser:**

```
push %ebp
```

**749-¿En qué generación, dentro de la historia de los computadores digitales, aparecieron la microprogramación, la segmentación de cauce, la memoria cache, los S.O. multiusuario y la memoria virtual?**

3ª generación (1965-75)  
en la 3ª generación se inventó casi todo

**750-Un cauce ("pipeline") de instrucciones con 5 etapas tarda 7 ciclos de reloj o más en ejecutar 3 instrucciones si éstas utilizan las cinco etapas.**

Verdadero

**751-PC y SP son dos registros de uso general (GPR).**

Falso

**752-Si se necesitan 60 ns para escribir una palabra de datos de caché en memoria principal y cada bloque de caché tiene 8 palabras, ¿cuántas veces seguidas se tiene que escribir en un mismo bloque para que una caché de postescritura sea más eficiente que una de escritura inmediata?**

Depende de la tasa de aciertos

**753-¿Cuál de las siguientes afirmaciones acerca de las**



**interrupciones en el PC (modo real) es cierta?**

**X** a)

Cada vector de interrupción es una palabra de 16 bits

b)

Existen 1024 vectores de interrupción

c)

La tabla de interrupciones tiene un tamaño de 256 bytes

• d)

Todas las interrupciones se pueden generar por software

|         |        |
|---------|--------|
| Nombre: |        |
| DNI:    | Grupo: |

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 4/20 si es correcta, 0 si está en blanco o claramente tachada, -4/60 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

- La dirección efectiva del primer parámetro de llamada a una función suele calcularse desde el código de la función como:
  - EBP+8
  - EBP-8
  - EBP+4
  - EBP-4
- El comienzo de un procedimiento que siga la convención cdecl es:
  - mov %ebp,%esp; push %ebp
  - mov %esp,%ebp; push %ebp
  - push %ebp; mov %ebp,%esp
  - push %ebp; mov %esp,%ebp
- Considere una función C declarada así:
 

```
void fun4arg (int a,int b,int c,int d);
```

 Suponiendo que fun4arg se ha compilado para una máquina x86 IA-32 con enteros de 4 bytes, ¿cuál sería la dirección del argumento b relativa a %ebp, en el marco de pila de fun4arg?
  - %ebp + 8
  - %ebp + 12
  - %ebp + 16
  - %ebp + 20
- ¿Cuál de las siguientes afirmaciones sobre las caches es **\*FALSA\***?
  - Casi ningún procesador actual tiene memoria cache L2
  - Las direcciones a las que accede un programa no son completamente aleatorias, sino que se rigen por ciertos patrones de localidad
  - Un procesador actual tiene varias cachés de nivel 1
  - La caché de nivel 3 no contiene toda la memoria que maneja el programa
- En un sistema Linux x86-64, ¿cuál de las siguientes variables ocupa más bytes en memoria?
  - char a[7]
  - short b[3]
  - int \*c
  - float d
- En la práctica "suma" se pide sumar una lista de 32 enteros SIN signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando acarrees. ¿Cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32bits (sin signo)?
  - 0xfc00 0000
  - 0xfbff ffff
  - 0x0800 0000
  - 0x07ff ffff
- En la práctica "suma" se pide sumar una lista de 32 enteros CON signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando

desbordamiento. ¿Cuál es el valor negativo más pequeño (en valor absoluto) que repetido en toda la lista causaría desbordamiento con 32bits (en complemento a 2)?

- a. 0xfc00 0000
- b. 0xfbff ffff**
- c. 0xf800 0000
- d. 0xf800 0001

8. ¿Qué valor contendrá edx tras ejecutar las siguientes instrucciones?

```
xor %eax, %eax
sub $1, %eax
cltd
idiv %eax
```

- a. 0**
- b. 1
- c. -1
- d. No puede saberse con los datos del enunciado

9. La práctica "popcount" debía calcular la suma de bits de los elementos de un array. Un estudiante entrega lo siguiente:

```
int popcount4(unsigned* array,
 int len) {
 int i, j, res = 0;
 for(i = 0; i < len; ++i) {
 unsigned x = array[i];
 int n = 0;
 do {
 n += x & 0x01010101L;
 x >>= 1;
 } while(x);
 for(j = 16; j == 1; j /= 2) {
 n ^= (n >>= j);
 }
 res += n & 0xff;
 }
 return res;
}
```

Esta función popcount4:

- a. produce el resultado correcto**
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}
- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

10. La práctica "paridad" debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity6:

```
int parity6(unsigned * array,
 int len) {
 int i, result = 0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 asm("mov %[x], %%edx \n\t"
 "shr $16, %%edx \n\t"
 "shr $8, %%edx \n\t"
 "xor %%edx, %%edx \n\t"
 "setp %%dl \n\t"
 "movzx %%dl, %[x] \n\t"
 : [x] "+r" (x)
 : "edx"
);
 result += x;
 }
 return result;
}
```

Esta función parity6:

- a. produce el resultado correcto
- b. fallaría con array={0,1,2,3}**
- c. fallaría con array={1,2,4,8}
- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

11. En la práctica "paridad" se pide calcular la suma de paridades de una lista de enteros sin signo. Suponer que un estudiante entrega la siguiente versión:

```
int paridad5(unsigned* array,
 int len) {
 int i, k, result = 0;
 unsigned x;
 for (i = 0; i < len; i++) {
 x = array[i];
 for (k = 16; k == 1; k /= 2)
 x ^= x >> k;
 result += (x & 0x01);
 }
 return result;
}
```

Esta función:

- a. es correcta

b. falla para `array={0,1,2,3}`

c. falla para `array={1,2,3,4}`

d. no se puede marcar una y sólo una de las opciones anteriores

12. Utilizando la sentencia `asm`, las denominadas restricciones que se indican al final de dicha sentencia, involucran a:

a. solamente las entradas

b. solamente las salidas

c. solamente los sobrescritos

d. Ninguna de las anteriores es cierta

13. En la realización de la práctica de la bomba digital, una parte del código máquina es el siguiente:

```
0x080486e8 <main+120>: call 0x8048524 <strcmp>
```

```
0x080486ed <main+125>: test %eax,%eax
```

```
0x080486ef <main+127>: je 0x80486f6<main+134>
```

```
0x080486f1 <main+129>: call 0x8048604 <boom>
```

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

a. `set $0x080486ef=0xeb`

b. `set *(char*)0x080486ef=0xeb`

c. `set *(char*)0x080486f6=jmp`

d. `set %0x080486ef=0xeb`

14. En una bomba como las estudiadas en prácticas, del tipo...

```
0x0804873f <main+207>: call 0x8048504 <scanf>
```

```
0x08048744 <main+212>: mov 0x24(%esp),%edx
```

```
0x08048748 <main+216>: mov 0x804a044,%eax
```

```
0x0804874d <main+221>: cmp %eax,%edx
```

```
0x0804874f <main+223>: je 0x8048756<main+230>
```

```
0x08048751 <main+225>: call 0x8048604 <boom>
```

```
0x08048756 <main+230>: ...
```

la contraseña es...

a. el entero `0x804a044`

b. el entero almacenado a partir de la posición de memoria `0x804a044`

c. el string almacenado a partir de la posición de memoria `0x24(%esp)`

d. ninguna de las anteriores

15. En una bomba como las estudiadas en prácticas, del tipo...

```
0x080486e8 <main+120>: call 0x8048524 <strcmp>
```

```
0x080486ed <main+125>: test %eax,%eax
```

```
0x080486ef <main+127>: je 0x80486f6 <main+134>
```

```
0x080486f1 <main+129>: call 0x8048604 <boom>
```

```
0x080486f6 <main+134>: ...
```

la contraseña es...

a. el valor que tenga `%eax`

b. el string almacenado a partir de donde apunta `%eax`

c. el entero almacenado a partir de donde apunta `%eax`

d. ninguna de las anteriores

16. El servidor de SWAD tiene dos procesadores Xeon E5540 con 4 núcleos cada uno. Cada procesador tiene 4 caches L1 de instrucciones de 32 KB, 4 caches L1 de datos de 32 KB, 4 caches unificadas L2 de 256 KB y una cache unificada L3 de 8MB. 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 observarían saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

a. 4 / 32 / 512 estudiantes

b. 8 / 64 / 2048 estudiantes

c. 16 / 32 / 64 estudiantes

d. 32 / 256 / 8192 estudiantes

17. En la práctica de la cache, el código de `line.cc` incluye la sentencia

```
for (unsigned line=1;line<=MAXLINE;
 line<=1) { ... }
```

¿Qué objetivo tiene la expresión `line<=1`?

- a. Salir del bucle si el tamaño de línea se volviera menor o igual que 1 para algún elemento del vector
- b. Duplicar el tamaño del salto en los accesos al vector respecto a la iteración anterior
- c. Volver al principio del vector cuando el índice exceda la longitud del vector
- d. Sacar un uno (1) por el stream line

18. Sea un computador de 32 bits con una memoria caché L1 para datos de 32 KB y líneas de 64 bytes asociativa por conjuntos de 2 vías. Dado el siguiente fragmento de código:

```
int v[262144];
for (i = 0; i < 262144; i += 2)
 v[i] = 9;
```

¿Cuál será la tasa de fallos aproximada que se obtiene en la primera ejecución del bucle anterior?

- a. 0 (ningún fallo)
- b. 1/2 (mitad aciertos, mitad fallos)
- c. 1/8 (un fallo por cada 8 accesos)
- d. 1 (todo son fallos)

19. Abajo se ofrece el listado de una función para multiplicar matrices  $C = A \times B$ .

```
void mult_matr(float A[N][N],
 float B[N][N], float C[N][N]) {
 /* Se asume valor inicial C = {0,0...} */
 int i,j,k;
 for (i=0; i<N; i++)
 for (j=0; j<N; j++)
 for (k=0; k<N; k++)
 C[i][j] += A[i][k] * B[k][j];
}
```

Suponer que:

- El computador tiene una cache de datos de 8 MB, 16-vías, líneas de 64 bytes.
- N es grande, una fila o columna no cabe completa en cache.
- El tamaño de los tipos de datos es como en IA32.
- El compilador optimiza el acceso a  $C[i][j]$  en un registro.

Aproximadamente, ¿qué tasa de fallos se podría esperar de esta función para valores grandes de N?

- a. 1/16
- b. 1/8
- c. 1/4
- d. 1/2

20. Con los mismos supuestos, imaginar que se modifica la última sentencia (el cuerpo anidado) por esta otra

$$C[i][j] += A[i][k] * B[j][k];$$

de manera que se calcule  $C = A \times B'$  ( $A$  por traspuesta de  $B$ ). Aproximadamente, ¿qué tasa de fallos se podría esperar de esta nueva función para valores grandes de N?

- a. 1/16
- b. 1/8
- c. 1/4
- d. 1/2

Nombre:

DNI:

Grupo:

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 4/20 si es correcta, 0 si está en blanco o claramente tachada, -4/60 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

- El switch de gcc para que únicamente compile de lenguaje C a ensamblador, y no realice ningún paso adicional (ensamblar, enlazar, etc), es...
  - c
  - S**
  - o
  - g
- Los switches --32 y --64 para trabajar en 32bit/64bit corresponden a la herramienta...
  - gcc
  - as**
  - ld
  - nm
- El switch -l para indicar librerías **\*NO\*** funciona con la herramienta...
  - gcc
  - as**
  - ld
  - no se puede marcar una y solo una de las anteriores
- ¿Cuál de las siguientes no es una sección de un fichero ELF?
  - .text
  - .static**
  - .data
  - .bss
- ¿Cuál de los siguientes contenidos no está incluido en un fichero ELF ejecutable?
  - código máquina
  - variables globales
  - pila del usuario**
  - tabla de símbolos
- En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor que se indica a continuación, ¿en qué caso ambos programas producen el mismo resultado?
  - 0x1111 1111**
  - 0x9999 9999
  - 0xAAAA AAAA
  - 0xFFFF FFFF
- En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x0400 0000, ¿en qué se diferencian los resultados de ambos programas?
  - no se diferencian**
  - en uno ocupa 32 bits, en otro 64 bits
  - en uno se interpreta como negativo, en otro como positivo
  - en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no

8. En la práctica "media" se suma una lista de 32 enteros de 4 B con signo para producir una media y un resto usando la instrucción IDIV. ¿Cuál de las siguientes afirmaciones es falsa?

- a. IDIV produce el mismo cociente que el operador / en lenguaje C
- b. IDIV produce el mismo resto que el operador % en lenguaje C
- c. La media se redondea al entero más próximo
- d. El resto siempre tiene el mismo signo que la suma

9. 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
cld
add %eax, %edi
adc %edx, %ebp
jnc nocarry
inc %edx
```

nocarry:

```
inc %esi
cmp %esi,%ecx
jne bucle
```

Estando bien programado todo lo demás, este código

- a. produce siempre el resultado correcto
- b. fallaría con lista: .int 0,1,2,3
- c. fallaría con lista: .int -1,-2,-4,-8
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

10. Alguno de los siguientes no es un nombre de registro en una máquina IA-32 en modo 32 bits

- a. ebp
- b. ax
- c. dh
- d. sil

11. Alguno de los siguientes no es un nombre de registro en una máquina x86-64 en modo 64 bits

- a. r8d
- b. r12w
- c. sih
- d. spl

12. Para comprobar si el entero almacenado en EAX es cero (y posiblemente saltar a continuación usando JZ/JNZ), gcc genera el código

- a. `cmp %eax, $0`
- b. `test %eax`
- c. `cmp %eax`
- d. `test %eax, %eax`

13. La práctica "paridad" debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity3:

```
int parity3(unsigned* array,
 int len){
 int i,res=0,val;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 val=0;
 do {
 val += x;
 x >>= 1;
 } while (x);
 val &= 0x1;
 res+=val;
 }
 return res;
}
```

Esta función parity3:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos



14. Un estudiante entrega la siguiente versión de parity4:

```
int parity4(unsigned* array,
 int len){
 int val,i,res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 val=0;
 asm("\n"
"ini3: \n\t"
"xor %[x],%[v] \n\t"
"shr %[x] \n\t"
"test %[x], %[x]\n\t"
"jne ini3 \n\t"
":[v]"+r" (val)
:[x] "r" (x)
);
 val = val & 0x1;
 res+=val;
 }
 return res;
}
```

Esta función parity4:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

15. La sentencia asm() del listado anterior tiene las siguientes restricciones

- a. ninguna
- b. arquitectura de 32 bits
- c. dos entradas y una salida
- d. un registro y dos sobrescritos (clobber)

16. Un estudiante entrega la siguiente versión de parity5:

```
int parity5(unsigned* array,
 int len){
 int i,j,res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 for (j=sizeof(unsigned)*4;
```

```
 j>0; j=j/2){
 x^=x>>j;
 }
 x = x & 0x1;
 res+=x;
 }
 return res;
}
```

Esta función parity5:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

17. Un estudiante entrega la siguiente versión de parity6:

```
int parity6(unsigned* array,
 int len){
 int i,j,res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x=array[i];
 asm("\n"
"mov %[x],%%edx \n\t"
"shr $16, %%edx \n\t"
"xor %%edx,%[x] \n\t"
"mov %[x],%%edx \n\t"
"mov %%dh, %%dl \n\t"
"xor %%edx, %[x]\n\t"
"setpo %%cl \n\t"
"movzx %%cl, %[x]
:[x] "+r" (x)
:
:"edx","ecx"
);
 res+=x;
 }
 return res;
}
```

Esta función parity6:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}



- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos
- 

18. La sentencia `asm()` del listado anterior tiene las siguientes restricciones

- a. ninguna
  - b. arquitectura de 32 bits
  - c. dos entradas y una salida
  - d. un registro y dos sobrescritos (clobber)
- 

19. En el programa "size" de la práctica de la cache, si el primer escalón pasa de tiempo = 1 para todos los tamaños de vector menores o iguales que 32 KB a tiempo = 3 para los tamaños 64 KB y 128 KB, podemos asegurar que:

- a. la cache L1 es al menos tres veces más rápida que la cache L2.
  - b. la cache L1 es como mucho tres veces más rápida que la cache L2.
  - c. la cache L2 es al menos el doble de rápida que la memoria principal.
  - d. la cache L2 es como mucho el doble de rápida que la memoria principal.
- 

20. El código del programa "size" de la práctica de la cache accede al vector saltando...

- a. de byte en byte.
  - b. de 64 en 64 bytes.
  - c. de 1 KB en 1 KB.
  - d. de 64 KB en 64 KB.
-

|                |               |
|----------------|---------------|
| <b>Nombre:</b> |               |
| <b>DNI:</b>    | <b>Grupo:</b> |

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

1. ¿Qué hace gcc -O0?

a. Compilar sin optimización

b. Compilar .c→.o (fuente C a objeto)

c. Compilar .s→.o (fuente ASM a objeto)

d. Compilar .c→.s (C→ASM sin generar objeto)

2. ¿Qué modificador (switch) de gcc hace falta para compilar una aplicación de 32 bits en un sistema de 64 bits?

a. -m32

b. -m64

c. -march32

d. -march64

3. La etiqueta del punto de entrada a un programa ensamblador en el entorno de las prácticas 1 a 4 (GNU/as Linux x86) es:

a. \_main

b. .L0

c. \_start

d. \_init

4. La siguiente línea en la sección de datos de un programa en ensamblador de IA32

```
result: .int 0,0
```

a. Reserva espacio para un único entero, inicializado a 0,0

b. Reserva espacio para un entero, inicializado a 0, seguido de un dato de tamaño indefinido, también inicializado a 0

c. Reserva espacio para dos enteros, inicializados ambos a 0

d. Reserva espacio para un único entero, inicializado a 0, en la posición de memoria 0

5. El volcado mostrado abajo se ha obtenido con el comando...

```
00000000 <main>:
```

```
0: 55 push %ebp
1: 89 e5 mov %esp,%ebp
3: 83 e4 f0 and $-16,%esp
6: 83 ec 10 sub $0x10,%esp
9: c7 44 24 04 03 movl $3, 4(%esp)
e: 00 00 00
11: c7 04 24 01 00 movl $0x1, (%esp)
16: 00 00
18: e8 fc ff ff ff call <main+0x19>
1d: c9 leave
1e: c3 ret
```

a. objdump -h p

b. objdump -d p

c. objdump -d p2.o

d. objdump -t p2.o

6. En la práctica "media" se desea invocar desde lenguaje ensamblador la función printf() de libC. Eso implica que este programa, como todo programa que use libC,

a. es obligatorio que contenga main (y entonces es más cómodo usar gcc para enlazar)

b. es obligatorio enlazarlo usando gcc (y entonces es más cómodo que contenga main)

c. es ventajoso para ensamblarlo que contenga main, y entonces es conveniente enlazarlo usando gcc (aunque ambas cosas son opcionales)

- d. es ventajoso para enlazarlo usar gcc, y entonces es conveniente que contenga main (aunque ambas cosas son opcionales)

7. En la práctica "media" se pide sumar una lista de 32 enteros SIN signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando perder acarreos. Un estudiante entrega la siguiente versión

```
$lista en EBX, longlista en ECX
suma:
 mov $0, %eax
 mov $0, %edx
 mov $0, %esi
bucle:
 add (%ebx,%edx,4), %eax
 jnc seguir
 inc %edx
seguir:
 inc %esi
 cmp %esi,%ecx
 jne bucle
 ret
```

Esta función presenta una única diferencia frente a la solución recomendada en clase, relativa al indexado en el array.

Esta función suma:

- a. Produce siempre el resultado correcto
- b. Fallaría con **lista: .int 1,1,1,1, 1,1,1,1, ...**
- c. Fallaría con **lista: .int 1,2,3,4, 1,2,3,4, ...**
- d. No es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

8. En la práctica "media" se pide sumar una lista de 32 enteros SIN signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando perder acarreos. De entre los siguientes, ¿cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32 bits (sin signo)? Se usa notación decimal y espacios como separadores de millares/millones/etc.

- a. 10 000 000
- b. 100 000 000
- c. 1 000 000 000
- d. 10 000 000 000

9. En la práctica "media" se pide sumar una lista de 32 enteros CON signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando desbordamiento. De entre los

siguientes, ¿cuál es el valor negativo más pequeño (en valor absoluto) que repetido en toda la lista causaría desbordamiento con 32 bits (en complemento a 2)? Se usa notación decimal y espacios como separadores de millares/millones/etc.

- a. -10 000 000
- b. -100 000 000
- c. -1 000 000 000
- d. -10 000 000 000

10. ¿Cuál es el popcount (peso Hamming, nº de bits activados) del número 19?

- a. 2
- b. 3
- c. 4
- d. 5

11. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:

```
int popcount3(unsigned* array,
 int len){
 int i, res = 0;
 unsigned x;
 for(i = 0; i < len; i++) {
 x = array[i];
 asm("ini3: \n"
 "shr %[x] \n"
 "adc $0, %[r] \n"
 "add $0, %[x] \n"
 "jne ini3 \n"
 : [r] "+r" (res)
 : [x] "r" (x));
 }
 return res;
}
```

Esta función sólo tiene una diferencia con la versión "oficial" recomendada en clase. En concreto, una instrucción máquina en la sección **asm()** es distinta.

Esta función popcount3:

- a. produce siempre el resultado correcto
- b. fallaría con **array={0,1,2,3}**
- c. fallaría con **array={1,2,4,8}**
- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

12. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:

```
int popcount3(int* array, int len){
 long val = 0;
 int i;
 unsigned x;
 for (i=0; i<len; i++){
 x= array[i];
 do{
 val += x & 0x1;
 x >>= 1;
 } while (x);
 val += (val >> 16);
 val += (val >> 8);
 }
 return val & 0xFF;
}
```

Esta función es una mezcla inexplicada de las versiones "oficiales" de **popcount2** y **popcount4**, incluyendo diferencias en 2 tipos de datos, la ausencia de la variable **res** y la diferente posición de la máscara **0xFF**.

Esta función popcount3:

- a. produce siempre el resultado correcto
- b. fallaría con **array={0,1,2,3}**
- c. fallaría con **array={1,2,4,8}**
- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

13. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount4:

```
int popcount4(unsigned* array,
 int len){
 int i,j;
 unsigned x;
 int result = 0;
 for(i=0;i<len;i++){
 x=array[i];
 for(j=0;j<8;j++){
 result += x & 0x01010101;
 x>>=1;
 }
 }
 result += (result >> 16);
 result += (result >> 8);
 return result & 0xFF;
}
```

}

Esta función presenta varias diferencias con la versión "oficial" recomendada en clase, incluyendo la ausencia de una variable auxiliar **val** y la diferente posición de los desplazamientos **>>** y máscara **0xFF**.

Esta función popcount4:

- a. produce siempre el resultado correcto
- b. fallaría con **array={0,1,2,3}**
- c. fallaría con **array={1,2,4,8}**
- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

14. La práctica "parity" debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity5:

```
int parity5(unsigned * array,
 int len){
 int i,j, result = 0;
 unsigned x;
 for(i = 0; i<len; i++){
 x=array[i];
 for(j=1; j<8*sizeof(int); j*=2)
 x ^= x >> j;
 result += x & 0x1;
 }
 return result;
}
```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase, en las condiciones del bucle **for** interno.

Esta función parity5:

- a. Produce siempre el resultado correcto
- b. Fallaría con **array={0,1,2,3}**
- c. Fallaría con **array={1,2,4,8}**
- d. No es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

15. La función **gettimeofday()** en la práctica de la "bomba digital" se utiliza para

- a. Para comparar las duraciones de las distintas soluciones del programa
- b. Para imprimir la fecha y hora
- c. Para cifrar la contraseña en función de la hora actual
- d. Para cronometrar lo que tarda el usuario en introducir la contraseña

16. Un fragmento de una “bomba” desensamblada es:

```
0x0804873f: call 0x8048504 <scanf>
0x08048744: mov 0x24(%esp),%edx
0x08048748: mov 0x804a044,%eax
0x0804874d: cmp %eax,%edx
0x0804874f: je 0x8048756 <main+230>
0x08048751: call 0x8048604 <boom>
0x08048756: ...
```

La contraseña/clave en este caso es...

- a. el string almacenado a partir de la posición de memoria 0x804a044
- b. el string almacenado a partir de la posición de memoria 0x24(%esp)
- c. el entero almacenado a partir de la posición de memoria 0x804a044
- d. el entero 0x804a044

17. Una de las “bombas” utiliza el siguiente bucle para cifrar la cadena con la contraseña introducida por el usuario:

