

TEST TEORÍA

- 1. 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 desplazamiento D puede ser una constante literal (1, 2 ó 4 bytes)
 - b. EBP no se puede usar como registro base
 - c. ESP no se puede usar como registro índice
 - d. El factor de escala S puede ser 1, 2, 4, 8
- 2. 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.
- 3. En IA32, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto al resto?**
- a. sub%edi,%edi
adc \$0xFFFFFFFF, %edi
 - b. mov \$-1, %edi
 - c. mov\$-1,%edi
add \$0, %edi
 - d. mov \$0, %edi

sub \$1, %edi
- 4. Si %rsp vale 0xdeadbeefdeadd0d0, ¿cuál será su nuevo valor después de que se ejecute pushq %rbx?**
- a. 0xdeadbeefdeadd0d4
 - b. 0xdeadbeefdeadd0d8
 - c. 0xdeadbeefdeadd0cc

d. 0xdeadbeefdeadd0c8

5. ¿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. Ninguna de esas formas es la correcta.

6. Comparando las convenciones de llamada de gcc Linux IA32 con x86-64 respecto a registros

- a. En IA32 %ebx es salva-invocante, pero en x86-64 %rbx es salva-invocado
- b. En IA32 %ecx es salva-invocante, y en x86-64 %rcx es salva-invocante también
- c. En IA32 %esi es salva-invocado, y en x86-64 %rsi es salva-invocado también
- d. En IA32 %ebp es especial (marco de pila), y en x86-64 %rbp también

7. 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 µPC
- c. el secuenciamiento de las instrucciones máquina
- d. todas las respuestas son ciertas

8. Respeto 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. 16bits
- 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. micro bifurcaciones**
- 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 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 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.Cuanta 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
idx 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 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(StackPointer)
- 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 Ai del nivel i de una jerarquía de memoria no depende de:

- a.La capacidad (tamaño) si del nivel i.
- b.La estrategia de administración de memoria.
- c.La unidad de la transferencia de información xi entre el nivel i y el i+1.
- d.El ancho de banda bi 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 $k*m$ (y mismo no 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

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

- a. La última versión es SPEC CPU2006 V1.2 de 2011
- b. Se cronometran unos 12 tests de enteros (CINT2006) y unos 17 tests de punto flotante (CFP2006)
- c. 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
- d. El resultado final es la media aritmética de las (12 ó 17) velocidades, bien sea de enteros ó de punto flotante (SPECint2006 ó SPECfp2006)

2. ¿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?

- a. 2a generación (1955-65)
 - b. 3a generación (1965-75)
 - c. 4a generación (1975-85)
 - d. esas innovaciones se repartieron a lo largo de varias generaciones, no sólo una
- 3. Respecto a tamaños de tipos integrales en x86 y x86-64, la excepción es que**
- a. int pasa de 4 B (x86) a 8 B (x86-64)
 - b. long int pasa de 4 B a 8 B
 - c. long long int pasa de 4 B a 8 B
 - d. ninguna de las anteriores

4. Con el repertorio IA32, para sumar %eax y %ebx dejando el resultado en %ecx se podría hacer lo siguiente:

- a. lea %eax, %ebx, %ecx
- b. lea (%eax, %ebx, 1), %ecx
- c. lea %ecx, [%eax, %ebx]

d. lea %ecx, %ebx, %eax

5. Cuál de las instrucciones máquina siguientes es incorrecta en x86-64:

- a. testl %edx, %edx
- b. **movl %r8, %eax**
- c. movl (%rdi,%rcx,4), %edx
- d. addq \$1, %rcx

6. 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:

- a. xorl %ebx,%eax
- b. **xorl %eax,%ebx**
- c. andl %ebx,%eax
- d. testl %eax,%ebx

7. Para poner a 1 el bit 5 del registro %edx sin cambiar el resto de bits podemos usar la instrucción máquina:

- a. and \$32, %edx
- b. and \$0x5, %edx
- c. or \$0b101, %edx
- d. **or \$0x20, %edx**

8. 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 con 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 llevan 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 213 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.210**
- b.212
- c.211
- d.213

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. 6ns
- 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

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 0xCAFEBAB0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al consultar la media palabra de la posición 0xCAFEBABE?

- 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.0xCAFEBABE
- 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?

C

- 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. $\ast(\ast(x + 16)) + 28$
- b. $\ast(\ast\ast x + 2) + 7$
- c. $\ast(\ast(x + 2) + 4) + 3$
- d. $\ast(\ast(x + 2) + 4) + 3$

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 acople (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

1.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:

a. 0xCAFEBAE1

b. **0xCAFEBAE0**

c. 0xCAFEBABF

d. 0xCAFEBAE

2.¿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

3.En X86-64, el registro contador de programa se denomina:

a.RIP

b.EIP

c.IP

d.R15

4.La primera letra (l) de la instrucción lea:

- a.indica que la instrucción usa ordenación de bytes little-endian
- b.indica que la instrucción trabaja con un operando destino de 32 bits (long word)
- c.indica que la instrucción afecta a los 16 bits menos significativos del operando destino (low word)
- d.forma parte del nemotécnico de la instrucción**

5. ¿Cuál es la diferencia entre las instrucciones mov y lea?

- a. lea accede a la posición indicada,mientras que mov no lo hace
- b. mov accede a la posición indicada, mientras que lea no lo hace**
- c. lea puede usarse para copiar un registro a otro, mientras que mov no
- d. mov puede usarse para copiar un registro a otro, mientras que lea no

6. La instrucción not:

- a. realiza el complemento ados
- b. realiza el complemento a uno (cambiar unos por ceros y ceros por unos)**
- c. realiza la operación no-or (or negada)
- d. realiza un salto condicional si negativo

7. La instrucción JGE / JNL provoca un salto si...

- a. SF = 1
- b. CF = 1
- c. SF = 0
- d. OF=SF**

8. Un overflow nunca puede ocurrir cuando:

- a. se suman dos números positivos
- b. se suman dos números negativos**

c. se suma un número positivo a un número negativo

d. se resta un número positivo de un número negativo

9. 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 cmove %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 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. %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 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 ahorraremos usando nanoprogramación en lugar de microprogramación?

- a.3744250
- b.259206
- c.287935
- d.Ninguno De Resultados Anterior 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 llenar 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 las cachés L1 y L2

30. ¿Cuál de las siguientes afirmaciones acerca de la jerarquía de memoria es *FALSA*?

a. Acceder a los discos es más lento que acceder a la RAM

b. Una memoria principal constituida por la tecnología más rápida es 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

1. 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. 1

c. -1

d. -2

2. Según la clasificación m/n, las máquinas de acumulador son de tipo

a. 0/0

b. 1/1

c. ½

d. 2/2 ó ¾

3. Según la clasificación m/n, las máquinas con arquitectura R/R son de tipo

a. 0/0

b. 0/x con x=2,3

c. x/0 con x=2,3

d. x/x con x=2,3

4. El bus del sistema es

a. el que conecta las distintas partes del sistema: UC, ALU, E/S, M

- b. en un sistema con bus único, todo el bus salvo la parte relacionada con E/S (SATA, GPU, USB, Ethernet, etc)
- c. en un sistema con buses separados, el que conecta el sistema E/S con el resto
- d. el que conecta CPU-M, ya sea un sistema con bus único o con múltiples buses

5. 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. 0x00 a 0x0f y 0x80 a 0x8f
- b. 0x00a0x1f
- c. 0x70 a 0x7f y 0xf0 a 0xff
- d. 0xe0a0xff

6. La ecuación básica de rendimiento calcula

- a. cuánto tiempo tarda en ejecutarse un programa concreto conociendo su número de instrucciones y el número de etapas (promedio) y la frecuencia del procesador
- b. cómo de mejor es un procesador frente a otro, conociendo las prestaciones de las respectivas UC, ALU, E/S y M
- c. el promedio de las ganancias obtenidas con una serie de programas de punto entero
- d. la media geométrica de los cocientes entre los tiempos de ejecución de una serie de programas predeterminados

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

- a. x86
- b. IA-32
- c. IA-64
- d. AMD64

8. ¿Qué combinación de *flags* aritmético- lógicos corresponde al código de condición b (*below*)?

- a. CF
- b. OF
- c. CFxorOF
- d. OFxorSF

9. En los modos de direccionamiento del tipo *Desplazamiento(Base,Indice,Factor Escala)*, puede usarse como

- a. desplazamiento,cualquierconstante de 1, 2 o 4 bytes (incluso el nombre de una variable,porsudirección)
- b. base, cualquiera de los 8 registros enteros salvo %esp
- c. índice,tambiéncualquierasalvo%ebp
- d. factor de escala, cualquier constante de 1, 2, 4 u 8 bytes

10. 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.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

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

- a. un salto condicional,según la condición expresada en el código C
- b. un salto condicional, según la condición opuesta a la del código C, y otro salto incondicional
- c. dos saltos condicionales (uno para la parte if y otro para la parte else)
- d. dos saltos condicionales y dos saltos incondicionales

12. Para traducir una construcción do-while de lenguaje C a lenguaje ensamblador, gcc utiliza generalmente

- a. un salto condicional hacia adelante, según la misma condición que en lenguaje C
- b. un salto condicional hacia atrás, según la misma condición que en lenguaje
- c. un salto condicional hacia adelante, según la condición opuesta a la de lenguaje C
- d. un salto condicional hacia atrás, según la condición opuesta a la de lenguaje C

13. Alguna de las siguientes señales no sirve de entrada a la unidad de control. ¿Cuál?

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

14. Alguna de las siguientes señales no es salida de la unidad de control. ¿Cuá

- a. señales decarga, habilitación y/o desplazamiento de registros (*Load, Enable, ShiftL, ShiftR*)
- b. códigos de selección en multiplexores, decodificadores, ALU, etc (00, 01, 10, 11...)
- c. señales de lectura y escritura en memoria (RD, WR)
- d. dirección de la siguiente microinstrucción (bits del campo DIR o Memoria B de Wilkes)

15. Alguna de las siguientes no es una operación básica de la unidad de control

- a. transferir un registro a otro
- b. (leer o escribir) un registro (de / a) memoria
- c. (guardar recuperar) un registro (en/de) la pila
- c. realizar operación ALU y guardar resultado en registro

16. Tipos de riesgos que hemos estudiado en cauces segmentados (señalar la opción incorrecta)

- a. riesgos estructurales
- b. riesgos de (dependencia de) datos
- c. riesgos control
- d. riesgos de transferencia**

17. ¿Cuál de los siguientes modos de direccionamiento es menos preferible para un procesador con segmentación de cauce?

- a. registro
- b. directo**
- c. indirecto a través de registro
- d. indexado (o relativo a base, o base+índice)

18. 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. $S = NKT/(N-K+1)T$
- c. $S=KN/(K+N-1)$**
- d. $S = NT/(N+K-1)T$

19. 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**

20. 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 escribir un mismo dato a varios puertos E/S
- d. usando decodificación centralizada es más fácil realizar expansiones al sistema de E/S

21. Ventajas de la E/S independiente (separada, aislada) (señalar la opción incorrecta)

- a. diseño del procesador más sencillo (E/S mapeada añade complejidad al diseño)
- b. protección de E/S más fácil (E/S mapeada añade dificultad a la protección de E/S)
- 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)

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

- a. la consulta del estado del dispositivo por parte de la CPU se suele/puede hacer con E/S programada y con E/S por IRQ
- b. 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
- c. sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo
- d. sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S

23. Tipos de interrupción que suelen contemplar las CPUs comerciales actuales (señalar la opción incorrecta)

- a. internas (excepciones o traps): generadas internamente por la CPU para indicar una condición que requiere atención (división por cero, codop inválido, etc)

- b.externas (IRQs *hardware*): generadas por un dispositivo externo a la CPU, activan la línea INTR# (o equivalente)
- c.software:generadas al ejecutar la instrucción INT (o equivalente)
- d.firmware (*faults*): generadas por el microcódigo de la CPU (*segmentation fault*, *page fault*, etc)

24. Técnicas que se pueden usar para determinar la causa de una interrupción (señalar la opción incorrecta)

- a.múltiples líneas de interrupción INT1#, INT2#...
- b.línea de reconocimiento INTA#
- c.consulta de estado,o polling
- d.interrupciones vectorizadas

25. Respecto a salvaguardar los registros de la CPU al inicio de una rutina de servicio de interrupción (ISR)

- a.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
- b.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
- c.se deben guardar los registros salva- invocado (p.ej. EBX, ESI, EDI en el caso de una CPU IA-32), los registros salva- invocante ya los guarda el programa interrumpido
- 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

26. En EC podemos usar la palabra *directo* para referirnos a... (señalar la opción incorrecta)

- a. dispositivo de almacenamiento secuencial directo (DASD)
- b. cache con correspondencia directa
- c. acceso directo a memoria
- d. modo de direccionamiento directo

27. Variación de los parámetros de los distintos niveles en una jerarquía de memoria (señalar la opción incorrecta)

- a. unidad de transferencia: x_i x_{i+1}
- b. ancho de banda: b_i b_{i+1}
- c. tamaño del nivel: s_i s_{i+1}
- d. tiempo de acceso: t_i t_{i+1}

28. Las técnicas *write-through* y *write-back* están relacionadas con

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

29. 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

- 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. repartiendolíneasdatos:el1oseconecta a $D_0 \dots D_{m-1}$, el 2o a $D_m \dots D_{2m-1}$, etc
- d. repartiendo líneas dirección: el 1o a $A_0 \dots A_{m-1}$, el 2o a $A_m \dots A_{2m-1}$, etc

30. En un sistema con direcciones de 32bits, memoria de bytes, cache de 1MB asociativa por conjuntos de 4 vías y líneas de 64B, el campo etiqueta en el formato de dirección cache es de

- a. 16bits
- b. 14bits
- c. 12bits

d. 10bits

1. ¿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 son capaces de lograr un mejor rendimiento cuando se asignan variables a registros.
- c.Porque la memoria es más cara que los registros.
- d.Porque las basadas en registros permiten reducir el tamaño del programa.

2. ¿Cuál de las siguientes afirmaciones es incorrecta?

- a.El direccionamiento indexado es útil para manejo de vectores.
- b.El direccionamiento indirecto indica la dirección del 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. ¿Cuál de las siguientes afirmaciones sobre el direccionamiento absoluto es falsa?

- a.La instrucción contiene la dirección de memoria en la que se encuentra el objeto.
- b.El objeto está en una posición de la memoria.
- c.El rango de posiciones direccionables queda limitado por el tamaño del campo de direccionamiento.
- d.El tamaño del operando direccionado queda limitado por el número de bits del campo de direccionamiento.

4. ¿Cuál es la diferencia entre los desplazamientos a la derecha lógico y aritmético?

- a.El lógico inserta siempre ceros en el bit más a la derecha
- b.El aritmético inserta en el bit más a la derecha una copia del bit de signo
- c.Insertan de forma distinta el bit más a la izquierda

d.Ninguna, la diferencia es entre los desplazamientos a la izquierda

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

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

b.Pone AL a 1 si CF=0 o ZF=0

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

d.Pone AL a 1 si CF=1 o ZF=1

6.Los arrays bidimensionales en lenguaje C se almacenan en orden...

a."mayor-de-columna" (column-major)

b."mayor-de-fila" (row-major)

c."de mayor a menor" (major-to-minor)

d."de fila a columna" (file-to-column)

7.¿Cuál sería el "equivalente x86-64" del "pseudo-código C" `rcx = ((int*)rax)[rcx]`?

a.`mov (%rax,%rcx,4),%rcx`

b.`lea (%rax,%rcx,4),%rcx`

c.`lea (%rax,4,%rcx),%rcx`

d.`mov (%rax,4,%rcx),%rcx`

8. El direccionamiento directo a memoria utiliza

a. un registro.

b. dos desplazamientos contenidos en la propia instrucción.

c. un registro y un desplazamiento contenidos en la propia instrucción.

d. un desplazamiento.

9. ¿Cuál de las siguientes afirmaciones es falsa?

a.Los programas necesitan una forma de pasar parámetros a las subrutinas y de recibir las salidas de vuelta.

b.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.

c.Las subrutinas necesitan recibir parámetros desde el programa que hace la llamada que indiquen qué registros pueden alterar y cuáles no.

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.

10. Respecto a la segmentación:

a. Cuanto más parecidos sean el tiempo de ejecución de una instrucción sin segmentar y el tiempo de una etapa en el procesador segmentado, mayor será la ganancia máxima que se puede obtener.

b.Cuando el número de instrucciones ejecutadas tiende al número de etapas de un procesador segmentado, la ganancia máxima que se puede obtener tiende a infinito.

c.Cuanto mayor sea la relación entre el tiempo de ejecución de una instrucción sin segmentar y el tiempo de una etapa en el procesador segmentado, mayor será la ganancia máxima que se puede obtener

d.Cuando el número de instrucciones ejecutadas en un procesador segmentado crece, la ganancia máxima que se puede obtener tiende a 1

11. 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.

12. 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)

13. La predicción de saltos 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)

14. ¿Cuál de los siguientes modos de direccionamiento es *menos* preferible para un procesador con segmentación de cauce?

- a. Registro
- b. Indirecto a través de memoria**
- c. Indirecto a través de registro
- d. Indexado (o relativo a base, o base+índice)

15. 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 (por ejemplo 0x0210) puede ser una posición de memoria o un puerto de E/S, según IO/M#
- b. La misma dirección puede usarse para transferir un byte o una palabra de mayor tamaño (ese byte y el siguiente)
- c. Un puerto será de sólo lectura, otro de sólo escritura, y ambos se decodifican en la misma dirección**
- d. La pregunta es capciosa, una dirección puede especificar un puerto, no dos

16. Respecto a si un computador dispone de E/S independiente (separada) o usa E/S mapeada a memoria:

- a.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
- b.Si el repertorio del procesador tiene instrucciones del tipo IN y OUT, es que el computador dispone de E/S separada
- c.Si el encapsulado del procesador no dispone de patilla IO/M# (ni equivalentes), el computador sólo dispone de E/S separada
- d.Si el repertorio del procesador tiene instrucciones del tipo LOAD y STORE, el computador sólo dispone de E/S mapeada a memoria

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

18. ¿Cuál de las siguientes afirmaciones es incorrecta?

- a. 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)
- b. 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)
- 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

19. 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 sin consulta de estado
- b. **E/S programada con consulta de estado**
- c. E/S por interrupciones
- d. E/S por DMA

20. Un procesador con E/S en el mapa de memoria tiene un bus de direcciones de 10 líneas y uno de datos de 8. El mapa de memoria tiene 512 posiciones para instrucciones, 256 para datos y 256 para E/S, en ese orden. Los rangos de direcciones para esas tres zonas serán:

- a. 000a1FF,200a2FFy300a3FF.
- b. 000a5FF,600a7FFy800a9FF.
- c. 000 a 7FF, 800 a BFF y C00 a FFF.
- d. 000a9FF,A00aCFFyD00aFFF.

21. El instrumento GIADA de la sonda espacial ROSETTA (diseñado en Granada) está basado en un microprocesador 8086 y el siguiente mapa de memoria:

RAM volátil: 00000 - 0FFFF

RAM no volátil: 10000 - 1FFFF

ROM: F0000 - FFFFF

¿Cuál es el tamaño total de la memoria?

- a. 48KB
- b. 192KB**
- c. 2MB

d.3MB

22.Una memoria que está estructurada en palabras de 8 bits tiene una capacidad de 64 Kbits. ¿Cuántas líneas de dirección tiene dicha memoria?

a.8

b.12

c.13

d.24

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

a.Referenciar elementos de un array sucesivamente

b.Iterar repetidamente el cuerpo de un bucle

c.Reservar dinámicamente (malloc) espacio para una estructura o union

d.Referenciar continuamente la misma variable local

24.¿Cuál de las siguientes es una idea fundamental de la jerarquía de memoria?

a.Crear una gran cantidad de almacenamiento que sea caro y rápido

b.Crear una pequeña cantidad de almacenamiento que sea caro y lento

c.Que dispositivos más pequeños y rápidos sirvan de cache para dispositivos más grandes y lentos

d.Que dispositivos más grandes y lentos sirvan de cache para dispositivos más pequeños y rápidos

25.¿Cuál de las siguientes afirmaciones acerca de las memorias SRAM es falsa?

a.Los datos almacenados se mantienen por un tiempo indefinido mientras se mantenga la alimentación.

b.El número de transistores necesario para implementar cada celda es mayor que en las memorias RAM dinámicas

c.Las operaciones de lectura son destructivas

d.Son más veloces que las memorias RAM dinámicas

26. Se dispone de un circuito integrado que actúa como módulo básico de memoria de $8K \times 4$. ¿Qué circuitos necesitamos para construir una memoria de $16K \times 8$?

- a. 2 módulos de memoria.
- b. 2 módulos de memoria y un decodificador.
- c. 4 módulos de memoria.
- d. 4 módulos de memoria y un inversor.**

27. En la cache L1 de instrucciones, la tasa de fallos:

- a. Siempre tiende a crecer si el tamaño total de L1 crece.
- b. Siempre tiende a disminuir si el tamaño total de L1 crece.**
- c. Siempre tiende a crecer si el número de vías crece.
- d. Siempre tiende a disminuir si el número de vías disminuye.

28. Supongamos que una memoria cache de tamaño 512B con política LRU está inicialmente vacía. Si 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 entonces el tamaño de bloque de la cache?

- a. 4B
- b. 8B**
- c. 16B
- d. Ninguno de los anteriores

29. Un sistema tiene una cache asociativa por conjuntos de 2-vías con 16 conjuntos y líneas de 64B. ¿A qué conjunto le corresponde el byte con dirección 0xdeadbeef?

- a. 7
- b. 11**
- c. 13
- d. 14

30. ¿Cuál es el tamaño de la etiqueta de caché en un ordenador capaz de direccionar por bytes 1 MB de memoria principal y 32 KB de memoria caché y correspondencia asociativa por conjuntos con 32 bytes por línea y 16 líneas por conjunto?

a. 6 bits

b. 7 bits

c. 8 bits

d. 9 bits

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:e807000000 callq400550<f>
```

```
400549:488903 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 micropogramación, 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 micropogramación 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 micropogramas ("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 ($256 \times 32 = 64 \times 4 \times 32$)

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 (entero ocupa 8)
- 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 micropogramada

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 (2^4+1 bit de control)

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(10/4)

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 (2^{12})

b. 10

c. 11

d. 12

TEST PRÁCTICAS

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

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 x 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)

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.olibC

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`:

```

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:

```
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 “\x61\x62\x72\x61\x63\x61\x64\x61\x62\x72\x61”, 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" ("cadabraabrabra")

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.

1.El switch de gcc para que únicamente compile de lenguaje C a ensamblador, y no realice ningún paso adicional (ensamblar, enlazar, etc), es...

a.-c

b.-S

c.-O

d.-g

2.Los switches --32 y --64 para trabajar en 32bit/64bit corresponden a la herramienta...

a.gcc

b.as

c.ld

d.nm

3.El switch -l para indicar librerías *NO* funciona con la herramienta...

a.gcc

b.as

c.ld

d.no se puede marcar una y solo una de las anteriores

4.¿Cuál de las siguientes no es una sección de un fichero ELF?

- a. .text
- b. .static
- c. .data
- d. .bss

5. ¿Cuál de los siguientes contenidos no está incluido en un fichero ELF ejecutable?

- a. código máquina
- b. variables globales
- c. pila del usuario
- d. tabla de símbolos

6. 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?

- a. 0x1111 1111
- b. 0x9999 9999
- c. 0xAAAA AAAA
- d. 0xFFFF FFFF

7. 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?

- a. se diferencian
- b. en uno ocupa 32 bits, en otro 64 bits
- c. en uno interpreta como negativo, en otro como positivo
- d. 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 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  
    cltd  
    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}

c.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("
            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 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 1KB en 1KB.
- d. de 64KB en 64KB.

1. La dirección efectiva del primer parámetro de llamada a una función suele calcularse desde el código de la función como:

- a. EBP+8**
- b. EBP-8
- c. EBP+4

d.EBP-4

2.El comienzo de un procedimiento que siga la convención cdecl es:

- a. mov%ebp,%esp;push%ebp
- b. mov %esp,%ebp; push %ebp push%ebp;
- c. mov%ebp,%esp push %ebp;
- d. mov %esp,%ebp**

3.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?

- a. %ebp+8
- b. %ebp + 12**
- c. %ebp+16
- d. %ebp+20

4.¿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 cachés de nivel 1
- d. La caché de nivel 3 no contiene toda la memoria que maneja el programa

5. En un sistema Linux x86-64, ¿cuál de las siguientes variables ocupa más bytes en memoria?

- a. char a[7]
- b. short b[3]
- c. int*c**

d. float d

6. 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 acarreos. ¿Cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32bits (sin signo)?

a. 0xfc00 0000

b. 0xfbff ffff

c. 0x0800 0000

d. 0x07ff ffff

7. 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 <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?

- 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>: ...
```

- a.el entero 0x804a044
- b.el entero almacenado a partir de la posición de memoria 0x804a044**
- c.elstringalmacenadoapartirdelaposición 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 <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 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 observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

- a. 4/32/512estudiantes
- b. 8 / 64 / 2048 estudiantes
- c. 16/32/64estudiantes
- 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 x 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

1. En contraposición a un ejecutable Linux ELF, un fichero objeto (obtenido con gcc -c)

- a.no contienen tablas de símbolos, que sólo se añaden al ejecutable tras enlazar
- b. ubica el código (y los datos) a partir de la posición 0x0, las direcciones definitivas sólo se calculan tras enlazar

c. contiene tablas de símbolos, pero sólo para los símbolos locales; p.ej., sólo al enlazar un programa que llame a printf se añadirá una entrada a la tabla con la dirección de printf

d. no reserva espacio para las direcciones de objetos desconocidos; p.ej., sólo al enlazar un CALL a printf se insertan los 4B de la dirección de printf entre el código de CALL y el de la siguiente instrucción máquina

2. Tras ejecutar las tres instrucciones que se muestran desensambladas a continuación, el registro EAX toma el valor

08048074 <_start>:

```
8048074: be 74 80 04 08 mov $_start, %esi
8048079: 46           inc  %esi
804807a: 8b 06         mov (%esi), %eax
```

a. 0x08048074

b. 0x08048075

c. 0x08048079

d. 0x0804807a

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

a. 0xf182

b. 0xf280

c. 0x1e180

d. Ninguna de las respuestas anteriores es correcta

4. ¿Cuál de las siguientes secuencias de instrucciones multiplica %eax por 10?

a. leal(%eax,%eax,4),%eax sall \$2, %eax

b. imull \$0x10, %eax

c. addl %eax,%eax shll \$5, %eax

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

5. ¿Cuál de las siguientes secuencias de instrucciones calcula $a=b-a$, suponiendo que %eax esay %ebx esb?

a. subl %eax, %ebx

b. subl %ebx, %eax

c. notl %eax

addl %ebx, %eax

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

6. Para desplazar %eax a la derecha un número variable de posiciones 32, indicado en %ebx, se puede hacer

a. sar %ebx, %eax

b. sar %bl, %eax

c. mov %ebx, %ecx sar %cl, %eax

d. No se puede, sar sólo admite un número fijo de posiciones (debe ser un valor inmediato)

7. 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

a. 08048b90

b. 08048b8b

c. 450a854e

d. No se puede deducir de la información proporcionada, faltan datos

8. El ajuste de marco de pila que gcc (Linux/IA-32) prepara para todas las funciones consiste en las instrucciones

a. movl %ebp, %esp pushl %ebp

b.`movl %esp, %ebp` `pushl %esp`

c.`pushl %ebp`

`movl %esp, %ebp`

d.`pushl %esp`

`movl %ebp, %esp`

9. 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

a.var (el nombre de la variable representa su posición de memoria)

b.-`k(%ebp)`, siendo k un número constante positivo relativamente pequeño

c.`k(%ebp)`, siendo k un número constante positivo relativamente pequeño

d. `k(%esp)`, siendo k un número constante positivo relativamente pequeño

10. En x86_64 se pueden referenciar los registros

a. `%rax, %eax, %ax, %ah, %al`

b. `%rsi, %esi, %si, %sih, %sil`

c. `%r8, %r8d, %r8w, %r8l`

d. `%r12q, %r12d, %r12w, %r12l`

11. Comparando las convenciones de llamada de gcc Linux IA-32 con x86_64 respecto a registros

a. En IA-32 `%ebx` es salva-invocante, pero en x86_64 `%rbx` es salva-invocado

b. En IA-32 `%ecx` es salva-invocante, y en x86_64 `%rcx` es salva-invocante también

c. En IA-32 `%esi` es salva-invocado, y en x86_64 `%rsi` es salva-invocado también

d. En IA-32 `%ebp` es especial (marco de pila), y en x86_64 `%rbp` también

12. Si declaramos int val[5]={1,5,2,1, 3}; entonces

a. `&val[2]` es de tipo `int*` y vale lo mismo que `val+8`

b. `val+4` es de tipo `int*` y se cumple que `*(val+4)==5`

- c. val+1 es de tipo int y vale 2
- d. val[5] es de tipo int y vale 3

13. Al traducir la sentencia C r->i = val; gcc genera el código ASM movl %edx, 12(%eax). Se deduce que

- a. resunpuntero que apunta a la posición de memoria 12
- b. el desplazamiento de i en *r es 12 c. i es un entero que vale 12
- d. val es un entero que vale 12

14. 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
```

- 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, no se pueden adivinar dimensiones
- d. var es un array bidimensional de enteros, con cinco columnas

15. 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 overflow. ¿Cuál es el menor valor positivo que repetido en toda la lista causaría overflow con 32bits?

- a. 0x04000000
- b. 0x0800 0000
- c. 0x40000000
- d. 0x8000 0000**

16. 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 overflow. ¿Cuál es el mayor valor negativo (menor en valor absoluto) que repetido en toda la lista causaría overflow con 32bits?

- a. 0xffff ffff
- b. 0xfc00 0000
- c. 0xfbff ffff
- d. 0xf000 0000

17. La práctica “parity” debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. La siguiente función contiene un único error (en realidad se han editado 2 líneas de código sobre la versión correcta) pero produce resultado correcto cuando se usa como test el array

```
int parity4(unsigned* array,
            int len){
    int i;
    unsigned x;
    int val=0, result=0;

    for (i=0; i<len; i++){
        x = array[i];

        asm("\n"
"ini:          \n\t"
"    xor %[x], %[v]  \n\t"
"    shr %[x]         \n\t"
"    jnz ini          \n\t"
": [v]" "+r" (val)
: [x] "r" (x)
);
        result += val & 0x1;
    }
    return result;
}
```

- a. array={0,1,2,3}
- b. array={1, 2, 4, 8}
- c. array={1,16,256,1024}
- d. array={5, 4, 3, 2}

18. 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 del línease 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

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 el programa “size.cc” 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:

- a. La cache L1 es cuatro veces más rápida que la cache L2.
- b. La cache L1 es seis veces más rápida que la cache L2.
- c. La cache L1 es al menos cuatro veces más rápida que la cache L2.
- d. La cache L1 es como mucho cuatro veces más rápida que la cache L2.

1. ¿De qué tipo son los procesadores Intel que usamos en los laboratorios?

- a. little-endian
- b. big-endian
- c. puede ajustarse mediante un bit de control en el registro CR0 que funcionen como little- o big-endian

d.el concepto de endian no es aplicable a estasmáquinas,yaqueunregistrodel procesador no cabe en una posición de memoria

2.En qué registro pasa el primer argumento a una función según el estándar _cdecl en una arquitectura IA32?

- a.edi
- b.esi
- c.eax

d.Las anteriores respuestas son erróneas

3.¿Cuál de las siguientes instrucciones es incorrecta en ensamblador GNU/as IA32?

- a.pop%eip
- b.pop %ebp
- c.mov(%esp),%ebp
- c.lea 0x10(%esp),%ebp

4.En un sistema Linux x86-64, ¿cuál de las siguientes variables ocupa más bytes en memoria?

- a. char a[7]
- b. short b[3]
- c. int*c
- d. float d

5. Si EBX=0x00002a00 y EDI=0x0000000a, ¿cuál es la dirección efectiva en la instrucción add 0x64(%ebx,%edi,2),%eax?

- a. 0x00002a70
- b. 0x00002a78
- c. 0x00005478
- d. 0x000054dc

6. Las instrucciones que asignan la dirección 0x78e00 a un puntero situado en la posición 0xff00 son:

- a. mov 0x78e00,%eax; mov %eax,0xff00
- b. mov 0x78e00,%eax; lea %eax,0xff00
- c. **mov\$0x78e00,%eax;mov%eax,0xff00**
- d. mov \$0x78e00,%eax; lea %eax,0xff00

7. Si ECX vale 0, la instrucción adc \$- 1,%ecx

- a. PoneCF=1
- b. Pone CF=0
- c. CambiaCF
- d. **No cambia CF**

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

- a. **0xABFC**
- b. 0xABFE
- c. 0xAC02
- d. 0xAC04

9. En IA32, tras dividir 0x640000 (64 bits) entre 0x8000 (32 bits), el resultado será:

- a. 0x0enAXy0xC8enDX.
- b. 0x0enEAXy0xC8enEDX.
- c. 0xC8enAXy0xC8enDX.
- d. **0xC8enEAXy0x0enEDX.**

10. 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:

- a. **0x0008d040y0x00000000**
- b. 0x0008d040 y 0x0008d040
- c. 0x0008d03cy0x0008d03c

d. 0x0008d044 y 0x00000000

11. La instrucción leave hace exactamente lo mismo que

a. mov%ebp,%esp;pop%ebp

b. pop %eip

c. mov%esp,%ebp;pop%esp

d. ret

12. Sean un int*a y un 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]?

a. ret(%ecx,%edx,4)

b. leal (%ecx,%edx,4),%eax; ret

c. mov(%ecx,%edx,4),%eax;ret

d. mov (%ecx,%edx,1),%eax; ret

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

a. .text _start:

b. .section .text .global _start _start:

c. .section .text .local _start _start:

d. section .text .start _global _start:

14. En la práctica 2 se pide sumar una lista de 32 enteros SIN signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando acarreos. ¿Cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32bits (sin signo)?

a. 0x07ff ffff_

b. 0x0800 0000

c. 0xfbff ffff

d. 0xfc00 0000

15. 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"
            "setpe %%dl\n\t"
            "movzx %%dl, %[x]\n\t"
            : [x] "+r" (x)
            :
            : "edx"
            );
        result += x;
    }
    return result;
}
```

a. Produce El Resultado Correcto

b. No es correcta, fallaría por ejemplo con
array={0,1,2,3}

c. No es correcta, fallaría por ejemplo 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

16. 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 incorrecta?

a. res un registro de entrada/salida

b. a es un registro de entrada

c.i es un registro de salida

d.se desea que el valor calculado por la instrucción ensamblador quede almacenado en la variable C result

17. 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 apartir de la posición de memoria 0x24(%esp)
- d. ninguna de las anteriores

18. 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

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 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:

- a. La cache L1 es cuatro veces más rápida que la cache L2.
- b. La cache L1 es seis veces más rápida que la cache L2.
- c. La cache L1 es al menos cuatro veces más de rápida que la cache L2.
- d. La cache L1 es como mucho cuatro veces más rápida que la cache L2

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:

- a. con `array={0x80000000, 0x00400000, 0x00000200, 0x00000001}`
 - b. con `array={0x10000000, 0x00200000, 0x00000400, 0x00000008}`
 - c. con ambos ejemplos
 - d. 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...

a. el entero 0x400590

b. el entero 0x400600

c. el entero almacenado a partir de 0xc(%rsp)

d. el entero alm. a partir de 0x1ab+0x400752

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.

a. 1/2 (un fallo por cada 2 accesos)

b. 1/4 (un fallo por cada 4 accesos)

c. 1/8 (un fallo por cada 8 accesos)

d. 1/16 (un fallo por cada 16 accesos)

9. En el programa line.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 line.cc ¿Qué forma tiene la gráfica que se puede obtener?
- a. Forma de U (o V), con un tramo descendente y otro ascendente
- b. Forma de /, una gráfica siempre creciente
- 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. ¿Cuál de las siguientes instrucciones es incorrecta?
- a. shr \$1,%rdx
- b. shr %rdx
- c. 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 (%0,%rdx,4), %%eax\n"  
        "    inc %%rdx\n"  
        "    cmp %%rdx,%rsi\n"  
        "    jne bucle\n"  
        "    : // output\n"  
        "    : // input\n"  
        "    : \"cc\", \"rax\", \"rdx\"\n"  
        );  
}
```

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. popcorn1 (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

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

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, 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 movz1q %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 micropogramada, bits del registro IR direccionan una ROM cuya salida puede cargarse en el registro μ 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 μ -op. $IR = ROM[MAR]$

d. dispone de la funcionalidad “`goto f(IR)`”

18. Para el procesador con unidad de control micropogramada 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/parallelismo. 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 micro-operaciones 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

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:

- a. ambos nos muestran el string (el mensaje “Hola mundo”)
- b. el primero (x) nos muestra el string, y el segundo (print) nos muestra otro valor distinto
- c. el segundo (print) nos muestra el string, y el primero (x) nos muestra otro valor distinto
- d. alguno o ambos contienen algún error gramatical (falta o sobra algún &, algún typecast (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?

- a. 0xFFFF FFFF
- b. 0x7FFF FFFF
- c. 0x1000 0000
- d. 0x0FFF FFFF

3. 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: .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...)

a. media = -1, resto = -8

b. media = -2, resto = 8

c. media = -1, resto = 8

d. media = -2, resto = -8

4. En la misma práctica "media" (cltq/cdqe 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/idivq %rcx no hace la division,

#quedá 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 0x7fffffff)

a. media = 0x0fffffff, resto = 0

b. media = 0x7fffffff, resto = 0

c. media = 0xffffffff, resto = 0

d. resto distinto de cero

5. En la misma práctica "media" (cltq/cdqe 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:

- a. para lista: .int 1, 2, 1, 2, ... (16 elementos)
- b. para lista: .int -1,-2,-1,-2, ... (16 elementos)
- c. para ambos ejemplos
- d. 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`:

- 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

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):

- 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 `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& 0x01010101010101L;  
            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`:

- 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 `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:

- a. con `array={0,1,2,3}`

b. con array={1,2,4,8}

c. con ambos ejemplos

d. 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 = 0x0f0f0f0f0f0f0f0f;  
    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:

- a. con array={3,2,1,0}
- b. con array={8,4,2,1}
- c. 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) recorremos 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

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 rex 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≠-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 111111101111000010. ¿Cuál será su contenido tras ejecutar la instrucción sar \$1,%ax?

a. 0x3fde1

b. 0x7fde1

c. 0xffde1

d. 0xfffffffffffffde1

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. 0xffffffff8a
- 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. 0xffffffff8a

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 micropogramada 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/parallelismo. 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 segundo 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...w-1) (los menos significativos)
- b. bits w...w+c-1 (con c=b-v)
- c. bits w...w+c-1 (siendo $2c=n^o$ conjuntos)
- d. bits b...b+c-1 (siendo $2c=n^o$ 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

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:

- a. ambos nos muestran la longitud del string (que es/vale/equivale a 28)
- b. el primero (x) nos muestra la longitud, y el segundo (print) nos muestra otro valor distinto
- c. el segundo (print) nos muestra la longitud, y el primero (x) nos muestra otro valor distinto
- d. 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?

- a. 0xFFFF FFFF
- b. 0x7FFF FFFF
- c. 0x1000 0000
- d. 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

jnl 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:

- a. con lista: .int 0x10000000, ... (16 elementos)
- b. con lista: .int 200000000, ... (16 elementos)
- c. con ambos ejemplos
- d. con ninguno de los dos ejemplos

4. En la práctica "media" se pide usar cltd/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 ffffffff0 (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:

- a. al inicio de main EAX vale 0 y R8 contiene un valor inferior a 0x7fffffff00000000

- b. al llamar a suma RBX contiene un valor inferior a 0x600000 y RCX vale 16
- c. al llamar a printf, ECX vale 16 y R8 contiene un valor superior a 0x80000000
- d. 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 0x7fffffff)

a. media = 2147483647 resto = 0

= 0x 7fffffff = 0x 00000000

b. media = 16 resto = -16

= 0x 7fffffff = 0x 00000000

c. media = 2147483647 resto = 0

= 0x 00000010 = 0x ffffff0

d. media = 16 resto = -16

= 0x 00000010 = 0x ffffff0

6. ¿Cuál expresión es cierta?