```
80485bb: rolb $0x4, (%eax)
80485be: add $0x1, %eax
80485c1: cmp %edx, %eax
80485c3: jne 80485bb <encrypt+0x20>
```

La instrucción **rolb** rota el byte destino hacia la izquierda tantos bits como indica el operando fuente. Si inicialmente `eax` apunta a la cadena del usuario, que se compara con otra cadena “\x16\x26\x27\x16\x36\x16\x46\x16\x26\x27\x16”, almacenada en el código, la contraseña es:

- a. “\x61\x62\x72\x61\x63\x61\x64\x61\x62\x72\x61” (“abracadabra”)
- b. “\x61\x72\x62\x61\x64\x61\x63\x61\x72\x62\x61” (“arbadacarba”)
- c. “\x63\x61\x64\x61\x62\x72\x61\x61\x62\x72\x61” (“cadabraabra”)
- d. “\x61\x62\x72\x61\x61\x62\x72\x61\x63\x61\x64” (“abraabracad”)

18. Una de las “bombas” utiliza el siguiente código para cifrar la clave numérica introducida por el usuario y ahora almacenada en `eax`:

```
804870d: xor $0xffff, %eax
8048712: mov $0x2, %ecx
8048717: cltd
8048718: idiv %ecx
804871a: cmp %eax, 0x804a034
```

Si el entero almacenado a partir de 0x804a034 es 0x7ff, la clave numérica puede ser:

- a. 0x10094F97 (269045655)
- b. 0xffff (4095)
- c. 0x7ff (2047)
- d. 1

19. En el programa `line.cc` de la práctica 5, si para cada tamaño de línea (`line`) recorremos una única vez el vector, la gráfica resultante es decreciente porque:

- a. Cada vez que `line` aumenta al doble, el número de aciertos por localidad temporal aumenta, porque ya habíamos accedido a cada posición  $i$  del vector cuando lo recorrimos en el punto anterior del eje  $X$ .
- b. Cada vez que `line` aumenta al doble, el número de aciertos por localidad espacial aumenta, porque ya habíamos accedido a cada posición  $i-1$  del vector cuando lo recorrimos en el punto anterior del eje  $X$ .
- c. Cada vez que `line` aumenta al doble, se accede con éxito a más posiciones del vector en niveles de la jerarquía de memoria más rápidos.
- d. Cada vez que `line` aumenta al doble, realizamos la mitad de accesos al vector que para el valor anterior.

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

- a. La diferencia de velocidades entre  $L2$  y  $L3$  es mayor que la diferencia de velocidades entre  $L1$  y  $L2$ .
- b. 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.
- c. 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.
- d. 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.



|         |        |
|---------|--------|
| Nombre: |        |
| DNI:    | Grupo: |

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

1. ¿Cuál de las siguientes instrucciones máquina copia en EAX la dirección efectiva resultante de la operación  $EDX*4 + EBX$ ?

- a. `leal 4(%edx, %edx), %eax`
- b. `leal (%ebx, %edx, 4), %eax`
- c. `movl 4(%edx, %edx), %eax`
- d. `movl (%ebx, %edx, 4), %eax`

2. ¿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  $EDX*4 + EBX$ ?

- a. `leal 4(%edx, %edx), %eax`
- b. `leal (%ebx, %edx, 4), %eax`
- c. `movl 4(%edx, %edx), %eax`
- d. `movl (%ebx, %edx, 4), %eax`

3. Los switches `-m elf_i386` y `-m elf_x86_64` para trabajar en 32 bits / 64 bits corresponden a la herramienta...

- a. gcc
- b. as
- c. ld
- d. nm

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

- a. Pone CF=1
- b. Pone CF=0
- c. Cambia CF
- d. No cambia CF

5. 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)`

6. 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 `movl longlista, %ecx` copia el siguiente valor:



- a. 4
- b. 8
- c. 16
- d. 32

7. 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"
```

y suponiendo que hemos llamado a una función *suma* que devuelve un número de 64 bits en la pareja EDX:EAX, las instrucciones que copian ese número en *resultado* son:

- a. `movl %eax, resultado`  
`movl %edx, resultado+4`
- b. `movl (%eax), resultado`  
`movl (%edx), resultado+4`
- c. `movl %eax, resultado+4`  
`movl %edx, resultado`
- d. `movl (%eax), resultado+4`  
`movl (%edx), resultado`

8. 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 llamada correcta a *printf* será:

- a. `push resultado+4`  
`push resultado`  
`push $formato`  
`call printf`  
`add $12, %esp`
- b. `push resultado+4`  
`push resultado`  
`push resultado+4`  
`push resultado`

```
push $formato
call printf
add $20, %esp
```

- c. `push resultado`  
`push resultado+4`  
`push $formato`  
`call printf`  
`add $12, %esp`
- d. `push resultado`  
`push resultado+4`  
`push resultado`  
`push resultado+4`  
`push $formato`  
`call printf`  
`add $20, %esp`

9. En la práctica “media” se pide sumar una lista de enteros **\*con\*** signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el menor valor positivo que repetido en los **\*dos\*** primeros elementos de la lista causaría overflow con 32 bits al realizar la suma de **\*esos dos\*** primeros elementos de la lista?

- a. 0x0400 0000
- b. 0x0800 0000
- c. 0x4000 0000
- d. 0x8000 0000

10. ¿Cuál de las siguientes afirmaciones es cierta respecto al lenguaje C?

- a. En lenguaje C, al llamar a una subrutina o función se introducen los parámetros en la pila y después se realiza una llamada a la subrutina
- b. Los parámetros se introducen en la pila en el orden en el que aparecen en la llamada de C, es decir, empezando

por el primero y acabando por el último

- c. Antes de volver de la rutina llamada, el programa en C se encarga de quitar de la pila los parámetros de llamada realizando varios pop
  - d. Pasar a una función un puntero a una variable se traduce en introducir en la pila el valor de la variable
- 

11. ¿Cuál de los siguientes registros tiene que ser salvaguardado (si va a modificarse) dentro de una subrutina según la convención cdecl para IA32?

- a. eax
  - b. ebx
  - c. ecx
  - d. edx
- 

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

- a. ecx
  - b. edx
  - c. esi
  - d. edi
- 

13. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount3:

```
int popcount3(unsigned* array,
 int len){
 int i, res = 0;
 unsigned x;
 for(i=0; i<len; i++) {
 x = array[i];
 asm("ini3: \n"
 "shr %[x] \n"
 "adc $0, %[r] \n"
 "add $0, %[x] \n"
 "jne ini3 \n"
```

```
: [r] "+r" (res)
: [x] "r" (x));
}
return res;
}
```

Esta función produce siempre el resultado correcto, a pesar de que una instrucción máquina en la sección **asm()** es distinta a la que se esperaba después de haber estudiado `pcount_r` en teoría. La instrucción distinta también se podría haber cambiado por...

- a. `sar %[x]`
  - b. `adc $0, %[x]`
  - c. `test %[x], %[x]`
  - d. `cmp %[x], %[r]`
- 

14. En la práctica de la bomba necesitamos estudiar el código máquina de la bomba del compañero. A veces dicho código no se visualiza directamente en el depurador ddd, y algunas de las técnicas que se pueden probar para conseguir visualizarlo son... (marcar la opción **\*falsa\***)

- a. comprobar que está activado el panel *View → Machine Code Window*
  - b. escribir `info line main` en el panel de línea de comandos gdb
  - c. recompilar con información de depuración, por si se nos había olvidado, ya que sin `-g` el ejecutable no contiene información de depuración
  - d. asegurarse de que se ha escrito correctamente el nombre del ejecutable
- 

15. En la práctica de la bomba, el primer ejercicio consistía en “saltarse” las “explosiones”, para lo cual se puede utilizar...



a. objdump o gdb

b. gdb o ddd

c. ddd o hexedit

d. hexedit u objdump

---

16. En la práctica de la bomba, el segundo ejercicio consistía en crear un ejecutable sin “explosiones”, para lo cual se puede utilizar...

a. objdump o gdb

b. gdb o ddd

c. ddd o hexedit

d. hexedit u objdump

---

17. En la práctica de la bomba, el tercer ejercicio consistía en usar un editor hexadecimal para crear un ejecutable sin “explosiones”. Para saber qué contenidos del fichero hay que modificar, se puede utilizar... (marcar la opción falsa)

a. objdump

b. gdb

c. ddd

d. hexedit

---

18. Suponer una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponer que para la siguiente secuencia de direcciones enviadas a la cache: 0, 2, 4, 8, 16, 32, la tasa de acierto es 0,33. ¿Cuál es el tamaño de bloque de la cache?

a. 4 bytes

b. 8 bytes

c. 16 bytes

d. Ninguno de los anteriores

---

19. Abajo se ofrece el listado de una función para multiplicar matrices  $C = A \times B$ .

```
void mult_matr(float A[N][N],
float B[N][N],float C[N][N]){
/*Asume val.ini. C={0,0...}*/
int i,j,k;
for (i=0; i<N; i++)
 for (j=0; j<N; j++)
 for (k=0; k<N; k++)
 C[i][j]+= A[i][k]*B[k][j];
}
```

Suponer que:

- El computador tiene una cache de datos de 8 MB, 16-vías, líneas de 64 bytes.

- N es grande, una fila o columna no cabe completa en cache.

- El tamaño de los tipos de datos es como en IA32.

- El compilador optimiza el acceso a  $C[i][j]$  en un registro.

Aproximadamente, ¿qué tasa de fallos se podría esperar de esta función para valores grandes de N?

a. 1/2

b. 1/4

c. 1/8

d. 1/16

---

20. Sea un computador de 32 bits con una memoria cache L1 para datos de 32 KB y líneas de 64 bytes asociativa por conjuntos de 2 vías. Dado el siguiente fragmento de código:

```
int v[262144];
for (i=0; i<262144; i+=8)
 v[i] = 9;
```

¿Cuál será la tasa de fallos aproximada que se obtiene en la ejecución del bucle anterior?

a. 0 (ningún fallo)

b. 1/2 (mitad aciertos, mitad fallos)

c. 1/8 (un fallo por cada 8 accesos)

d. 1 (todo son fallos)

---

|                |               |
|----------------|---------------|
| <b>Nombre:</b> |               |
| <b>DNI:</b>    | <b>Grupo:</b> |

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

- ¿Cuál de los siguientes fragmentos es correcto para comenzar un programa en ensamblador que conste de un solo archivo .s?
  - .text  
\_start:
  - .text  
.local \_start  
\_start:
  - .text  
.start \_global  
\_start:
  - .text  
.global \_start  
\_start:
- Suponga una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponga que para la siguiente secuencia de direcciones enviadas a la cache: 1, 2, 4, 8, 16, 32, la tasa de acierto es 0,333. ¿Cuál es el tamaño de línea de la cache?
  - 4 bytes
  - 8 bytes
  - 16 bytes
  - 32 bytes
- En la convención cdecl estándar para arquitecturas x86 de 32 bits, cuál de las siguientes afirmaciones es cierta:
  - Los 6 primeros parámetros se pasan a través de registros
  - Solamente es necesario salvar el registro EAX
  - Los registros EBX, ESI y EDI son salva - invocante
  - Ninguna de las anteriores es cierta
- ¿Cuál es el popcount (peso Hamming, nº de bits activados) del número 0x10101010?
  - 4
  - 8
  - 16
  - 32
- Compilar de fuente C a ejecutable usando sólo as y ld, sin gcc...
  - Se puede, repartiendo entre as y ld los modificadores (switches) que corresponda
  - Basta usar ld, con los modificadores de gcc que corresponda, y añadiéndole el runtime de C
  - Se puede, repartiendo modificadores entre as y ld, y añadiendo al comando ld el runtime de C
  - No se puede
- La función gettimeofday() en la práctica de popcount y parity se utiliza para
  - Comparar las duraciones de las distintas soluciones del programa
  - Imprimir la fecha y hora
  - Cifrar el código en función de la hora actual
  - Cronometrar lo que tarda el usuario en pulsar una tecla
- ¿Cuál de los siguientes registros tiene que ser salvaguardado (si va a modificarse) dentro de

una subrutina según la convención cdecl para IA32?

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

8. ¿Qué hace `gcc -O1`?

- a. Compilar `.s`→`.o` (fuente ASM a objeto)
- b. Compilar con optimización
- c. Compilar `.c`→`.o` (fuente C a objeto)
- d. Compilar `.c`→`.s` (C→ASM sin generar objeto)

9. Dada la siguiente definición de datos:

```
lista: .int 0x10000000, 0x50000000,
 0x10000000, 0x20000000
longlista: .int (.-lista)/4
resultado: .quad 0x123456789ABCDEF
formato: .ascii "%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)`

10. En la práctica "media" se programa la suma de una lista de 32 enteros de 4 B para producir un resultado de 8 B, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x8000 0000, ¿en qué se diferencian los resultados de ambos programas?

- a. no se diferencian
- b. en uno ocupa 32 bits, en otro 64 bits
- c. en uno los 16 bits superiores son 0xFFFF, en el otro no
- d. en uno los 16 bits inferiores son 0x0000, en el otro no

11. En la práctica "media" se pide sumar una lista de 32 enteros CON signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando desbordamiento. Un estudiante entrega un programa que se diferencia de la versión recomendada en el siguiente bucle, en particular en la instrucción `adc`.

```
bucle:
 mov (%ebx,%esi,4), %eax
 cltd
 add %eax, %edi
 adc %eax, %ebp
```

```
inc %esi
cmp %esi, %ecx
jne bucle
```

Esta versión de la suma CON signo

- a. produce siempre el resultado correcto
- b. fallaría con lista: `.int 1, 1, 1, 1, ...`
- c. fallaría con lista: `.int -1,-1,-1,-1, ...`
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

12. 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
 jne bucle
```

Este código es una mezcla de las soluciones recomendadas para suma sin signo y para suma con signo. Estando bien programado todo lo demás, este código

- a. produce siempre el resultado correcto
- b. fallaría con lista: `.int 0,1,2,3`
- c. fallaría con lista: `.int -1,-2,-4,-8`
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

13. ¿Cuál es el popcount (peso Hamming, nº de bits activados) de una lista de N números inicializada con los valores 0..N-1?

- a.  $(N-1)*N/2$
- b.  $N*(N+1)/2$
- c. si N es par,  $N*(N/2)$
- d. si N es potencia de 2,  $\log_2(N)*N/2$

14. ¿Cuál es la paridad (XOR "lateral" de todos los bits) del número 199?

- a. 0
- b. 1
- c. 2
- d. 4

15. Comparando los popcounts (pop(199) vs. pop(99)) y paridades (par(199) vs. par(99)) de los números 199 y 99, se verifica que

- a.  $\text{pop}(199) > \text{pop}(99)$
- b.  $\text{par}(199) < \text{par}(99)$
- c.  $\text{pop}(199) < \text{par}(99)$
- d.  $\text{par}(199) > \text{pop}(99)$

16. ¿Cuál es la suma de paridades (suma de los XOR "laterales" de los bits de cada número) de una lista de N números inicializada con los valores 0..N-1?

- a.  $(N-1)*N/2$
- b.  $N*(N+1)/2$
- c. si N es par,  $N/2$
- d. si N es potencia de 2,  $(\log_2(N)*N)/2$

17. En la práctica "popcount/paridad", para cronometrar sistemáticamente las diversas versiones necesitamos una función crono() a la que se le pueda pasar como argumento cuál versión queremos cronometrar. En lenguaje C esto se puede hacer con punteros a funciones. Sabiendo que todas las versiones devuelven un valor entero, el prototipo de la función crono() debería ser:

- a. `void crono( int * func (), char* msg);`
- b. `void crono( int (* func)(), char* msg);`
- c. `void crono((int *)func (), char* msg);`
- d. `void crono( int * func , char* msg);`

18. Para corregir la práctica "bomba digital", un profesor dispone de 26 ejecutables (y la lista de claves correspondientes) para asignar en el Laboratorio una bomba distinta a cada estudiante. Cuando un estudiante diga que la ha resuelto el profesor exigirá ver al estudiante ejecutando la bomba y tecleando la contraseña y el pin correctos para comprobar que no explota, para así anotarla como resuelta y que le puntúe al estudiante.

Un estudiante (usando ordenador del Laboratorio con Ubuntu 10.04) dice que ha resuelto su bomba, y cuando el profesor pide que se tecleen las claves, el ddd se "bloquea" y le empieza a "parpadear" al estudiante. Para poder puntuar, lo más recomendable para el estudiante sería...

- a. teclear las claves (contraseña y pin), aunque ddd esté "bloqueado" y "parpadeando"

b. probar en orden los remedios básicos: pulsar <Ctrl>-C varias veces, pulsar <Enter> repetidamente, comprobar que está seleccionada "Machine Code Window", teclear "info line main", y si todo falla, ejecutar "`rm -rf ~/ .ddd`"

c. matar la ventana ddd, abrir un terminal con un shell, ejecutar la bomba desde línea de comandos y teclear las claves

d. reinstalar un paquete ddd más actualizado usando "sudo apt-get install"

19. Suponer el mismo contexto de la pregunta anterior, donde el profesor tiene una lista de las 26+26 claves (contraseñas y pines)

Un estudiante dice que ha resuelto su bomba, y cuando el profesor pide que se tecleen las claves, la contraseña coincide con la que tiene anotada el profesor, pero el pin no, y de todas formas la bomba no explota. Debería hacerse lo siguiente:

- a. la bomba tiene que puntuarle al estudiante porque no ha explotado
- b. el profesor puede pedirle que vuelva a descargar la bomba original e intente repetir con ese ejecutable las claves que acaba de teclear
- c. la bomba debe marcarse como inválida y no hay que hacer más comprobaciones
- d. no puede suceder lo que dice el enunciado, y en ningún caso el profesor tiene derecho a hacer comprobaciones adicionales como pedir que se vuelva a descargar la bomba

20. En la práctica de la cache, el código de "line.cc" incluye la sentencia

```
for (unsigned long long line=1;
 line<=LINE; line<=1) { ... }
```

¿Qué objetivo tiene la expresión `line<=1`?

- a. salir del bucle si el tamaño de línea se volviera menor o igual que 1 para algún elemento del vector
- b. duplicar el tamaño del salto en los accesos al vector respecto a la iteración anterior
- c. volver al principio del vector cuando el índice exceda la longitud del vector
- d. sacar un uno (1) por el stream line

Nombre:

DNI:

Grupo:

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

1. El punto de entrada de un programa ensamblador en GNU/as Linux x86 se llama

- a. main
- b. begin
- c. **\_start**
- d. \_init

2. En un programa en ensamblador queremos crear espacio para una variable entera var inicializada a 1. La línea que hemos de escribir en la sección de datos es:

- a. .int var 1
- b. **var: .int 1**
- c. .int: var 1
- d. int var 1

3. En la práctica "media" se pide sumar una lista de 32 enteros SIN signo de 32 bits en una plataforma de 32 bits sin perder precisión, esto es, evitando perder acarreos. Un estudiante entrega la siguiente versión

```
$lista en EBX, longlista en ECX
suma:
 mov $0, %eax
 mov $0, %edx
 mov $0, %esi

bucle:
 add (%ebx,%esi,4), %eax
 jne nocarry
 inc %edx

nocarry:
 inc %esi
 cmp %esi,%ecx
 jne bucle
 ret
```

Esta función presenta una única diferencia frente a la solución recomendada en clase, relativa al salto condicional.

Esta función suma:

- a. produce siempre el resultado correcto
- b. fallaría con lista: .int 1,1,1,1, 1,1,1,1, ...
- c. fallaría con lista: .int 1,2,3,4, 1,2,3,4, ...
- d. **no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos**

4. En la misma práctica "media" un estudiante entrega la siguiente versión de suma sin signo:

```
main: .global main
 ...
 call suma
 ...
 mov $1, %eax
 mov $0, %ebx
 int $0x80

suma:
 ...
bucle:
 ...
nocarry:
 inc %esi
 cmp %esi,%ecx
 jne bucle
```

Notar que falta la instrucción ret final. Al desensamblar el código ejecutable se obtiene

```
08048445 <nocarry>:
8048445: 46 inc %esi
8048446: 39 f1 cmp %esi,%ecx
8048448: 75 f5 jne 804843f <bucle>
804844a: 90 nop
804844b: 90 nop
..... 90 nop
804844f: 90 nop
```



```

08048450 <__libc_csu_fini>:
8048450: 55 push %ebp
8048451: 89 e5 mov %esp,%ebp
8048453: 5d pop %ebp
8048454: c3 ret

```

Este programa:

- está correctamente diseñado, la instrucción ret final es optativa, y no es concebible que quitar el ret cause algún error
- produce un error "Segmentation fault" cuando empieza a acceder a memoria que no le corresponde (EIP)
- funciona bien, pero si pusiéramos en el código fuente primero la definición de suma y luego la de main, el ejecutable terminaría accediendo a memoria que no le corresponde (ESP) y hará "Segmentation fault"
- no se puede marcar ninguna de las opciones anteriores

5. ¿Cuál de las siguientes sumas con signo produce desbordamiento con 32 bits?

- 0xFFFFFFFF + 0xFFFFFFFF
- 0x7FFFFFFFF + 0xFFFFFFFF
- 0x7FFFFFFFF + 0x000000001
- 0xFFFFFFFF + 0x000000001

6. En la práctica "media" se pide sumar una lista de 32 enteros \*con\* signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el menor valor positivo que repetido en toda la lista causaría overflow con 32bits?

- 0x0400 0000
- 0x0800 0000
- 0x4000 0000
- 0x8000 0000

7. En la práctica "media" se pide sumar una lista de 32 enteros \*con\* signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando overflow. ¿Cuál es el mayor valor negativo (menor en valor absoluto) que repetido en toda la lista causaría overflow con 32bits?

- 0xffff ffff
- 0xfc00 0000
- 0xfbff ffff
- 0xf000 0000

8. ¿Cuál es el popcount (peso Hamming, n° de bits activados) del número 29?

- 2
- 3
- 4
- 5

9. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount4:

```

int popcount4(unsigned* array, int len){
 int val = 0;
 int i, j;
 unsigned x;
 int res = 0;
 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 for (j=0; j<8; j++){
 val += x & 0x01010101;
 x >>= 1;
 }
 val += (val>>16);
 val += (val>>8);
 res += val;
 }
 return (res & 0xFF);
}

```

Esta función presenta varias diferencias con la versión "oficial" recomendada en clase, incluyendo la "doble inicialización" de val y la acumulación y retorno de res. Esta popcount4:

- produce siempre el resultado correcto
- fallaría con array={0,1,2,3}
- fallaría con array={1,2,4,8}
- no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

10. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount4:

```

int popcount4(unsigned* array, int len){
 int i, j;
 unsigned x;
 int result = 0;
 long val;
 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 for (j=0; j<8*sizeof(int);j++){
 val += x & 0x01010101;
 x >>= 1;
 }
 val += (val>>16);
 val += (val>>8);
 result += val & 0xFF;
 }
 return result;
}

```

```
}
```

Esta función presenta varias diferencias con la versión "oficial" recomendada en clase, incluyendo el tipo de val y las condiciones del bucle for.

Esta función popcount4:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}
- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

11. Comparando los popcounts (pop(129) vs. pop(29)) y paridades (par(129) vs. par(29)) de los números 129 y 29, se verifica que

- a.  $\text{pop}(129) > \text{pop}(29)$
- b.  $\text{par}(129) > \text{par}(29)$
- c.  $\text{pop}(129) < \text{par}(29)$
- d.  $\text{par}(129) < \text{pop}(29)$

12. La práctica "parity" debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. Un estudiante entrega la siguiente versión de parity4:

```
int parity4(unsigned* array, int len){
 int val;
 int i;
 unsigned x;
 int result = 0;
 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 asm("\n"
 "ini3: \n\t"
 "shr $0x1, %[x] \n\t"
 "adc $0x0, %[r] \n\t"
 "test %[x], %[x] \n\t"
 "jnz ini3 "
 : [r]" +r"(val)
 : [x]"r"(x)
);
 result += val & 0x1;
 }
 return result;
}
```

Esta función presenta una sentencia asm distinta de la versión "oficial" recomendada en clase. En concreto son distintas la etiqueta y las instrucciones adc/test.

Esta función parity4:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}

- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

13. En la misma práctica "parity" un estudiante entrega la siguiente versión de parity4:

```
int parity4(unsigned* array, int len){
 int i;
 unsigned x;
 int val, result = 0;
 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 asm("\n"
 "ini4: \n\t"
 "xor %[x], %[y] \n\t"
 "shr $1, %[x] \n\t"
 "cmpl $0, %[x] \n\t"
 "jnz ini4 \n\t"
 : [y] "+r"(val)
 : [x] "r"(x)
);
 result += val & 0x1;
 }
 return result;
}
```

Esta función presenta dos diferencias con la versión "oficial" recomendada en clase, relativas a la instrucción cmp y al nombre [y] escogido para la restricción val. Esta parity4:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}
- d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

14. En la misma práctica "parity" un estudiante entrega la siguiente versión de parity6:

```
int parity6(int *array, int len){
 int i, res=0;
 unsigned x;
 for (i = 0; i < len; i++){
 x = array[i];
 asm(
 "mov %[x], %%edx \n\t"
 "shr $16, %[x] \n\t"
 "xor %[x], %%edx \n\t"
 "xor %%dh, %%dl\n\t"
 "setpo %%dl \n\t"
 "movzx %%dl, %[x]\n\t"
 : [x] "+r"(x)
 :
 : "edx"
);
 res += (x & 0x1);
 }
 return res;
}
```

}

Esta función presenta dos diferencias con la versión "oficial" recomendada en clase, relativas al tipo del array y a la máscara y paréntesis usados al acumular.

Esta función parity6:

- a. produce siempre el resultado correcto
  - b. fallaría con array={0,1,2,3}
  - c. fallaría con array={1,2,4,8}
  - d. no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos
- 

15. En la práctica de la bomba, el primer ejercicio consistía en “saltarse” las “explosiones”, para lo cual se puede utilizar...

- a. objdump o gdb
  - b. gdb o ddd
  - c. ddd o hexedit
  - d. hexedit u objdump
- 

16. En la práctica de la bomba, el tercer ejercicio consistía en usar un editor hexadecimal para crear un ejecutable sin “explosiones”. Para saber qué contenidos del fichero hay que modificar, se puede utilizar... (marcar la opción **\*FALSA\***)

- a. objdump
  - b. gdb
  - c. ddd
  - d. hexedit
- 

17. Suponer una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponer que para la siguiente secuencia de direcciones enviadas a la cache: 0, 2, 4, 8, 16, 32, la tasa de acierto es 0.33. ¿Cuál es el tamaño de bloque de la cache?

- a. 4 bytes
  - b. 8 bytes
  - c. 16 bytes
  - d. Ninguno de los anteriores
- 

18. Sea un computador de 32 bits con una memoria cache L1 para datos de 32 KB y líneas de 64 bytes asociativa por conjuntos de 2 vías. Dado el siguiente fragmento de código:

```
int v[262144];
for (i = 0; i < 262144; i += 8)
 v[i] = 9;
```

¿Cuál será la tasa de fallos aproximada que se obtiene en la ejecución del bucle anterior?

- a. 0 (ningún fallo)
  - b. 1/2 (mitad aciertos, mitad fallos)
  - c. 1/8 (un fallo por cada 8 accesos)
  - d. 1 (todo son fallos)
- 

19. En la práctica de la cache, el código de size.cc accede al vector saltando de 64 en 64. ¿Por qué?

- a. Porque cada elemento del vector ocupa 64 bytes
  - b. Para recorrer el vector más rápidamente
  - c. Porque el tamaño de cache L1 de todos los procesadores actuales es de 64KB
  - d. Para anular los aciertos por localidad espacial, esto es, que sólo pueda haber aciertos por localidad temporal
- 

20. ¿En qué unidades se suelen medir las capacidades de almacenamiento de los niveles de cache L1, L2 y L3 de un microprocesador actual (2017-2018)?

- a. L1 en KB, L2 en KB o MB, L3 en MB.
  - b. L1 en MB, L2 en GB, L3 en GB o TB.
  - c. L1 en MB, L2 en MB, L3 en GB.
  - d. L1 en KB, L2 en MB, L3 en GB.
-



Nombre:

DNI:

Grupo:

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | c  |    | d  |    |    |

1. Habiendo definido en código fuente ASM  
`longsal: .quad .-saludo` justo detrás de un  
string `saludo` que ocupaba 28 bytes, si se  
comparan los comandos gdb siguientes: `x/1xg`  
`&longsal` frente a `print (long) &longsal`:

- ambos nos muestran la longitud del string  
(que es/vale/equivalente a 28)
- el primero (x) nos muestra la longitud, y el  
segundo (print) nos muestra otro valor distinto
- el segundo (print) nos muestra la longitud, y  
el primero (x) nos muestra otro valor distinto
- alguno o ambos contienen algún error  
gramatical (falta o sobra algún &, algún  
typecast (char\*) o (long), etc.)

2. En la práctica "media" se pide sumar una lista  
de 16 enteros SIN signo de 32 bits evitando  
acarreo. ¿Cuál es el menor valor que repetido  
en toda la lista causaría acarreo en 32 bits?

- 0xFFFF FFFF
- 0x7FFF FFFF
- 0x1000 0000
- 0x0FFF FFFF

3. En la práctica "media" se pide usar `adc` para  
sumar una lista de 16 enteros SIN signo de 32  
bits en dos registros de 32 bits mediante  
extensión con ceros. Un estudiante entrega la  
siguiente versión:

```
...
resultado: .quad 0
...
main: .global main
 mov $lista, %rbx
 mov $16, %ecx
 call suma
```

```
mov %eax, resultado
código para printf ...
código para _exit ...
```

```
suma:
 mov $0, %eax
 mov $0, %rdx
bucle:
 adc (%rbx, %rdx, 4), %eax
 inc %rdx
 cmp %rdx, %rcx
 jnle bucle
 ret
```

Este programa no usa la variable `longlista`,  
guarda el resultado con una instrucción MOV,  
usa como índice RDX, no usa la instrucción  
ADD, y usa JNLE para el salto condicional.

Al empezar un programa CF no está activado.  
Esta versión de la suma SIN signo mediante  
extensión con ceros da resultado correcto:

- con lista: `.int 0x10000000, ...` (16 elementos)
- con lista: `.int 200000000, ...` (16 elementos)
- con ambos ejemplos
- con ninguno de los dos ejemplos

4. En la práctica "media" se pide usar `cld/cdq`  
para sumar una lista de 16 enteros CON signo  
de 32 bits en dos registros de 32 bits mediante  
extensión de signo. Un estudiante entrega la  
siguiente versión:

```
...
main: .global main
 mov $lista, %rbx
 mov longlista, %ecx
 call suma
 mov %eax, resultado
 mov %edx, resultado+4
```

```

movq $formato, %rdi
movq resultado,%rsi
movq resultado,%rdx
movl $0,%eax
call printf
...

```

El programa produce la siguiente salida con el test #01 (16 elementos con valor -1):

```

__TEST01__ -----
resultado = -16 (sng)
 = 0x ffffffff00000000 (hex)
 = 0x 00000010 9f816d80

```

Recordar que todo el texto aparecía tal cual literalmente en el formato (ignorar la errata sng) y los números llevaban especificación de formato (%18ld, %18lx, etc). De esta versión de la suma CON signo mediante extensión de signo se puede afirmar que:

- al inicio de main EAX vale 0 y R8 contiene un valor inferior a 0x7ffffff00000000
  - al llamar a suma RBX contiene un valor inferior a 0x600000 y RCX vale 16
  - al llamar a printf, ECX vale 16 y R8 contiene un valor superior a 0x80000000
  - tras volver de printf RAX contiene un valor superior a 60 y RDI superior a 0x600000
5. En la práctica "media" se pide usar `cltq/cdqe` y `cqto/cqo` para hallar la media y resto de una lista de 16 enteros CON signo de 32 bits usando registros de 64 bits. Un estudiante entrega la siguiente versión:

```

...
media: .double 0
resto: .double 0
formatoq:
.ascii "media = %11d resto = %11d\n"
.asciz "\t = 0x %08x \t = 0x %08x\n"
...
mov $lista, %rbx
mov longlista, %ecx
call sumaq

mov $formatoq, %rdi
mov media,%rsi
mov resto,%rdx
mov $0,%eax
call printf

...
sumaq:
push %rdx
push %rsi
mov $0, %rax
mov $0, %rsi
mov $0, %rdx
mov $0, %r8

bucleq:
mov (%rbx,%rsi,4), %eax
cdqe # EAX -> RAX

```

```

add %rax, %r8
inc %rsi
cmp %rsi,%rcx
jne bucleq

mov %r8, %rax
cqo # RAX -> RDX:RAX
idivq %rsi
mov %rdx, %r10
mov %rdx, resto
mov %rax, media
pop %rdx
pop %rsi
ret

```

Este programa es muy diferente a la versión "oficial" recomendada en clase. Notar los `push/pop`, los `mov $0` adicionales, `idiv %rsi` en lugar de `%rcx`, los `mov media/resto` al final de la subrutina en lugar de tras la llamada en `main`, y el tipo de ambas variables.

¿Qué media y resto imprime esta versión para el test #03? (16 elementos con valor 0x7ffffff)

- media = 2147483647 resto = 0  
= 0x 7fffffff = 0x 00000000
- media = 16 resto = -16  
= 0x 7fffffff = 0x 00000000
- media = 2147483647 resto = 0  
= 0x 00000010 = 0x ffffffff0
- media = 16 resto = -16  
= 0x 00000010 = 0x ffffffff0

6. ¿Cuál expresión es cierta?

- `popcount(15) < popcount(51)`
- `popcount( 2) == popcount(64)`
- `popcount( 7) > popcount(60)`
- `popcount(96) != popcount( 3)`

7. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de `popcount1`:

```

int pcl(unsigned* array, size_t len){
 size_t i,j;
 int res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 for (j=0; j<8*sizeof(int);j++){
 x >>= 1;
 unsigned bit = x & 0x1;
 res+=bit;
 }
 }
 return res;
}

```

Esta función se diferencia de la versión "oficial" recomendada en clase en el cuerpo del bucle interno. Esta función `popcount1`:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}

d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

8. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount2:

```
int pcount2(int* array, size_t len){
 size_t i;
 int res=0;
 unsigned x;
 unsigned bit;
 for (i=0; i<len; i++){
 x = array[i];
 while(x){
 bit += x & 0x1;
 x >>= 1;
 res = res + bit;
 }
 }
 return res;
}
```

Esta función se diferencia de la versión "oficial" recomendada en clase en el tipo del **array**, la variable **bit** y el cuerpo del bucle interno. Esta función popcount1:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={1,2,4,8}

d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

9. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount4:

```
int pc4(unsigned* array, size_t len){
 size_t i;
 int res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 asm("\n\t"
 "clc\n\t"
 "ini4:\n\t"
 "adc $0, %[r]\n\t"
 "test %[x],%[x]\n\t"
 "shr %[x]\n\t"
 "jne ini4\n\t"
 "adc $0, %[r]\n\t"
 : [r]">+r" (res)
 : [x] "r" (x));
 }
 return res;
}
```

Esta función se diferencia de la versión "oficial" en que tiene una instrucción ensamblador adicional. Este popcount4:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}

- c. fallaría con array={1,2,4,8}

- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

10. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount4:

```
int pc4(unsigned* array, size_t len){
 size_t i;
 int res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 asm("\n\t"
 "clc\n\t"
 "ini4:\n\t"
 "adc $0, %[r]\n\t"
 "fin4:\n\t"
 "shr %[x]\n\t"
 "jne ini3\n\t"
 : [r]">+r" (res)
 : [x] "r" (x));
 }
 return res;
}
```

Esta función es muy diferente a la versión "oficial". Notar el salto condicional a "ini3" en la función popcount3 (en donde sí se hizo bien el **ini3:/shr/adc/test/jne** recomendado). Esta función popcount4:

- a. produce siempre el resultado correcto

- b. fallaría con array={0,1,2,3}

- c. fallaría con array={1,2,4,8}

- d. no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

11. En la práctica de la bomba, el primer ejercicio consistía en saltarse las explosiones, para lo cual se puede utilizar... (marcar opción **falsa**)

- a. objdump

- b. gdb

- c. ddd

- d. eclipse

12. ¿Para qué se utiliza la función **gettimeofday()** en la práctica de la "bomba digital"?

- a. Para cronometrar y poder comparar lo que tardan las distintas versiones del programa

- b. Para imprimir la hora en la pantalla

- c. Para cifrar la clave en función de la hora actual

- d. Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la clave

13. ¿Para qué se utiliza la función **scanf()** en la práctica de la "bomba digital"?

- a. Para escanear el fichero ejecutable "bomba" y asegurarse de que no contenga virus

- b. Para leer la contraseña (clave alfanumérica)
- c. Para leer el PIN (clave numérica)
- d. Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la clave

14. Respecto a las bombas estudiadas en la práctica "bomba digital", ¿en cuál de los siguientes tipos de bomba sería más **difícil** descubrir la contraseña? Se distingue entre strings definidos en el código fuente de la bomba, y strings solicitados al usuario por teclado. Por "cifrar" podemos entender la cifra del César, por ejemplo.

- a. 1 string del fuente se cifra, se invierte y se compara con el string del usuario
- b. el string del usuario se cifra y se compara con 1 string del fuente
- c. 2 strings del fuente se invierten, se concatenan, se cifra el resultado, y se compara con el string del usuario
- d. el string del usuario se concatena con 1 string del fuente, luego se invierte 1 string del fuente, se cifra y se compara con el concatenado

15. En una bomba como las estudiadas en prácticas, del tipo...

```
0x40080f <main+180> lea 0xc(%rsp),%rsi
0x400814 <main+185> lea 0x1dd(%rip),%rdi
 # 0x4009f8
0x40081b <main+192> mov $0x0,%eax
0x400820 <main+197> call 0x400620<scanf>
0x400825 <main+202> mov %eax,%ebx
0x400827 <main+204> test %eax,%eax
0x400829 <main+206> jne 0x40083c <m+225>
0x40082b <main+208> lea 0x1c9(%rip),%rdi
 # 0x4009fb
0x400832 <main+215> mov $0x0,%eax
0x400837 <main+220> call 0x400620<scanf>
0x40083c <main+225> cmp $0x1,%ebx
0x40083f <main+228> jne 0x4007f9 <m+158>
0x400841 <main+230> mov 0x200819(%rip),
 %eax # 0x601060
0x400847 <main+236> cmp %eax,0xc(%rsp)
0x40084b <main+240> je 0x400852 <m+247>
0x40084d <main+242> call 0x400727 <boom>
0x400852 <main+247> lea 0x10(%rsp),%rdi
```

...el código numérico (pin) es...

- a. el entero 0x601060
- b. el entero cuya dirección está almacenada en la posición de memoria 0x4009f8
- c. el entero almacenado a partir de la posición de memoria 0x4009fb
- d. el entero almacenado a partir de la posición de memoria 0x200819+0x400847

16. La función **setup()** de Arduino es llamada:

- a. Al principio de cada iteración de la función loop()

- b. Cuando se sube desde el **kit-entorno** de desarrollo o se pulsa el botón de reset, pero no cuando se conecta la alimentación
- c. Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, o se sube desde el **kit-entorno** de desarrollo
- d. Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, pero no cuando se sube desde el **kit-entorno** de desarrollo

17. ¿Qué sentencia usamos en el programa **blink** (led intermitente) para encender el led integrado en la placa Elegoo Mega2560?

- a. digitalWrite (LED\_BUILTIN, HIGH);
- b. pinMode (LED\_BUILTIN, OUTPUT);
- c. analogWrite (LED\_BUILTIN, HIGH);
- d. pulseIn (LED\_BUILTIN, HIGH);

18. Sobre el resultado devuelto por la función de Arduino **map(sensorValue, sensorLow, sensorHigh, 50, 4000)**; usada en el programa del Theremín de luz:

- a. Si sensorValue vale 50, devuelve sensorLow
- b. Si sensorValue vale 50, devuelve sensorHigh
- c. Si sensorValue vale **25202025**, devuelve la mitad entre sensorLow y sensorHigh
- d. Si sensorValue vale la mitad entre sensorLow y sensorHigh, devuelve **25202025**

19. En el programa line.cc de la práctica de cache, si para cada tamaño de línea (line) recorremos una única vez el vector, la gráfica resultante es decreciente porque:

- a. hay un mayor historial de accesos y es más probable que un nuevo acceso sea un acierto
- b. cada vez los tamaños de línea escogidos van decreciendo y se tarda menos en leerlos
- c. cada vez los tamaños de línea escogidos van decreciendo y hay menor localidad espacial
- d. el vector se indexa con la variable de control del bucle, con un incremento o paso de line

20. En la práctica de la cache, en size.cc se accede al vector saltando de 64 en 64. ¿Por qué?

- a. Para recorrer el vector más rápidamente
- b. Porque con un salto menor que 64 habría aciertos por localidad espacial y haría menos clara la gráfica
- c. Porque cada elemento del vector ocupa 64 B
- d. Para evitar aciertos por localidad temporal y que sólo haya aciertos por localidad espacial

Nombre:

DNI:

Grupo:

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

1. Habiendo definido en código fuente ensamblador `saludo: .ascii "Hola mundo\n"`, si comparamos los comandos gdb siguientes:  
`x/s &saludo` frente a `print (char*) &saludo`, se puede decir que:

- ambos nos muestran el string (el mensaje "Hola mundo")
- el primero (x) nos muestra el string, y el segundo (print) nos muestra otro valor distinto
- el segundo (print) nos muestra el string, y el primero (x) nos muestra otro valor distinto
- alguno o ambos contienen algún error gramatical (falta o sobra algún &, algún `typedef` (char\*) ó (long), etc)

2. En la práctica "media" se pide sumar una lista de 16 enteros SIN signo de 32 bits evitando acarreo. ¿Cuál es el mayor valor que repetido en toda la lista no causaría acarreo en 32 bits?

- 0xFFFF FFFF
- 0x7FFF FFFF
- 0x1000 0000
- 0x0FFF FFFF

3. En la práctica "media" se pide usar `cltq/cdq` y `cqto/cqo` para hallar la media y resto de una lista de 16 enteros CON signo de 32 bits usando registros de 64 bits. Un estudiante entrega la siguiente versión:

```
...
media: .int 0
resto: .int 0
...
main: .global main
 mov $lista, %rbx
 mov longlista, %ecx
```

```
call suma
mov %rax, media
mov %edx, resto
...
suma:
mov $0, %rax
mov $0, %rsi
mov $0, %edx
mov $0, %rdi
mov $0, %ebp
bucle:
movl (%rbx,%rsi,4), %eax
cltq # EAX -> RAX
add %rax, %rdi
#adc %edx, %ebp
inc %rsi
cmp %rsi,%rcx
jne bucle
mov %rdi, %rax
#mov %ebp, %edx
idiv %rcx
ret
```

Este programa se diferencia de la versión "oficial" recomendada en clase en que se guarda como media todo RAX, la subrutina no se llama **sumaq**, se inicializan a 0 más registros y no se usa CQTO.

¿Qué media y resto calcula este programa para el test #02? (16 elems. con valor -1,-2,-1,-2...)

- media = -1, resto = -8
- media = -2, resto = 8
- media = -1, resto = 8
- media = -2, resto = -8

4. En la misma práctica "media" (`cltq/cdq` y `cqto/cqo`), un estudiante entrega la siguiente versión:

```
sumaq:
mov $0, %edi
mov $0, %esi
```



```

bucleq:
 mov (%rbx,%rsi,4), %eax
 cltq # EAX -> RAX
 add %rax, %rdi
 inc %esi
 cmp %esi, %ecx
 jne bucleq

 mov %rdi, %rax
 cqto # RAX -> RDX:RAX
#idiv/idiq %rcx no hace la division,
#queda todo como resto
 idiv %ecx
 ret

```

Esta subrutina se diferencia de la versión "oficial" recomendada en clase en que se nombran EDI, ESI y ECX. El propio estudiante comentó alguna de las diferencias. ¿Qué media y resto calcula este programa para el test #03? (16 elementos con valor 0x7ffffff)

- media = 0x0ffffff, resto = 0
- media = 0x7ffffff, resto = 0
- media = 0xffffffff, resto = 0
- resto distinto de cero

5. En la misma práctica "media" (cltq/cdq y cqto/cqo), un estudiante entrega la siguiente versión:

```

...
media: .int 0
resto: .int 0
formato: .asciz "..."
formatoq: .asciz "..."
...
 mov $lista, %rbx
 mov longlista, %ecx
 call sumaq
 mov %rax, media
 mov %rdx, resto
...
sumaq:
 mov $0, %rax
 mov $0, %rsi
 mov $0, %rdx
 mov $0, %rdi
 mov $0, %rbp
bucleq:
 mov (%rbx,%rsi,4), %rax
 cqo # RAX -> RDX:RAX
 add %rax, %rdi
 adc %rdx, %rdi

 inc %rsi
 cmp %rsi,%rcx
 jne bucleq

 mov %rdi, %rax
mov %rbp, %rdx
 mov $0, %ebp
 mov %ecx,%ebp
 idiv %ebp
 ret

```

Este programa es muy diferente a la versión "oficial" recomendada en clase, destacando especialmente los mov media/resto tras la

llamada a sumaq, los mov \$0 adicionales superfluos, mover elementos a %rax, usar cqo en lugar de cltq y no usarlo antes de idiv %ebp, que tampoco está bien, y además sobra el adc.

Este programa calcula el resultado correcto:

- para lista: .int 1, 2, 1, 2, ... (16 elementos)
- para lista: .int -1,-2,-1,-2, ... (16 elementos)
- para ambos ejemplos
- para ninguno de los dos ejemplos

6. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount4:

```

int pc4(unsigned* array, size_t len){
 size_t i;
 int res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 asm("\n\t"
 "clc \n"
 "ini4: \n\t"
 "shr %[x] \n\t"
 "adc $0, %[r] \n\t"
 "test %[x],[x] \n\t"
 "jne ini4 \n"
 "fin4: \n\t"
 "clc \n\t"
 : [r]">+r" (res)
 : [x] "r" (x));
 }
 return res;
}

```

Esta función es una mezcla de las versiones "oficiales" recomendadas en clase para popcount3 y popcount4, con alguna instrucción cambiada. Esta función popcount4:

- produce siempre el resultado correcto
- fallaría con array={0,1,2,3}
- fallaría con array={1,2,4,8}
- no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

7. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount4:

```

int pc4(unsigned* array, size_t len){
 size_t i;
 int res=0;
 unsigned x;
 for (i=0; i<len; i++){
 x = array[i];
 asm("\n\t"
 "clc \n"
 "ini4: \n\t"
 "adc $0, %[r] \n\t"
 "shr %[x] \n\t"
 "jnz ini4 \n\t"

```

```

 : [r]"+"r" (res)
 : [x] "r" (x));
 }
 return res;
}

```

Esta función se diferencia de la versión "oficial" recomendada en clase en que le falta una instrucción ensamblador final (última).

Esta función popcount4 (pensar bien qué pasa con el elemento 0 y con el elemento 1):

- produce siempre el resultado correcto
- fallaría con array={0,1,2,3}
- fallaría con array={1,2,4,8}
- no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

8. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount5:

```

int pc5(unsigned* array, int len){
 size_t i, k, x;
 int res=0;
 for (i=0; i<len; i++){
 long val = 0;
 x = array[i];
 for (k=0; k<8; k++){
 val+= x& 0x0101010101010101L;
 x >>= 1;
 }
 val += (val >> 32);
 val += (val >> 16);
 val += (val >> 8);
 res += (val & 0xFF);
 }
 return res;
}

```

Esta función no sigue la adaptación a 32 bits recomendada en clase como versión "oficial", y conserva muchos vestigios de la versión de 64 bits, como el tipo de `val`, `len` y `x`, la constante `0x01...01L` y el desplazamiento `>>32`. Esta función popcount5:

- produce siempre el resultado correcto
- fallaría con array={0,1,2,3}
- fallaría con array={1,2,4,8}
- no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

9. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount5:

```

int pc5(unsigned* array, size_t len){
 size_t i,j;
 long val=0;
 unsigned long x;
 for (i=0; i<len; i++){
 x = array[i];
 for (j=0; j<8; j++){
 val+= x& 0x0101010101010101L;
 x >>= 1;
 }
 }
}

```

```

 val += (val >> 16);
 val += (val >> 8);
 val += (val >> 4);
 return val & 0xFF;
}

```

Esta función es muy diferente a la versión "oficial" recomendada en clase, destacando especialmente el tipo de `val` y `x`, la inicialización de `val`, la variable `res`, la constante `0x01...01L`, el anidamiento de los desplazamientos y el desplazamiento `>>4`.