- a. $\text{popcount}(15) < \text{popcount}(51)$
- b. $\text{popcount}(2) == \text{popcount}(64)$
- c. $\text{popcount}(7) > \text{popcount}(60)$
- d. $\text{popcount}(96) != \text{popcount}(3)$

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 pc1(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 $\text{array}=\{0,1,2,3\}$
- c. fallaría con $\text{array}=\{1,2,4,8\}$
- 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"
"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"
"ini4: \n\t" "adc $0, %[r] \n"
"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 $\text{array}=\{0,1,2,3\}$
- c. fallaría con $\text{array}=\{1,2,4,8\}$
- 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 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

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 2025, devuelve la mitad entre sensorLow y sensorHigh
- d. Si sensorValue vale la mitad entre sensorLow y sensorHigh, devuelve 2025**

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

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 EDX 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

cmove \$17, %rax

realiza la operación:

a. valor = cond ? 13 : 17;

b. valor = 17;

c. valor = cond ? 17 : 13;

d. 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 acople (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

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

a. .text: _

_start:

b. .text _

.local _start

_start:

c. .text _

.start _global

_start:

d. .text _

.global _start

e. _start:

2. 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?

a. 4 bytes

b. 8 bytes

c. 16 bytes

d. 32 bytes

3. En la convención cdecl estándar para arquitecturas x86 de 32 bits, cuál de las siguientes afirmaciones es cierta:

a. Los 6 primeros parámetros se pasan a través de registros

b. Solamente es necesario salvar el registro EAX

c. Los registros EBX, ESI y EDI son salvainvocante

d. Ninguna de las anteriores es cierta

4. ¿Cuál es el **popcount (peso Hamming, nº de bits activados) del número 0x10101010?**

a. 4

b. 8

c. 16

d. 32

5. Compilar de fuente C a ejecutable usando sólo as y ld, sin gcc...

a. Se puede, repartiendo entre as y ld los modificadores (switches) que corresponda

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

c. Se puede, repartiendo modificadores entre as y ld, y añadiendo al comando ld el runtime de C

d. No se puede

6. La función gettimeofday() en la práctica de **popcount y **parity** se utiliza para**

a. Comparar las duraciones de las distintas soluciones del programa

b. Imprimir la fecha y hora

c. Cifrar el código en función de la hora actual

d. Cronometrar lo que tarda el usuario en pulsar una tecla

7. ¿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. `pop(199) > pop(99)`
- b. `par(199) < par(99)`
- c. `pop(199) < par(99)`
- d. `par(199) > 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 excede la longitud del vector
- d. sacar un uno (1) por el stream line

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?

- a. Si asignamos f1=bit2float(u1); entonces f1== 2147483648.00
- b. Si asignamos f1=bit2float(u1); entonces f1== -0.0**
- c. Si asignamos f2= (float)u1 ; entonces f2== 4294967296.00
- d. 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:

- a. Intercambiar el elemento 0 del vector con el 99, el 1 con el 98, el 2 con el 97 y así sucesivamente.
- b. Intercambiar el elemento 0 del vector con el 1, el 1 con el 2, el 3 con el 4 y así sucesivamente.
- c. 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.
- d. 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*)

- a. registros de acople (coste de segmentación)
- b. fragmentación desigual (duración de etapas)
- c. riesgos (hazards)
- d. 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?

- a. 2 ns
- b. 3 ns**
- c. 4.5 ns
- d. 9 ns

19. ¿Qué es un controlador de E/S?

- a. Un circuito electrónico que implementa la memoria del computador.
- b. Un circuito impreso del tipo DIMM.
- c. Un circuito electrónico que puede guardar temporalmente datos enviados desde el procesador al periférico o viceversa.**
- d. Un bus que permite interconectar distintos periféricos entre sí.

20. Respecto a la interfaz de E/S, ¿cuál de las siguientes afirmaciones es *FALSA*?

- a. Involucra tareas que se pueden implementar parte en hardware y parte en software.
- b. Permite configurar el funcionamiento del periférico en un momento determinado, y además conocer su estado.
- c. 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.**

d. 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?

a. Determinadas zonas del espacio de direccionamiento del procesador se asignan por convenio a controladores de E/S.

b. Un ejemplo de mecanismo de E/S mapeada en memoria es la instrucción IN de los procesadores Intel.

c. Una misma dirección se usa alternativamente para E/S y para memoria en distintos momentos de ejecución de un programa.

d. Un pin IO/M# del procesador permite distinguir si accedemos a E/S o a memoria

22. ¿Cuál de las siguientes afirmaciones es *FALSA*?

a. 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).

b. 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).

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 permitir 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 Dn-1-D0.
- c. Direcciones An-1-A0.

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.

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:

- a. está correctamente diseñado, la instrucción ret final es optativa, y no es concebible que quitar el ret cause algún error
- b. produce un error "Segmentation fault" cuando empieza a acceder a memoria que no le corresponde (EIP)
- c. 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"
- d. no se puede marcar ninguna de las opciones anteriores

5. ¿Cuál de las siguientes sumas con signo produce desbordamiento con 32 bits?

- a. 0xFFFFFFFF + 0xFFFFFFFF
- b. 0x7FFFFFFF + 0xFFFFFFFF
- c. 0x7FFFFFFF + 0x00000001
- d. 0xFFFFFFFF + 0x00000001

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?

- a. 0x0400 0000
- b. 0x0800 0000
- c. 0x4000 0000
- d. 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?

- a. 0xffff ffff
- b. 0xfc00 0000
- c. 0xfbff ffff
- d. 0xf000 0000

8. ¿Cuál es el **popcount (peso Hamming, nº de bits activados) del número 29?**

- a. 2
- b. 3
- c. 4
- d. 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:**

- 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 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 $\text{array}=\{0,1,2,3\}$
- c. fallaría con $\text{array}=\{1,2,4,8\}$
- 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.

1. En el direccionamiento indirecto a través de registro, la dirección efectiva...

a. se encuentra en una dirección de memoria

b. se encuentra en un registro general del procesador

c. se encuentra en el registro de instrucción

d. 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:

a. $N \cdot F/C$

b. $N \cdot C/F$

c. $N \cdot F \cdot C$

d. N/(C·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)?

- a. long
- b. char *
- c. ninguno de los dos
- d. ambos

4. La secuencia de instrucciones

leaq (%rdi, %rdi, 2), %rax

salq \$3, %rax

produce el efecto de...

- a. leer en RAX a partir de la posición $2 * \text{RDI} + 2$ y multiplicar dicho contenido por 8
- b. calcular $\text{RAX} = (2 * \text{RDI} + 2) * 3 = 6 * (\text{RDI} + 1)$
- c. calcular $\text{RAX} = 24 * \text{RDI}$
- d. calcular $\text{RAX} = 9 * \text{RDI}$

5. ¿Qué valor contendrá el registro RDX tras ejecutar las dos instrucciones siguientes?

movq \$-1, %rdx

movb \$1, %dl

- a. 0xFFFF FFFF FFFF FFF1
- b. 0xFFFF FFFF FFFF FF01
- c. 0xFFFF FFFF FFFF 0001
- d. 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?

- a. 0x400544
- b. 0x400549
- c. 0x40054b
- d. 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:

- a. 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 An-1-A0

d. Datos Dn-1-D0

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*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 (callee) 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

e. 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.

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:

- a. acumulación de desbordamientos (OF, overflow flag)
- b. acumulación de acarreos (CF, carry flag)
- c. extensión con ceros
- d. extensión de signo

2. En la misma práctica "media", 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

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`:

a. produce siempre el resultado correcto

b. fallaría con `array={0,1,2,3}`

c. fallaría con `array={0,-1,-2,-3}`

d. 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 4^a banda es un multiplicador y la 5^a banda es la tolerancia

b. Son de color beige y tienen 4 bandas de color: las 2 primeras indican un valor, la 3^a banda es un multiplicador y la 4^a 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.

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) 0x40054b
no, eso sería la dirección de inicio de mult2, mal calculada
(desde CALL)
 $400544+7=40054b$
- b) 0x400544
no, eso sería la dirección de la propia instrucción CALL
- c) 0x400550
no, eso sería la dirección de inicio de mult2, bien calculada
(desde MOV)
 $400549+7=400550$

- 0x400549
- d) sí, se guarda la dirección de retorno, la dirección de inicio de la siguiente instrucción

Suponga la expresión: unsigned int val = x & 0x01010101; donde x es un unsigned int. ¿Cuál de los siguientes valores de x da como resultado val = 0x01010101?

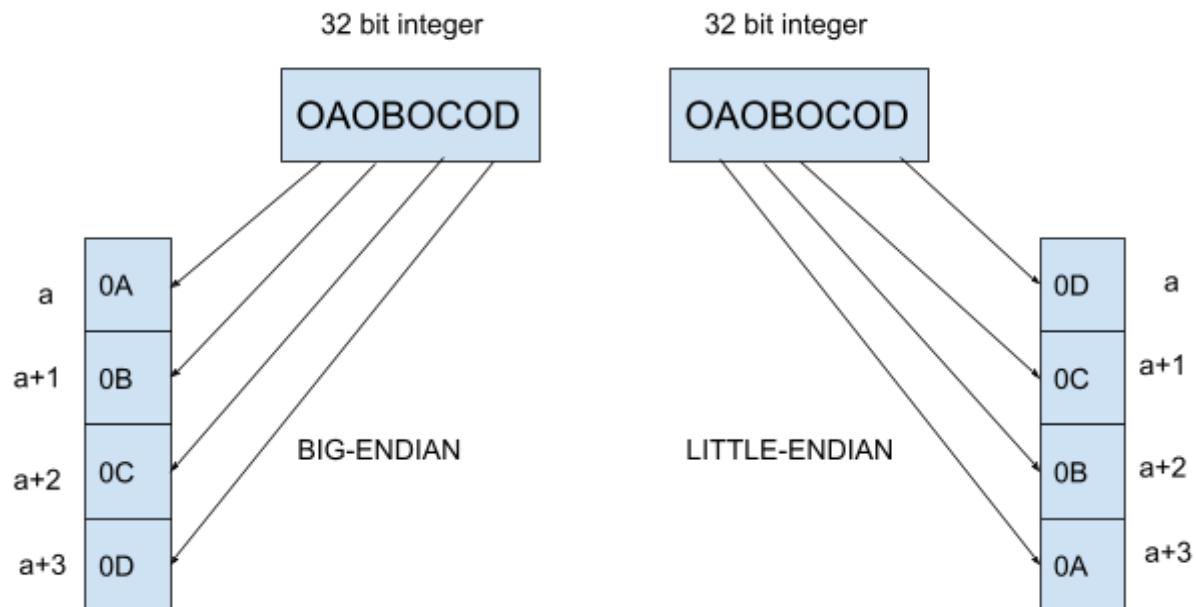
- a) 0x1010101
0
- b) 0aaaaaaaa
a
- c) 0eeeeeee
e
- d) 0x13579bdf

ENSAMBLADOR:

```
leaq (%rdi, %rdi, 2), %rax -> rdi^3
salq $3, %rax-> rxax*3
movq $-1, %rdx (64 bits)
movb $1, %dl
400544: e8 07 00 00 00 callq 400550 <f> -> saltar a la posición llamada con call (lo que hace es empujar en el STACK la posición de memoria en la que va. Luego salta hacia donde el operando de CALL se lo ordena. )
400549: 48 89 03 mov %rax,(%rbx)
addq %rax, %rsi -> suma a rxax rsi
ret -> RET realiza un retorno de subrutina. La dirección de retorno es cargada de la pila.
shr -> Utilice SHR para desplazar el operando de la izquierda, que es siempre tratado como sin signo
```

Datos alineados

IA-32		AMD64	
char	1 byte	char	1 byte
short	2 bytes	short	2 bytes
int, long int, float, punteros	4 bytes	int, float	4 bytes
double, long long int	8 bytes	double, long int , punteros	8 bytes
long double	12 bytes	long double, long long int	16 bytes

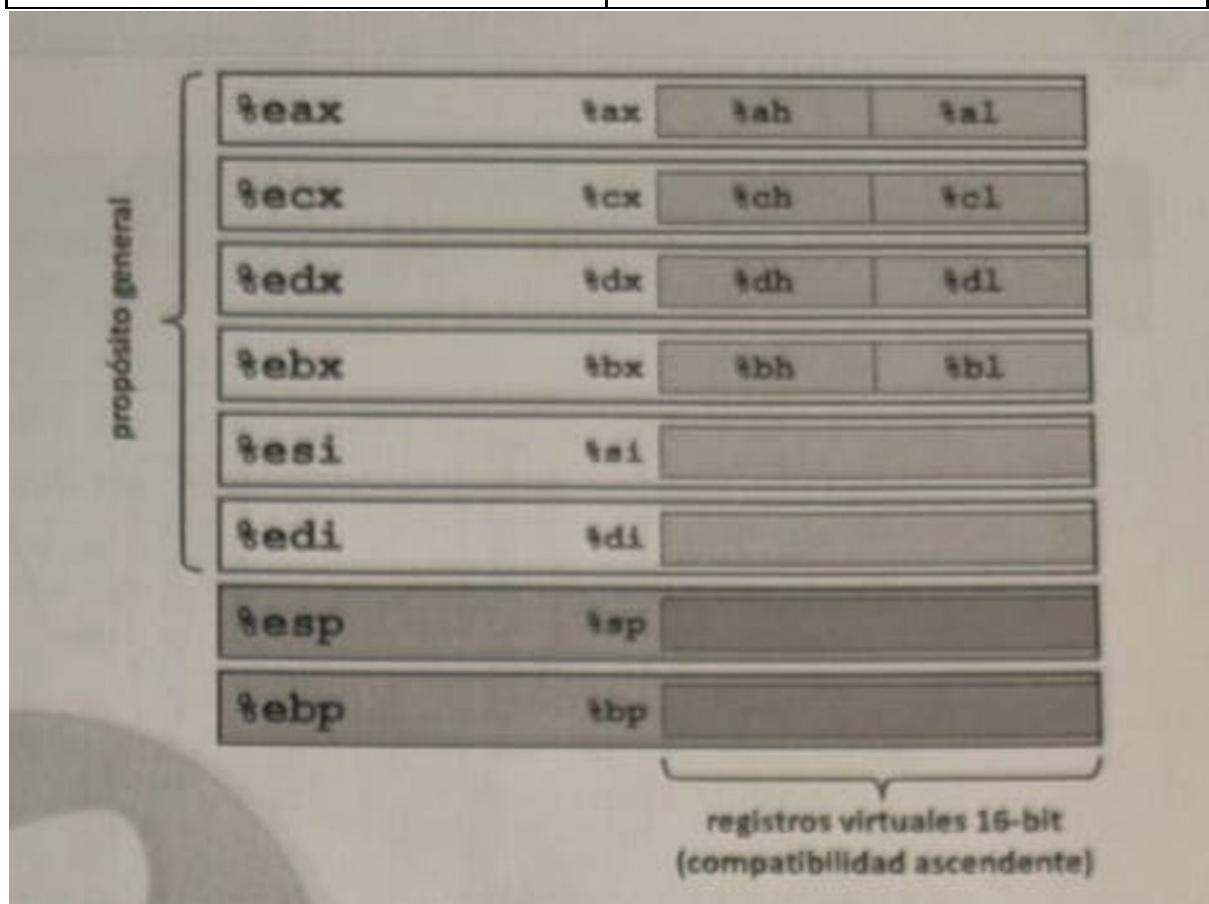


NOMBRE	COMENTARIO	CÓDIGO	OPERACIÓN
ADD	suma	addl fte , dst	dst = dst + fte
ADC	suma con acarreo	adcl fte , dst	dst = dst + fte + CF
SUB	resta	subl fte , dst	dst = dst - fte
DIV IDIV	división sin signo división entera con signo	div op idiv op	op -> byte AL = AX/op AH = Resto
DIV IDIV	división sin signo división entera con signo	idv op idiv op	op -> word AL = DX:AX/op DX = Resto
DIV IDIV	división sin signo división entera con signo	div op idiv op	op -> doubleword EAX = EDX:EAX/op EDX = Resto

MUL IMUL	multiplicac sin signo multiplicación entera con signo	mul op imul op	op -> byte AX =AL*op CF=0 , OF=0 si AH=0 CF=1, OF=1 en otro caso
MUL IMUL	multiplicac sin signo multiplicación entera con signo	mul op imul op	op -> word DX:AX =AX*op CF=0 , OF=0 si DX=0 CF=1, OF=1 en otro caso
MUL IMUL	multiplicac sin signo multiplicación entera con signo	mul op imul op	op -> doubleword EDX:EAX =EAX*op CF=0 , OF=0 si EDX=0 CF=1, OF=1 en otro caso
INC	incrementar	incl dst	dst = dst + 1(CF no es afectado)
DEC	decrementar	decl dst	dst = dst - 1(CF no es afectado)
NOT	negación a nivel de bit	notl dst	dst = -dst
NEG	negación	negl dst	dst = -dst
AND &	y a nivel de bit	and fte , dst	dst = dst & fte
OR	o a nivel de bit	orl fte , dst	dst = dst fte
XOR ^	o exclusivo a nivel de bit (sirve para averiguar la paridad)	xorl fte , dst	dst = dst ^ fte
SAR >>	desplazamiento aritmético a la derecha	sar fte , dst	dst = dst << fte
SAL <<	desplazamiento aritmético a la izquierda	sall fte , dst	dst = dst >> fte
SHR >>	desplazamiento lógico a la derecha	shr fte , dst	dst = dst << fte
SHL <<	desplazamiento lógico a la izquierda	shll fte , dst	dst = dst >> fte
LEA	mueve dirección de memoria	leaq (%rdi, %rdi, 2), %rax	rdi ^ 3
MOV	mueve contenido	movb \$1, %dl	
RET	realiza un retorno de subrutina. La dirección de retorno es cargada de la pila.		
CALL	empujar en el STACK la posición de memoria	400544: e8 07 00 00 00 callq 400550 <f>	

	en la que va. Luego salta hacia donde el operando de CALL se lo ordena.		
--	---	--	--

FLAG	NOMBRES
PF	flag de paridad
ZF	flag zero
SF	signo flag
OF	desbordamiento overflow
CF	acarreo



- SALTOS CONDICIONALES

SALTO	DESCRIPCIÓN	FLAGS	OP
JO	saltar si overflow	OF=1	70
JNO	saltar si no overflow	OF=0	71
JS	saltar si signo	SF=1	78
JNS	saltar si no signo	SF=0	79
JE JZ	saltar si es igual saltar si es cero	ZF=1	74
JNE JNZ	saltar si no es igual saltar si no es cero	ZF=0	75
JP JPE	saltar si la paridad saltar si paridad aun	PF=1	7A
JNP JPO	saltar si no paridad jump if parity odd	PF=0	7B
JB JNAE JC	jump if below jump if no above or equal jump if carry	CF=1	72
JNB JAE JNC	jump if not below jump if no above or equal jump if not carry	CF=0	73
JBE JNA	jump if below or equal jump if not above	CF=1 or ZF=1	76
JA JNBE	jump if above jump if not below or equal	CF=0 or ZF=0	77
JL JNGE	jump if less jump if not greater or equal	SF<>OF	7C
JGE JNL	jump if greater or equal jump if not less	SF=OF	7D
JLE JNG	jump if less or equal jump if not greater	ZF=1 or SF<>OF	7E
JG JNLE	jump if greater jump if not less or equal	ZF=0 and SF=OF	TF

- COMPARACIONES

INSTRUCCIÓN	OPERACIÓN
cmp op1, op2	op2 - op1
test op1, op2	op2 & op1

- EXTENSIÓN DE REGISTRO

INSTRUCCIÓN	OPERACIÓN
movs{bw}{wl}	extensión de signo
movz{bw}{wl}	extensión para usigned
cltd	extiende el signo de eax a edx
cmoveXX{wlq}	move condicional

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) Sigue 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 produciría 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) $x \leq -1 \text{ || } x \geq 11$
- d) $x \geq 0 \text{ && } x \leq 10$

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 4 e 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 micropogramada.
- 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) Sigue 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 $RDX*8 + RBX$?

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

`lea` calcula la dirección efectiva $[RBX+RDX*8]$ 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 $8+2*RDX$

- d) `movq 8(%rdx, %rdx), %rax`

`mov` mueve desde memoria, y además la dirección efectiva sería $8+2*RDX$

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 NTeo>=2.5p y NPrá>=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?
- Un protocolo sirve para “ponerse de acuerdo” en cosas como velocidad, paridad, nº de bits, etc.
 - La mayoría de los periféricos trabajan a velocidad muy superior a la CPU; por eso es necesario sincronizar.
 - Un controlador se encarga de la comunicación con la CPU.
 - 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?
- Sí, permite averiguar el estado de un periférico y también alterar la prioridad de los periféricos
 - No, porque no permite ni averiguar el estado de un periférico ni alterar la prioridad de los periféricos
 - No, porque no permite averiguar el estado de un periférico
 - 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?
- 23 ns
 - 32 ns
 - 57.5 ns
 - 92 ns
67. Cuando dos o más instrucciones necesitan un recurso hardware en el mismo ciclo, se trata de un riesgo:
- estructural
 - de control
 - de salto
 - por dependencia de datos
68. En el arbitraje de un bus...
- los dispositivos pasivos pueden requerir el uso del bus para iniciar una transferencia
 - si hay un único dispositivo pasivo, siempre funciona como esclavo
 - si hay varios dispositivos activos, siempre funcionan como maestros
 - 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?
(Ob 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) 0aaaaaaaaa
 - c) 0eeeeeeeee
 - 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,Indice,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 CR0 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 únicamente 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 acople, 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) `movzlq %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
cltd
- 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) 0xFFFF
- d) 0xFF00

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 ahorraremos 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 ahorraremos 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 0xCAFEBAE, el byte 0xCE quedará almacenado en la dirección:
- a) 0xCAFEBAC1
 - b) 0xCAFEBAC0
 - c) 0xCAFEBABF
 - d) 0xCAFEBABE
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.