Esta función popcount4 da resultado correcto:

- con array={0,1,2,3}
- con array={1,2,4,8}
- con ambos ejemplos
- con ninguno de los dos ejemplos

10. En la misma práctica "popcount" un estudiante entrega la siguiente versión de popcount7:

```

int pc7(unsigned* array, size_t len){
 size_t i;
 unsigned long x1,x2;
 int res=0;
 const unsigned long m1 = 0x5555555555555555;
 const unsigned long m2 = 0x3333333333333333;
 const unsigned long m4 = 0xf0f0f0f0f0f0f0f;
 const unsigned long m8 = 0x00ff00ff00ff00ff;
 const unsigned long m16= 0x0000ffff0000ffff;
 const unsigned long m32= 0x00000000ffffffff;

 if (len & 0x3)
 printf("len no múltiplo de 4\n");

 for (i=0; i<len; i+=4){
 x1= *(unsigned long*)&array[i];
 x2= *(unsigned long*)&array[i+2];

 x1 = (x1 & m1) + ((x1 >> 1) & m1);
 x1 = (x1 & m2) + ((x1 >> 2) & m2);
 x1 = (x1 & m4) + ((x1 >> 4) & m4);
 x1 = (x1 & m8) + ((x1 >> 8) & m8);
 x1 = (x1 & m16) + ((x1 >>16) & m16);
 x1 = (x1 & m32) + ((x1 >>32) & m32);

 x2 = (x2 & m1) + ((x2 >> 1) & m1);
 x2 = (x2 & m2) + ((x2 >> 2) & m2);
 x2 = (x2 & m4) + ((x2 >> 4) & m4);
 x2 = (x2 & m8) + ((x2 >> 8) & m8);
 x2 = (x2 & m16) + ((x2 >>16) & m16);
 x1 = (x1 & m32) + ((x1 >>32) & m32);

 res += x1+x2;
 }
 return res;
}

```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en la última máscara, suma y desplazamiento `>>32`, que se realizan sobre una variable distinta. Esta función popcount4 da resultado correcto:

- con array={3,2,1,0}
- con array={8,4,2,1}
- con ambos ejemplos

d. con ninguno de los dos ejemplos

11. Invocando a printf de libC (SystemV AMD64) desde ensamblador...

- a. el primer argumento debe ponerse en %rax
- b. el segundo argumento es el formato
- c. si se desean imprimir tres **long**, el tercero debe ponerse en %rdx

d. si se desean imprimir cuatro **int**, el cuarto debe ponerse en %r8d

12. Recordando que los argumentos de `_start` (argc/argv y variables de entorno) se pasan en pila en SysV AMD64, si en el Ejemplo 1 de la Práctica 3 se hace **`gdb -tui --args suma_01 uno dos tres`**, se lanza con **`br _start`** y **`run`**, y se teclea **`p * (char**)(rsp+40)`**, ¿qué obtenemos?

- a. tres
- b. 0x0
- c. una variable de entorno
- d. un error de gdb

13. En la práctica de la bomba, el primer ejercicio consistía en “saltarse” las “explosiones”, para lo cual se puede utilizar...

- a. objdump o gdb
- b. gdb o eclipse
- c. eclipse o ghex
- d. ghex u objdump

14. En una bomba como las estudiadas en prácticas, del tipo...

```
0x40079b <main+64> lea 0x30(%rsp),%rdi
0x4007a0 <main+69> mov 0x2008d9(%rip),
 %rdx # 0x601080
0x4007a7 <main+76> mov $0x64,%esi
0x4007ac <main+81> call 0x400600 <fgets>
0x4007b1 <main+86> test %rax,%rax
0x4007b4 <main+89> je 0x400785 <main+42>
0x4007b6 <main+91> lea 0x30(%rsp),%rdi
0x4007bb <main+96> mov $0xd,%edx
0x4007c0 <main+101> lea 0x2008a1(%rip),
 %rsi # 0x601068
0x4007c7 <main+108> call 0x4005d0 <strncmp>
0x4007cc <main+113> test %eax,%eax
0x4007ce <main+115> je 0x4007d5 <main+122>
0x4007d0 <main+117> call 0x400727 <boom>
0x4007d5 <main+122> lea 0x20(%rsp),%rdi
```

...la contraseña (alfanumérica) es...

- a. el string almacenado a partir de 0x601068
- b. el string alm. a partir de 0x2008a1+0x4007c7
- c. el string almacenado a partir de 0x30(%rsp)
- d. no se puede marcar una y sólo una de las respuestas anteriores

15. En la práctica de E/S en Arduino, la instrucción CBI del repertorio del microcontrolador realiza...

- a. una comparación
- b. un complemento a uno
- c. una operación de bits
- d. una llamada a subrutina

16. En la práctica de E/S en Arduino, la instrucción SBI del repertorio del microcontrolador realiza...

- a. una suma de un valor inmediato
- b. una resta de un valor inmediato
- c. una operación de bits
- d. un salto incondicional

17. En la práctica de E/S en Arduino, DDRB es

- a. el segundo canal de memoria DDR
- b. el registro de dirección de datos del puerto B
- c. el registro de datos y direcciones B
- d. el registro buffer de datos D

18. En la práctica de E/S en Arduino, la instrucción SBIW del repertorio del microcontrolador realiza...

- a. una suma de un valor inmediato
- b. una resta de un valor inmediato
- c. una operación de bits
- d. un salto incondicional

19. En el programa line.cc de la práctica de cache, para cada tamaño de línea (line) recorreremos una única vez el vector, pero podríamos haber realizado un número fijo y grande de accesos. Al dibujar la gráfica del tiempo de iteración en función del tamaño de línea...

- a. de ambas formas sale gráfica creciente
- b. recorriendo una vez sale gráfica creciente
- c. con núm. fijo accesos sale gráfica creciente
- d. de ninguna de las dos formas sale gr. creciente

20. En la práctica de la cache, en size.cc se realiza un número fijo y grande de accesos, pero podríamos haber recorrido una única vez el vector (saltando también de 64 en 64). Al dibujar la gráfica del tiempo de bucle en función del tamaño del vector...

- a. de ambas formas sale gráfica creciente
- b. recorriendo una vez sale gráfica creciente
- c. con núm. fijo accesos sale gráfica creciente
- d. de ninguna de las dos formas sale gr. creciente



|                |               |
|----------------|---------------|
| <b>Nombre:</b> |               |
| <b>DNI:</b>    | <b>Grupo:</b> |

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

1. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros en doble precisión. alguna de las siguientes técnicas no está relacionada con la aritmética en doble precisión:

- acumulación de desbordamientos (OF, overflow flag)
- acumulación de acarreo (CF, carry flag)
- extensión con ceros
- extensión de signo

2. En la misma práctica "media", el programa esqueleto ofrecido (suma.s) no es válido, en concreto... (marcar la opción FALSA)

- no está preparado para sumar más de 9 elementos
- no hace extensión con ceros de los elementos
- no hace extensión de signo de los elementos
- no consulta ni el flag de acarreo CF ni el de overflow OF

3. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount5:

```
int popcount5(int* array, size_t len){
 size_t i,j;
 int x, val, result=0;
 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 for (j=0; j<8; j++){
 val += x & 0x01010101;
 x >>= 1;
 }
 val += (val >> 16);
 val += (val >> 8);
 result+= val & 0xFF;
 }
}
```

```
}
return result;
}
```

Esta función se diferencia de la versión "oficial" en los tipos de array y x.

Esta función popcount5:

- produce siempre el resultado correcto
- fallaría con array={0,1,2,3}
- fallaría con array={0,-1,-2,-3}
- no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

4. Otro estudiante entrega la siguiente versión de popcount5:

```
int pcnt5(unsigned* array, size_t len){
 size_t i,j;
 unsigned x;
 int val, result=0;

 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 for (j=0; j<=8; j++){
 val += x & 0x01010101;
 x >>= 1;
 }
 val += (val >> 16);
 val += (val >> 8);
 result+= val & 0xFF;
 }
 return result;
}
```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en las condiciones del for interno.

Esta función popcount5 fallaría:

- a. con array={1, 16, 256, 4096}
- b. con array={1, 32, 1024, 32768}
- c. con ambos ejemplos
- d. con ninguno de los dos ejemplos

5. En una bomba como las estudiadas en prácticas, del tipo...

```
4006fa: lea 0x10(%rsp),%rbx
4006ff: mov 0x20096a(%rip),%rdx
400706: mov $0x64,%esi
40070b: mov %rbx,%rdi
40070e: callq 400570 <fgets@plt>
400713: mov $0xd,%edx
400718: lea 0x200939(%rip),%rsi
40071f: mov %rbx,%rdi
400722: callq 400560 <strncmp@plt>
400727: test %eax,%eax
400729: je 400730 <main+0x51>
40072b: callq 400697 <boom>
400730: lea 0x1b5(%rip),%rsi
```

...la contraseña (alfanumérica) es...

- a. el string almacenado a partir de 0x10(%rsp)
- b. el string alm. a partir de 0x20096a+0x4006ff
- c. el string almacenado a partir de 0x601058
- d. el string alm. a partir de 0x200939+0x400718

6. En una bomba como las estudiadas en prácticas, del tipo...

```
400746: lea 0xc(%rsp),%rsi
40074b: lea 0x1ae(%rip),%rdi
400752: mov $0x0,%eax
400757: callq 400590 <_scanf@plt>
40075c: mov 0x2008ee(%rip),%eax
400762: cmp %eax,0xc(%rsp)
400766: je 40076d <main+0x8e>
400768: callq 400697 <boom>
40076d: callq 4006bb <defused>
```

...el código numérico (pin) es...

- a. el entero almacenado a partir de 0xc(%rsp)
- b. el entero alm. a partir de 0x1ae+0x40074b
- c. el entero almacenado a partir de 0x6010560
- d. el entero alm. a partir de 0x2008ee+0x400762

7. La función setup() de Arduino es llamada:

- a. Al principio de cada iteración de la función loop()
- b. Cuando se sube desde el entorno de desarrollo o se pulsa el botón de reset, pero no cuando se conecta la alimentación

- c. Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, o se sube desde el entorno de desarrollo
- d. Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, pero no cuando se sube desde el entorno de desarrollo

8. Suponga una memoria cache con las siguientes propiedades: Tamaño: 512 bytes. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Suponga que para la siguiente secuencia de direcciones enviadas a la cache: 0, 1, 2, 4, 8, 16, 32, 64, la tasa de acierto es 0,25. ¿Cuál es el tamaño de línea de la cache?

- a. 4 bytes
- b. 8 bytes
- c. 16 bytes
- d. Ninguno de los anteriores

9. En el programa size.cc de la práctica "cache", se accede al vector saltando...

- a. de byte en byte
- b. de 64 en 64 bytes
- c. de 1 KB en 1 KB
- d. de line en line bytes, donde line barre los valores desde 1 hasta 1K en un bucle for

10. En la práctica de cache hemos hecho una gráfica con el código size.cc ¿Qué forma tiene la gráfica que se debe obtener?

- a. Forma de U (o V), con un tramo descendente y otro ascendente
- b. Forma de /, una gráfica siempre creciente y sin escalones
- c. Forma de media U seguida de \, es decir, un tramo descendente suave, un pequeño tramo horizontal, y un tramo descendente lineal
- d. Una escalera con varios tramos horizontales

11. La línea de código ensamblador: **mov \$msg, %rsi**

- a. Copia en rsi los primeros 64 bits de memoria desde la posición apuntada por la etiqueta msg.
- b. Copia en rsi todo el contenido de la cadena apuntada por msg.
- c. Copia en rsi la dirección de memoria de 64 bits almacenada en memoria a partir de la posición indicada por la etiqueta msg.
- d. Copia en rsi los 64 bits de la dirección msg.

12. En la práctica "media" se programa la suma de una lista de 16 enteros de 4 bytes para producir un resultado de 8 bytes, primero sin signo y luego con signo. Si la lista se rellena con el valor 0x0400 0000, ¿en qué se diferencian los resultados de ambos programas?

- a. no se diferencian
- b. en uno ocupa 32 bits, en otro 64 bits
- c. en uno se interpreta como negativo, en otro como positivo
- d. en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no

13. ¿Cuál de las siguientes líneas declara un puntero a función en C?

- a. `int *func;`
- b. `int func();`
- c. `int *func();`
- d. `int (*func)();`

14. Si `val` es una variable de tipo `unsigned long`, entonces la sentencia: `val += (val >> 32);`

- a. Pone siempre `val` a 0.
- b. Deja siempre `val` al mismo valor que tuviera antes de la sentencia.
- c. Su traducción incluye una instrucción `shr` seguida de una suma.
- d. Su traducción incluye una instrucción `sar` seguida de una suma.

15. ¿Para qué se utiliza la función `gettimeofday` en la práctica de la "bomba digital"?

- a. Para cronometrar y poder comparar las duraciones de las distintas soluciones del programa.
- b. Para imprimir la hora en la pantalla.
- c. Para cifrar la contraseña en función de la hora actual.
- d. Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la contraseña o el PIN.

16. En la práctica de la bomba, el primer ejercicio consiste en ir saltando las "explosiones" mientras se depura el código, para lo cual se puede utilizar...

- a. `objdump` o `gdb`
- b. `gdb` o `ddd`
- c. `ddd` o `hexedit`

d. `hexedit` u `objdump`

17. En la placa del kit de Arduino, las patillas de tierra vienen etiquetadas con la leyenda:

- a. A0
- b. 5V
- c. GND
- d. 3.3V

18. Las resistencias utilizadas en la práctica de Arduino

- a. Son de color azul claro y tienen 5 bandas de color: las 3 primeras indican un valor, la 4ª banda es un multiplicador y la 5ª banda es la tolerancia
- b. Son de color beige y tienen 4 bandas de color: las 2 primeras indican un valor, la 3ª banda es un multiplicador y la 4ª banda es la tolerancia
- c. Tienen polaridad y el cátodo (polo negativo) es el extremo de la banda de color con una separación mayor respecto a las otras.
- d. Tienen polaridad y el ánodo (polo positivo) es el extremo de la banda de color con una separación mayor respecto a las otras.

19. ¿Cuál de las siguientes afirmaciones sobre las caches es FALSA?

- a. Casi ningún procesador actual tiene memoria cache L2.
- b. Las direcciones a las que accede un programa no son completamente aleatorias, sino que se rigen por ciertos patrones de localidad.
- c. Un procesador actual tiene varias caches de nivel 1.
- d. La cache de nivel 3 no contiene toda la memoria que maneja el programa.

20. En el programa `line.cc`, si para cada tamaño de línea (`line`) recorremos una única vez el vector, la gráfica resultante es decreciente porque:

- a. Cada vez que `line` aumenta al doble, el número de aciertos por localidad temporal aumenta, porque ya habíamos accedido a cada posición  $i$  del vector cuando lo recorrimos en el punto anterior del eje X.
- b. Cada vez que `line` aumenta al doble, el número de aciertos por localidad espacial aumenta, porque ya habíamos accedido a cada posición  $i-1$  del vector cuando lo recorrimos en el punto anterior del eje X.

- c. Cada vez que line aumenta al doble, se accede con éxito a más posiciones del vector en niveles de la jerarquía de memoria más rápidos.
  - d. Cada vez que line aumenta al doble, realizamos la mitad de accesos al vector que para el valor anterior.
-

|                |               |
|----------------|---------------|
| <b>Nombre:</b> |               |
| <b>DNI:</b>    | <b>Grupo:</b> |

## Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 0.2p si es correcta, 0 si está en blanco o claramente tachada, -0.06p 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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |

1. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros sin signo en doble precisión acumulando acarreos. Para aplicar esta técnica se sugiere utilizar la(s) instrucción(es)...

- a. ADC \$0, ...
- b. CLTD, ADD y ADC
- c. JNC y INC
- d. CLTQ, CQTO y IDIV

2. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros sin signo en doble precisión extendiendo con ceros. Para aplicar esta técnica se sugiere utilizar la(s) instrucción(es)...

- a. ADC \$0, ...
- b. CLTD, ADD y ADC
- c. JNC y INC
- d. CLTQ, CQTO y IDIV

3. En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros en doble precisión extendiendo el signo. Para aplicar esta técnica se sugiere utilizar la(s) instrucción(es)...

- a. ADC \$0, ...
- b. CLTD, ADD y ADC
- c. JNC y INC
- d. CLTQ, CQTO y IDIV

4. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount5:

```
int popcount5(int* array, size_t len){
 size_t i,j,val;
 long x;
 int result=0;
 for (i=0; i<len; i++){
 x = array[i];
 val = 0;
 for (j=0; j<8; j++){
 val += x & 0x0101010101010101L;
 x >>= 1;
 }
 val += (val >> 32);
 val += (val >> 16);
 val += (val >> 8);
 result+= val & 0xFF;
 }
 return result;
}
```

Esta función no sigue la adaptación a 32 bits recomendada en clase como versión "oficial", y conserva muchos vestigios de la versión de 64 bits, como los tipos de **array**, **x**, **val**, la máscara **0x...01L** y el desplazamiento **>>32**.

Esta función popcount5 fallaría:

- a. con array={0x80000000, 0x00400000, 0x00000200, 0x00000001}
- b. con array={0, -1, -2, -3}
- c. con ambos ejemplos
- d. con ninguno de los dos ejemplos

5. La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcount6:

```
int pc6(unsigned* array, size_t len){
 size_t i;
 unsigned x;
```

```

int result=0;

const unsigned m1 = 0x55555553;
const unsigned m2 = 0x33333333;
const unsigned m4 = 0x0f0f0f0f;
const unsigned m8 = 0x00ff00ff;
const unsigned m16 = 0x0000ffff;

for (i=0; i<len; i++){
 x = array[i];

 x = (x & m1) + ((x >> 1) & m1);
 x = (x & m2) + ((x >> 2) & m2);
 x = (x & m4) + ((x >> 4) & m4);
 x = (x & m8) + ((x >> 8) & m8);
 x = (x & m16) + ((x >> 16) & m16);

 result+= x;
}
return result;
}

```

Esta función sólo se diferencia de la versión "oficial" recomendada en clase en algún dígito de alguna máscara.

Esta función popcount6 fallaría:

- con array={0x80000000, 0x00400000, 0x000000200, 0x00000001}
- con array={0x10000000, 0x00200000, 0x000000400, 0x00000008}
- con ambos ejemplos
- con ninguno de los dos ejemplos

6. En una bomba como las estudiadas en prácticas, del tipo...

```

400746: lea 0xc(%rsp),%rsi
40074b: lea 0x1ab(%rip),%rdi
 # 4008fd <stdin_u+0x10d>
400752: mov $0x0,%eax
400757: callq 400590 <scanf@plt>
40075c: cmpl $0x400600,0xc(%rsp)
400764: je 40076b <main+0x8c>
400766: callq 400697 <boom>
40076b: callq 4006bb <defused>

```

...el código numérico (pin) es...

- el entero 0x400590
  - el entero 0x400600
  - el entero almacenado a partir de 0xc(%rsp)
  - el entero alm. a partir de 0x1ab+0x400752
7. En la práctica "blink" de Arduino, un estudiante muestra a los profesores lo que le han enseñado en algún otro lugar: conectando un led directamente entre las patillas *Digital pin 13* (LED\_BUILTIN) y GND (justo al lado) también parpadea. Cabe esperar la siguiente respuesta de los profesores:

- Exactamente esa solución viene en el guión, no hace falta aprenderla fuera de clase
- Nosotros conectamos los led en serie con una resistencia como dice el guión
- Hay que revisar ese equipo en concreto, debe tener algún defecto, ese led no debe parpadear
- Es cierto, es un error de diseño del guión, es más sencillo conectarlo así

8. Suponga una memoria cache con las siguientes propiedades: Tamaño: 32 KB. Política de reemplazo: LRU. Estado inicial: vacía (todas las líneas inválidas). Tamaño de línea 64 B. Dado el siguiente fragmento de código:

```

int x[262144], y[262144], a=0;
for (i = 0; i < 262144; i++)
 a += x[i] * y[i];

```

¿Cuál será la tasa de fallos aproximada que se obtiene en la ejecución del bucle anterior?

Suponer que el tamaño de los tipos de datos es como en x86-64, y que el compilador optimiza el acceso a las variables a/i en un registro.

- 1/2 (un fallo por cada 2 accesos)
- 1/4 (un fallo por cada 4 accesos)
- 1/8 (un fallo por cada 8 accesos)
- 1/16 (un fallo por cada 16 accesos)

9. En el programa line.cc de la práctica "cache", se accede al vector saltando...

- de byte en byte
- de 64 en 64 bytes
- de 1 KB en 1 KB
- de line en line bytes, donde line barre los valores desde 1 hasta 1K en un bucle for

10. En la práctica de cache hemos hecho una gráfica con el código line.cc ¿Qué forma tiene la gráfica que se puede obtener?

- Forma de U (o V), con un tramo descendente y otro ascendente
- Forma de /, una gráfica siempre creciente
- Forma de media U seguida de \, es decir, un tramo descendente suave, un pequeño tramo horizontal, y un tramo descendente lineal
- Una escalera con varios tramos horizontales

11. ¿Cuál de las siguientes instrucciones es incorrecta?

- shr \$1,%rdx
- shr %rdx
- shr %cl,%rdx



d. `shr %rcx,%rdx`

12. 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. `0xffff ffff`
- b. `0xfc00 0000`
- c. `0xf7ff ffff`
- d. `0xf000 0000`

13. La función siguiente, que aparece en el esqueleto `suma_09` presentado en la práctica “popcount”

```
int suma3(int* array, int len){
 asm("mov $0, %%eax \n"
 " mov $0, %%rdx \n"
 "bucle: \n"
 " add (%rdi,%rdx,4), %%eax \n"
 " inc %rdx \n"
 " cmp %rdx,%rsi \n"
 " jne bucle \n"
 " : // output
 " : // input
 " : "cc", // clobber
 " "rax", "rdx"
);
}
```

no incluye la instrucción **ret** dentro de la sentencia **asm** porque:

- a. porque la función no devuelve nada
- b. porque `suma3` es un puntero a función
- c. porque la incluye automáticamente el compilador
- d. porque `suma3` no se llamará con `call` sino con `jmp`

14. Respecto a la inicialización de los elementos del array en el esqueleto `suma_09` de la práctica “popcount”:

```
#define SIZE (1<<16)
for (i=0; i<SIZE; i++)
 lista[i]=i;
```

- a. Cualquiera de las tres funciones suma del programa `suma_09_Casm` irán más rápido con esta inicialización que con números aleatorios.
- b. Las tres funciones suma del programa `suma_09_Casm` irán igual de rápido con esta inicialización que con números aleatorios.

- c. `popcount1` (bucle interno `for`) irá más rápido con esta inicialización que con números aleatorios
- d. `popcount2` (bucle interno `while`) irá más lento con esta inicialización que con números aleatorios.

15. Un estudiante entrega la solución de una bomba y en ella aparece el comando de gdb “**p(char\*)\$rdi**”. Este comando:

- a. muestra únicamente el contenido de `%rdi`
- b. es erróneo sintácticamente
- c. muestra el valor de la dirección de memoria contenida en `%rdi` y el string almacenado a partir de esa dirección de memoria
- d. muestra el contenido de `%rdi` interpretado como un string de 8 caracteres

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

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

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

- a. `set $0x080486ef=0xeb`
- b. `set *(char*)0x080486ef=0xeb`
- c. `set *(char*)0x080486f6=jmp`
- d. `set %0x080486ef=0xeb`

17. ¿Cuál de las siguientes patillas (pin) no se menciona (no se pide usar/no se debe usar) en la 1ª práctica del Theremín de luz de Arduino? (Proyecto p06 / sólo fotocélula y zumbador)

- a. Analog In A0
- b. AREF
- c. `ledPin = 13`
- d. GND

18. En la práctica del Theremín de luz de Arduino se pide combinar el código del Proyecto p06 (sólo fotocélula y zumbador) con el de la Lección 26 (fotocélula y leds). Para conseguir que se enciendan y apaguen correctamente todos los led, lo mejor sería:

- a. Reutilizar la variable `pitch` calculada mediante `map()` como la nueva `numLEDSLit`

- b. Fusionar las variables sensorValue y reading de forma que sólo se llame una vez a analogRead()
  - c. Calcular numLEDSLit con otra función map() de 0 a 8, en lugar de barrer de 50 a 4000
  - d. Copiar uno y otro código en las secciones correspondientes, no se puede fusionar nada
- 

19. El código de size.cc accede al vector saltando de 64 en 64. ¿Por qué?

- a. Porque cada elemento del vector ocupa 64 bytes.
  - b. Para recorrer el vector más rápidamente.
  - c. Porque el tamaño de cache L1 de todos los procesadores actuales es de 64 KB.
  - d. Para anular los aciertos por localidad espacial, es decir, para que sólo pueda haber aciertos por localidad temporal.
- 

20. Un servidor 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, que se ejecuta en un único núcleo, tiene que ordenar un vector de datos de usuarios accediendo repetidamente a sus elementos. Cada elemento es una estructura de datos y tiene un tamaño de 4 KB. Según <http://www.cpu-world.com/> los tamaños de cache son:

| Cache: | L1 data  | L1 instr. | L2        | L3   |
|--------|----------|-----------|-----------|------|
| Size:  | 6 x 32KB | 6 x 32KB  | 6 x 256KB | 15MB |

Si representamos en una gráfica las prestaciones en función del número de usuarios a ordenar, ¿para qué límites teóricos en el número de usuarios se observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

- a. 4 / 32 / 512 usuarios
  - b. 8 / 64 / 3840 usuarios
  - c. 48 / 384 / 23040 usuarios
  - d. 32 / 256 / 8192 usuarios
-



Nombre:

DNI:

Grupo:

## Test de Teoría (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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

- Si almacenamos según el criterio little-endian la palabra de 64 bits 0xFACEB00C a partir de la dirección 0xCAFEBAFE, el byte 0xCE quedará almacenado en la dirección:
  - 0xCAFEBAF1
  - 0xCAFEBAF0
  - 0xCAFEBAFF
  - 0xCAFEBAFE
- ¿Qué novedad se desarrolló en la tercera generación de computadores?
  - Los circuitos integrados
  - Los primeros lenguajes de programación de alto nivel
  - Los microprocesadores CISC
  - Los microprocesadores RISC
- En X86-64, el registro contador de programa se denomina:
  - RIP
  - EIP
  - IP
  - R15
- La primera letra (l) de la instrucción lea:
  - indica que la instrucción usa ordenación de bytes little-endian
  - indica que la instrucción trabaja con un operando destino de 32 bits (long word)
  - indica que la instrucción afecta a los 16 bits menos significativos del operando destino (low word)
  - forma parte del nemotécnico de la instrucción
- ¿Cuál es la diferencia entre las instrucciones mov y lea?
  - lea accede a la posición indicada, mientras que mov no lo hace
  - mov accede a la posición indicada, mientras que lea no lo hace
  - lea puede usarse para copiar un registro a otro, mientras que mov no
  - mov puede usarse para copiar un registro a otro, mientras que lea no
- La instrucción not:
  - realiza el complemento a dos
  - realiza el complemento a uno (cambiar unos por ceros y ceros por unos)
  - realiza la operación no-or (or negada)
  - realiza un salto condicional si negativo
- La instrucción JGE / JNL provoca un salto si...
  - SF = 1
  - CF = 1
  - SF = 0
  - OF = SF
- Un overflow nunca puede ocurrir cuando:
  - se suman dos números positivos
  - se suman dos números negativos
  - se suma un número positivo a un número negativo
  - se resta un número positivo de un número negativo
- Después de ejecutar una instrucción de suma sobre dos números con signo de la que

sabemos que no provocará overflow (los dos números son pequeños en valor absoluto), queremos comprobar si el resultado de la suma es menor que 0. ¿Qué flag necesita comprobar la instrucción de salto condicional equivalente a “if (resultado<0) then goto label”?

- a. CF
- b. OF
- c. SF
- d. ZF

10. La instrucción cmovb %edx, %eax

- a. copia en %eax el contenido de %edx si el indicador de acarreo es 1
- b. copia el byte bajo de %edx en el byte bajo de %eax
- c. copia en %eax el byte de memoria apuntado por la dirección contenida en %edx
- d. copia en %eax el contenido de %edx si %eax es menor que %edx

11. ¿Cuál de las siguientes afirmaciones sobre la instrucción leave es cierta?

- a. Se ejecuta justo después de retornar de un procedimiento
- b. Equivale a pop %ebp seguida de mov %ebp,%esp
- c. Equivale a mov %esp,%ebp seguida de pop %ebp
- d. No es obligatorio usarla. En su lugar puede realizarse una secuencia explícita de operaciones mov y pop

12. Para crear espacio en la pila para variables locales sin inicializar suele realizarse la siguiente operación:

- a. Restar una cantidad positiva a EBP
- b. Sumar una cantidad positiva a EBP
- c. Restar una cantidad positiva a ESP
- d. Sumar una cantidad positiva a ESP

13. ¿Cuál de los siguientes lenguajes no permite el paso de parámetros por referencia?

- a. Pascal
- b. C
- c. C++
- d. FORTRAN

14. En la secuencia de programa siguiente:

```
804854e:e8 3d 06 00 00 call 8048b90 <main>
8048553:50 pushl %eax
```

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

- a. 804854e
- b. 804854f
- c. 8048b90
- d. 8048553

15. En el fragmento de código

```
804854e:e8 3d 06 00 00 call 8048b90 <main>
8048553:50 pushl %eax
```

la instrucción call suma al contador de programa la cantidad:

- a. 0000063d
- b. 08048553
- c. 0804854e
- d. 50

16. Es responsabilidad del procedimiento llamado salvaguardar los registros:

- a. %ebx, %esi, %edi
- b. %eax, %edx, %ecx
- c. %eax, %ebx, %ecx, %edx
- d. %esi, %edi

17. Al llamar a una función de 2 argumentos foo(arg1, arg2), ¿cuál es el orden correcto en el que se ejecutan las instrucciones? (suponiendo convención de llamada x86 cdecl, y que foo requiere ajustar marco de pila, esto es, salvar %ebp)

- a. push arg1, push arg2, call foo, push %ebp
- b. push arg1, push arg2, push %ebp, call foo
- c. push arg2, push arg1, call foo, push %ebp
- d. push arg2, push arg1, push %ebp, call foo

18. Cuando se ejecuta la instrucción ret al final de una subrutina:

- a. la dirección de comienzo de la pila se transfiere al puntero de pila
- b. la dirección de memoria de la instrucción ret se transfiere al contador de programa
- c. la dirección almacenada en la cima de la pila se transfiere al contador de programa
- d. la dirección almacenada en la cima de la pila se transfiere al puntero de pila

19. En IA-32 la pila es:

- a. un registro de 16 bits en el microprocesador
- b. un registro de 32 bits en el microprocesador
- c. una dirección de memoria de 32 bits almacenada en el contador de programa

- d. un conjunto de posiciones de memoria usadas para almacenar información temporal durante la ejecución del programa
- 

20. En IA-32 el puntero de pila es:

- a. un registro de 16 bits en el microprocesador
  - b. un registro de 32 bits en el microprocesador
  - c. una dirección de memoria de 32 bits almacenada en el contador de programa
  - d. un conjunto de posiciones de memoria usadas para almacenar información temporal durante la ejecución del programa
- 

21. ¿Cuál de las siguientes afirmaciones **\*NO\*** es cierta? (entender que x86=IA-32)

- a. x86-64 proporciona un espacio de memoria virtual mayor que x86
  - b. Las disciplinas de pila para x86 y x86-64 son diferentes
  - c. x86 usa %ebp como puntero base para el marco de pila
  - d. x86-64 usa %rbp como puntero base para el marco de pila
- 

22. alguna de las siguientes **\*NO\*** es una operación básica de la unidad de control:

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

23. Un computador tiene una memoria de control de 16000 palabras de 250 bits, de las que 447 son diferentes. ¿Cuántos bits ahorramos usando nanoprogramación en lugar de microprogramación?

- a. 3744250
  - b. 259206
  - c. 287935
  - d. Ninguno de los resultados anteriores es exacto
- 

24. Un sistema no segmentado tarda 20 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce (pipeline) de 4 segmentos con un ciclo de reloj de 5 ns. Cuando se procesan muchas tareas, la ganancia máxima de velocidad que se obtiene se aproxima a:

- a. 5
- b. 4
- c. 0,25

- d. 20
- 

25. En la técnica de salto retardado:

- a. el compilador puede reorganizar el código para rellenar los huecos de retardo con instrucciones útiles
  - b. el compilador no puede insertar operaciones NOP en los huecos de retardo
  - c. el salto se realiza varios ciclos antes de la instrucción de salto
  - d. las instrucciones en los huecos de retardo se ejecutan unas veces y otras no
- 

26. Al método de interacción con los periféricos, en los que el procesador vigila periódicamente el estado de los dispositivos mediante una encuesta activa se le denomina:

- a. daisy-chain
  - b. interrupción
  - c. polling
  - d. DMA
- 

27. ¿Cuántas señales de control se necesitan como mínimo para implementar un sistema de gestión de interrupciones?

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

28. Se dispone de un procesador con una frecuencia de reloj de 1 GHz. Se le conecta un dispositivo que genera 100.000 interrupciones por segundo. La rutina de servicio de interrupción ejecuta 500 instrucciones. El número medio de ciclos por instrucción es 2. ¿Qué porcentaje del tiempo dedica el procesador al dispositivo?

- a. 1%
  - b. 10%
  - c. 50%
  - d. 90%
- 

29. ¿Cuál de las siguientes afirmaciones acerca de la memoria es **\*FALSA\***?

- a. La memoria dinámica usa señales de control RAS# y CAS#
- b. Las celdas de memoria dinámica están constituidas por un transistor y un condensador

- c. Las celdas de memoria estática tienen que ser constantemente refrescadas
  - d. La memoria estática se emplea en los cachés L1 y L2
- 

30. ¿Cuál de las siguientes afirmaciones acerca de la jerarquía de memoria es **\*FALSA\***?

- a. Acceder a los discos es órdenes de magnitud más lento que acceder a la RAM
  - b. Una memoria principal constituida por la tecnología más rápida es órdenes de magnitud más cara que la DRAM
  - c. La velocidad de acceso a la memoria principal ha crecido proporcionalmente a la velocidad del procesador
  - d. Un computador puede tener una pequeña cantidad de memoria rápida además de una gran cantidad de memoria más lenta
-

Nombre:

DNI:

Grupo:

## Test de Teoría (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. El conjunto de todos los atributos de un sistema que son visibles para el programador y son necesarios para programar en lenguaje máquina se denomina:

- a. arquitectura del computador
- b. conjunto de componentes físicos del computador
- c. organización del computador
- d. repertorio de instrucciones máquina

2. ¿Cuál de las siguientes afirmaciones es cierta?

- a. la arquitectura Von Neumann de los computadores tradicionales consiste en tener almacenados los datos separados de las instrucciones en memorias distintas
- b. el registro de estado (flags) es un registro de propósito específico cuyo contenido puede ser visto directa o indirectamente por el usuario mediante el uso de ciertas instrucciones específicas
- c. la unidad de control necesita como entrada el registro contador de programa para saber cuál es la instrucción que debe ejecutar a continuación
- d. el registro de direcciones de memoria es un registro de propósito general que puede contener tanto direcciones como datos

3. 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), se almacenan a partir de la posición 0xCAFEBA0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al

consultar la media palabra de la posición 0xCAFEBA0?

- a. -1
- b. -4
- c. no se puede saber, faltan datos
- d. ninguna de las anteriores

4. 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. 0xCAFEBA0E
- d. 0xABADF00D

5. En una arquitectura de acumulador, la instrucción LOAD X:

- a. transfiere el contenido del registro X a la memoria
- b. suma M(X) al acumulador
- c. transfiere el contenido del acumulador a la posición de memoria X
- d. transfiere el contenido de la posición de memoria X al acumulador

6. Una instrucción máquina del tipo "Add M,R" podría formar parte del repertorio de

- a. una máquina pila
- b. una máquina de acumulador
- c. una máquina con arquitectura R/R
- d. una máquina con arquitectura M/M

7. ¿Cuál de los siguientes no es un modo de direccionamiento IA-32?

- a. Registro
  - b. Memoria
  - c. Cache
  - d. Inmediato
- 

8. Un bus se compone de:

- a. líneas de datos y líneas de dirección
  - b. líneas de alimentación
  - c. líneas de estado y líneas de control
  - d. líneas de control/estado, líneas de dirección y líneas de datos
- 

9. ¿Cuál de los siguientes no es un tipo de bus?

- a. Secuencial
  - b. Paralelo
  - c. E/S
  - d. Sistema
- 

10. Si en un bus de direcciones de 32 bits se decodifica parcialmente la dirección de un dispositivo de 32 posiciones usando 22 bits, ¿cuántas veces aparecerá repetido en el mapa de memoria?

- a. 10
  - b. 16
  - c. 32
  - d. 1024
- 

11. Para obtener una única velocidad comparativa final, el benchmark SPEC CPU combina las velocidades de ejecución de una serie de tests, respecto a un ordenador de referencia, usando la media...

- a. aritmética
  - b. geométrica
  - c. armónica
  - d. ponderada
- 

12. El primer computador electrónico basaba su funcionamiento en:

- a. tubos de vacío
  - b. circuitos integrados LSI
  - c. amplificadores operacionales
  - d. núcleos de ferrita
- 

13. En Linux IA-32, si gcc usa la instrucción leave se puede asegurar que en ese punto del programa

- a. correspondería emitir la secuencia de salida pop/ret, pero leave hace lo mismo y ocupa menos espacio
  - b. ya no hay registros salva-invocado que recuperar
  - c. ya no hay variables locales que destruir
  - d. ya no se hacen llamadas anidadas y por tanto no hay parámetros que ocupen espacio en pila
- 

14. Usando el repertorio IA-32, para intercambiar el valor de 2 variables (por ejemplo A: .int 1 y B: .int 2) se pueden usar...

- a. dos instrucciones mov
  - b. una instrucción mov y una instrucción lea
  - c. 3 mov, no menos (se le llama "intercambio circular")
  - d. 4 mov, no menos (debido a la arquitectura R/M)
- 

15. Respecto a registros base e índice en IA-32, la excepción es que

- a. EBP no puede ser registro base
  - b. EBP no puede ser registro índice
  - c. ESP no puede ser registro base
  - d. ESP no puede ser registro índice
- 

16. El registro SP / ESP / RSP...

- a. es un registro transparente al usuario y contiene la instrucción que se está ejecutando
  - b. es un registro de propósito específico y contiene la dirección de la cima de la pila
  - c. es un registro transparente al usuario y contiene la dirección de memoria a la que se está accediendo
  - d. es un registro de propósito específico y contiene la dirección de la siguiente instrucción a ejecutar
- 

17. Diferencias gcc Linux IA-32/x86-64: marcar la respuesta falsa

- a. los enteros largos (long) pasan de 32 a 64 bits
  - b. los punteros (void\*) pasan de 32 a 64 bits
  - c. el tipo double pasa de 4 B a 8 B
  - d. long double pasa de 10/12 B a 16 B
- 

18. ¿Cuál de los siguientes fragmentos de código deja en %eax un resultado distinto a los otros tres fragmentos?



- a. `mov $-1, %edx`  
`sub %eax, %edx`  
`mov %edx, %eax`
- b. `not %eax`  
`add $1, %eax`
- c. `xor %edx, %edx`  
`sub %eax, %edx`  
`mov %edx, %eax`
- d. `neg %eax`

19. Si A y B son dos enteros almacenados respectivamente en %eax y %ebx, ¿cuál de las siguientes implementaciones de `if (!A && !B) {...then part...}` es incorrecta?

- a. `or %ebx, %eax`  
`jne not_true`  
`...then part...`  
`not_true:`  
`...`
- b. `cmp $0, %eax`  
`jne not_true`  
`cmp $0, %ebx`  
`jne not_true`  
`...then part...`  
`not_true:`  
`...`
- c. `test %ebx, %eax`  
`jne not_true`  
`...then part...`  
`not_true:`  
`...`
- d. `test %eax, %eax`  
`jne not_true`  
`test %ebx, %ebx`  
`jne not_true`  
`...then part...`  
`not_true:`  
`...`

20. Dada la siguiente declaración en lenguaje C, una estructura de este tipo podría ocupar en un sistema Linux IA-32 o bien en uno x86-64 un total de...

```
struct a{
 int i;
 double d;
 char c;
 short s; };
```

- a. 18B
- b. 20B
- c. 22B
- d. 24B

21. En un sistema Linux x86-64, ¿cuál de las siguientes expresiones es equivalente a la expresión C `x[2] + 4)[3]`? Suponer que previamente se ha declarado `int **x`.

- a. `*((*(x + 16)) + 28)`
- b. `*(((x) + 2) + 7)`
- c. `(***(x + 2) + 4) + 3)`
- d. `*((*(x + 2) + 4) + 3)`

22. Una unidad de control microprogramada se denomina "con secuenciamiento de microinstrucciones explícito" según tenga o no tenga

- a. ROM/PLA para traducir el codop en dirección de inicio de microprograma (goto f(IR))
- b. un multiplexor para seleccionar la fuente de la dirección de la memoria de control
- c. micro-contador de programa atacando a las líneas de dirección de la memoria de control
- d. microcódigo de decodificación que analice el codop bit a bit de izquierda a derecha

23. Dado un camino de datos concreto, un posible formato de microprogramación se caracteriza como horizontal o vertical según tenga más o menos (señalar la respuesta falsa)

- a. codificación
- b. solapamiento
- c. microbifurcaciones
- d. longitud relativa de microinstrucción

24. Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señalar la respuesta falsa)

- a. registros de acoplo (coste de la segmentación)
- b. fragmentación desigual (duración desigual de etapas)
- c. riesgos (hazards)
- d. cola de instrucciones (precaptación)

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

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

26. Las técnicas principales de E/S son (señalar la respuesta falsa)

- a. DMA (por acceso directo)
  - b. E/S programada
  - c. E/S cableada (hardwired)
  - d. IRQ (por interrupciones)
- 

27. Para determinar la causa de una interrupción se pueden usar las siguientes técnicas: (señalar la respuesta falsa)

- a. múltiples líneas de interrupción INT1#, INT2#...
  - b. línea de reconocimiento INTA#
  - c. consulta de estado, o polling
  - d. interrupciones vectorizadas
- 

28. Utilizar una cache en el mismo chip del procesador:

- a. aumenta el tamaño de los bloques enviados entre cache y procesador
  - b. reduce los tiempos de ejecución
  - c. reduce el tamaño del bus
  - d. aumenta la tasa de aciertos
- 

29. En un sistema Linux IA-32, ¿cuántos enteros se podrían almacenar en una línea de cache, si la cache del procesador fuera de 4 KB, asociativa por conjuntos de 4-vías, y contuviera 4 conjuntos?

- a. 16
  - b. 32
  - c. 64
  - d. 128
- 

30. La cache con correspondencia directa se puede considerar como un caso límite de la asociativa por conjuntos, en donde...

- a. solo hay 1 línea por conjunto
  - b. solo hay 1 palabra por bloque
  - c. solo hay 1 conjunto por cache
  - d. ninguna de las anteriores
-



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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

- ¿Cuál de las siguientes afirmaciones sobre el benchmark SPEC CPU es falsa?
  - La última versión es SPEC CPU2006 V1.2 de 2011
  - Se cronometran unos 12 tests de enteros (CINT2006) y unos 17 tests de punto flotante (CFP2006)
  - Se usa como referencia un computador UltraSPARC II 300MHz, y para cada test se calcula el cociente entre el tiempo de ejecución en el computador a testear y en el de referencia
  - El resultado final es la media aritmética de las (12 ó 17) velocidades, bien sea de enteros ó de punto flotante (SPECint2006 ó SPECfp2006)
- ¿En qué generación, dentro de la historia de los computadores digitales, aparecieron la microprogramación, la segmentación de cauce, la memoria cache, los S.O. multiusuario y la memoria virtual?
  - 2ª generación (1955-65)
  - 3ª generación (1965-75)
  - 4ª generación (1975-85)
  - esas innovaciones se repartieron a lo largo de varias generaciones, no sólo una
- Respecto a tamaños de tipos integrales en x86 y x86-64, la excepción es que
  - int pasa de 4 B (x86) a 8 B (x86-64)
  - long int pasa de 4 B a 8 B
  - long long int pasa de 4 B a 8 B
  - ninguna de las anteriores
- Con el repertorio IA32, para sumar %eax y %ebx dejando el resultado en %ecx se podría hacer lo siguiente:
  - lea %eax, %ebx, %ecx
  - lea (%eax, %ebx, 1), %ecx
  - lea %ecx, [%eax, %ebx]
  - lea %ecx, %ebx, %eax
- Cuál de las instrucciones máquina siguientes es incorrecta en x86-64:
  - testl %edx, %edx
  - movl %r8, %eax
  - movl (%rdi,%rcx,4), %edx
  - addq \$1, %rcx
- Si la variable val está almacenada en ebx y la variable x está almacenada en eax, la sentencia `val ^= x;` se puede traducir a ensamblador como:
  - xorl %ebx,%eax
  - xorl %eax,%ebx
  - andl %ebx,%eax
  - testl %eax,%ebx
- Para poner a 1 el bit 5 del registro %edx sin cambiar el resto de bits podemos usar la instrucción máquina:
  - and \$32, %edx
  - and \$0x5, %edx
  - or \$0b101, %edx
  - or \$0x20, %edx
- Si tenemos un número  $n$ , de 64 bits, almacenado en la pareja de registros EDX:EAX (EDX contiene los 32 bits más

significativos y EAX los 32 bits menos significativos) y queremos realizar la división  $n / 2^{32}$  entonces:

- a. Podemos quedarnos con EDX, pero sólo en el caso de que  $n$  sea un número sin signo.
- b. Podemos quedarnos con EDX tanto si  $n$  es un número con signo como sin signo.
- c. Podemos usar las instrucciones siguientes, pero sólo en el caso de que  $n$  sea un número sin signo:

```
mov $0x100000000,%ecx
div %ecx
```

- d. Podemos usar las instrucciones siguientes tanto si  $n$  es un número con signo como sin signo:

```
mov $0x100000000,%ecx
div %ecx
```

9. ¿Dónde está ubicado el primer argumento a una función (suponer código ensamblador cdecl generado por gcc para Linux/x86) inmediatamente después de ejecutar la instrucción call?

- a. %ebp + 0x4
- b. %ebp - 0x4
- c. %esp + 0x4
- d. %esp - 0x4

10. Dado el código C siguiente:

```
struct data {
 char str[16];
};
char *f(struct data *ptr) {
 return &(ptr->str[2]);
}
```

la función se traducirá a ensamblador de x86-64 como:

- a. leaq 2(%rdi), %rax  
ret
- b. movq (,%rdi,2), %rax  
ret
- c. movq 2(%rdi), %rax  
ret
- d. leaq (,%rdi,2), %rax  
ret

11. Respecto a requisitos de alineamiento de structs en gcc/IA32 x86 y x86-64, alguna de las siguientes afirmaciones es falsa

- a. en x86 Linux alinea double a 4x (Windows no)
- b. en x86 Linux alinea long double a 4x (Windows también)
- c. en x86-64 Linux alinea double a 8x (Windows también)
- d. en x86-64 Linux alinea float a 8x (Windows también)

12. Si la estructura **struct a** ocupa un espacio de 28 bytes en memoria, ¿cuántos bytes ocupa la siguiente estructura **struct b** cuando se compila en 64 bits?

```
struct b {
 struct a a1;
 int i;
 struct a a2;
};
```

- a. 24 bytes
- b. 60 bytes
- c. 64 bytes
- d. 84 bytes

13. Respecto a los términos microinstrucción y microcódigo:

- a. Son equivalentes, llamamos microcódigo o microinstrucción a una palabra de la memoria de control
- b. Una microinstrucción está programada en microcódigo, que es un lenguaje para programar señales de control
- c. Un microcódigo controla una serie de señales de control relacionadas (por ejemplo, el código 000 para que la ALU realice la suma), y varios microcódigos juntos forman una microinstrucción
- d. Microcódigo es el contenido de la memoria de control, y una microinstrucción es una palabra de dicha memoria

14. ¿Cuál de las siguientes afirmaciones es verdadera?

- a. La unidad de control necesita como entrada el registro de estado para poder controlar la ejecución de las instrucciones de salto condicional.
- b. El registro de instrucción es un registro de propósito específico que contiene la dirección de la siguiente instrucción a ejecutar.

- c. Las únicas instrucciones en las que algunas de sus fases de ejecución conllevan un acceso a memoria son las instrucciones load y store.
  - d. El registro puntero de pila es un registro de propósito general que suele contener tanto direcciones como datos.
- 

15. Un procesador con una unidad de control microprogramada tiene una memoria de control de 300 palabras de 100 bits, de las que 200 son diferentes. Si se rediseñara como unidad de control nanoprogramada, ¿qué tamaño ocuparía la nanomemoria que contiene las microinstrucciones completas sin repeticiones?