RullenCastro

www.wuolah.com/student/RullenCastro



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 <strcmp>

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

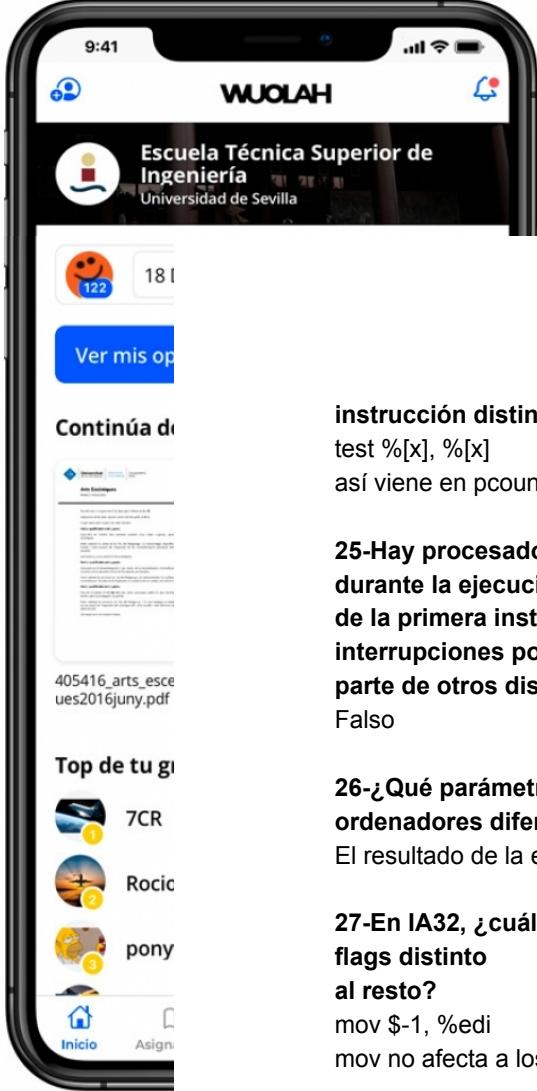
23-En la práctica "media" se suma una lista de 32 enteros de 4 B con signo para producir una media y un resto usando la instrucción IDIV. ¿Cuál de las siguientes afirmaciones es falsa?

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.

Available on the
App Store

GET IT ON
Google Play

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 -

"I da 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

direcciónamiento 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 direcciónamiento 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

**Ids IOBuf ; Apuntar puntero pila a
; ...área mem.intermedia
Idx 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·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

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 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.

Available on the
App Store

GET IT ON
Google Play

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 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.

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 int*a y un 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

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

1^a 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 llenar 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 envia 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)  
);
```

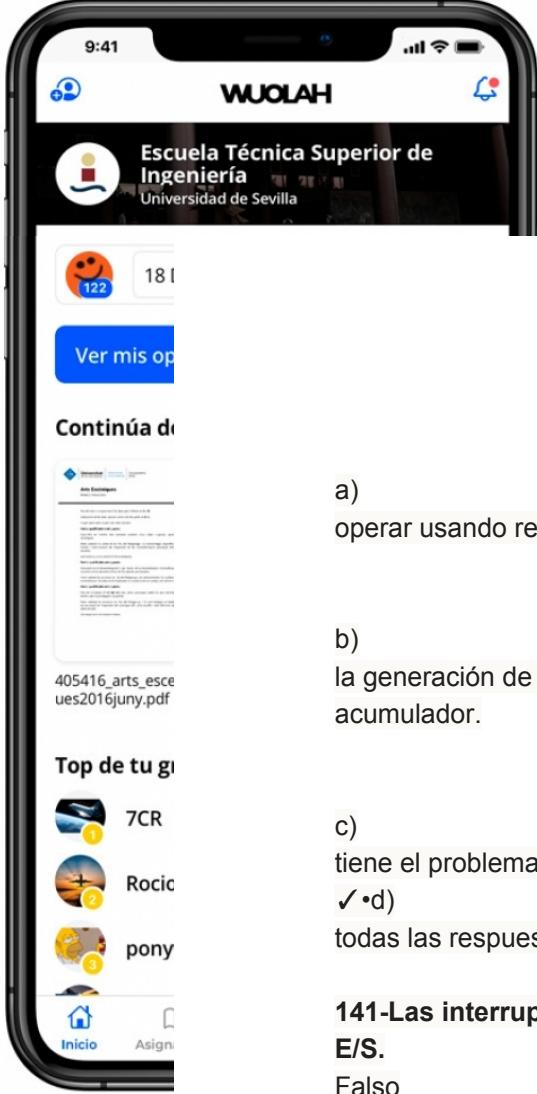
¿Cuál de las siguientes afirmaciones es correcta?

- 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.

Available on the
App Store

GET IT ON
Google Play

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.

Verdadero

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

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 0xCAFEBAB0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al consultar la media palabra de la posición 0xCAFEBABE?

-1

los contenidos son

CAFEBAB0: FF FF FF FF

CAFEBAB4: FE FF FF FF

CAFEBAB8: FD FF FF FF

CAFEBABC: FC FF FF FF

las últimas dos posiciones, a partir de CAFEBABC, 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

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 micropograma 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^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

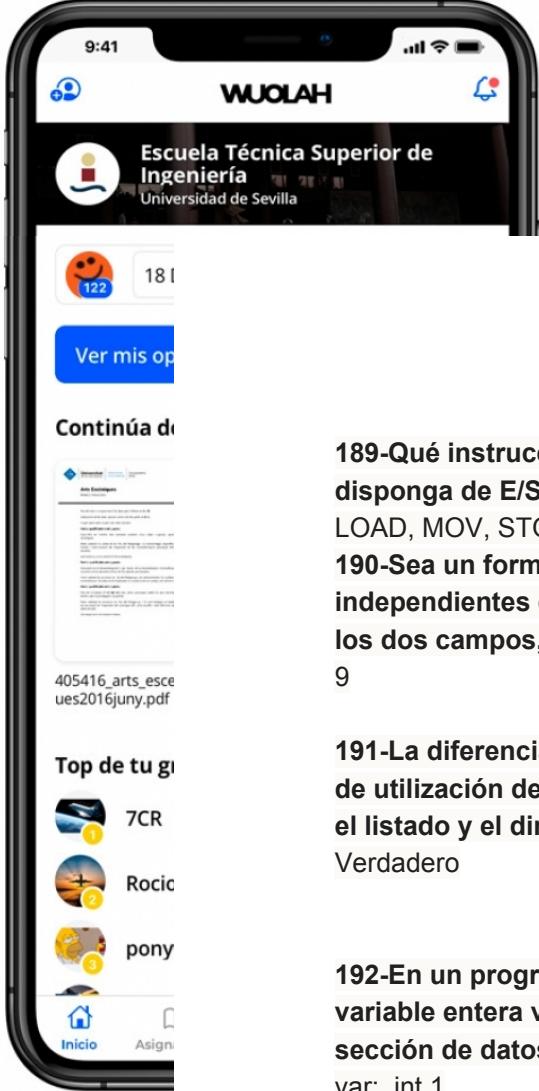
opuestos a programas secuenciales son los programas paralelos

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.

Available on the
App Store

GET IT ON
Google Play

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:

0xCAFEBAC0

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
- 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

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

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 micropograma.

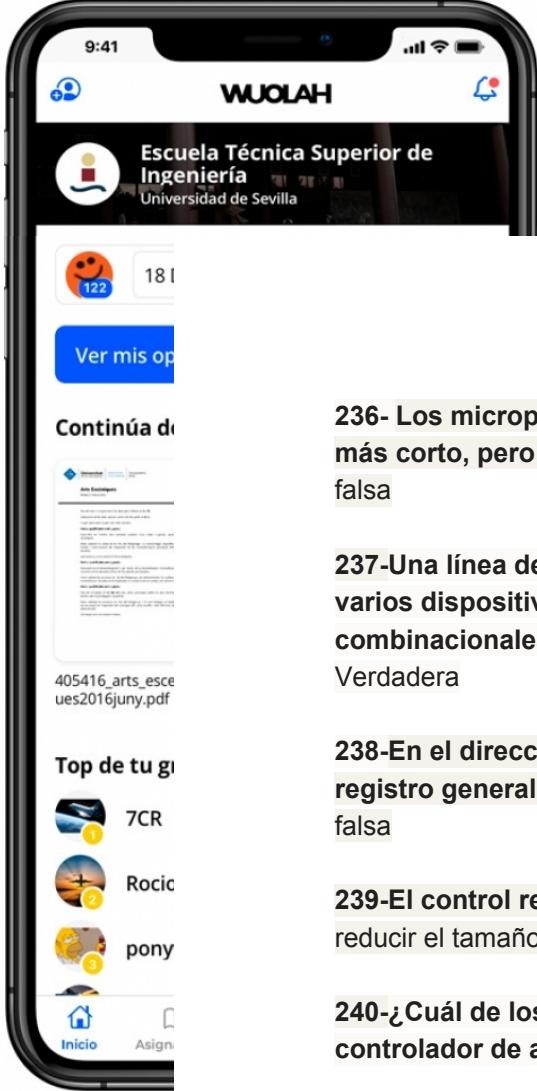
Verdadero

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.

Available on the
App Store

GET IT ON
Google Play

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.
veradero

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

veradero

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

veradero

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



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 ffffffe0 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/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 5^a 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)

$\ast(\ast(x + 2)) + 7$

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.

Available on the
App Store

GET IT ON
Google Play

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-EI 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 del 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



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

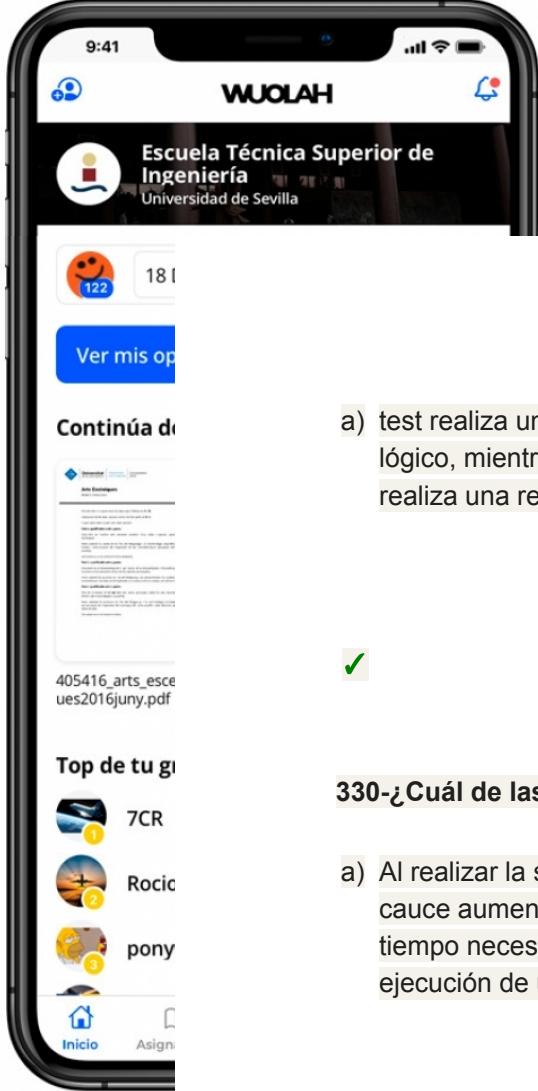
327-Más de la cuarta parte de las instrucciones máquina que se ejecutan en un programa típico son de movimiento o transferencia de datos.

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.

Available on the
App Store

GET IT ON
Google Play

- 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



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.

✓

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.

✓

376-Dada la siguiente definición de datos:

```
lista: .int 0x10000000, 0x50000000,  
        0x10000000, 0x20000000
```

```
longlista: .int (.lista)/4
```

```
resultado: .quad 0x123456789ABCDEF
```

```
formato: .ascii "suma=%lu=%lx 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.

✓

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

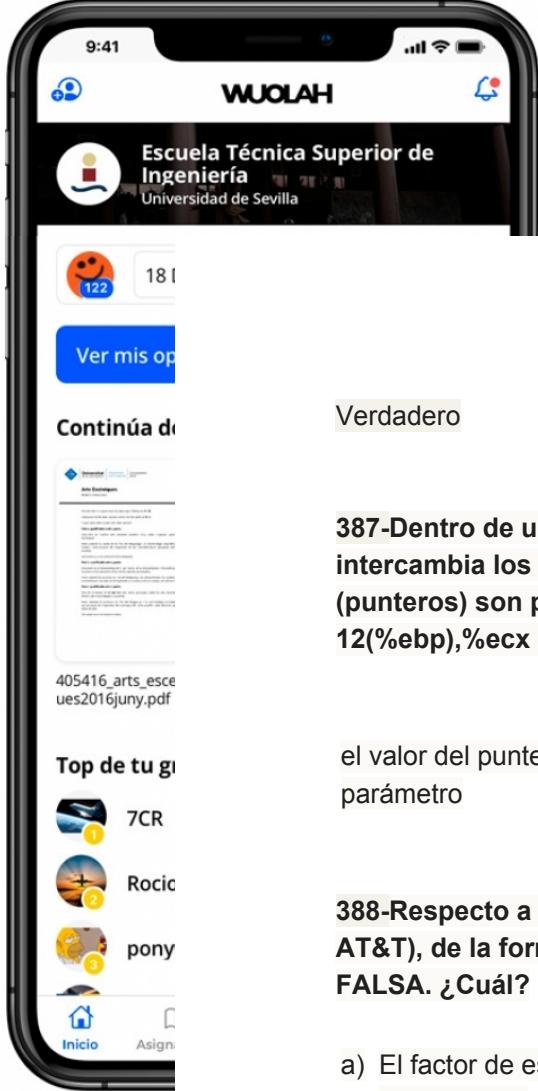
384-¿Cuántos controladores de interrupciones 8259 hacen falta como mínimo para manejar 25 líneas de interrupción?

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.

Available on the
App Store

GET IT ON
Google Play

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) geométric
- a

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, una 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

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

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 Id 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

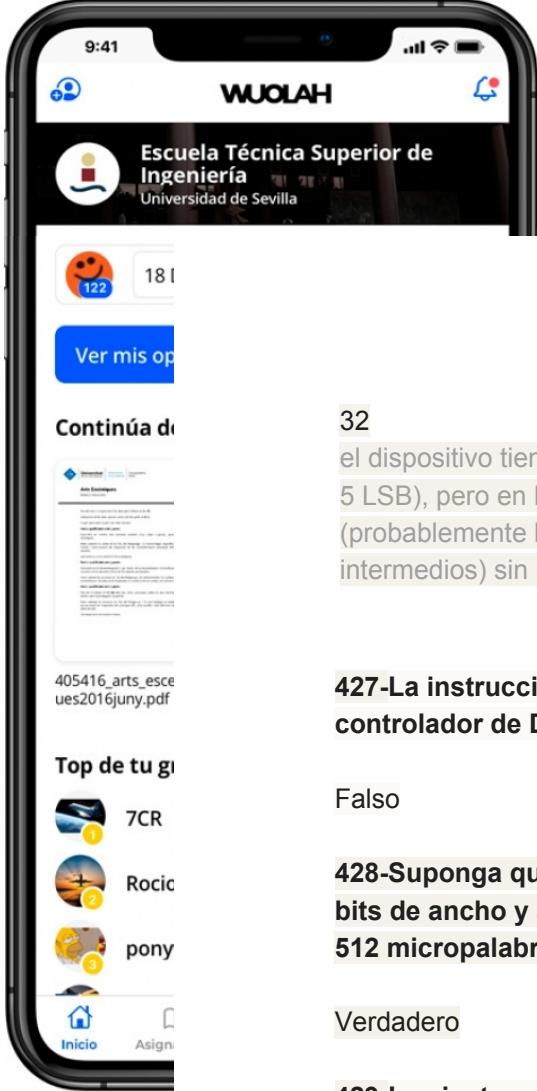
VERDADERO

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.

Available on the
App Store

GET IT ON
Google Play

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"
        );
    }
    return res;
}
```

```
"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

ringteacher



447-Para desplazar %eax a la derecha un número variable de posiciones <= 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 ^ 0b1010 en lenguaje C?

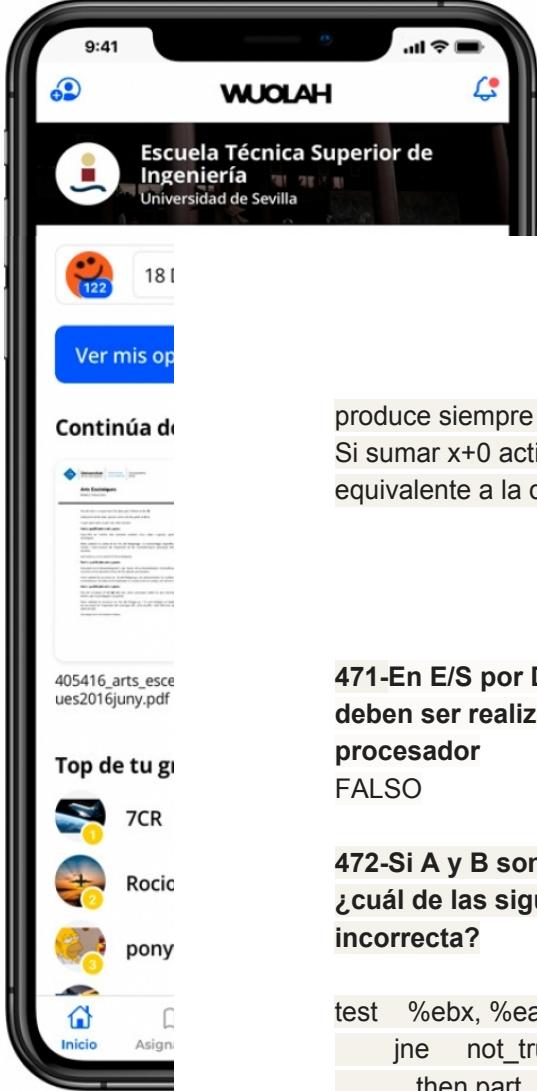
0b0100 (hace el AND y pone 1 cuando no es AND, 1 y 1 = 0, 0 y 1 = 1)

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.

Available on the
App Store

GET IT ON
Google Play

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.

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

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...

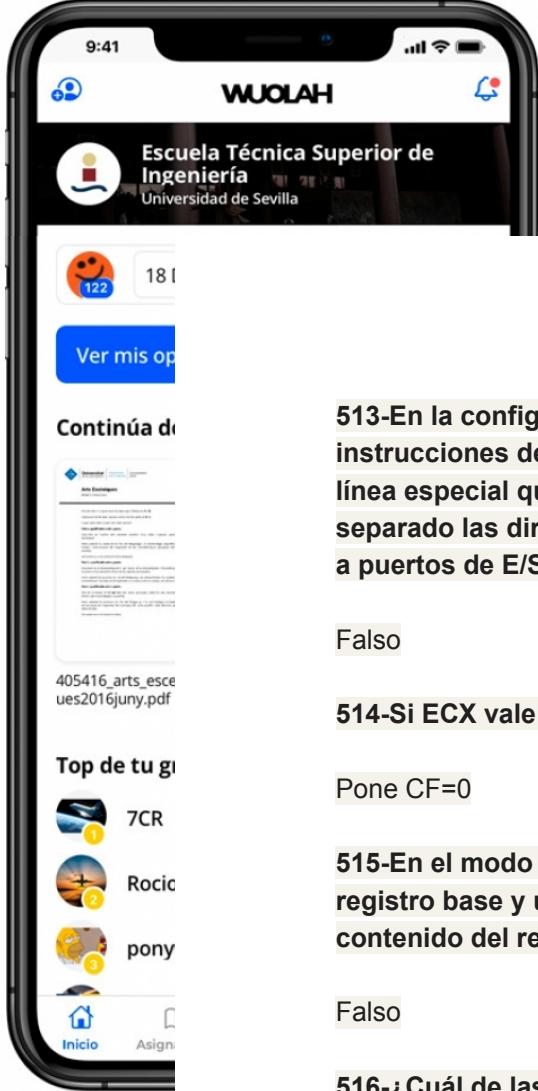
Realizar la operación and lógico bit-a-bit (a&b) pero no guardar el resultado, sino simplemente ajustar los flags

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.