- a. 20000 bits
  - b. 21600 bits
  - c. 22400 bits
  - d. 30000 bits
- 

16. En el pseudocódigo usado para representar las microinstrucciones, la expresión “goto f(IR)”:

- a. Se utiliza para realizar un microsalto condicional en función del registro de estado.
  - b. Realiza una llamada a una microsubrutina.
  - c. Salta a una dirección de memoria de control que depende de la instrucción máquina actual.
  - d. Permite saltar a la dirección de memoria de control del principio de un microbucle.
- 

17. Respecto a la predicción de saltos, alguna de las siguientes afirmaciones es falsa

- a. si se toma la misma decisión para cada tipo de instrucción, se trata de "predicción estática"
  - b. si la predicción cambia según la historia de ejecución del programa, se trata de "predicción dinámica"
  - c. para predicción estática, es conveniente decidir que los saltos hacia adelante siempre se cumplen, y hacia atrás no
  - d. para predicción dinámica, existen, entre otros, algoritmos de dos o cuatro estados, que requieren 1 o 2 bits por instrucción
- 

18. Respecto a los conceptos de procesamiento segmentado y superescalar, una de las siguientes afirmaciones es falsa

- a. idealmente, con el segmentado se intenta ejecutar una instrucción por ciclo, y con el superescalar más de una por ciclo (al combinarlo con segmentado)
  - b. en cualquier procesador resulta ventajoso usar una cola de instrucciones, pero es más importante para uno segmentado (fundamental) que para uno superescalar (conveniente)
  - c. por definición, un procesador superescalar debe tener varias unidades funcionales (más de una)
  - d. implícitamente, se presupone que un procesador superescalar emitirá más de una instrucción por ciclo
- 

19. Respecto a los conceptos de interfaz de dispositivo, controlador(a), puerto de E/S:

- a. La controladora o interfaz contiene los puertos necesarios para utilizar el dispositivo
  - b. Cada puerto o interfaz es una línea de comunicación con el procesador. El conjunto de ellos forma el controlador.
  - c. El puerto, o interfaz, contiene los controladores necesarios para comunicar el dispositivo con el procesador
  - d. El interfaz contiene las controladoras necesarias para conectar los puertos con el procesador
- 

20. Respecto a los conceptos de procesador de E/S, canal de E/S, dispositivos de E/S:

- a. Un procesador o canal tiene un repertorio de instrucciones específico para manejar los dispositivos E/S
  - b. Cada canal es una línea de comunicación entre el procesador y un dispositivo de E/S.
  - c. Al conjunto de conexiones entre el procesador y los dispositivos se le denomina canal de E/S (de ese ordenador)
  - d. La pregunta es capciosa, el procesador no es E/S, son otros dos componentes von Neumann distintos (ALU+UC)
- 

21. La E/S programada:

- a. Mejora las prestaciones globales del sistema respecto a la E/S por interrupciones porque la CPU tiene el control de toda la operación.
- b. Mejora las prestaciones globales del sistema respecto a la E/S por interrupciones porque la CPU es más rápida que el controlador de interrupciones y la interfaz del periférico.
- c. Empeora las prestaciones globales del sistema respecto a la E/S por interrupciones porque una

instrucción de transferencia individual de datos con la interfaz del periférico (por ej. IN, OUT) es más lenta en E/S programada que en E/S por interrupciones.

- d. Empeora las prestaciones globales del sistema respecto a la E/S por interrupciones porque la CPU debe encargarse de la sincronización con la interfaz del periférico haciendo una espera activa.

22. Una puerta AND con 16 entradas conectada a un bus de direcciones de 16 bits, con todos los bits negados excepto A10 y A6, permite seleccionar un dispositivo (con CS activa en alta) en la dirección:

- a. 0xFDDF
- b. 0xFBBF
- c. 0x0220
- d. 0x0440

23. Un computador con 15 líneas de direcciones tiene 3 módulos de memoria de  $2^{13}$  palabras y utiliza E/S mapeada en memoria. ¿Cuál es el número máximo de periféricos que pueden conectarse, si cada uno de ellos utiliza 8 direcciones?

- a.  $2^{10}$
- b.  $2^{12}$
- c.  $2^{11}$
- d.  $2^{13}$

24. Un procesador accede en el instante de tiempo  $t$  a una posición de memoria  $d(t)$ . Poco tiempo después (en el instante de tiempo  $t+k$ ) accede a la posición anterior  $d(t)-1$ . Esos dos accesos son un ejemplo de...

- a. Localidad espacial
- b. Localidad temporal
- c. No tiene nombre, ese tipo de localidad con incremento negativo ( $d(t)-1$ ) no se ha estudiado en clase
- d. No es una localidad, esa condición no guarda relación con el concepto de localidad

25. Una jerarquía de memoria consta de una cache de con una tasa de aciertos del 92% y 4 ns de tiempo de acceso y una memoria principal con una tasa de aciertos del 100% y 100 ns de tiempo de acceso. ¿Cuál es el tiempo promedio estimado de acceso a memoria?

- a. 6 ns

- b. 8 ns
- c. 10 ns
- d. 12 ns

26. Una SRAM de 1Mx4bit (4Mbit) puede venir organizada en 2048 filas, dedicando por tanto al decodificador de columnas...

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

27. Un sistema basado en un microprocesador con un bus de datos de  $n$  bits y un bus de direcciones de 16 bits direcciona la memoria por palabras de  $n$  bits y dispone de una memoria SRAM formada por dos módulos de 16 K x  $n$  cada uno. ¿Qué porcentaje del mapa de memoria está ocupado por la SRAM?

- a. 12,5%
- b. 25%
- c. 50%
- d. 100%

28. Un módulo de memoria de 16 GB está formado por varios chips DRAM de 1024Mx4. ¿Cuántos chips DRAM necesita el módulo?

- a. 4
- b. 8
- c. 16
- d. 32

29. Una cache de 256 B asociativa por conjuntos de 4-vías con líneas de 16 B tendría

- a. 4 conjuntos
- b. 16 conjuntos
- c. 64 conjuntos
- d. ningún conjunto

30. En un sistema con memoria de bytes, ¿cuál sería el tamaño de una línea de cache, si la cache del procesador fuera de 4MB, asociativa por conjuntos de 16-vías, y contuviera 4096 conjuntos?

- a. 16 B
- b. 32 B
- c. 64 B
- d. 128 B



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 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

- Respecto a direccionamiento a memoria en ensamblador IA32 (sintaxis AT&T), de la forma  $D(Rb, Ri, S)$ , sólo una de las siguientes afirmaciones es FALSA. ¿Cuál?
  - El desplazamiento  $D$  puede ser una constante literal (1, 2 ó 4 bytes)
  - EBP no se puede usar como registro base
  - ESP no se puede usar como registro índice
  - El factor de escala  $S$  puede ser 1, 2, 4, 8
- La extensión de signo a  $m$  bits de un número original  $N$  de  $n$  bits, con  $m > n$ , consiste en:
  - Realizar la operación  $2^m - N$
  - Realizar la operación  $2^m - N - 1$
  - Incrementar la cantidad de bits a  $m$  preservando el signo y el valor del número.
  - Incrementar la cantidad de bits a  $m$  rellenando con unos por la izquierda.
- En IA32, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto al resto?
  - sub %edi, %edi  
adc \$0xFFFFFFFF, %edi
  - mov \$-1, %edi
  - mov \$-1, %edi  
add \$0, %edi
  - mov \$0, %edi
- Si %rsp vale 0xdeadbeefdeadd0d0, ¿cuál será su nuevo valor después de que se ejecute pushq %rbx?
  - 0xdeadbeefdeadd0d4
  - 0xdeadbeefdeadd0d8
  - 0xdeadbeefdeadd0cc
  - 0xdeadbeefdeadd0c8
- ¿Cómo se devuelve en ensamblador x86-64 Linux gcc el valor de retorno de una función long int al terminar ésta?
  - La instrucción RET lo almacena en un registro especial de retorno.
  - Por convención se guarda en %eax.
  - Se almacena en pila justo encima de los argumentos de la función.
  - Ninguna de esas formas es la correcta.
- Comparando las convenciones de llamada de gcc Linux IA32 con x86-64 respecto a registros
  - En IA32 %ebx es salva-invocante, pero en x86-64 %rbx es salva-invocado
  - En IA32 %ecx es salva-invocante, y en x86-64 %rcx es salva-invocante también
  - En IA32 %esi es salva-invocado, y en x86-64 %rsi es salva-invocado también
  - En IA32 %ebp es especial (marco de pila), y en x86-64 %rbp también
- Son funciones de la unidad de control:



- a. la codificación de las instrucciones máquina
  - b. la lectura de memoria principal de la instrucción apuntada por el  $\mu$ PC
  - c. el secuenciamiento de las instrucciones máquina
  - d. todas las respuestas son ciertas
- 

8. Respecto a MBR y MAR

- a. Ambos son accesibles por el programador
  - b. MAR contiene el dato/instrucción que se leerá o escribirá en memoria
  - c. MAR requiere menos señales de control que MBR
  - d. Ambos permiten guardar información sobre el marco de pila
- 

9. Una instrucción máquina puede desglosarse en las siguientes operaciones elementales:

`sp := sp - 1; m[sp] := pc; pc := x`

Probablemente se trate de una instrucción de:

- a. apilamiento
  - b. llamada a subrutina
  - c. carga local
  - d. almacenamiento local
- 

10. En una unidad de control

microprogramada con formato de microinstrucciones vertical, un subcampo que deba especificar 16 señales de control codificadas de tal forma que pueda activarse sólo una o ninguna habrá de tener una anchura mínima de

- a. 4 bits
  - b. 5 bits
  - c. 16 bits
  - d. 17 bits
- 

11. Dado un camino de datos concreto, un posible formato de microprogramación se caracteriza como horizontal o vertical según tenga más o menos (señalar la respuesta falsa)

- a. codificación
- b. solapamiento

c. microbifurcaciones

d. longitud relativa de microinstrucción

---

12. El control residual se utiliza para:

- a. reducir el tiempo de ejecución de las instrucciones máquina
  - b. eliminar los bits residuales de la ejecución de las microinstrucciones
  - c. reducir el tamaño de la memoria de control
  - d. ninguna de las anteriores es cierta
- 

13. Un procesador está segmentado en las etapas F, D, E, M y W. Cada una de ellas consume un tiempo  $t$ . La aceleración ideal (si no hay riesgos) al ejecutar  $n$  instrucciones respecto a un procesador no segmentado será:

- a.  $5n / (4+n)$
  - b.  $(4+n) / 5t$
  - c.  $4n / (5+n)$
  - d.  $(5+n) / 4t$
- 

14. En un procesador con segmentación de cauce, aumentar el número de etapas (p.ej. de 2 a 4, o de 4 a 8), tiene en general como consecuencia:

- a. Un incremento de las prestaciones
  - b. Un mayor retraso en la ejecución de los programas debido al incremento del número de etapas
  - c. Una disminución en la posible dependencia de datos
  - d. Una disminución de la máxima frecuencia de reloj a la que puede operar el cauce
- 

15. En la secuencia de instrucciones siguiente, siendo el primer registro el destino, ¿cuántos riesgos se dan?

`sub r2,r1,r3`  
`or r8,r6,r2`

- a. Un riesgo estructural
  - b. Un riesgo por dependencia de datos
  - c. Un riesgo estructural y dos por dependencia de datos
  - d. Dos riesgos por dependencia de datos y uno de control
-

16. La precaptación (cola de instrucciones) está relacionada con...

- a. Los riesgos estructurales (intenta evitar el efecto de un fallo de cache)
- b. Los riesgos de (dependencia de) datos (intenta que el dato esté disponible anticipadamente)
- c. Los riesgos de control (intenta determinar de antemano el flujo de control)
- d. Los riesgos de transferencia (intenta agrupar las posibles transferencias de un conjunto de instrucciones)

17. Respecto a la segmentación, ¿cuál de las siguientes afirmaciones es falsa?

- a. La técnica de register forwarding habilita una serie de caminos (buses) que se añaden al cauce para permitir que los resultados de una etapa pasen como entradas a la etapa donde son necesarias
- b. La reorganización del código y la introducción de instrucciones nop permite evitar dependencias de datos
- c. Retrasar la fase de decisión saltar/no saltar de las instrucciones de salto condicional contribuye a mejorar el rendimiento del procesador
- d. Cuantas más etapas tenga un cauce, más instrucciones se estarán ejecutando en distintas fases y más posibilidades se presentan de que existan riesgos entre ellas

18. ¿Cuál de los siguientes modos de direccionamiento es menos preferible para un procesador de 32 bits y con tamaño de instrucción de 32 bits?

- a. registro
- b. indexado
- c. indirecto a través de registro
- d. directo (o absoluto)

19. La conexión entre un dispositivo de E/S y el procesador mediante bus:

- a. Es difícil de expandir
- b. Permite conectar en paralelo varios dispositivos

c. Requiere mucha circuitería

d. Requiere multiplexores y demultiplexores para las señales de datos

20. El fragmento de código ensamblador de un microprocesador de 8 bits

```
lds IOBuf ; Apuntar puntero pila a
 ; ...área mem.intermedia
ldx Count ; Inicializar X-contador
poll lda a Status; Leer estado en A
bpl poll ; Signo(A)!=1 => repetir
lda a Data ; Leer dato en A
psh a ; Transferir dato a pila
dex ; Decrementar contador X
bne poll ; Seguir leyendo si X!=0
```

corresponde a:

- a. Entrada programada con consulta de estado
- b. Salida programada sin consulta de estado
- c. Entrada programada sin consulta de estado
- d. Salida programada con consulta de estado

21. En la E/S controlada por interrupciones:

- a. El controlador de DMA transfiere bloques de datos por el bus del sistema.
- b. El controlador de DMA envía una petición de interrupción a la CPU.
- c. La CPU lee y comprueba el estado de los dispositivos de E/S (en el caso de consulta de estado).
- d. La CPU transfiere el control a una rutina de servicio cuando recibe una interrupción.

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

- a. Todas las interrupciones enmascarables
- b. Las interrupciones de inferior o igual prioridad a una dada
- c. Determinados niveles de interrupción de forma selectiva
- d. Las interrupciones software

23. Con nueve controladores de interrupciones 8259 se pueden manejar exactamente:

- a. 8 niveles de prioridad
  - b. 16 niveles de prioridad
  - c. 24 niveles de prioridad
  - d. Ninguna de las anteriores es cierta
- 

24. ¿Cuál de los siguientes es un registro de un controlador de DMA?

- a. IR (Instruction Register)
  - b. PC (Program Counter)
  - c. SP (Stack Pointer)
  - d. WC (Word Count)
- 

25. Respecto al refresco de memorias DRAM, ¿cuál de las siguientes afirmaciones es falsa?

- a. Una operación de refresco consiste en dar un impulso /CAS junto con una dirección de columna.
  - b. Los chips DRAM refrescan automáticamente la fila accedida en cualquier ciclo de lectura o escritura.
  - c. Se precisa una circuitería auxiliar, externa al chip DRAM o integrada en él, que produzca ciclos de refresco.
  - d. Los ciclos de refresco deben producirse cada pocos ms (milisegundos).
- 

26. La tasa de aciertos  $A_i$  del nivel  $i$  de una jerarquía de memoria no depende de:

- a. La capacidad (tamaño)  $s_i$  del nivel  $i$ .
  - b. La estrategia de administración de memoria.
  - c. La unidad de la transferencia de información  $x_i$  entre el nivel  $i$  y el  $i+1$ .
  - d. El ancho de banda  $b_i$  del nivel  $i$ .
- 

27. La política de correspondencia de una memoria cache con 1 único conjunto es:

- a. Directa
  - b. Totalmente asociativa
  - c. Asociativa por conjuntos con una única línea
  - d. Asociativa por conjuntos de una única vía
- 

28. La política de correspondencia de una memoria cache con la mitad de conjuntos que líneas es:

- a. Asociativa por conjuntos de 2 vías
  - b. Totalmente asociativa de media vía
  - c. Asociativa por conjuntos con 2 líneas
  - d. Directa con 2 líneas
- 

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

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

30. Para diseñar una memoria con ancho de palabra  $k \times m$  (y mismo  $n^\circ$  palabras que los módulos) a partir de módulos con ancho de palabra  $m$ , se utilizan  $k$  módulos

- a. repartiendo las líneas de datos entre los  $k$  módulos: el primero se conecta a  $D_0 \dots D_{k-1}$ , el segundo a  $D_k \dots D_{2k-1}$ , etc
  - b. repartiendo las líneas de dirección: el 1º se conecta a  $A_0 \dots A_{k-1}$ , el 2º a  $A_k \dots A_{2k-1}$ , etc
  - c. repartiendo líneas datos: el 1º se conecta a  $D_0 \dots D_{m-1}$ , el 2º a  $D_m \dots D_{2m-1}$ , etc
  - d. repartiendo líneas dirección: el 1º se conecta a  $A_0 \dots A_{m-1}$ , el 2º a  $A_m \dots A_{2m-1}$ , etc
-



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. ¿Cuál es el valor mínimo (más negativo) que puede tomar un entero de 32 bits en complemento a dos?

- a.  $-2^{32}$
- b.  $-2^{32} + 1$
- c.  $-2^{31}$
- d.  $-2^{31} + 1$

2. Un datapath con bus de direcciones de 32 bits y bus de datos de 16 bits tiene un registro de 16 bits conectado al bus de datos y a la unidad de control. Puede tratarse del registro

- a. IR
- b. SP
- c. MAR
- d. PC

3. ¿Cuál de las siguientes características sobre RISC es **\*FALSA\***?

- a. Para acelerar un procesador RISC se deberían emplear técnicas de segmentación.
- b. Las instrucciones máquina en un procesador RISC deberían ser complejas y potentes.
- c. La decodificación de las instrucciones debe ser simple: un procesador RISC debería emplear pocos formatos de instrucción.
- d. La unidad de control de un procesador RISC debería ser cableada, no microprogramada.

4. ¿Cuál de las siguientes instrucciones máquina copia en el registro EDI la dirección efectiva resultante de la operación  $EAX * 8 + EBX$ ?

- a. `movl (%ebx, %eax, 8), %edx`
- b. `movl 8(%edx, %eax), %edx`
- c. `leal (%ebx, %eax, 8), %edx`
- d. `leal 8(%edx, %eax), %edx`

5. En el contexto general del lenguaje máquina, el acrónimo ISA suele referirse a:

- a. Internal Standard Architecture
- b. Integrated Set Assembly
- c. Instruction System Architecture
- d. Instruction Set Architecture

6. En una suma de dos números en complemento a dos, se produce desbordamiento cuando

- a. Sumamos dos positivos y el resultado es negativo o bien sumamos dos negativos y el resultado es positivo.
- b. Sumamos dos positivos y el resultado es positivo.
- c. Sumamos un número positivo y uno negativo.
- d. Sumamos dos negativos y el resultado es negativo.

7. Usando el repertorio IA32, para intercambiar el valor de 2 registros se pueden usar...

- a. 4 mov, no menos (debido a la arquitectura R/M)
- b. 3 mov, no menos (se le llama "intercambio circular")
- c. dos instrucciones mov
- d. una instrucción mov y una instrucción lea

8. Al ejecutar el fragmento de código:

```
leal -1(%eax), %edx
cmpl $9, %edx
ja .L2
```

se salta a .L2 si el contenido del registro %eax:

- a. es menor o igual que 1
- b. es mayor o igual que 10
- c. está fuera del intervalo [1,10]
- d. está dentro del intervalo [1,10]

9. ¿Cuál de las siguientes instrucciones convierte  $\%eax = 5 * \%eax$ ?

- 1) `mov 4(%eax, %eax), %eax`
- 2) `lea 4(%eax, %eax), %eax`

- a. Sólo la 2
- b. Sólo la 1
- c. Ambas, la 1 y la 2
- d. Ninguna de las dos

10. Si el registro **r12b** contiene la variable booleana **cond**, y **rax** la variable **valor**, la secuencia de instrucciones:

```
testb %r12b, %r12b
movq $13, %rax
cmovle $17, %rax
```

realiza la operación:

- a.  $\text{valor} = \text{cond} ? 13 : 17;$
- b.  $\text{valor} = 17;$
- c.  $\text{valor} = \text{cond} ? 17 : 13;$
- d.  $\text{valor} = 13;$

11. En una matriz declarada como “`int a[n][n];`” en lenguaje C...

- a. los n elementos de una columna se almacenan en memoria de manera contigua
- b. los n elementos de una fila se almacenan en memoria de manera contigua
- c. podría haber huecos de relleno al final de cada columna para alineamiento, dependiendo de n
- d. podría haber huecos de relleno al final de cada fila para alineamiento, dependiendo de n

12. ¿Cuáles de las siguientes señales son entradas a la unidad de control?

- a. El contenido del contador de programa
- b. Las señales de habilitación de buffers triestado entre registros y buses

c. El contenido del registro de instrucción

d. Las señales de control de la ALU

13. Una CPU con bus de direcciones de 64 bits y bus de datos de 32 bits tiene un registro de 64 bits conectado al bus de direcciones de la memoria. Probablemente se trata del registro

- a. IR
- b. MBR
- c. Acumulador
- d. MAR

14. En la secuencia de instrucciones siguiente, siendo el primer registro el destino, ¿cuántos riesgos se dan?

```
sub r2, r1, r3
or r8, r6, r1
```

- a. Un riesgo por dependencia de datos
- b. Un riesgo estructural
- c. Un riesgos por dependencia de datos y uno de control
- d. Ninguno

15. Un sistema no segmentado tarda 10 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce (pipeline) con un ciclo de reloj de 5 ns. Cuando se procesan muchas tareas, la ganancia máxima de velocidad que se obtiene se aproxima a:

- a. 2
- b. 5
- c. 10
- d. 20

16. ¿Cuál de las siguientes técnicas no se puede usar para determinar la causa de una interrupción?

- a. línea de reconocimiento INTA#
- b. interrupciones vectorizadas
- c. consulta de estado, o polling
- d. múltiples líneas de interrupción INT1#, INT2#... con un dispositivo en cada línea

17. Señale cuál de las siguientes opciones es una técnica para llevar a cabo la transferencia de datos entre el computador y los dispositivos de E/S externos:

- a. E/S por flanco
- b. E/S programada
- c. Acceso indirecto a memoria
- d. E/S por nivel

18. ¿Cuál de las siguientes afirmaciones es **\*FALSA\***?

- a. La operación de lectura de una celda DRAM es destructiva
- b. Las memorias DRAM son en general más lentas que las SRAM
- c. Una celda DRAM no pierde la información al desconectar la alimentación
- d. Las memorias DRAM presentan generalmente una capacidad de almacenamiento mayor que las SRAM

19. La memoria DRAM:

- a. Se inventó en la década de los 2000
- b. Necesita 6 transistores por cada celda
- c. Se denomina dinámica porque su contenido puede alterarse, al contrario que la SRAM
- d. Es más densa que la memoria SRAM

20. Indique cuál es la dirección de la instrucción **mov** en el siguiente desensamblado, donde se ha borrado parte de la dirección

```
0804xxxx: 74 12 je 08048391
0804xxxx: b8 00 00 00 00 mov $0, %eax
```

- a.  $08048391 + 12 = 08048403$
- b.  $08048391 - 12 = 08048379$
- c. 0804837d
- d. 0804837f

21. Dada la siguiente declaración en lenguaje C, una estructura de este tipo podría ocupar, bien sea en un sistema Linux IA32 o bien en uno x86-64, un total de...

```
struct a{
 int i;
 double d;
 char c;
 short s; };
```

- a. 18 B
- b. 20 B
- c. 22 B
- d. 24 B

22. Dado un camino de datos concreto, un posible formato de microprogramación se caracteriza como horizontal o vertical según tenga más o menos (señalar la respuesta **\*FALSA\***)

- a. codificación
- b. solapamiento
- c. microbifurcaciones

d. longitud relativa de microinstrucción

23. En una unidad de control microprogramada con formato de microinstrucciones vertical, un subcampo que deba especificar 16 señales de control codificadas de tal forma que pueda activarse sólo una o ninguna habrá de tener una anchura mínima de

- a. 4 bits
- b. 5 bits
- c. 16 bits
- d. 17 bits

24. Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señale la respuesta **\*FALSA\***)

- a. registros de acoplo (coste de la segmentación)
- b. fragmentación desigual (duración desigual de etapas)
- c. riesgos (*hazards*)
- d. cola de instrucciones (precaptación)

25. Un procesador de 1 GHz sin segmentación de cauce tarda 4 ns en ejecutar 4 instrucciones. ¿Cuánto tardaría en ejecutar 9 instrucciones una versión de dicho procesador con segmentación de cauce de 4 etapas si no existiera ningún retraso en ninguna de las instrucciones?

- a. 2 ns
- b. 3 ns
- c. 4,5 ns
- d. 9 ns

26. Respecto al salto retardado y al salto anulante, ¿cuál permite que se ejecute la siguiente instrucción, y cuál no?

- a. el retardado ejecuta la siguiente instrucción (con el correspondiente retraso), el anulante no la ejecuta (de hecho la anula)
- b. el retardado la ejecuta sólo si se cumple la condición de salto, el anulante sólo si no se cumple
- c. el retardado la ejecuta sólo si no se cumple la condición de salto, el anulante no la ejecuta nunca
- d. el retardado la ejecuta siempre, el anulante la ejecuta sólo si se cumple la condición de salto

27. Sobre la E/S mapeada en memoria podemos decir que:

- a. Usa el espacio común de direccionamiento para acceder a puertos de E/S
  - b. La CPU necesita el pin IO/M#
  - c. Dispone de instrucciones especiales de E/S
  - d. Todas las respuestas anteriores son falsas
- 

28. ¿Cuál de las siguientes tareas no es responsabilidad de un circuito de interfaz o controlador de periféricos sencillo?

- a. Adaptar el formato de las señales
  - b. Ajustar la temporización entre el procesador y los dispositivos de E/S
  - c. Recibir señales de control desde el procesador
  - d. Ejecutar el programa de transferencia de información entre el procesador y los dispositivos de E/S
- 

29. ¿Cuál es el ancho del bus de direcciones de un chip DRAM de 1G palabra, siendo la longitud de palabra de 16 bits?

- a. 20
  - b. 16
  - c. 30
  - d. 15
- 

30. Sea un computador de 32 bits que dispone de una memoria cache de 512 KB y líneas de 64 bytes. ¿Cuántas líneas tiene la cache?

- a. 64
  - b. 1024
  - c. 8192
  - d. 65536
-

|                |               |
|----------------|---------------|
| <b>Nombre:</b> |               |
| <b>DNI:</b>    | <b>Grupo:</b> |

## 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. ¿Cuál es el valor mínimo (más negativo) que puede tomar un entero de 32 bits en complemento a dos? (el punto se usa como separador)
  - a. -2.147.483.647
  - b. -2.147.483.648
  - c. -4.294.967.295
  - d. -4.294.967.296

---

2. ¿Cómo se representa el valor -1 como entero con signo en 14 bits?
  - a. 0xFFFF
  - b. 0x3FFF
  - c. las respuestas anteriores no son válidas porque usan hexadecimal; habría que usar binario
  - d. no se puede porque 14 no es múltiplo de 4

---

3. ¿Cuál de las siguientes no es una unidad de la arquitectura Von Neumann?
  - a. Unidad central de proceso
  - b. Memoria principal
  - c. Sistema de entrada/salida
  - d. Núcleo del sistema operativo

---

4. ¿Cuál de las siguientes afirmaciones es verdadera?
  - a. La arquitectura Von Neumann en la que se basan los computadores tradicionales consiste en tener los datos separados de las instrucciones en memorias distintas.
  - b. El registro de estado es un registro transparente al usuario, ya que éste no puede utilizarlo en las instrucciones máquina.
  - c. El registro de instrucción es un registro transparente al usuario, ya que éste no puede utilizarlo en las instrucciones máquina.
  - d. La unidad de control necesita como entrada el registro contador de programa, para saber cuál es la instrucción que debe ejecutar a continuación.

---

5. ¿Qué es el lenguaje máquina?
  - a. Conjunto de datos binarios que representan señales eléctricas internas de la unidad de control de un microprocesador.
  - b. Conjunto de sentencias en un lenguaje escrito que se utilizan para generar programas codificados en lenguaje ensamblador.
  - c. Conjunto formado por las siglas asignadas a las instrucciones del repertorio de instrucciones más un conjunto de directivas que facilitan la generación del código binario.
  - d. Conjunto de instrucciones en formato binario que entiende un determinado procesador.

---

6. ¿Cuál de los siguientes elementos no forma parte de la Arquitectura del Repertorio de Instrucciones (ISA)?
  - a. Descripción del espacio de direccionamiento de la memoria y de la E/S.

- b. Descripción de los campos de bits en los que están organizadas conceptualmente las microinstrucciones.
  - c. Descripción de los registros de datos, registros de estado y control.
  - d. Descripción de los tipos de datos sobre los que opera el lenguaje máquina.
- 

7. ¿Cuál de las siguientes definiciones de modos de direccionamiento es **\*incorrecta\***?
- a. Inmediato: el dato está codificado dentro de la propia instrucción, en uno de los campos en los que se divide el formato de instrucción.
  - b. Registro: el dato se encuentra en un registro de propósito general.
  - c. Directo: la dirección se calcula como la suma de un dato codificado en la propia instrucción y el contenido de un registro de propósito general.
  - d. Indirecto: el dato está contenido en una posición de memoria que es apuntada por un registro de propósito general.
- 

8. Respecto a los registros enteros en arquitectura IA32 de 32bits (x86)
- a. Se puede acceder a 8, y en cada uno de esos 8 registros enteros, se puede acceder a todos los 32 bits (p.ej. EAX), a los 16 bits menos significativos (p.ej. AX) ó a los 8 LSBs (p.ej. AL)
  - b. Se puede acceder a 8, y en cada uno de esos 8 registros enteros, se puede acceder a todos los 32 bits (p.ej. EAX), a los 16 bits menos significativos (p.ej. AX), a los 8 LSBs (p.ej. AL) o a los bits 8-15 (p.ej. AH)
  - c. Se puede acceder a 8 de cada tamaño (32, 16, 8 bits), aunque no todos los registros tienen versión de 8 y 16 bits
  - d. No hay distintos tamaños, son sólo registros de 32 bits, como corresponde a dicha arquitectura
- 

9. ¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar):
- a. `movw %dx, (%eax)`
  - b. `movb $0xFF, (%dl)`
  - c. `movswl (%eax), %edx`
  - d. `movzbl %dl, %eax`
- 

10. ¿Qué modo de direccionamiento usa el operando fuente en la instrucción `mov (%rcx), %al`?

- a. Directo a memoria
  - b. Indirecto a memoria a través de registro
  - c. Registro
  - d. Inmediato
- 

11. Si el contenido del registro `%rax` es `0x10` antes de ejecutar la instrucción `shl $0xc, %rax`, ¿cuánto es su contenido tras ejecutarla?

- a. `0x10000`
  - b. `0x1000`
  - c. `0x4000`
  - d. `0x800`
- 

12. En el fragmento de código

```
804854e:e8 3d 06 00 00 call 8048b90
8048553:50 pushl %eax
```

la instrucción `call` suma al contador de programa la cantidad:

- a. `0x0000063d`
  - b. `0x08048553`
  - c. `0x0804854e`
  - d. `0x50`
- 

13. ¿Cuál de los siguientes registros x86-64 es distinto del resto en convenio de uso? (salva-invocante/invocado)

- a. `RBX`
  - b. `RCX`
  - c. `RSI`
  - d. `R8`
- 

14. Respecto a requisitos de alineamiento de structs en gcc/IA32 x86 y x86-64, una de las siguientes afirmaciones es **\*FALSA\***

- a. en x86 Linux alinea `double` a 4x
  - b. en x86 Linux alinea `long double` a 4x
  - c. en x86-64 Linux alinea `double` a 8x
  - d. en x86-64 Linux alinea `float` a 8x
- 

15. Se definen las variables, unión y función C siguientes:

```
float f1;
unsigned u1=0x80000000;
float f2;
typedef union {
 float f;
 unsigned u;
} bit_float_t;
```



```
float bit2float(unsigned u) {
 bit_float_t arg;
 arg.u = u;
 return arg.f;
}
```

¿Cuál afirmación es verdadera?

- Si asignamos `f1=bit2float(u1)`; entonces `f1== 2147483648.00`
- Si asignamos `f1=bit2float(u1)`; entonces `f1== -0.0`
- Si asignamos `f2= (float)u1` ; entonces `f2== 4294967296.00`
- Si asignamos `f2= float(u1)`; entonces `f2== 0.0`

16. Convertir un vector de 100 shorts de formato little endian a formato big endian consiste en:

- Intercambiar el elemento 0 del vector con el 99, el 1 con el 98, el 2 con el 97 y así sucesivamente.
- Intercambiar el elemento 0 del vector con el 1, el 1 con el 2, el 3 con el 4 y así sucesivamente.
- Cambiar el orden en memoria de los 4 bytes de cada elemento, es decir, en cada elemento intercambiar el byte 0 con el 3 y el 1 con el 2.
- Cambiar el orden en memoria de los 2 bytes de cada elemento, es decir, en cada elemento intercambiar el byte 0 con el 1.

17. Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señalar la respuesta **\*FALSA\***)

- registros de acoplo (coste de segmentación)
- fragmentación desigual (duración de etapas)
- riesgos (hazards)
- cola de instrucciones (precaptación)

18. Un procesador de 1GHz tarda 4ns en realizar 4 instrucciones sin realizar segmentación de cauce. ¿Cuanto tardaría en realizar 9 instrucciones con segmentación de cauce de 4 etapas si no existiera ningún retraso en ninguna de las instrucciones?

- 2 ns
- 3 ns
- 4.5 ns
- 9 ns

19. ¿Qué es un controlador de E/S?

- Un circuito electrónico que implementa la memoria del computador.
- Un circuito impreso del tipo DIMM.
- Un circuito electrónico que puede guardar temporalmente datos enviados desde el procesador al periférico o viceversa.
- Un bus que permite interconectar distintos periféricos entre sí.

20. Respecto a la interfaz de E/S, ¿cuál de las siguientes afirmaciones es **\*FALSA\***?

- Involucra tareas que se pueden implementar parte en hardware y parte en software.
- Permite configurar el funcionamiento del periférico en un momento determinado, y además conocer su estado.
- Puede guardar temporalmente en registros internos tanto datos generados por el periférico para ser enviados al procesador, como datos que son enviados desde el procesador al periférico.
- Una interfaz de entrada recibe los datos desde el procesador y los transforma y envía al periférico en formato digital.

21. ¿Cuál de las siguientes características corresponde a E/S mapeada en memoria?

- Determinadas zonas del espacio de direccionamiento del procesador se asignan por convenio a controladores de E/S.
- Un ejemplo de mecanismo de E/S mapeada en memoria es la instrucción `IN` de los procesadores Intel.
- Una misma dirección se usa alternativamente para E/S y para memoria en distintos momentos de ejecución de un programa.
- Un pin `IO/M#` del procesador permite distinguir si accedemos a E/S o a memoria.

22. ¿Cuál de las siguientes afirmaciones es **\*FALSA\***?

- La consulta del estado del dispositivo por parte de la CPU se suele hacer con E/S programada (salvo con dispositivos que siempre están listos para transferir) y con E/S por IRQ (cuando se usa polling para determinar el origen de la IRQ).
- Se suele avisar a la CPU (mediante una IRQ) de que debe realizar alguna tarea, tanto en E/S por IRQ (obligatoriamente, la tarea es la transferencia) como en E/S por DMA (opcionalmente, el controlador DMA puede avisar de que acabó).

- c. Sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo de E/S.
  - d. Sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S.
- 

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

- a. Todas las interrupciones enmascarables
  - b. Las interrupciones de inferior o igual prioridad a una dada
  - c. Determinados niveles de interrupción de forma selectiva
  - d. Las interrupciones software
- 

24. ¿Cuál de los siguientes es un registro de un controlador de DMA?

- a. IR (Instruction Register)
  - b. PC (Program Counter)
  - c. SP (Stack Pointer)
  - d. WC (Word Count)
- 

25. El ancho de banda de memoria es:

- a. el número de bits que se pueden transferir entre ésta y la CPU en paralelo en una sola operación de lectura o escritura
  - b. el número de bytes que se pueden leer/escribir por unidad de tiempo
  - c. el tiempo que se tarda en transferir una palabra entre memoria y CPU
  - d. el intervalo de frecuencias de reloj permitidas entre memoria y CPU
- 

26. ¿Cuál de las siguientes afirmaciones sobre la memoria DRAM es **\*incorrecta\***?

- a. El principio de funcionamiento de los circuitos electrónicos de la memoria DRAM consiste en cargar o descargar un transistor.
  - b. Los bits de memoria se organizan dentro del circuito integrado en forma de matriz de celdas de bit, en la que se pueden diferenciar filas y columnas.
  - c. Un transistor en cada celda permite o no permite circular la corriente eléctrica a través de él. Cuando el transistor no deja pasar la corriente, la información queda almacenada durante un tiempo en el condensador. Cuando el transistor deja pasar corriente, el condensador se carga o se descarga.
  - d. Cada celda de memoria está compuesta por un transistor y un condensador y almacena un bit de información.
- 

27. ¿Cuál de las siguientes afirmaciones sobre memorias es correcta?

- a. La memoria cache se construye con tecnología electrónica de tipo DRAM.
  - b. La memoria principal se construye con tecnología electrónica de tipo SRAM.
  - c. Los chips de memoria DRAM se conectan entre sí en un circuito impreso constituyendo lo que se denomina DIMM.
  - d. Las memorias SRAM no son volátiles; es decir, cuando no están alimentadas eléctricamente siguen guardando toda la información.
- 

28. ¿Cuál de los siguientes grupos de señales no se usa en un chip de memoria SRAM?

- a. Selección de filas RAS# y de columnas CAS#.
  - b. Datos  $D_{n-1}-D_0$ .
  - c. Direcciones  $A_{n-1}-A_0$ .
  - d. Selección de chip CS# y habilitación de escritura WE#.
- 

29. ¿Qué es el tiempo de refresco de memoria?

- a. La cantidad de datos transferidos por segundo entre dos niveles de la jerarquía de memoria.
  - b. El tiempo que se tarda en recargar los condensadores que almacenan los bits de datos para que no se pierdan.
  - c. El tiempo que transcurre entre la solicitud de una operación en un determinado nivel de la jerarquía de memoria (lectura o escritura) y la recepción de todos los datos solicitados.
  - d. El tiempo que tiene que transcurrir entre sucesivas solicitudes de acceso a un determinado nivel de la jerarquía de memoria.
- 

30. ¿Cuál de las siguientes políticas está **\*menos\*** relacionada con la jerarquía memoria?

- a. Política de escritura: determina cómo se actualiza el nivel de la memoria  $i+1$  cuando se ejecutan instrucciones de almacenamiento en el nivel  $i$ .
  - b. Política de reemplazo: qué bloque se tiene que sustituir (reemplazar) cuando se trae un bloque desde otro nivel.
  - c. Política de planificación: en qué orden se ejecutarán los procesos pendientes.
  - d. Política de colocación: dónde se almacena un bloque de datos dentro de la memoria.
-



|         |        |
|---------|--------|
| 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. ¿Cuál es el complemento a 2 del número binario 1110 1101 1000?

- a. 0001 0010 0110
- b. 0001 0010 0101
- c. 0001 0010 0111
- d. 0001 0010 1000

2. Escoger de entre las 4 operaciones la de mayor valor que pueda calcularse con enteros de 4B con signo sin problemas

- a. 100.000.000 + 100.000.000
- b. 300.000.000 + 300.000.000
- c. 1.000.000.000 + 1.000.000.000
- d. 3.000.000.000 + 3.000.000.000

3. Si **rcx** vale -1, tras ejecutar las instrucciones

```
rol $1, %cl
rcr $2, %rcx
```

el nuevo valor de **RCX** y del flag **CF** es

- a. hay algún fallo de sintaxis o gramática en esas instrucciones
- b.  $RCX \neq -1$ , **CF** mantiene su valor
- c.  $RCX = -1$ , **CF** = 1
- d. no se puede marcar ninguna de las opciones anteriores

4. El registro **RAX** contiene el número binario 111111110111000010. ¿Cuál será su contenido tras ejecutar la instrucción **sar \$1,%ax**?

- a. 0x3fde1
- b. 0x7fde1
- c. 0xffde1

d. 0xffffffffffffde1

5. Para comprobar si el contenido del registro **RDX** es 0 (y posiblemente saltar a continuación usando la instrucción **je**), el compilador **gcc** genera:

- a. `cmpq %rdx, %rdx`
- b. `testq %rdx`
- c. `testq %rdx, %rdx`
- d. `cmpq %rdx`

6. 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 salto de la instrucción **je** en el siguiente desensamblado, donde se ha tachado precisamente dicha dirección.

```
40042f: 74 f4 je xxxxxx
400431: 5d pop %rbp
```

- a. 400431
- b. 400525
- c. 400425
- d. 40043d

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

- a. Un salto incondicional, según la condición expresada en el código C, y otro salto incondicional
- b. Un salto condicional, según la condición opuesta a la del código C, y otro salto condicional

- c. Una instrucción de movimiento condicional, pero sólo si el procesador es Pentium Pro/II o superior
- d. Una instrucción de movimiento incondicional, pero sólo si el S.O. es de 64bits

---

8. La instrucción **cmovb %rdx,%rax**

- a. copia el byte bajo de rdx en el byte bajo de rax
- b. copia en rax el byte de memoria apuntado por la dirección contenida en rdx
- c. copia en rax el contenido de rdx si rax es menor que rdx

d. copia en rax el contenido de rdx si CF= 1

---

9. Uno de los puntos clave de la traducción que gcc hace de una construcción switch-case de lenguaje C a lenguaje ensamblador es...

- a. el salto condicional hacia atrás
- b. el salto relativo a contador de programa
- c. el salto directo

d. el salto indirecto

---

10. El procesador utiliza el puntero de pila...

- a. En las instrucciones de llamadas y retornos de subrutinas
- b. En todo tipo de instrucciones de saltos, incluyendo llamadas y retornos a subrutinas
- c. En todas las instrucciones que tengan al menos dos accesos a memoria
- d. En todas las instrucciones

---

11. ¿Cuál de las siguientes instrucciones situada al principio de una función se utilizará probablemente para crear espacio en la pila para variables locales sin inicializar?

- a. sub      \$0x30, %rsp
- b. add      \$0x30, %rsp
- c. sub      \$0x30, %rbp
- d. add      \$0x30, %rbp

---

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

- a. RAX es un registro salva-invocante, por eso en cualquier función hay que salvarlo antes de modificarlo
- b. R10 es un registro salva-invocante, por eso si es necesario hay que salvarlo antes de llamar a función

- c. R11 es un registro salva-invocado, por eso en cualquier función hay que salvarlo antes de modificarlo
- d. RBP es un registro salva-invocado, por eso si es necesario hay que salvarlo antes de llamar a función

---

13. Un procedimiento llamado por una instrucción call debe guardar y restaurar los registros siguientes siempre que los altere:

- a. %rsi, %rdi
- b. %rax, %rbx, %rcx, %rdx
- c. %rax, %rdx, %rcx
- d. %rbx, %rbp

---

14. Dada una función que devuelve la suma de 8 enteros en x86-64, ¿cuál de las siguientes instrucciones suma el 7º argumento?

- a. add      -0x8(%rsp), %eax
- b. add      0x8(%rsp), %eax
- c. add      -0x4(%rsp), %eax
- d. add      0x4(%rsp), %eax

---

15. En el fragmento de programa siguiente:

```
66b: e8 8a ff ff ff callq 5fa <f>
670: 48 83 c4 10 add $0x10,%rsp
```

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

- a. 0x670
- b. 0xfffff8a
- c. 0x66b
- d. 0x5fa

---

16. En el fragmento de programa siguiente:

```
66b: e8 8a ff ff ff callq 5fa <f>
670: 48 83 c4 10 add $0x10,%rsp
```

la instrucción callq suma al contador de programa la cantidad:

- a. -0x76
- b. 0x5fa
- c. 0xfffff8a
- d. 0x76

---

17. Suponga la siguiente llamada a una función f de 4 argumentos:

```
mov $0x1, %ecx
mov $0x2, %edx
mov $0x3, %esi
mov $0x4, %edi
callq 5fa <f>
```

El primer parámetro de llamada a la función:

- a. Es el valor inmediato 1
- b. Es el contenido de la dirección de memoria 0x1
- c. Es el valor inmediato 4
- d. Es el contenido de la dirección de memoria 0x4

18. Habiendo declarado **int array={0,1,2,3};** y **long long \*ptr=array;** ¿cuánto vale **ptr[1]**?

- a. 0x0000 0001 0002 0003
- b. 0x0000 0001 0000 0000
- c. 0x0003 0002 0001 0000
- d. 0x0000 0003 0000 0002

19. Una función C llamada **get\_el(...)** genera el siguiente código ensamblador.

```
leaq (%rdi,%rdi,4), %rax
addq %rax, %rsi
movl arr(,%rsi,4), %eax
ret
```

Se puede adivinar que:

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

20. Las microoperaciones de la fase de captación de una instrucción:

- a. Son comunes para todas las instrucciones
- b. Dependen del código de operación de la instrucción que se encuentra en el registro de instrucción
- c. Dependen de los indicadores de estado y del código de operación de la instrucción que se encuentra en el registro de instrucción
- d. Dependen del valor del contador de programa

21. Para el procesador con unidad de control microprogramada estudiado en clase, Tanenbaum propone codificar los 16 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. El ahorro de bits en cada microinstrucción debido a esta técnica es de

- a. 40 bits
- b. 39 bits
- c. 35 bits
- d. 29 bits

22. Un procesador está segmentado en k etapas. Cada una de ellas consume un tiempo t. La aceleración ideal (si no hay riesgos) al ejecutar 5 instrucciones respecto a un procesador no segmentado será:

- a.  $5k / (4+k)$
- b.  $(4+k) / 5t$
- c.  $4k / (5+k)$
- d.  $(5+k) / 4t$

23. ¿Cuál de las siguientes afirmaciones sobre la E/S programada con consulta de estado es cierta?

- a. Si se emplea E/S programada puede hacerse con consulta de estado o sin consulta de estado
- b. Un programa que realice salida programada con consulta de estado no ejecutará ninguna instrucción de entrada o carga
- c. Sólo la E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo de E/S
- d. La escritura de un led requiere consulta de estado

24. En un sistema de interrupciones vectorizado y en daisy-chain, ¿cuál de las siguientes afirmaciones es cierta?

- a. El procesador informa de un ciclo de reconocimiento de interrupción con la señal de reconocimiento de interrupción (INTA) y la identificación de los dispositivos se realiza por consulta de estado
- b. La gestión de prioridades queda establecida por el orden en que los dispositivos reciben la señal INTA y el dispositivo se identifica por un dato que deposita en el bus
- c. La gestión de prioridades queda establecida por el orden en que los dispositivos reciben la señal INTA y la identificación de los dispositivos se realiza leyendo sus registros de estado
- d. El daisy-chain asigna a todos los dispositivos la misma prioridad y la identificación de los dispositivos se realiza leyendo sus registros de estado

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

- a. dos direcciones (origen y destino)
  - b. dos tamaños (copia origen y copia destino)
  - c. cuál de las dos direcciones es de E/S (si alguna lo es) en lugar de Memoria
  - d. si se desea producir una IRQ al terminar
- 

26. En un computador con una jerarquía de memoria de dos niveles se observa experimentalmente que el tiempo medio de acceso a la memoria es de 300 ns cuando en realidad el tiempo medio de acceso al primer nivel es de 6 ns. Sabiendo que el tiempo de acceso al segundos nivel es de 3 microsegundos, ¿cuál sería aproximadamente el porcentaje de fallos en los accesos al primer nivel?

- a. 90%
  - b. 1%
  - c. 10%
  - d. 99%
- 

27. El orden de magnitud del tiempo de acceso a la memoria DRAM de un computador es de:

- a. Picosegundos
  - b. Nanosegundos
  - c. Microsegundos
  - d. Milisegundos
- 

28. Una memoria estática tiene un bus de datos de 32 bits y su bus de direcciones es de 20 bits, ¿cuál es su capacidad?

- a. 4 MBytes
  - b. 1 MByte
  - c. 32 MBytes
  - d. 80 GBytes
- 

29. En una cache asociativa por conjuntos de  $2^v$  vías con  $2^b$  líneas (marcos de bloque) 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 ( $0 \dots w-1$ ) (los menos significativos)
  - b. bits  $w \dots w+c-1$  (con  $c=b-v$ )
  - c. bits  $w \dots w+c-1$  (siendo  $2^c = n^\circ$  conjuntos)
  - d. bits  $b \dots b+c-1$  (siendo  $2^c = n^\circ$  conjuntos)
- 

30. Para obtener una única velocidad comparativa final, el benchmark SPEC CPU combina las ganancias en velocidad de ejecución de una serie de tests, respecto a un ordenador de referencia, usando...

- a. la mediana
  - b. la media aritmética
  - c. la media geométrica
  - d. la moda
-

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 direccionan 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  $w...w+p-1$
  - c. bits  $w...p-1$
  - 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
-



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 el direccionamiento indirecto a través de registro, la dirección efectiva...

- se encuentra en una dirección de memoria
- se encuentra en un registro general del procesador
- se encuentra en el registro de instrucción
- se calcula como la suma del contenido de dos registros

2. 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:

- $N \cdot F / C$
- $N \cdot C / F$
- $N \cdot F \cdot C$
- $N / (C \cdot F)$

3. ¿Cuál/es de los siguientes tipos de datos tienen distinto tamaño en arquitecturas x86 y x86-64 (Intel 32 bits y 64 bits)?

- long
- char \*
- ninguno de los dos
- ambos

4. La secuencia de instrucciones

```
leaq (%rdi, %rdi, 2), %rax
salq $3, %rax
```

produce el efecto de...