Available on the
App Store

GET IT ON
Google Play

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  
cltd  
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

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.

Available on the
App Store

GET IT ON
Google Play

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=%lu=%lx 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
- X
- 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

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)

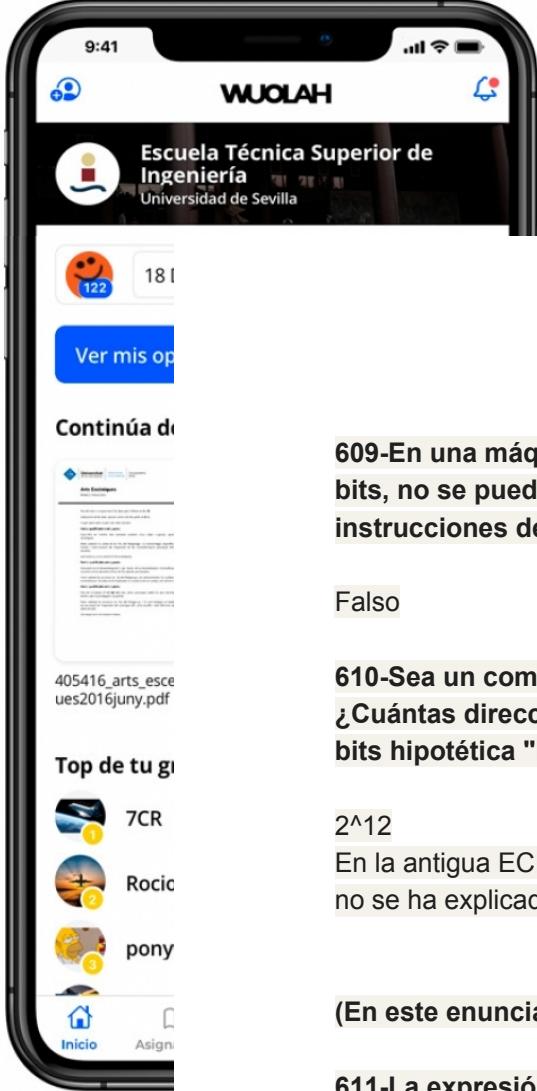
redacción original era "x86-64 Linux long double 8x, Windows no", motivo por el cual notamos errata transparencias (no dicen nada de x86-64 Windows, y dicen que x86-64 Linux long double 8x, cuando libro pág.325 dice que 16x). Posible corrección: "alinea long double a 12x (Windows no)". Redacción final 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.

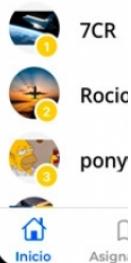
Available on the
App Store

GET IT ON
Google Play

Continúa dí



Top de tu gi



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 != 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:
muchas codificaciones

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

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?

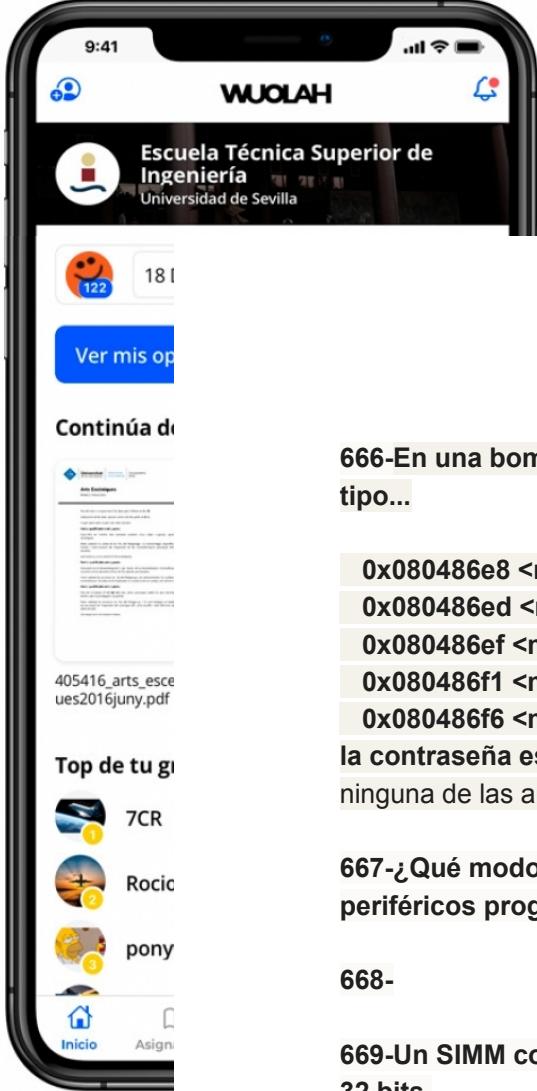
662-Una cache de 256 B asociativa por conjuntos de 4-vías con líneas de 16 B tendría

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.

Available on the
App Store

GET IT ON
Google Play

666-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...

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

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 <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

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-El 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".

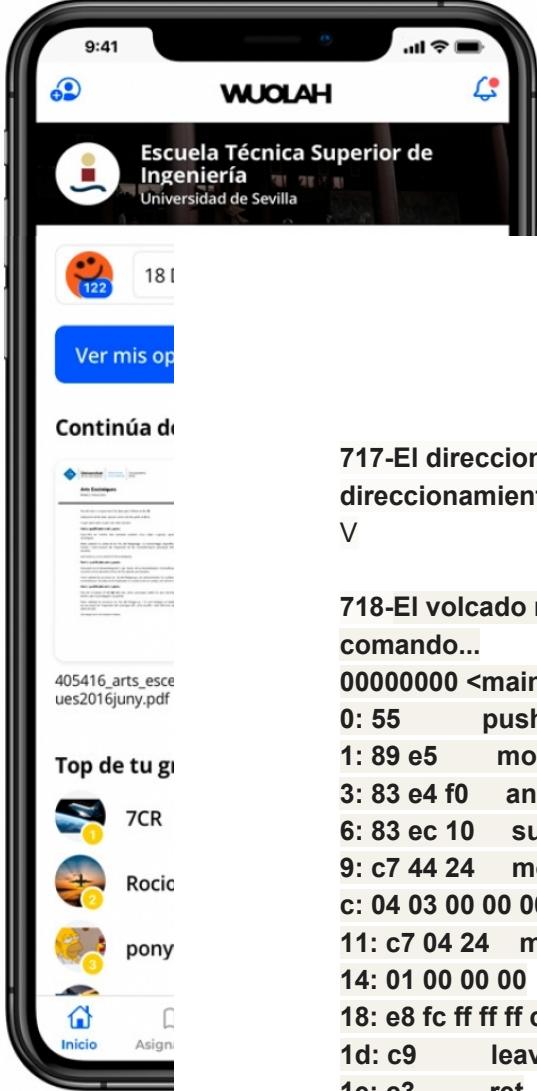
FALSO

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.

Available on the
App Store

GET IT ON
Google Play

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\n0"
```

```
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 <- 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
sal \$2, %eax
sería x5x4=x20

b)

imull \$0x10, %eax
sería x16

c)

addl %eax, %eax
shll \$5, %eax
sería x2x32=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=%lu=%lx 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 $0, 1, \dots, 2M-1$ 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

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?

3^a generación (1965-75)

en la 3^a 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?

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

1. La dirección efectiva del primer parámetro de llamada a una función suele calcularse desde el código de la función como:
 - a. EBP+8
 - b. EBP-8
 - c. EBP+4
 - d. EBP-4

 2. El comienzo de un procedimiento que siga la convención cdecl es:
 - a. mov %ebp,%esp; push %ebp
 - b. mov %esp,%ebp; push %ebp
 - c. push %ebp; mov %ebp,%esp
 - d. push %ebp; mov %esp,%ebp

 3. 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?
 - a. %ebp + 8
 - b. %ebp + 12
 - c. %ebp + 16
 - d. %ebp + 20

 4. ¿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 cachés de nivel 1
 - d. La caché de nivel 3 no contiene toda la memoria que maneja el programa
-
5. En un sistema Linux x86-64, ¿cuál de las siguientes variables ocupa más bytes en memoria?
 - a. char a[7]
 - b. short b[3]
 - c. int *c
 - d. float d

 6. 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 acarreos. ¿Cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32bits (sin signo)?
 - a. 0xfc00 0000
 - b. 0xfbff ffff
 - c. 0x0800 0000
 - d. 0x07ff ffff

 7. 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"
            "movzxx %%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 observarán 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

1. El switch de gcc para que únicamente compile de lenguaje C a ensamblador, y no realice ningún paso adicional (ensamblar, enlazar, etc), es...
 - a. -c
 - b. -S
 - c. -o
 - d. -g

2. Los switches --32 y --64 para trabajar en 32bit/64bit corresponden a la herramienta...
 - a. gcc
 - b. as
 - c. ld
 - d. nm

3. El switch -l para indicar librerías ***NO*** funciona con la herramienta...
 - a. gcc
 - b. as
 - c. ld
 - d. no se puede marcar una y solo una de las anteriores

4. ¿Cuál de las siguientes no es una sección de un fichero ELF?
 - a. .text
 - b. .static
 - c. .data
 - d. .bss

5. ¿Cuál de los siguientes contenidos no está incluido en un fichero ELF ejecutable?
 - a. código máquina
 - b. variables globales
 - c. pila del usuario
 - d. tabla de símbolos

6. 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?
 - a. 0x1111 1111
 - b. 0x9999 9999
 - c. 0xAAAA AAAA
 - d. 0xFFFF FFFF

7. 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?
 - 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

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
    cltd
    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 sobreescritos (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 sobreescritos (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.
-

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. ¿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 x 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)

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 los siguientes fragmentos es correcto para comenzar un programa en ensamblador que conste de un solo archivo .s?
- a. `.text:`
`_start:`
 - b. `.text`
`.local _start`
`_start:`
 - c. `.text`
`.start _global`
`_start:`
 - d. `.text`
`.global _start`
`_start:`
-
2. 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?
- a. 4 bytes
 - b. 8 bytes
 - c. 16 bytes
 - d. 32 bytes
-
3. En la convención cdecl estándar para arquitecturas x86 de 32 bits, cuál de las siguientes afirmaciones es cierta:
- a. Los 6 primeros parámetros se pasan a través de registros
 - b. Solamente es necesario salvar el registro EAX
-
- c. Los registros EBX, ESI y EDI son salvados por el invocante
- d. Ninguna de las anteriores es cierta
-
4. ¿Cuál es el popcorn (peso Hamming, nº de bits activados) del número 0x10101010?
- a. 4
 - b. 8
 - c. 16
 - d. 32
-
5. Compilar de fuente C a ejecutable usando sólo as y ld, sin gcc...
- a. Se puede, repartiendo entre as y ld los modificadores (switches) que corresponda
 - b. Basta usar ld, con los modificadores de gcc que corresponda, y añadiéndole el runtime de C
 - c. Se puede, repartiendo modificadores entre as y ld, y añadiendo al comando ld el runtime de C
 - d. No se puede
-
6. La función gettimeofday() en la práctica de popcorn y parity se utiliza para
- a. Comparar las duraciones de las distintas soluciones del programa
 - b. Imprimir la fecha y hora
 - c. Cifrar el código en función de la hora actual
 - d. Cronometrar lo que tarda el usuario en pulsar una tecla
-
7. ¿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\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. `pop(199) > pop(99)`
 - b. `par(199) < par(99)`
 - c. `pop(199) < par(99)`
 - d. `par(199) > 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 excede 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:

- a. está correctamente diseñado, la instrucción ret final es optativa, y no es concebible que quitar el ret cause algún error
 - b. produce un error "Segmentation fault" cuando empieza a acceder a memoria que no le corresponde (EIP)
 - c. 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"
 - d. no se puede marcar ninguna de las opciones anteriores
-

5. ¿Cuál de las siguientes sumas con signo produce desbordamiento con 32 bits?

- a. 0xFFFFFFFF + 0xFFFFFFFF
 - b. 0x7FFFFFFF + 0xFFFFFFFF
 - c. 0x7FFFFFFF + 0x00000001
 - d. 0xFFFFFFFF + 0x00000001
-

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?

- a. 0x0400 0000
 - b. 0x0800 0000
 - c. 0x4000 0000
 - d. 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?

- a. 0xffff ffff
 - b. 0xfc00 0000
 - c. 0xfbff ffff
 - d. 0xf000 0000
-

8. ¿Cuál es el popcorn (peso Hamming, nº de bits activados) del número 29?

- a. 2
 - b. 3
 - c. 4
 - d. 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 popcorn4:

```

int popcorn4(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 popcorn4:

- 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 popcorn4:

```

int popcorn4(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 popcorn4:

- 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. pop(129) > pop(29)
- b. par(129) > par(29)
- c. pop(129) < par(29)
- d. par(129) < 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;
```

```
}
```

```
int v[262144];
for (i = 0; i < 262144; i += 8)
    v[i] = 9;
```

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:

¿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:**

- a. ambos nos muestran la longitud del string (que es/vale/equivale a 28)
- b. el primero (x) nos muestra la longitud, y el segundo (print) nos muestra otro valor distinto
- c. el segundo (print) nos muestra la longitud, y el primero (x) nos muestra otro valor distinto
- d. 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?

- a. 0xFFFF FFFF
- b. 0x7FFF FFFF
- c. 0x1000 0000
- d. 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:

- a. con lista: **.int 0x10000000, ... (16 elementos)**
- b. con lista: **.int 200000000, ... (16 elementos)**
- c. con ambos ejemplos
- d. con ninguno de los dos ejemplos

4. En la práctica "media" se pide usar **cltd/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 ffffffff0 (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 0x7fffffff00000000
 - 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
-
- 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: .double 0
resto: .double 0
formatoq:
.ascii "media = %lld resto = %lld\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 0x7fffffff)

- `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
-

- ¿Cuál expresión es cierta?

- `popcount(15) < popcount(51)`
 - `popcount(2) == popcount(64)`
 - `popcount(7) > popcount(60)`
 - `popcount(96) != popcount(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 `popcount1`:

```

int pc1(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 pc2(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"  
            "ini4:\n"                "\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"  
            "ini4:\n"                "\n\t"  
            "adc $0, %[r]\n\t"  
            "fin4:\n"                "\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:

- a. ambos nos muestran el string (el mensaje “Hola mundo”)
- b. el primero (x) nos muestra el string, y el segundo (print) nos muestra otro valor distinto
- c. el segundo (print) nos muestra el string, y el primero (x) nos muestra otro valor distinto
- d. alguno o ambos contienen algún error gramatical (falta o sobra algún &, algún typecast (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?

- a. 0xFFFF FFFF
- b. 0x7FFF FFFF
- c. 0x1000 0000
- d. 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
    #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...)

- a. media = -1, resto = -8
- b. media = -2, resto = 8
- c. media = -1, resto = 8
- d. 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/idivq %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 0x7fffffff)

- a. media = 0x0fffffff, resto = 0
 - b. media = 0x7fffffff, resto = 0
 - c. media = 0xffffffff, resto = 0
 - d. 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:

- a. para lista: `.int 1, 2, 1, 2, ... (16 elementos)`
 - b. para lista: `.int -1,-2,-1,-2, ... (16 elementos)`
 - c. para ambos ejemplos
 - d. 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)          \n"
            "    : [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`:

- 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
-

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"

```

```

        : [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):

- 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 `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`:

- 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 `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:

- a. con `array={0,1,2,3}`
 - b. con `array={1,2,4,8}`
 - c. con ambos ejemplos**
 - d. 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 = 0x0f0f0f0f0f0f0f0f;
    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:

- a. con `array={3,2,1,0}`
- b. con `array={8,4,2,1}`
- c. 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**) recorremos 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

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. 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:

- a. acumulación de desbordamientos (OF, overflow flag)
- b. acumulación de acarreos (CF, carry flag)
- c. extensión con ceros
- d. extensión de signo

2. En la misma práctica "media", 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

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;
    }
}
```

```
    }
```

Esta función se diferencia de la versión "oficial" en los tipos de array y x.

Esta función popcorn5:

- a. produce siempre el resultado correcto
- b. fallaría con array={0,1,2,3}
- c. fallaría con array={0,-1,-2,-3}
- d. 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 popcorn5:

```
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 4^a banda es un multiplicador y la 5^a banda es la tolerancia
 - b. Son de color beige y tienen 4 bandas de color: las 2 primeras indican un valor, la 3^a banda es un multiplicador y la 4^a 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.
-

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. 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:

- a. con `array={0x80000000, 0x00400000, 0x00000200, 0x00000001}`
 - b. con `array={0x10000000, 0x00200000, 0x00000400, 0x00000008}`**
 - c. con ambos ejemplos
 - d. 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...

- a. el entero 0x400590
 - b. el entero 0x400600
 - c. el entero almacenado a partir de 0xc(%rsp)
 - d. 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:

- a. Exactamente esa solución viene en el guión, no hace falta aprenderla fuera de clase
 - b. Nosotros conectamos los led en serie con una resistencia como dice el guión
 - c. Hay que revisar ese equipo en concreto, debe tener algún defecto, ese led no debe parpadear
 - d. 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.

- a. 1/2 (un fallo por cada 2 accesos)
 - b. 1/4 (un fallo por cada 4 accesos)
 - c. 1/8 (un fallo por cada 8 accesos)
 - d. 1/16 (un fallo por cada 16 accesos)
-

9. En el programa `line.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 `line.cc` ¿Qué forma tiene la gráfica que se puede obtener?

- a. Forma de U (o V), con un tramo descendente y otro ascendente
 - b. Forma de /, una gráfica siempre creciente
 - 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. ¿Cuál de las siguientes instrucciones es incorrecta?

- a. `shr $1,%rdx`
- b. `shr %rdx`
- c. `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\n"  
        "    : : : // input\n"  
        "    : \"cc\", // clobber\n"  
        "    \"rax\", \"rdx\"\n";  
}
```

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 %dil
 - 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 `numLEDSSlit` 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

1. 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:
 - a. 0xCAFEBAC1
 - b. 0xCAFEBAC0
 - c. 0xCAFEBABF
 - d. 0xCAFEBABE

2. ¿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

3. En X86-64, el registro contador de programa se denomina:
 - a. RIP
 - b. EIP
 - c. IP
 - d. R15

4. La primera letra (l) de la instrucción lea:
 - a. indica que la instrucción usa ordenación de bytes little-endian
 - b. indica que la instrucción trabaja con un operando destino de 32 bits (long word)
 - c. indica que la instrucción afecta a los 16 bits menos significativos del operando destino (low word)
 - d. forma parte del nemotécnico de la instrucción

5. ¿Cuál es la diferencia entre las instrucciones mov y lea?
 - a. lea accede a la posición indicada, mientras que mov no lo hace
 - b. mov accede a la posición indicada, mientras que lea no lo hace
 - c. lea puede usarse para copiar un registro a otro, mientras que mov no
 - d. mov puede usarse para copiar un registro a otro, mientras que lea no

6. La instrucción not:
 - a. realiza el complemento a dos
 - b. realiza el complemento a uno (cambiar unos por ceros y ceros por unos)
 - c. realiza la operación no-or (or negada)
 - d. realiza un salto condicional si negativo

7. La instrucción JGE / JNL provoca un salto si...
 - a. SF = 1
 - b. CF = 1
 - c. SF = 0
 - d. OF = SF

8. Un overflow nunca puede ocurrir cuando:
 - a. se suman dos números positivos
 - b. se suman dos números negativos
 - c. se suma un número positivo a un número negativo
 - d. se resta un número positivo de un número negativo

9. Despues 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 ahorraremos usando nanoprogramación en lugar de micropogramación?

- 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 llenar 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 las 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 0xCAFEBAE?

- 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. 0xCAFEBAE
 - 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?
- Registro
 - Memoria
 - Cache
 - Inmediato
-
8. Un bus se compone de:
- líneas de datos y líneas de dirección
 - líneas de alimentación
 - líneas de estado y líneas de control
 - líneas de control/estado, líneas de dirección y líneas de datos
-
9. ¿Cuál de los siguientes no es un tipo de bus?
- Secuencial
 - Paralelo
 - E/S
 - 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?
- 10
 - 16
 - 32
 - 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...
- aritmética
 - geométrica
 - armónica
 - ponderada
-
12. El primer computador electrónico basaba su funcionamiento en:
- tubos de vacío
 - circuitos integrados LSI
 - amplificadores operacionales
 - núcleos de ferrita
-
13. En Linux IA-32, si gcc usa la instrucción leave se puede asegurar que en ese punto del programa
- correspondería emitir la secuencia de salida pop/ret, pero leave hace lo mismo y ocupa menos espacio
 - ya no hay registros salva-invocado que recuperar
 - ya no hay variables locales que destruir
 - 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...
- dos instrucciones mov
 - una instrucción mov y una instrucción lea
 - 3 mov, no menos (se le llama "intercambio circular")
 - 4 mov, no menos (debido a la arquitectura R/M)
-
15. Respecto a registros base e índice en IA-32, la excepción es que
- EBP no puede ser registro base
 - EBP no puede ser registro índice
 - ESP no puede ser registro base
 - ESP no puede ser registro índice
-
16. El registro SP / ESP / RSP...
- es un registro transparente al usuario y contiene la instrucción que se está ejecutando
 - es un registro de propósito específico y contiene la dirección de la cima de la pila
 - es un registro transparente al usuario y contiene la dirección de memoria a la que se está accediendo
 - 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
- los enteros largos (long) pasan de 32 a 64 bits
 - los punteros (void*) pasan de 32 a 64 bits
 - el tipo double pasa de 4 B a 8 B
 - 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. $\ast((\ast(x + 16)) + 28)$
b. $\ast((\ast x) + 2) + 7)$
c. $(\ast(\ast(x + 2) + 4) + 3)$
d. $\ast((\ast(x + 2) + 4) + 3)$

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 acople (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

1. ¿Cuál de las siguientes afirmaciones sobre el benchmark SPEC CPU es falsa?
 - a. La última versión es SPEC CPU2006 V1.2 de 2011
 - b. Se cronometran unos 12 tests de enteros (CINT2006) y unos 17 tests de punto flotante (CFP2006)
 - c. 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
 - d. El resultado final es la media aritmética de las (12 ó 17) velocidades, bien sea de enteros ó de punto flotante (SPECint2006 ó SPECfp2006)

2. ¿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?
 - a. 2ª generación (1955-65)
 - b. 3ª generación (1965-75)
 - c. 4ª generación (1975-85)
 - d. esas innovaciones se repartieron a lo largo de varias generaciones, no sólo una

3. Respecto a tamaños de tipos integrales en x86 y x86-64, la excepción es que
 - a. int pasa de 4 B (x86) a 8 B (x86-64)
 - b. long int pasa de 4 B a 8 B
 - c. long long int pasa de 4 B a 8 B
 - d. ninguna de las anteriores

4. Con el repertorio IA32, para sumar %eax y %ebx dejando el resultado en %ecx se podría hacer lo siguiente:
 - a. lea %eax, %ebx, %ecx
 - b. lea (%eax, %ebx, 1), %ecx
 - c. lea %ecx, [%eax, %ebx]
 - d. lea %ecx, %ebx, %eax

5. Cuál de las instrucciones máquina siguientes es incorrecta en x86-64:
 - a. testl %edx, %edx
 - b. movl %r8, %eax
 - c. movl (%rdi,%rcx,4), %edx
 - d. addq \$1, %rcx

6. 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:
 - a. xorl %ebx,%eax
 - b. xorl %eax,%ebx
 - c. andl %ebx,%eax
 - d. testl %eax,%ebx

7. Para poner a 1 el bit 5 del registro %edx sin cambiar el resto de bits podemos usar la instrucción máquina:
 - a. and \$32, %edx
 - b. and \$0x5, %edx
 - c. or \$0b101, %edx
 - d. or \$0x20, %edx

8. 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 $0x10000000,%ecx  
div %ecx
```

- d. Podemos usar las instrucciones siguientes tanto si n es un número con signo como sin signo:

```
mov $0x10000000,%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 micropogramada 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. 0xFBFF
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

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

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

2. 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.

3. En IA32, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto al resto?

- a. sub %edi, %edi
 adc \$0xFFFFFFFF, %edi
- b. mov \$-1, %edi
- c. mov \$-1, %edi
 add \$0, %edi
- d. mov \$0, %edi

sub \$1, %edi

- 4. Si %rsp vale 0xdeadbeefdeadd0d0, ¿cuál será su nuevo valor después de que se ejecute pushq %rbx?
 - a. 0xdeadbeefdeadd0d4
 - b. 0xdeadbeefdeadd0d8
 - c. 0xdeadbeefdeadd0cc
 - d. 0xdeadbeefdeadd0c8
- 5. ¿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. Ninguna de esas formas es la correcta.
- 6. Comparando las convenciones de llamada de gcc Linux IA32 con x86-64 respecto a registros
 - a. En IA32 %ebx es salva-invocante, pero en x86-64 %rbx es salva-invocado
 - b. En IA32 %ecx es salva-invocante, y en x86-64 %rcx es salva-invocante también
 - c. En IA32 %esi es salva-invocado, y en x86-64 %rsi es salva-invocado también
 - d. En IA32 %ebp es especial (marco de pila), y en x86-64 %rbp también
- 7. 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 µPC
 - 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  
idx 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 $k*m$ (y mismo n^o 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 micropogramada.

4. ¿Cuál de las siguientes instrucciones máquina copia en el registro EDX 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
cmpb   $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
cmovc $17, %rax
```

realiza la operación:

- a. valor = cond ? 13 : 17;
- b. valor = 17;
- c. valor = cond ? 17 : 13;
- d. 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 riesgo 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***?
- La operación de lectura de una celda DRAM es destructiva
 - Las memorias DRAM son en general más lentas que las SRAM
 - Una celda DRAM no pierde la información al desconectar la alimentación
 - Las memorias DRAM presentan generalmente una capacidad de almacenamiento mayor que las SRAM
-

19. La memoria DRAM:

- Se inventó en la década de los 2000
 - Necesita 6 transistores por cada celda
 - Se denomina dinámica porque su contenido puede alterarse, al contrario que la SRAM
 - 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

- $08048391 + 12 = 08048403$
 - $08048391 - 12 = 08048379$
 - 0804837d
 - 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; };
```

- 18 B
 - 20 B
 - 22 B
 - 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***)

- codificación
 - solapamiento
 - 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
- 4 bits
 - 5 bits
 - 16 bits
 - 17 bits
-
- 24.** Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señale la respuesta ***FALSA***)
- registros de acople (coste de la segmentación)
 - fragmentación desigual (duración desigual de etapas)
 - riesgos (*hazards*)
 - 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?
- 2 ns
 - 3 ns
 - 4,5 ns
 - 9 ns
-
- 26.** Respecto al salto retardado y al salto anulante, ¿cuál permite que se ejecute la siguiente instrucción, y cuál no?
- el retardado ejecuta la siguiente instrucción (con el correspondiente retraso), el anulante no la ejecuta (de hecho la anula)
 - el retardado la ejecuta sólo si se cumple la condición de salto, el anulante sólo si no se cumple
 - el retardado la ejecuta sólo si no se cumple la condición de salto, el anulante no la ejecuta nunca
 - 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
-

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? (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?

- a. Si asignamos f1=bit2float(u1); entonces f1== 2147483648.00
 - b. Si asignamos f1=bit2float(u1); entonces f1== -0.0**
 - c. Si asignamos f2= (float)u1 ; entonces f2== 4294967296.00
 - d. 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:

- a. Intercambiar el elemento 0 del vector con el 99, el 1 con el 98, el 2 con el 97 y así sucesivamente.
 - b. Intercambiar el elemento 0 del vector con el 1, el 1 con el 2, el 3 con el 4 y así sucesivamente.
 - c. 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.
 - d. 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*)

- a. registros de acople (coste de segmentación)
 - b. fragmentación desigual (duración de etapas)
 - c. riesgos (hazards)
 - d. 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?

- a. 2 ns
 - b. 3 ns**
 - c. 4.5 ns
 - d. 9 ns
-

19. ¿Qué es un controlador de E/S?

- a. Un circuito electrónico que implementa la memoria del computador.
 - b. Un circuito impreso del tipo DIMM.
 - c. Un circuito electrónico que puede guardar temporalmente datos enviados desde el procesador al periférico o viceversa.**
 - d. Un bus que permite interconectar distintos periféricos entre sí.
-

20. Respecto a la interfaz de E/S, ¿cuál de las siguientes afirmaciones es *FALSA*?

- a. Involucra tareas que se pueden implementar parte en hardware y parte en software.
 - b. Permite configurar el funcionamiento del periférico en un momento determinado, y además conocer su estado.
 - c. 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.
 - d. 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?

- a. Determinadas zonas del espacio de direccionamiento del procesador se asignan por convenio a controladores de E/S.**
 - b. Un ejemplo de mecanismo de E/S mapeada en memoria es la instrucción IN de los procesadores Intel.
 - c. Una misma dirección se usa alternativamente para E/S y para memoria en distintos momentos de ejecución de un programa.
 - d. Un pin IO/M# del procesador permite distinguir si accedemos a E/S o a memoria.
-

22. ¿Cuál de las siguientes afirmaciones es *FALSA*?

- a. 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).
- b. 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).

- 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 permitir 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}-D₀.
- c. Direcciones A_{n-1}-A₀.
- 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** ≠ -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 111111101111000010. ¿Cuál será su contenido tras ejecutar la instrucción **sar \$1,%ax**?

- a. 0x3fde1
- b. 0x7fde1
- c. 0xffde1

d. 0xfffffffffffffde1

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 xxxxxxxx
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. 0xfffffff8a
 - 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. 0xfffffff8a
 - 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 segundo 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...w-1) (los menos significativos)
 - b. bits w...w+c-1 (con c=b-v)
 - c. bits w...w+c-1 (siendo $2^c=n$ conjuntos)
 - d. bits b...b+c-1 (siendo $2^c=n$ 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, 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 **movzq %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):
- pone AL a 1 si ZF=1 y CF=1
 - pone AL a 1 si ZF=1 o CF=0
 - pone AL a 1 si ZF=0 o CF=1
 - pone AL a 1 si ZF=0 y CF=0
-
- 9.** La instrucción JNGE / JL provoca un salto si...
- SF = 1
 - CF = 1
 - SF ≠ OF
 - SF = CF
-
- 10.** Para traducir una asignación condicional (**a=b ? c : d ;**) de lenguaje C a lenguaje ensamblador, gcc puede que utilice...
- Un salto condicional, según la condición expresada en el código C, y otro salto incondicional
 - Un salto incondicional, según la condición opuesta a la del código C, y otro salto condicional
 - Una instrucción de movimiento incondicional, pero sólo si el procesador es Pentium Pro/II o superior
 - 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...
- el salto incondicional hacia adelante
 - el salto incondicional hacia atrás
 - el salto condicional hacia adelante
 - 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...
- el salto condicional hacia adelante, según la misma condición expresada en lenguaje C
 - el salto condicional hacia atrás, según la misma condición expresada en lenguaje C
 - el salto condicional hacia adelante, según la condición opuesta a la expresada en leng. C
 - 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...
- R12 es un registro salva-invocado, por eso en cualquier función hay que salvarlo antes de modificarlo
 - R11 es un registro salva-invocante, por eso en cualquier función hay que salvarlo antes de modificarlo
 - RBX es un registro salva-invocado, por eso si es necesario conservar su valor hay que salvarlo antes de llamar a función
 - 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?
- 0x670
  - 0xe8000000
  - 0x66b
  - 0x58
- 
- 15.** ¿Cuál sería el “equivalente x86-64” del “pseudo-código C” **rcx=((long\*)rax)[rbx]**?
- mov (%rax,%rbx,4), %rcx
  - lea (%rax,%rbx,4), %rcx
  - lea (%rax,%rbx,8), %rcx
  - mov (%rax,%rbx,8), %rcx
- 
- 16.** Habiendo declarado **int array={-4,-3,-2,-1};** y **char \*ptr=array;** ¿cuánto vale **ptr[1]**?
- 1
  - 2
  - 3
  - 4
- 
- 17.** En una unidad de control microprogramada, bits del registro IR direccionan una ROM cuya salida puede cargarse en el registro μPC (micro-PC). Esa unidad de control...
- es una UC con secuenciamiento explícito
  - puede realizar la micro-operación IR=μPC
  - puede realizar la μ-op. IR=ROM[MAR]
  - 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...

- a. se encuentra en una dirección de memoria
- b. se encuentra en un registro general del procesador
- c. se encuentra en el registro de instrucción
- d. 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:

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

**3.** ¿Cuál/es de los siguientes tipos de datos tienen distinto tamaño en arquitecturas x86 y x86-64 (Intel 32 bits y 64 bits)?

- a. long
- b. char \*
- c. ninguno de los dos
- d. ambos

**4.** La secuencia de instrucciones

```
leaq (%rdi,%rdi,2), %rax
salq $3, %rax
```

produce el efecto de...

- a. leer en RAX a partir de la posición  $2 * \text{RDI} + 2$  y multiplicar dicho contenido por 8

b. calcular  $\text{RAX} = (2 * \text{RDI} + 2) * 3 = 6 * (\text{RDI} + 1)$

c. calcular  $\text{RAX} = 24 * \text{RDI}$

d. calcular  $\text{RAX} = 9 * \text{RDI}$

**5.** ¿Qué valor contendrá el registro RDX tras ejecutar las dos instrucciones siguientes?