- leer en RAX a partir de la posición  $2 \cdot \text{RDI} + 2$  y multiplicar dicho contenido por 8

b. calcular  $\text{RAX} = (2 \cdot \text{RDI} + 2) \cdot 3 = 6 \cdot (\text{RDI} + 1)$

c. calcular  $\text{RAX} = 24 \cdot \text{RDI}$

d. calcular  $\text{RAX} = 9 \cdot \text{RDI}$

5. ¿Qué valor contendrá el registro RDX tras ejecutar las dos instrucciones siguientes?

```
movq $-1, %rdx
movb $1, %dl
```

- 0xFFFF FFFF FFFF FFF1
- 0xFFFF FFFF FFFF FF01
- 0xFFFF FFFF FFFF 0001
- 0x0000 0000 0000 0001

6. En la secuencia de programa siguiente:

```
400544: e8 07 00 00 00 callq 400550 <f>
400549: 48 89 03 mov %rax, (%rbx)
```

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

- 0x400544
- 0x400549
- 0x40054b
- 0x400550

7. Una función C declarada como `int get_var_digit(size_t index, size_t digit)` genera como código ensamblador

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

Se puede adivinar que:

- var es un array multi-nivel (punteros a enteros) de cuatro filas

- b. var es un array bidimensional de enteros, con cinco columnas
- c. var es un array multi-nivel pero no se pueden adivinar las dimensiones
- d. var es un array bidimensional de enteros, no se pueden adivinar dimensiones
- 
8. El microcódigo de un procesador consiste en 640 microinstrucciones de 70 bits, de las cuales 280 son únicas. ¿Cuántos bits se podría llegar a ahorrar usando nanoprogramación?
- a.  $640 \times 70 - 280 \times 70$
- b.  $640 \times 70 - 640 \times 9 - 280 \times 70$
- c.  $640 \times 70 - (640 \times \log_2(70) + 280 \times 70)$
- d.  $640 \times 280 - 640 \times 70 - 280 \times 70$
- 
9. ¿A qué instrucción de lenguaje máquina podría corresponder la siguiente secuencia de microinstrucciones del camino de datos con un bus estudiado en clase?
- Enable R1, Load Y*  
*Enable R2, Select Y, Add, Load Z*  
*Enable Z, Load R2*
- a. load (R1,R2), R2
- b. add R1, R2
- c. store R1+R2, (R2)
- d. move Y(R1,R2), Z(R2)
- 
10. Suponer que un procesador ideal que ejecuta cada instrucción en T segundos se segmenta en cuatro etapas ideales de duración T/4. Con ello se consigue que
- a. 4 instrucciones se ejecuten en 4T segundos
- b. una instrucción se ejecute en T/4 segundos
- c. cada 4T segundos se terminen de ejecutar 4 instrucciones
- d. cada T/4 segundos se termine de ejecutar una instrucción
- 
11. Precaptar instrucciones antes de que sean necesarias y almacenarlas en una cola de instrucciones, es una técnica que se usa para...
- a. evitar cierto tipo de riesgos estructurales
- b. reducir riesgos por dependencias de datos
- c. corregir algunos riesgos de control
- d. calcular las predicciones de saltos
- 
12. La técnica de Consulta de Estado (polling) puede usarse para... (señalar la opción INCORRECTA)
- a. identificar el origen de una interrupción
- b. consultar si el dispositivo está dispuesto para entregar o recibir datos
- c. consultar el sentido del DMA en curso (desde memoria / hacia memoria)
- d. establecer un mecanismo software de asignación de prioridades a los dispositivos
- 
13. Técnicas que suelen contemplar los procesadores para gestionar el sistema de prioridades entre peticiones de interrupción (señalar la opción INCORRECTA)
- a. gestión de prioridades centralizada
- b. gestión de prioridades distribuida
- c. gestión de prioridades simultánea
- d. gestión de prioridades híbrida
- 
14. Una SRAM de 256Kx4bit (1Mbit) puede venir organizada en 1024 filas, dedicando por tanto al decodificador de columnas...
- a. 6 bits
- b. 7 bits
- c. 8 bits
- d. 10 bits
- 
15. ¿Cuál de los siguientes grupos de señales no se usa en un chip de memoria SRAM?
- a. Selección de chip CS# y habilitación de escritura WE#
- b. Selección de filas RAS# y de columnas CAS#
- c. Direcciones  $A_{n-1}-A_0$
- d. Datos  $D_{n-1}-D_0$
- 
16. ¿En qué pareja de registros están la dirección de memoria y el dato que se leerá o escribirá en memoria?
- a. pc y mar
- b. ir y pc
- c. mar y mdr/mbr
- d. mdr/mbr y pc
- 
17. 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 0xA6 y en el 0x1001 0x91
- b. en el byte 0x1000 se guarda 0x89 y en el 0x1001 0x65

- c. en el byte 0x1000 se guarda 0x91 y en el 0x1001 0xA6
  - d. en el byte 0x1000 se guarda 0x65 y en el 0x1001 0x89
- 

18. En x86-64, el registro contador de programa se denomina:

- a. rip
  - b. eip
  - c. pcr
  - d. pc
- 

19. Si rax contiene x, ¿cuál de las siguientes instrucciones calcula  $x \cdot 9$ ?

- a. `leaq 8(%rax,%rax),%rdx`
  - b. `leaq (%rax,%rax,8),%rdx`
  - c. `leaq 3(%rax,%rax,2),%rdx`
  - d. `leaq 5(%rax,%rax,4),%rdx`
- 

20. La instrucción `jbe / jna` provoca un salto si...

- a. `SF == 1 || ZF == 1`
  - b. `CF == 1 || ZF == 1`
  - c. `CF == 1`
  - d. `SF != OF`
- 

21. En x86-64, es responsabilidad del procedimiento llamado (calle) salvaguardar, entre otros, los registros:

- a. `%rbx, %rsi, %rdi`
  - b. `%rax, %rdx, %rcx`
  - c. `%rax, %rbx, %rcx, %rdx`
  - d. `%rbx, %rbp`
- 

22. 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 16(%rsp), %rax`  
`ret`
  - c. `movq %r10, %rax`  
`ret`
  - d. `movq %r8, %rax`  
`ret`
- 

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

- a. Muchos campos solapados
- b. Poca codificación
- c. Microinstrucciones cortas

d. Poca capacidad para expresar paralelismo entre microoperaciones

---

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. Sobre la segmentación:

- a. La frecuencia de reloj viene impuesta por la etapa más corta.
  - b. Existen limitaciones al rendimiento provocadas por las instrucciones de salto y por las dependencias de datos.
  - c. Es una técnica para lanzar a ejecutar simultáneamente varias instrucciones con el fin de reducir el tiempo de ejecución.
  - d. Un procesador superescalar no puede estar segmentado.
- 

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

- a.  $S = (K \cdot N) / (K - N + 1)$
  - b.  $S = (N \cdot K \cdot T) / ((N - K + 1) \cdot T)$
  - c.  $S = (K \cdot N) / (K + N - 1)$
  - d.  $S = (N \cdot T) / ((N + K - 1) \cdot T)$
- 

27. ¿Cuál de las siguientes tareas NO es responsabilidad de un circuito de interfaz o controlador de periféricos sencillo?

- a. Adaptar el formato de las señales.
  - b. Ajustar la temporización entre el procesador y los dispositivos de E/S.
  - c. Recibir señales de control desde el procesador.
  - d. Ejecutar el programa de transferencia de información entre el procesador y los dispositivos de E/S.
- 

28. ¿Con cuál de los siguientes dispositivos tendría sentido utilizar E/S programada sin consulta de estado?

- a. Salida a un display de 7 segmentos
  - b. Entrada desde un disco duro
  - c. Salida a una impresora
  - d. Con ningún dispositivo tiene sentido
- 

**29.** ¿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 secuencial
  - c. Localidad temporal
  - d. Localidad asociativa
- 

**30.** ¿Cuál de las siguientes afirmaciones acerca de la memoria es FALSA?

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

|                |               |
|----------------|---------------|
| <b>Nombre:</b> |               |
| <b>DNI:</b>    | <b>Grupo:</b> |

## 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 el direccionamiento inmediato el operando reside en:
  - a. en un registro del procesador
  - b. en la instrucción tras el código de operación
  - c. en memoria, en la dirección indicada
  - d. en la pila

---

2. Una instrucción máquina del tipo "Add M,R" podría formar parte del repertorio de
  - a. una máquina pila
  - b. una máquina de acumulador
  - c. una máquina con arquitectura R/R
  - d. una máquina con arquitectura M/M

---

3. La traducción a ASM de una función C con prototipo **void fun(<tipo> arg1, <tipo> arg2);** por parte de gcc empieza con el siguiente texto:
 

```
fun:
 movl (%rdi), %eax
 movl (%rsi), %edx
 ...
```

¿Cuál es el posible <tipo> mencionado en el prototipo?

  - a. long \*
  - b. unsigned int \*
  - c. short \*
  - d. unsigned char \*

---

4. Se puede describir paso a paso la ejecución de la instrucción **add (%rbx, %rdx, 4), %eax;** de la siguiente manera (marcar la opción **FALSA**):
  - a. primero: se calcula la dirección efectiva  $EA = RBX + RDX * 4$
  - b. segundo: se leen ocho bytes a partir de dicha dirección de memoria
  - c. tercero: se yuxtaponen los bytes leídos de forma que el de dirección más baja ("el primero leído") sea el menos significativo y el de dirección más alta ("el último leído") sea el más significativo
  - d. cuarto: se suma el valor obtenido con el registro EAX

---

5. ¿Qué valor contendrá el registro rdx tras ejecutar las dos instrucciones siguientes?
 

```
movq $-1, %rdx
movl $1, %edx
```

  - a. 0xFFFF FFFF FFFF 0001
  - b. 0xFFFF FFFF 0000 0001
  - c. 0xFFFF 0000 0000 0001
  - d. 0x0000 0000 0000 0001

---

6. En el fragmento de código
 

```
400544: e8 07 00 00 00 callq 400550 <f>
400549: 48 89 03 mov %rax, (%rbx)
```

la instrucción call suma al contador de programa la cantidad:

  - a. 0x00000007
  - b. 0x00400549
  - c. 0x400544
  - d. 0x48

---

7. Una función C declarada como **int get\_var\_digit(size\_t index, size\_t digit)** genera como código ensamblador

```

movq var(,%rdi,8), %rax
movl (%rax,%rsi,4), %eax
ret

```

Se puede adivinar que:

- a. var es un array multi-nivel (punteros a enteros) de cuatro filas
  - b. var es un array multi-nivel pero no se pueden adivinar las dimensiones
  - c. var es un array bidimensional de enteros, con ocho columnas
  - d. var es un array bidimensional de enteros, con cinco columnas
- 
8. En una unidad de control microprogramada se tiene un campo de 14 señales de control de las cuales sólo se activaría una o ninguna en un ciclo de reloj, nunca dos o más en el mismo ciclo de reloj. Sería entonces posible...
- a. codificarlas con 4 bits, y sobraría un código que quedaría sin uso
  - b. codificarlas con 5 bits, y sobrarían dos códigos que quedarían sin uso
  - c. solaparlas en un solo campo de 5 bits, ahorrando por tanto 9 bits
  - d. solaparlas en un solo campo de 7 bits, ahorrando por tanto 6 bits
- 
9. En el contexto de microprogramación, el control residual...
- a. intenta disminuir la cantidad de "bits residuales", usando las técnicas de codificación y/o solapamiento de campos, como opuestas a la microprogramación directa o "inmediata".
  - b. se refiere a que cuanto más codificación y/o solapamiento se use, menos capacidad para expresar paralelismo se tiene, siendo ese menor control un "residuo" o consecuencia no deseada de dichas técnicas no "inmediatas"
  - c. clasifica las microinstrucciones del microcódigo según formen parte de microprogramas ("microinstr. inmediatas") o no ("microinstr. residuales")
  - d. consiste en almacenar señales de control en un "registro de control residual" para usarlas en ciclos posteriores, a diferencia del "control inmediato", en donde los bits se utilizan inmediatamente
- 

10. Suponer que un procesador ideal que ejecuta cada instrucción en T segundos se segmenta en cuatro etapas ideales de duración T/4. ¿Cuál razonamiento es correcto?
- a. Se espera una reducción de prestaciones porque además de ejecutar las instrucciones hay que segmentarlas (coste de la segmentación)
  - b. Se espera una reducción de prestaciones porque la duración del ciclo de reloj vendrá impuesta por la etapa más lenta
  - c. Se espera un aumento de prestaciones debido al efecto de los riesgos (hazards) sobre el avance de las instrucciones en el cauce
  - d. Se espera un aumento de prestaciones debido a que las cuatro etapas solapan su funcionamiento, con una aceleración ideal de 4x
- 
11. Un salto condicional del tipo "delayed branch", o salto retardado, ejecuta la(s) instrucción(es) siguiente(s)...
- a. sólo si el salto se produce (las ignora si NO se produce), de manera que instrucción(es) en el destino del salto podrían adelantarse tras la propia instrucción de salto
  - b. sólo si el salto NO se produce (las ignora si se produce), de manera que instrucción(es) en el destino del salto podrían adelantarse tras la propia instrucción de salto
  - c. siempre, de manera que instrucción(es) anterior(es) al salto podrían colocarse tras la propia instrucción de salto
  - d. nunca, de manera que instrucción(es) anterior(es) al salto no podrían colocarse tras la propia instrucción de salto
- 
12. Un computador con 20 líneas de dirección y memoria de bytes tiene 640KB de RAM, 128KB de ROM, y utiliza E/S mapeada en memoria. ¿Cuál es el número máximo de periféricos que pueden conectarse, si cada uno de ellos utiliza 32 direcciones?
- a.  $2^{10}$
  - b.  $2^{11}$
  - c.  $2^{12}$
  - d.  $2^{13}$
- 
13. La consulta de estado que se puede llevar a cabo en una operación de salida mediante E/S programada sirve para...



- a. consultar si el dispositivo tiene algún dato de salida disponible
  - b. consultar si el dispositivo está aún ocupado, por ejemplo con alguna operación de salida anterior
  - c. consultar si el dispositivo funciona correctamente
  - d. ninguna de las respuestas anteriores es correcta
- 

14. ¿Qué conjunto de componentes permite construir una memoria 256Mx32? (sin que sobren componentes)

- a. 16 chips 64Mx4
  - b. 32 chips 64Mx4
  - c. 16 chips 64Mx16
  - d. Ninguna de las anteriores
- 

15. ¿Cuántas líneas de dirección (patillas) son necesarias para direccionar un chip de memoria DRAM de 256K x 4?

- a. 9
  - b. 13
  - c. 18
  - d. 22
- 

16. ¿Qué arquitectura es típica en procesadores RISC?

- a. registro-registro
  - b. registro-memoria
  - c. memoria-registro
  - d. memoria-memoria
- 

17. ¿Cuál de las siguientes características es posterior a la segunda generación de computadores?

- a. Memoria de núcleos de ferrita
  - b. Compilador
  - c. Memoria cache
  - d. Lenguaje ensamblador
- 

18. Si el registro rax contiene **x**, la sentencia en C **x &= 0x1;** se traducirá a ensamblador como:

- a. andq \$1, %rax
  - b. orq \$0x1, %rax
  - c. shrq %rax
  - d. sarq %rax
- 

19. Para crear espacio en la pila para variables locales sin inicializar suele realizarse la siguiente operación:

- a. Restar una cantidad positiva a rbp.
  - b. Sumar una cantidad positiva a rbp.
  - c. Restar una cantidad positiva a rsp.
  - d. Sumar una cantidad positiva a rsp.
- 

20. Si la estructura struct a ocupa un espacio de 26 bytes en memoria, ¿cuántos bytes ocupa la siguiente estructura struct b cuando se compila en 64 bits?

```
struct b {
 struct a a1;
 int i;
 struct a a2;
};
```

- a. 24
  - b. 58
  - c. 60
  - d. 64
- 

21. En la secuencia de programa siguiente:

```
628: e8 cd ff 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. 0xffffffffcd
  - b. 0x5fa
  - c. 0x628
  - d. 0x62d
- 

22. Un archivo .o que contiene código objeto reubicable:

- a. Contiene instrucciones máquina binarias.
  - b. Contiene instrucciones máquina y directivas en ensamblador.
  - c. Puede ejecutarse directamente.
  - d. Contiene las direcciones definitivas de las variables globales.
- 

23. ¿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 manera semiautomática
- 

24. En una unidad de control microprogramada con formato de microinstrucciones vertical, un subcampo que deba especificar 16 señales de control, codificadas de tal forma que pueda activarse sólo una o ninguna señal de control, habrá de tener una anchura mínima de:

- a. 4 bits
  - b. 5 bits
  - c. 16 bits
  - d. 17 bits
- 

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

- a. estructurales
  - b. organizativos
  - c. de control
  - d. de datos
- 

26. Un sistema no segmentado tarda 10 ns en procesar una tarea. La misma tarea puede ser procesada en un cauce (pipeline) de 4 segmentos con un ciclo de reloj de 4 ns. Cuando se procesan muchas tareas, la ganancia máxima de velocidad que se obtiene se acerca a:

- a. 10
  - b. 4
  - c. 40
  - d. 2,5
- 

27. ¿Cuál de las siguientes afirmaciones es cierta?

- a. La E/S en memoria emplea la patilla IO/M#.
  - b. En E/S independiente, las instrucciones de acceso a memoria suelen ser más largas que las de E/S.
  - c. La E/S en memoria facilita la protección.
  - d. En la E/S en memoria las instrucciones de E/S son fácilmente reconocibles.
- 

28. La instrucción máquina di (Disable Interrupts), conocida como cli (Clear Interrupt Flag) en x86:

- a. Desactiva todas las interrupciones enmascarables.

- b. Desactiva las interrupciones de inferior o igual prioridad a una dada.
  - c. Desactiva determinados niveles de interrupción de forma selectiva.
  - d. Desactiva las interrupciones software.
- 

29. ¿Cuál de las siguientes afirmaciones sobre la jerarquía de memoria es cierta?

- a. Para aumentar la eficiencia se transfieren bloques completos.
  - b. Toda la información que el procesador necesita está en el nivel 1.
  - c. Si una palabra no se encuentra en el tercer nivel entonces se busca en el segundo nivel.
  - d. A medida que nos alejamos del procesador, el tamaño de memoria disminuye.
- 

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

- a. 6
  - b. 10
  - c. 11
  - d. 12
-



11. Un salto condicional del tipo "delayed branch", o salto retardado, ejecuta la(s) instrucció(n)es siguiente(s)...

- a. sólo si el salto se produce (las ignora si NO se produce), de manera que instrucció(n)es en el destino del salto podrí(=)an adelantarse tras la propia instrucció(n) de salto
- b. sólo si el salto NO se produce (las ignora si se produce), de manera que instrucció(n)es en el destino del salto podrí(=)an adelantarse tras la propia instrucció(n) de salto
- c. siempre, de manera que instrucció(n)es <----- anterior(es) al salto podrí(=)an colocarse tras la propia instrucció(n) de salto
- d. nunca, de manera que instrucció(n)es anterior(es) al salto no podrí(=)an colocarse tras la propia instrucció(n) de salto

12. Indicar cuál es la dirección de la instrucció(n) pop en el siguiente desensamblado, donde se ha borrado parte de la dirección

```
400XXX: 77 02 ja 400b41
400XXX: 5d pop %rbp
```

- a) 400b3d
- b)  $400b41 + 2 = 400b43$
- c)  $400b3f <-----$
- d)  $400b41 - 2 = 400b39$

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 instrucció(n), el tiempo de ejecución del programa será:

- a)  $N \rightarrow F/C$
- b)  $N \rightarrow F \rightarrow C$
- c)  $N/(F \rightarrow C)$
- d)  $N \rightarrow C/F <-----$

En el siguiente código, ¿qué reordenamiento de los bucles muestra mejor localidad?

```
// X, Y, Z ctes #define previo
int a[X][Y][Z]
int i, j, k, sum = 0;
for (i = 0; i < Y; i++)
 for (j = 0; j < Z; j++)
 for (k = 0; k < X; k++)
 sum += a[k][i][j];
```

- a) i externo, j central, k interno (el orden en que están ahora)
- b) j externo, k central, i interno
- c) k externo, i central, j interno <-----
- d) El orden de los bucles no afecta a la localidad

¿Cuál es el tamaño de la marca de cache si el bus de direcciones es de 48 bits (256 TB de memoria principal) y hay 8MB de cache L3, con un tamaño de línea de 64 B y correspondencia asociativa por conjuntos con 16 vías?

- a) 29 bits <-----
- b) 6 bits
- c) 13 bits
- d) 48 bits

Supongamos dos procesadores con bus de direcciones con idéntico número de líneas. Si uno de ellos emplea E/S mapeada en memoria y el otro E/S independiente, ¿cuál podrá acceder a una mayor cantidad de memoria?

- a) Ambos podrán acceder a la misma cantidad de memoria
- b) Depende del tamaño del bus de direcciones
- c) El que tiene E/S independiente <-----
- d) El que tiene E/S mapeada en memoria

La técnica de Consulta de Estado (polling) puede usarse para...

- a) identificar el origen de una interrupción
- b) consultar si el dispositivo está dispuesto para entregar o recibir datos
- c) consultar el sentido del DMA en curso (desde memoria / hacia memoria) <-----
- d) establecer un mecanismo software de asignación de prioridades a los dispositivos

¿Cuál de las siguientes afirmaciones sobre memorias cache es cierta?

Recordar que llamamos "latencia" al tiempo transcurrido desde que se envía una dirección a la cache hasta que se obtiene el dato (suponiendo que se trata de un acierto), y depende por tanto del tiempo que emplee la circuitería hardware en obtenerlo. Suponer similares condiciones para ambas caches (tamaño de bloque, tamaño total, tecnología, frecuencia...)

- a) Las caches totalmente asociativas ofrecen mejor latencia, mientras que las de correspondencia directa tienen tasas de fallo más bajas
- b) Ambas tienen en general similar latencia y tasa de fallos
- c) Las caches totalmente asociativas ofrecen tasas de fallo más bajas, mientras que las de correspondencia directa tienen mejor latencia <-----
- d) Las caches de correspondencia directa tienen mejor latencia y tasa de fallos

Para el procesador con unidad de control microprogramada estudiado en clase, Tanenbaum propone codificar los 16 registros y añadir una señal, ÁPERC, para habilitar la carga desde el bus C (recordar que era un diseño típico con 3 buses) y así no perder expresividad/paralelismo. El ahorro de bits en cada microinstrucción debido a esta técnica es de

- a) 29 bits
- b) 40 bits
- c) 35 bits<-----
- d) 39 bits

La consulta de estado que se puede llevar a cabo en la E/S programada sirve para...

- a) consultar si el dispositivo está dispuesto para recibir datos o tiene datos disponibles<-----
- b) consultar qué dispositivo ha solicitado la interrupción
- c) consultar si el dispositivo funciona correctamente
- d) Ninguna de las respuestas anteriores es correcta

En las políticas anticipativas de extracción de cache, ¿cuál de ellas se caracteriza por preextraer el bloque i+1 si se referencia al bloque i y se produce falta de bloque?

- a) Preextracción indexada
- b) Preextracción siempre
- c) Preextracción marcada
- d) Preextracción por falta<-----

¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC (modo real) es cierta?

- a) Cada vector de interrupción es una palabra de 16 bits
- b) Existen 1024 vectores de interrupción
- c) La tabla de interrupciones tiene un tamaño de 256 bytes
- d) Todas las interrupciones se pueden generar por software<-----

Si el contenido de RSP es 0xEC00AC00 antes de ejecutar la instrucción push %rbx. ¿Cuál será su contenido tras ejecutarla?

- a) 0xEC00AC08
- b) 0xEC00AC04
- c) 0xEC00ABFC
- d) 0xEC00ABF8<-----

Se ha declarado en un programa C la variable int val[5]={1,5,2,1,3}; ¿Cuál de las siguientes afirmaciones es cierta?

- a) val[1] == 1
- b) &val[3] == (char \*)val + 12<-----
- c) sizeof(val) == 5
- d) &val[2] es de tipo int \* y vale lo mismo que val + 8

Parecidos y diferencias entre los métodos de E/S (señalar la opción incorrecta)

- a) se suele avisar a la CPU (con una IRQ) de que debe realizar alguna tarea, tanto en E/S por IRQ como en E/S por DMA
- b) sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo <-----
- c) la consulta del estado del dispositivo por parte de la CPU se suele/puede incluir en E/S programada y en E/S por IRQ
- d) sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S

Respecto a saltos retardados y anulantes, condicionales o no, NO será 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 (y avanzando el destino del salto)
- c) en el destino del salto condicional anulante para ponerlas después (en mem.) del salto (y avanzando el destino del salto)
- d) anteriores al condicional anulante para ponerlas después (en memoria) <-----