```
movq $-1, %rdx
movb $1, %dl
```

- a. 0xFFFF FFFF FFFF FFF1
- b. 0xFFFF FFFF FFFF FF01
- c. 0xFFFF FFFF FFFF 0001
- d. 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?

- a. 0x400544
- b. 0x400549
- c. 0x40054b
- d. 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:

- a. 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<sub>n-1</sub>-A<sub>0</sub>
  - d. Datos D<sub>n-1</sub>-D<sub>0</sub>
- 

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\*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 (callee) 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.
-

|         |  |
|---------|--|
| Nombre: |  |
| DNI:    |  |

## 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**):
- Test de Teoría (3.0p)
- 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:

- var es un array multi-nivel (punteros a enteros) de cuatro filas
  - var es un array multi-nivel pero no se pueden adivinar las dimensiones
  - var es un array bidimensional de enteros, con ocho columnas
  - var es un array bidimensional de enteros, con cinco columnas
- 
- 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...
    - codificarlas con 4 bits, y sobraría un código que quedaría sin uso
    - codificarlas con 5 bits, y sobrarían dos códigos que quedarían sin uso
    - solaparlas en un solo campo de 5 bits, ahorrando por tanto 9 bits
    - solaparlas en un solo campo de 7 bits, ahorrando por tanto 6 bits
- 
- En el contexto de microprogramación, el control residual...
    - 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".
    - 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"
    - clasifica las microinstrucciones del microcódigo según formen parte de micropogramas ("microinstr. inmediatas") o no ("microinstr. residuales")
    - 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?

- Se espera una reducción de prestaciones porque además de ejecutar las instrucciones hay que segmentarlas (coste de la segmentación)
  - Se espera una reducción de prestaciones porque la duración del ciclo de reloj vendrá impuesta por la etapa más lenta
  - Se espera un aumento de prestaciones debido al efecto de los riesgos (hazards) sobre el avance de las instrucciones en el cauce
  - 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)...

- 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
  - 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
  - siempre, de manera que instrucción(es) anterior(es) al salto podrían colocarse tras la propia instrucción de salto
  - 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?

- $2^{10}$
  - $2^{11}$
  - $2^{12}$
  - $2^{13}$
- 

**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) instrucción(es) siguiente(s) ...

- a. si el salto se produce (las ignora si NO se produce), de manera que instrucción(es) en el destino del salto podrán adelantarse tras la propia instrucción de salto
- b. si el salto NO se produce (las ignora si se produce), de manera que instrucción(es) en el destino del salto podrán adelantarse tras la propia instrucción de salto
- c. siempre, de manera que instrucción(es) <----- anterior(es) al salto podrán colocarse tras la propia instrucción de salto
- d. nunca, de manera que instrucción(es) anterior(es) al salto no podrán colocarse tras la propia instrucción de salto

12. Indicar cuál es la dirección de la instrucción 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 , Äì 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 instrucción, el tiempo de ejecución del programa será:

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

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 linea 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, ÄUPERC, Ä para habilitar la carga desde el bus C (recordar que era un diseño típico con 3 buses) y así no perder expresividad/parallelismo. 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 servía apropiado intentar reordenar instrucciones, Ä¶

- 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)<-----

## PREGUNTAS EC TEORÍA

### Normas:

- **Arial 11 y la correcta se señala en negrita.**
  - **No valen fotos, ponerla en texto.**
  - **Comprobad que la pregunta ya no esté puesta.**
  - **No entrar con la cuenta go.ugr y si se entra que al cambiar de cuenta se quite también el archivo de la zona de compartidos.**
  - **Cuidado con pisar preguntas de otra gente, escribid y ya voy entrando y ordeno.**
  - **El pdf de 750 preguntas se adjuntará al final el día de antes.**
  - **Copiad y pegad de swad que queda bonito.**
  - **Este documento se realiza para estudiar, en ningún caso se debe usar para cometer actos fraudulentos en el examen.**
- 
- 

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

### Usuario Profesores

- ✓      •      a) **MDR/MBR y MAR**  
              b) IR y PC  
              c) MDR/MBR y PC  
              d) MAR y PC
- 

---  
¿Cuál es el valor mínimo (más negativo) que puede tomar un entero de 32bits en complemento a dos?

(^ simboliza potenciación)

- a)  $-2^{32}$   
              b)  $-2^{31} + 1$   
              c)  **$-2^{31}$**   
              d)  $-2^{32} + 1$
- 

---  
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) **0xe0 a 0xff**
  - b) 0x00 a 0x0f y 0x80 a 0x8f
  - c) 0x00 a 0x1f
  - d) 0x70 a 0x7f y 0xf0 a 0xff
- 
- 

La exo-arquitectura de una máquina especifica el conjunto de instrucciones, los modos de direccionamiento, la longitud de palabra, el nº de palabras de memoria, etc.

### **VERDADERA**

---

R-----

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?

Usuaria Profesores

- a)  $2^{10}$  palabras
  - **b)  $2^{12}$  palabras**
  - c)  $2^{13}$  palabras
  - d) Ninguna de las anteriores
- 
- 

¿Cuál de las siguientes expresiones representa un direccionamiento inmediato?

1. %eax
  2. 8(%ebp)
  - 3. \$0x400**
  4. (%eax)
- 
- 

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 MBR lo recogemos**
- c) en MAR indicamos la dirección donde está y en IR lo recogemos
- d) en MBR indicamos la dirección donde está y en IR lo recogemos
- 
- 

Se dice que las máquinas con arquitectura Von Neumann siguen un modelo de programa...

- a) Externo.
- b) Cableado.
- c) Micropogramado.
- ✓ • d) **Almacenado.**
- 
- 

¿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 RISC
- d) Los microprocesadores CISC
- 
- 

Según la clasificación m/n, las máquinas con arquitectura R/R son de tipo

- a) 0/0
- b) x/0 con  $x=2,3$
- c) x/x con  $x=2,3$
- ✓ • d) **0/x con  $x=2,3$**
- 
- 

¿En qué generación, dentro de la historia de los computadores digitales, aparece la

segmentación de cauce?

Usuaria Profesores

- a) primera
  - b) segunda
  - ✓ • c) **tercera**
  - d) cuarta
- 

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

Usuaria Profesores

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

---  
Una máquina superescalar es aquella que:

Usuaria Profesores

- 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.
- 

---  
¿Cuál de las siguientes expresiones toma el valor 0x01 si x es múltiplo de 32 y 0x0 en caso contrario? Asumir que x es unsigned int.

- a)  $(x \mid 0x3f)$
  - b)  $!(x \& 0x3f)$
  - c)  **$!(x \& 0x1f)$**
  - d)  $(x \& 0x1f)$
- 

---  
Las unidades de control microprogramadas utilizan habitualmente memorias RAM para

almacenar los microprogramas.

**FALSA**

---

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) 7  
        b) 8  
        c) 9  
        d) 1
- 

Un modo de vídeo de 1680 x 1050 píxeles y 4 canales de color por píxel (R, G, B, alpha) y 256 niveles de intensidad por canal (0 a 255), ocupa aproximadamente una memoria de:

Usaria Profesores

- a) 54 MB
  - b) **7 MB**
  - c) 1,7 MB
  - d) 1,7 GB
- 

Alguno de los siguientes NO es un motivo de que no se alcance la ganancia ideal en un cauce segmentado

Usaria Profesores

- 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 acople, etc...)
  - c) Los riesgos (hazards)
  - d) **La emisión múltiple (y posiblemente desordenada) de instrucciones**
- 

Según la clasificación m/n, las máquinas de acumulador son de tipo

- a) 1/2
- b) 0/0

- c) 1/1
- d) 2/2 ó 2/3

---

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.

**FALSA**

---

---

El marco de pila en x86-64 Linux...

- a) se crea para funciones en las que GCC no puede evitar que RBP baje más, como por ejemplo: que haya demasiadas variables locales (y no quepan en registros), o que haya que salvar algún registro salva-invocado
- b) sólo se crea para funciones que invocan anidadamente a otra función (procedimientos padre, no hojas)
- c) **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**
- d) no existe, porque RBP no es registro especial en x86-64

---

---

En una estructura de computador de bus único (bus del sistema):

- a) ninguna de las otras respuestas es cierta
- b) **sólo una unidad funcional puede tener el control del bus en cada momento**
- c) la UC concede el acceso al bus, por lo que éste funciona a la velocidad de la CPU
- d) es la estructura más usada en los PC actuales

---

---

Un bus se compone de:

- a) líneas de estado y líneas de control
- b) líneas de datos y líneas de dirección
- c) líneas de alimentación
- d) **líneas de control/estado, líneas de dirección y líneas de datos**

¿En qué generación, dentro de la historia de los computadores digitales, aparecen las memorias de semiconductores?

- a) Tercera generación.
  - b) Cuarta generación.
  - c) Segunda generación.
  - d) Quinta generación.
- 

Un procesador con E/S mapeada a memoria tiene un bus de direcciones de 10 líneas y uno de datos de 8. El mapa de memoria tiene 512 posiciones para código (ROM), 256 para datos (RAM) y 256 para E/S, en ese orden. Los rangos de direcciones para esas tres zonas serán:

- a) 000 a 9FF, A00 a CFF y D00 a FFF
  - b) 000 a 7FF, 800 a BFF y C00 a FFF
  - c) 000 a 1FF, 200 a 2FF y 300 a 3FF
  - d) 000 a 5FF, 600 a 7FF y 800 a 9FF
- 

El ancho de palabra de una memoria corresponde a:

- a) El número de posiciones que la componen.
  - b) La cantidad de bits que caben en una sola posición
  - c) El número que identifica únicamente cada posición de la memoria.
  - d) La longitud del registro de direcciones de la memoria.
- 

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

- a) Contaba con más de 17000 tubos de vacío.
  - b) La idea del programa almacenado surgió durante el desarrollo del ENIAC, pero no fue implementada en este computador.
  - c) 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.
  - d) Un inconveniente era que tenía que ser programado manualmente mediante conmutadores, y conectando y desconectando cables.
- 

¿Qué tipo de instrucciones se emplean más en una arquitectura de acumulador?

- a) de desplazamiento y rotación
- b) de transferencia de datos con memoria
- c) de transferencia de datos entre registros
- d) aritmético-lógicas

---

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 0xCAFEBAB0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al consultar la media palabra de la posición 0xCAFEBABE?

- a) -1

los contenidos son

CAFEBAB0: FF FF FF FF

CAFEBAB4: FE FF FF FF

CAFEBAB8: FD FF FF FF

CAFEBABC: FC FF FF FF

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

- b) -4

- c) no se puede saber, faltan datos

- d) ninguna de las anteriores

---

¿Qué modelo de programa se ejecuta en las arquitecturas de Von Neumann?

- a) cableado

- b) almacenado

- c) nanoprogramado

- d) microprogramado

---

El ancho de palabra de una memoria corresponde a:

- a) El número que identifica únicamente cada posición de la memoria.

- b) La longitud del registro de direcciones de la memoria.

- c) La longitud del registro de datos de la memoria.

- d) El número de posiciones que la componen.

---

Una instrucción máquina del tipo "Add M,R" podría formar parte del repertorio de

- a) una máquina con arquitectura R/R

sería ADD R1,R2

- b) una máquina pila

sería ADD

- c) una máquina de acumulador

sería ADD M

- d) una máquina con arquitectura M/M

podría ser también ADD M,M

---

¿Cuál de las siguientes afirmaciones es incorrecta?

- a) En lenguaje ensamblador cada instrucción se corresponde con una instrucción máquina.
  - b) El lenguaje de alto nivel es más portable que el lenguaje máquina.
  - c) Un programa de alto nivel interpretado es más lento que el mismo programa en lenguaje máquina.
  - d) **En lenguaje ensamblador las instrucciones se escriben en binario.**
- 

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) De direcciones
  - b) Puntero de pila
  - c) Contador de programa
  - d) **De instrucción**
- 

Una palabra está formada por 2 bytes (16 bits) en cualquier ordenador.

**FALSA**

---

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

- a) Direccionamiento relativo a registro base
  - b) **Direccionamiento a registro**
  - c) Direccionamiento indexado
  - d) Direccionamiento inmediato
- 

¿Cuál de los siguientes no es un tipo de bus?

- a) Sistema
  - b) **Secuencial**
    - opuestos a buses paralelos son los buses serie
    - opuestos a programas secuenciales son los programas paralelos
  - c) E/S
  - d) Paralelo
- 

¿De qué tipo son los procesadores Intel que usamos en los laboratorios?

- a) little-endian
  - b) big-endian
  - c) puede ajustarse mediante un bit de control en el registro CR0 que funcionen como little- o big-endian
  - d) el concepto de endian no es aplicable a estas máquinas, ya que un registro del procesador no cabe en una posición de memoria
- 

El instrumento GIADA de la sonda espacial ROSETTA (diseñado en Granada) está basado en un microprocesador 8086 y el siguiente mapa de memoria:

RAM volátil: 00000 - 0FFFF

RAM no volátil: 10000 - 1FFFF

ROM: F0000 - FFFFF

¿Cuál es el tamaño total de la memoria?

- a) 48KB
  - b) 3MB
  - c) 2MB
  - d) 192KB
- 

En las arquitecturas RISC hay...

- a) pocas instrucciones muy rápidas con muchos modos de direccionamiento.
  - b) muchos registros y pocos modos de direccionamiento.
  - c) pocos modos de direccionamiento y muchos formatos de instrucción.
  - d) pocos registros y muchos tipos de instrucciones.
- 

¿Cuál de los siguientes registros se utiliza para guardar la dirección de memoria donde se localiza la instrucción siguiente?

- a) Memory Data Register
  - b) Memory Address Register
  - c) Program Counter
  - d) Instruction Register
- 

PC y SP son dos registros de uso general (GPR).

**FALSA**

---

La primera generación de computadores se caracteriza por el uso de:

- a) Microprocesadores.
  - b) Transistores.
  - c) **Tubos de vacío.**
  - d) Fibra óptica.
- 

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:

- a)  $M[0x8600]=0x89$  y  $M[0x8601]=0x65$
  - b)  $M[0x8600]=0x69$  y  $M[0x8601]=0x85$
  - c)  $M[0x8600]=0x85$  y  $M[0x8601]=0x69$
  - d)  **$M[0x8600]=0x65$  y  $M[0x8601]=0x89$**
- 

¿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)

- a) **1110110101110100)2**
  - b) 1110110101101000)2
  - c) 1110110101110000)2
  - d) Todas están alineadas a double
- 

¿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: 80
  - b) **0:80 1:FF 2:FF 3:FF**
  - c) 0:00 1:01 2: 00 3:80
  - d) Ninguna de las anteriores
- 

La idea de desarrollar máquinas CISC surgió para:

- a) simplificar el diseño hardware de la UC.
  - b) **tener instrucciones cercanas al lenguaje de alto nivel.**
  - c) conseguir un conjunto de instrucciones cortas y sencillas de decodificar.
  - d) ninguna de las respuestas anteriores es cierta.
- 

¿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 las basadas en registros son capaces de lograr un mejor rendimiento cuando se asignan variables a registros**
- d) Porque la memoria es más cara que los registros
- 

¿De qué depende el tamaño del contador de programa?

- - a) el tamaño no importa
    - b) ninguna de las otras respuestas es cierta**
    - c) del ancho del bus de datos
    - d) de la longitud del código de operación
- 

Una dirección de memoria se refiere siempre a:

- - a) una palabra
    - b) 16 bits
    - c) un byte
    - d) ninguna de las anteriores**
- 

Para direccionar una memoria de bytes en la que quepan 2G palabras de 32 bits se necesitarán:

- - a) 33 bits como mínimo**
    - b) 21 bits como máximo
    - c) 31 bits como mínimo
    - d) 32 bits exactamente
- 

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

- - a) El número medio de ciclos de reloj por instrucción.
    - b) El resultado de la ejecución de un conjunto de programas de prueba.**
    - c) La arquitectura del procesador.
    - d) La frecuencia de reloj del procesador.
- 

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.**
- 

¿Cuál de las siguientes no es una característica de los procesadores RISC?

- a) Para acelerar el computador RISC se emplean técnicas de pipelining.
  - b) Un computador RISC no debe emplear microprogramación.
  - c) La decodificación de las instrucciones debe ser simple: un computador RISC debería emplear un único formato de instrucción
  - d) **Las funciones que realizan los computadores RISC deben ser lo más complejas y potentes que sea posible.**
- 

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

---

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

- a) avances en las tecnologías de fabricación.
  - b) **avances en tecnología y avances en la estructura y arquitectura del computador.**
  - c) avances en la estructura y arquitectura del computador.
  - d) avances en los sistemas operativos y aplicaciones.
- 

En el arbitraje de un bus...

Usuaria Profesores

- 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
- 

En una arquitectura RISC típica:

- a) no puede usarse segmentación.
  - b) la UC es más compleja que en una arquitectura CISC.
  - c) la programación resulta mucho más simple que en una arquitectura CISC.
  - d) **se usan muchas instrucciones de las disponibles en el conjunto de instrucciones.**
- 

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) 0,538
  - b) **1,857**
  - c) 1,538
  - d) 12
- 

¿En qué generación, dentro de la historia de los computadores digitales, aparece la memoria virtual?

- a) primera
  - b) segunda
  - c) **tercera**
  - d) cuarta
- 

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

Usuaria      Profesores

- a) Un inconveniente era que tenía que ser programado manualmente mediante conmutadores, y conectando y desconectando cables.
  - b) **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.**
  - c) La idea del programa almacenado surgió durante el desarrollo del ENIAC, pero no fue implementada en este computador.
  - d) Contaba con más de 17000 tubos de vacío.
- 

El programador de lenguaje ensamblador necesita conocer:

- - a) la microarquitectura del procesador.
    - b) la arquitectura del ordenador.**
    - c) el diseño RTL del procesador.
    - d) todas las anteriores son ciertas.
- 

En la captación de la instrucción:

- a) en MBR indicamos la dirección donde está la instrucción y en IR recogemos la instrucción
  - b) en MAR indicamos la dirección donde está la instrucción y en MAR recogemos la instrucción
  - - c) en MAR indicamos la dirección donde está la instrucción y en MBR recogemos la instrucción**
    - d) en MBR indicamos la dirección donde está la instrucción y en MAR recogemos la instrucción
- 

La exo-arquitectura de una máquina especifica el conjunto de instrucciones, los modos de direccionamiento, la longitud de palabra, el nº de palabras de memoria, etc. **Verdadero**

---

¿Qué tipo de instrucciones se emplean más en una arquitectura de acumulador?

- - a) de desplazamiento y rotación
    - b) de transferencia de datos entre registros
    - c) aritmético-lógicas
    - d) de transferencia de datos con memoria**
- 

El programador de lenguaje ensamblador necesita conocer:

- - a) la microarquitectura del procesador.
    - b) todas las respuestas son ciertas.
    - c) la arquitectura del ordenador.**
    - d) el diseño RTL del procesador.
- 

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

---

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

- - a) 0xCAFEbac1
    - b) 0xCAFEbac0**
    - c) 0xCAFEBABE
    - d) 0xCAFEBABF
- 

Si queremos almacenar la palabra de 16 bits 0x9660 en una memoria de bytes según "little-endian", quedará almacenada a partir de la posición 0x1000 como:

- - a) M[0x1000]=0x96 y M[0x1001]=0x60
    - b) M[0x1000]=0x60 y M[0x1001]=0x96**
    - c) M[0x1000]=0x69 y M[0x1001]=0x06
    - d) M[0x1000]=0x06 y M[0x1001]=0x69
- 

El programador de lenguaje ensamblador necesita conocer la microarquitectura del ordenador.

**Falso**

---

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

- - a) Direccionamiento indexado
    - b) Direccionamiento a registro**
    - c) Direccionamiento relativo a registro base
    - d) Direccionamiento inmediato
- 

¿Cuál es el valor mínimo (más negativo) que puede tomar un entero de 32bits en complemento a dos?

(^ simboliza potenciación)

Usuaria Profesores

- - a) -2^31**
    - b) -2^31 + 1
    - c) -2^32
    - d) -2^32 + 1
- 

¿Cuál de los siguientes no es un tipo de bus?

**Secuencial**

- - a)** opuestos a buses paralelos son los buses serie  
opuestos a programas secuenciales son los programas paralelos

- 
- b) Paralelo
  - c) Sistema
  - d) E/S
- 

¿Cuál de las siguientes no es una característica de los computadores RISC?

Usuaria Profesores

- a) Un computador RISC no debe emplear microprogramación.
  - La decodificación de las instrucciones debe ser simple: un
  - b) computador RISC debería emplear un único formato de
  - instrucción
  - c) Para acelerar el computador RISC se emplean técnicas de
  - pipelining.
  - **Las funciones que realizan los computadores RISC deben**
  - ser lo más complejas y potentes que sea posible.**
- 

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

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

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:

Usuario Profesores

- 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

---

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

Usuario Profesores

- a) Memoria de núcleos de ferrita
- b) Compilador
- c) **Memoria cache**
- d) Lenguaje ensamblador

---

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

Usuaria Profesores

- a) **De instrucción**
- b) Puntero de pila
- c) Contador de programa
- d) De direcciones

---

*Práctica Febrero 19/20*

---

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

---

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

---

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
- 

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 & 0x01010101010101L;
 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:

- a. con `array={0x80000000, 0x00400000, 0x00000200, 0x00000001}`
  - b. con `array={0x10000000, 0x00200000, 0x00000400, 0x00000008}`**
  - c. con ambos ejemplos
  - d. 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...

```

- a. el entero 0x400590
  - b. el entero 0x400600**
  - c. el entero almacenado a partir de 0xc(%rsp)
  - d. 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:

- a. Exactamente esa solución viene en el guión, no hace falta aprenderla fuera de clase
  - b. Nosotros conectamos los led en serie con una resistencia como dice el guión**
  - c. Hay que revisar ese equipo en concreto, debe tener algún defecto, ese led no debe parpadear
  - d. 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++) media"
 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.

- a. 1/2 (un fallo por cada 2 accesos)
  - b. 1/4 (un fallo por cada 4 accesos)
  - c. 1/8 (un fallo por cada 8 accesos)
  - d. 1/16 (un fallo por cada 16 accesos)**
- 

9. En el programa line.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 line.cc ¿Qué forma tiene la gráfica que se puede obtener?

- a. Forma de U (o V), con un tramo descendente y otro ascendente
  - b. Forma de /, una gráfica siempre creciente
  - 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. ¿Cuál de las siguientes instrucciones es incorrecta?

- a. shr \$1,%rdx
  - b. shr %rdx
  - c. 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. 0xfffff ffff
  - b. 0xfc00 0000
  - c. 0xf7ff ffff**
  - d. 0x0000 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 "
 : // 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. popcorn1 (bucle interno for) irá más rápido con esta inicialización que con números aleatorios
- d. popcorn2 (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 1<sup>a</sup> 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
- 

¿En qué generación, dentro de la historia de los computadores digitales, aparecen las memorias de semiconductores?

Usuario Profesores

- a) Segunda generación.
  - b) Quinta generación.
  - c) Tercera generación.
  - **d) Cuarta generación.**
- 

Es posible diseñar un microprocesador que ejecute más de 100 MIPS, trabajando a una frecuencia de reloj de 100 MHz.

**Falso**

---

Se dice que las máquinas con arquitectura Von Neumann siguen un modelo de programa...

Usuario Profesores

- a) Cableado.
  - b) Externo.
  - **c) Almacenado.**
  - d) Microprogramado.
- 

Una máquina superescalar es aquella que:

Usuario Profesores

- 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.
- 

¿En qué registro está contenido el último dato (o instrucción) leído de memoria, o el dato que se va a escribir en memoria?

Usuario Profesores

- a) Acumulador.
  - b) PC.
  - c) MAR.
  - d) **MBR.**
- 

En una arquitectura de acumulador, la instrucción LOAD X:

Usuario Profesores

- a) transfiere el contenido del acumulador a la posición de memoria X  
sería STORE X
  - b) transfiere el contenido del registro X a la memoria  
no existe registro X en una máquina acumulador pura  
el argumento de LOAD es una posición de memoria X
  - c) el 6502/6510 sí tenía, además de acumulador A, índices X e Y, y correspondientemente tenía instrucciones LDA, LDX, LDY  
[https://es.wikipedia.org/wiki/MOS\\_6502](https://es.wikipedia.org/wiki/MOS_6502)  
<http://www.6502.org/tutorials/6502opcodes.html>
  - d) suma M(X) al acumulador  
sería ADD X  
**transfiere el contenido de la posición de memoria X al acumulador**
- 

En un procesador de la familia 80x86 una variable de 32 bits, entera con signo, almacenada a partir de la dirección n contiene: 0xFF en la dirección n, 0xFF en la dirección n+1, 0xFF en la dirección n+2 y 0xF0 en la dirección n+3. ¿Cuánto vale dicha variable?

Usuario Profesores

- a) 16  
**-251658241**
  - b) **Como es little-endian, se trata de un número negativo de gran magnitud, y éste es el único con ese aspecto**
  - c) -16
  - d) 4294967280
- 

Para crear espacio en la pila para variables locales sin inicializar suele realizarse la siguiente operación:

- Usuario Profesores
- - a) Sumar una cantidad positiva a ESP
    - b) Restar una cantidad positiva a EBP
    - c) **Restar una cantidad positiva a ESP**
    - d) Sumar una cantidad positiva a EBP
- 

En el RISC-I, una ventana de registros contiene, entre otros registros,...

- Usuario Profesores
- - a) **registros para recibir parámetros del procedimiento**
    - b) **llamador**
    - c) registros para almacenar matrices de enteros
    - d) registros para enviar parámetros a otros procesos
    - d) todas las respuestas son falsas
- 

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?

- Usuario Profesores
- Se espera una reducción de prestaciones porque además de
    - a) 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
    - Se espera un aumento de prestaciones debido a que las cuatro etapas solapan su funcionamiento, con una aceleración ideal de 4x**
- 

La instrucción "desplazar a la izquierda aritméticamente" suele ser idéntica a la "desplazar a la izquierda lógicamente".

**Verdadera**

---

¿Cuál es el complemento a 2 del número binario 1110 1101 1000?

1. 0001 0010 0110
2. 0001 0010 0101

3. 0001 0010 0111

**4. 0001 0010 1000**

---

¿Qué valor contendrá el registro rdx tras ejecutar las dos instrucciones siguientes?

movq \$-1, %rdx

movl \$1, %edx

Usuario Profesores

- a) 0xFFFF FFFF FFFF 0001
- b) 0xFFFF FFFF 0000 0001
- c) 0xFFFF 0000 0000 0001
- d) **0x0000 0000 0000 0001**

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?

Usuario Profesores

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

Escoger de entre las 4 operaciones la de mayor valor que pueda calcularse con enteros de 4B con signo sin problemas

1. 100.000.000 + 100.000.000

2. 300.000.000 + 300.000.000

**3. 1.000.000.000 + 1.000.000.000**

4. 3.000.000.000 + 3.000.000.000

---

Si rcx vale -1, tras ejecutar las instrucciones

rol \$1, %cl

rcr \$2, %rcx

el nuevo valor de RCX y del flag CF es:

1. Hay algún fallo de sintaxis o gramática en esas instrucciones

- 
- 2. RCX $\neq$ -1, CF mantiene su valor
  - 3. RCX=-1, CF=1**
  - 4. Ninguna
- 

El registro RAX contiene el número binario 111111101111000010. ¿Cuál será su contenido tras ejecutar la instrucción `sal $1,%ax`?

- 1. 0x3fde1
  - 2. 0x7fde1**
  - 3. 0xffde1
  - 4. 0xfffffffffffffde1
- 

Para comprobar si el contenido del registro RDX es 0 (y posiblemente saltar a continuación usando la instrucción `je`), el compilador gcc genera:

- 1. `cmpq %rdx, %rdx`
  - 2. `testq %rdx`
  - 3. `testq %rdx, %rdx`**
  - 4. `cmpq %rdx`
- 

Los buses de expansión son relativamente lentos y generalmente requieren terminadores.

**Verdadero**

---

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

**Verdadero**

---

Cuando se utiliza "daisy-chain", será necesario leer los biestables de concesión de interrupción asociados a los dispositivos, a no ser que se usen vectores de interrupción.

**Verdadero**

---

Si varios dispositivos comparten una única línea de solicitud de interrupción:

- a) se podría usar "polling" para identificar el origen de una interrupción
- b) se podría utilizar "daisy-chain" para establecer la prioridad de los dispositivos

- - c) se podría utilizar interrupciones vectorizadas
    - d) **todas son ciertas**
- 

Ventajas de la E/S independiente (señalar la opción incorrecta)

- - a) **diseño del procesador más sencillo (E/S mapeada añade complejidad al diseño)**
    - b) mayor aprovechamiento del espacio de memoria (E/S mapeada resta espacio a la memoria)
    - c) protección de E/S más fácil (E/S mapeada añade dificultad a la protección de E/S)
    - d) decodificación de memoria más elegante, limpia, sencilla (E/S mapeada añade complejidad a la decodificación)
- 

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
- 

Tipos de interrupción que suelen contemplar las CPUs comerciales actuales (señalar la opción incorrecta)

- - a) software: generadas al ejecutar la instrucción INT (o equivalente)
    - b) externas (IRQs hardware): generadas por un dispositivo externo a la CPU, activan la línea INTR# (o equivalente)
    - c) internas (excepciones o traps): generadas internamente por la CPU para indicar una condición que requiere atención (división por cero, codop inválido, etc)
    - d) **firmware (faults): generadas por el microcódigo de la CPU (segmentation fault, page fault, etc)**
- 

El controlador de interrupciones programable 8259 permite enmascarar selectivamente algunas líneas de interrupción.

**Falso**

---

Para realizar la petición y concesión del bus no son necesarias dos patillas de arbitraje como BUS\_REQUEST y BUS\_GRANT, ya que a veces se usa una única patilla.

**Verdadero**

---

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

- a) Instruction Set Architecture
  - b) Industry Standard Architecture
  - c) Intel Standard Architecture
  - d) Information Security Architecture
- 

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 µPC
  - c) el secuenciamiento de las instrucciones máquina
  - d) todas las respuestas son ciertas
- 

En la ejecución de una instrucción...

- a) la UC activa las señales de control que envía por el bus de direcciones

b) el registro de instrucción se va incrementando para apuntar a la siguiente instrucción

**c) la ALU realiza las operaciones aritméticas y lógicas**

d) siempre se altera el registro de estado

---

¿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) Los operandos siempre están almacenados en memoria.**

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

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

---

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

a) en el byte 0x1000 se guarda 0x65 y en el 0x1001 0x89

b) en el byte 0x1000 se guarda 0xA6 y en el 0x1001 0x91

c) en el byte 0x1000 se guarda 0x91 y en el 0x1001 0xA6

**d) en el byte 0x1000 se guarda 0x89 y en el 0x1001 0x65**

---

¿Cuál de las siguientes afirmaciones es incorrecta?

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

¿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) 20**

b) 4

c) 4, 7, 48

d) 4, 7, 8, 10

---

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
- 

¿Cuál de las siguientes afirmaciones es falsa?

- a) el bus de control transporta señales de estado
  - b) el bus de datos es bidireccional
  - c) el bus de direcciones es unidireccional
  - d) la anchura del bus de datos es de 16 bits**
- 

¿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) 4, 11, 1

**b) 1**

c) 11, 1

d) 4, 7, 10

---

¿Cuál de las siguientes afirmaciones es incorrecta?

a) Todas las instrucciones deben tener código de operación

b) El formato de una instrucción nos indica el significado de cada bit de la instrucción

**c) Todas las instrucciones deben tener operando fuente y operando destino**

d) No siempre es necesario indicar la dirección de la siguiente instrucción

---

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

a) pueden estar en cualquier posición de la pila.

b) son dos registros del procesador.

c) se introducen en la pila tras realizar la operación.

**d) son la cima de la pila y el elemento siguiente de la cima de la pila.**

---

El direccionamiento relativo a registro base utiliza...

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

¿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

- a) 10
  - b) 1 y 2
  - c) 1, 2, 3 y 4
  - d) 1**
- 

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

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

En una jerarquía de memoria con estrategia "Write-Through" en todos los niveles, el tiempo medio de acceso en escritura no es mejor que si sólo existiera el último nivel.

**Verdadero**

---

¿En qué tipo de refresco de memoria DRAM CAS# permanece a 0 después del ciclo de lectura o escritura precedente?

- a) RAS# antes de CAS#

- - b) Sólo RAS#
    - c) **Refresco transparente**
    - d) Ninguna de las anteriores respuestas es correcta
- 

Una placa madre de un 486 con un único SIMM de 30 contactos con 8 chips de 1M x 1...

- - a) tiene 8 M palabras de memoria principal
    - b) **no funciona correctamente**
    - c) tiene 1 M palabras de memoria principal
    - d) tiene 256 K palabras de memoria principal
- 

La tasa de aciertos  $A_i$  del nivel  $i$  de una jerarquía de memoria no depende de:

(mediante  $_$  se representa subíndice)

- - a) La estrategia de administración de memoria.
    - b) La capacidad (tamaño)  $s_i$  del nivel  $i$ .
    - c) **El ancho de banda  $b_i$  del nivel  $i$ .**
    - d) La unidad de la transferencia de información  $x_i$  entre el nivel  $i$  y el  $i+1$ .
- 

La política de postescritura ("Write Back") se utiliza en caché, nunca en paginación, debido a su lentitud.

**Falso**

---

¿Cuál de las siguientes afirmaciones acerca de las memorias RAM dinámicas es cierta?

- - a) Los datos permanecen en cada celda indefinidamente
    - b) Las operaciones de lectura no son destructivas
    - c) Las celdas de almacenamiento son complejas
    - d) **Las operaciones de escritura sirven como operaciones de refresco**
- 

Supongamos una memoria asociativa con un registro indicador de  $x$  bits y otro de máscara de  $y$  bits. ¿Cuántas palabras puede almacenar dicha memoria?

- - a)  $x$

- 
- b) y
  - c)  $x + y$
  - d) Ninguna de las anteriores respuesta es cierta
- 

La política de sustitución o reemplazo de bloques dice cuándo hay que llevar un bloque del nivel i al i+1 en una jerarquía de memoria.

**Falso**

---

Si se usa la estrategia de escritura Write-Through en dos niveles de memoria i e i+1, cuando se escribe una palabra en el nivel i se retrasa su actualización en el nivel i+1 hasta que la palabra tenga que ser modificada o reemplazada en el nivel i.

**Falso**

---

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

- a) 11
  - b) 10
  - c) **6 ( $4096 = 2^{12}$ , multiplexados 6+6)**
  - d) 12
- 

Una caché de 32 palabras y correspondencia por sectores utiliza bloques de una palabra. Si las direcciones de memoria principal son de 24 bits, el campo etiqueta debe tener 19 bits.

**Verdadero**

---

El refresco sólo RAS consiste en activar la señal RAS a la vez que se proporciona la dirección de

fila a refrescar, y a continuación activar la señal CAS sin proporcionar dirección de columna.

**Falso**

---

¿En qué generación, dentro de la historia de los computadores digitales, aparece la microprogramación?

- a) primera
  - b) segunda
  - c) **tercera**
  - d) cuarta
- 

La 5<sup>a</sup> generación de computadores comenzó aproximadamente en 1971 con el primer microprocesador.

**Falso**

---

En una arquitectura Von Neumann la CPU no puede determinar si una posición de memoria contiene una palabra de instrucción o una palabra de dato.

**VERDADERO**

---

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**

---

La VRAM es más cara que la DRAM normal.

**Verdadero**

---

Hay sistemas de caché que realizan prebúsqueda o preextracción, y no búsqueda o extracción por demanda.

**Falso**

---

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
- 

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) geométrica  
b) aritmética  
c) ponderada  
d) armónica
- 

El direccionamiento directo a memoria utiliza

- a) un registro y un desplazamiento contenidos en la propia instrucción
  - b) dos desplazamientos contenidos en la propia instrucción
  - c) un registro
  - ✓ • d) **un desplazamiento**
- 

¿De qué depende el tamaño del contador de programa?

Usaria Profesores

- ✓ • a) **Del número de direcciones de memoria.**  
Del número de instrucciones diferentes y de los tipos de dirección posibles.
- b) De la longitud del código de operación de las instrucciones.  
Del ancho del bus de datos.
- d)

En qué generación, dentro de la historia de los computadores digitales, aparecieron la microporgramación, la segmentación de cauce, la memoria cache, los S.O. multiusuario y

la memoria virtual?

Usaria Profesores

- a) 2<sup>a</sup> generación (1955-65)
- ✓     •     **b)** **3<sup>a</sup> generación (1965-75)**  
               **en la 3<sup>a</sup> generación se inventó casi todo**
- c) 4<sup>a</sup> generación (1975-85)  
               esas innovaciones se repartieron a lo largo de varias
- d) generaciones, no sólo una

¿Cuál de los siguientes grupos de señales son entradas a la unidad de control?

Usuario Profesores

- 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
- 

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?

Usaria Profesores

- **a) 0xFACEB00C**
  - b) 0xABADF00D
  - c) 0xCAFEBABE
  - d) 0xDEADBEEF
- 

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

Usaria Profesores

- a) en MAR está el resultado y en MBR 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  
**en MBR está el resultado y en MAR la dirección donde se**  
• d) **almacenará**
- 

Una computadora puede funcionar prescindiendo de:

Usaria Profesores

- a) del IR  
b) de los biestables de condición  
• c) **de un acumulador**  
d) del PC
- 

El objetivo de un diseño CISC es...

Usaria Profesores

- a) disminuir el tamaño medio de instrucción.  
• b) **disminuir el número de instrucciones a ejecutar por un programa.**  
c) disminuir el número medio de ciclos por instrucción.  
d) disminuir la frecuencia de reloj.
- 

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?

Usaria Profesores

- ✓ • a)  $2^{10}$  palabras  
b)  **$2^{12}$  palabras**  
c)  $2^{13}$  palabras  
d) Ninguna de las anteriores
- 

En una estructura de computador de bus único (bus del sistema):

Usaria Profesores

- ✓ • a) **sólo una unidad funcional puede tener el control del bus en cada momento**  
b) es la estructura más usada en los PC actuales  
c) ninguna de las otras respuestas es cierta  
d) la UC concede el acceso al bus, por lo que éste funciona a la

velocidad de la CPU

---

¿En qué generación, dentro de la historia de los computadores digitales, se alcanzaron tiempos de conmutación del orden de nanosegundos?

Usaria Profesores

- a) primera
  - b) segunda
  - ✓      • c) **tercera**
  - d) cuarta
- 

La ecuación básica de rendimiento calcula

Usaria Profesores

- a) el promedio de las ganancias obtenidas con una serie de programas de punto entero
  - b) cómo de mejor es un procesador frente a otro, conociendo las prestaciones de las respectivas UC, ALU, E/S y M
  - c) la media geométrica de los cocientes entre los tiempos de ejecución de una serie de programas predeterminados
  - ✓      • d) **cuánto tiempo tarda en ejecutarse un programa concreto**
  - ✓      • d) **conociendo su número de instrucciones y el número de etapas (promedio) y la frecuencia del procesador**
- 

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?

Usaria Profesores

**32**

el dispositivo tiene 32 puertos, usa 5 bits de direccionamiento (probablemente los 5 LSB), pero en lugar

- a) 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
  - b) 10
  - c) 16
  - d) 1024
- 

El espacio direccionable de memoria de un computador depende del diseño del:

Usaria Profesores

- - a) **Bus de direcciones**
    - b) Bus de datos
    - c) a) y b) son correctas
    - d) Ninguna de las anteriores es correcta
- 

El computador EDVAC, propuesto por John Von Neumann, presentaba dos importantes diferencias respecto al ENIAC:

Usuario Profesores

- - a) Se programaba enchufando centenares de clavijas y empleaba aritmética octal.
    - b) Era electromecánico y de propósito específico.
    - c) **Empleaba aritmética binaria y permitía trabajar con un programa almacenado.**
    - d) Utilizaba transistores y memoria de semiconductor.
- 

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?

Usuario Profesores

- - a) Más de 8 veces.
    - b) **Depende de la tasa de aciertos.**
    - c) La caché de postescritura no puede ser más eficiente que la de escritura inmediata.
    - d) La caché de postescritura siempre será más eficiente que la de escritura inmediata.
- 

Una cache de 256 B asociativa por conjuntos de 4-vías con líneas de 16 B tendrá

Usuario Profesores

- - a) ningún conjunto
    - b) 16 conjuntos  
**4 conjuntos**
    - c)  $256 \text{ B} = 2^8 \text{ B} = 2^4 \text{ lineas} \cdot 2^4 \text{ B/lin}$   
 $2^4 \text{ lin} = 2^2 \text{ conj.} \cdot 2^2 \text{ vías}$
    - d) 64 conjuntos
- 

En la cache L1 de instrucciones, la tasa de fallos:

Usuario Profesores

- a) Siempre tiende a crecer si el número de vías crece
  - b) Siempre tiende a disminuir si el número de vías disminuye
  - c) Siempre tiende a crecer si el tamaño total de L1 crece
  - d) **Siempre tiende a disminuir si el tamaño total de L1 crece**
- 

Las celdas de memoria estática...

Usuario Profesores

- a) son más pequeñas y lentas que las celdas de memoria dinámica
  - b) almacenan la información en un condensador
  - c) **mantienen la información almacenada por tiempo indefinido mientras se mantenga la alimentación**
  - d) Ninguna de las respuestas anteriores es cierta
- 

¿En qué se diferencian las estrategias de mantenimiento de coherencia en memoria "escritura directa" y "post-escritura"?

Usuario Profesores

- a) **En cuándo tiene lugar la actualización**
  - b) En cómo tiene lugar la actualización
  - c) Tanto en a) como en b)
  - d) Ni en a) ni en b)
- 

En la arquitectura Von Neumann...

Usuaria Profesores

- a) los bloques principales son la unidad de control, la ALU y la CPU.
  - b) **el programa se encuentra residente en memoria.**
  - c) los registros se encuentran en la memoria principal.
  - d) Todas son ciertas.
- 

En el direccionamiento inmediato, tras captarse completamente la instrucción:

Usuaria Profesores

- a) el código de operación contiene el operando.
- ✓ • b) **se accede al operando, que es una constante contenida en la propia instrucción.**
- c) se accede al operando, que se encuentra almacenado en uno

- de los registros programables.
- d) se accede al operando, que está contenido en una posición de memoria principal.
- 

La memoria cache en un sistema computador es:

Usuario Profesores

- 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
- 

Una instrucción máquina del tipo "Add M,R" podría formar parte del repertorio de

Usuaria Profesores

**una máquina con arquitectura**

- a) **M/M**  
podría ser también ADD M,M
  - b) una máquina con arquitectura R/R  
sería ADD R1,R2
  - c) una máquina de acumulador  
sería ADD M
  - d) una máquina pila
- 

Respecto a los dispositivos activos y pasivos en un bus podemos decir que:

Usuaria Profesores

- a) Los dispositivos pasivos sólo pueden convertirse en esclavos
  - b) Sólo los dispositivos activos pueden convertirse en maestros
  - ✓ • c) **Las respuestas a y b son ciertas**  
Las respuestas a y b son falsas
  - d)
- 

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:

Usuaria Profesores

- a) **arquitectura del computador**
  - b) conjunto de componentes físicos del computador
  - c) repertorio de instrucciones máquina
  - d) organización del computador
-

El registro MBR...

Usaria Profesores

- ✓ •
    - a) especifica la dirección en memoria de la palabra que va a ser escrita o leída
    - b) contiene el código de operación de la instrucción que se está ejecutando
    - c) **contiene el valor que va a ser almacenado en la memoria, o bien se usa para recibir un valor procedente de la memoria**
    - d) contiene la dirección de la próxima instrucción que va a ser captada de memoria
- 

Una memoria que está estructurada en palabras de 8 bits tiene una capacidad de 64 Kbits.

¿Cuántas líneas de dirección tiene dicha memoria?

Usaria Profesores

- - a) 8
    - b) **13**
    - c) 12
    - d) 24
- 

¿En qué generación, dentro de la historia de los computadores digitales, aparecen los sistemas operativos multiusuario?

Usaria Profesores

- ✓ •
    - a) primera
    - b) segunda
    - c) **tercera**
    - d) cuarta
- 

Si usamos una estructura de bus con DMA:

Usaria Profesores

- - a) la velocidad de este controlador establece la velocidad del bus del sistema.
    - la CPU puede dejar las transferencias entre MP y periféricos en manos de este controlador (DMA) y seguir ejecutando otras instrucciones.**
    - c) al bus del sistema sólo se conecta la CPU y la MP, ya que el DMA se conecta directamente a MP para realizar las transferencias de datos. podemos prescindir de controladores de E/S ya que el controlador de
    - d) DMA se ocupa de controlar las transferencias hacia/desde los periféricos.
- 

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

Usaria Profesores

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

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

Usaria Profesores

- a) La última versión es SPEC CPU2006 V1.2 de 2011
- b) Se cronometran unos 12 tests de enteros (CINT2006) y unos 17 tests de punto flotante (CFP2006)
- c) 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
- d) **El resultado final es la media aritmética de las (12 ó 17) velocidades, bien sea de enteros ó de punto flotante (SPECint2006 ó SPECfp2006) es la media geométrica**

La segmentación de cauce...

Usuario Profesores

- 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**
- 

¿Cuáles de las siguientes señales son entradas a la unidad de control?

Usuario Profesores

- 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
- 

El bus del sistema es

Usaria Profesores

- a) en un sistema con bus único, todo el bus salvo la parte

- 
- - b) relacionada con E/S (SATA, GPU, USB, Ethernet, etc)
    - el que conecta CPU-M, ya sea un sistema con bus único o con múltiples buses**
    - c) en un sistema con buses separados, el que conecta el sistema E/S con el resto
    - d) el que conecta las distintas partes del sistema: UC, ALU, E/S, M
- 

En una arquitectura RISC típica:

Usaria Profesores

- a) la programación resulta mucho más simple que en una arquitectura CISC.
  - b) no puede usarse segmentación.
  - c) la UC es más compleja que en una arquitectura CISC.  
**se usa un porcentaje elevado de las instrucciones del repertorio.**
  - d) **repertorio.**
- 

Un computador con 8 bits en el bus de direcciones puede direccionar como máximo:

Usaria Profesores

- a) 1024 palabras
  - b) 8192 palabras
  - c) **256 palabras**
  - d) 16384 palabras
- 

En un sistema con un único bus...

Usaria Profesores

- 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**
- 

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?

Usaria Profesores

- a) -1
  - b) -2
  - c) 1
  - d) 0
- 

¿Cuál de las siguientes afirmaciones acerca de la arquitectura Von Neumann es cierta?

Usuaria Profesores

- ✓ • a) Para ejecutar un programa en una máquina Von Neumann, es necesario volver a cablear o incluso rediseñar la máquina.  
**La separación entre almacenamiento y unidad de procesamiento es una de los ideas contempladas en la arquitectura Von Neumann.**
  - b) La arquitectura Von Neumann es un diseño que sitúa el programa en un almacenamiento distinto al de los datos. Existe un consenso general en considerar justo el término "arquitectura von Neumann", ya que las ideas de esta
  - c) d) arquitectura fueron completamente originales de John Von Neumann y no influenciadas por sus colaboradores o predecesores.
- 

¿Cuál de las siguientes afirmaciones es cierta?

Usuario Profesores

- a) la unidad de control necesita como entrada el registro contador de programa para saber cuál es la instrucción que debe ejecutar a continuación
  - b) el registro de direcciones de memoria es un registro de propósito general que puede contener tanto direcciones como datos
  - 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
  - ✓ • d) la arquitectura von Neumann de los computadores tradicionales consiste en tener almacenados los datos separados de las instrucciones en memorias distintas
- 

En una CPU de 32 bits con memoria de bytes, el problema es que...

Usuario Profesores

- X
- 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) No tiene sentido, un registro no cabría en memoria
  - d) Hay que respetar el ordenamiento de bytes y reglas de alineamiento con que se diseñó la CPU
- 

¿Cuál de las siguientes no es una característica de los procesadores RISC?

Usuario Profesores

- X
- a) Un computador RISC no debe emplear microprogramación. La decodificación de las instrucciones debe ser simple: un
  - b) computador RISC debería emplear un único formato de instrucción
  - c) Las funciones que realizan los computadores RISC deben ser lo más complejas y potentes que sea posible. Para acelerar el computador RISC se emplean técnicas de
  - d) pipelining.

---

¿Cuál de las siguientes listas está correctamente ordenada temporalmente?

Usuaria Profesores

- ✓
- a) 8088, 80386, Pentium, Pentium Pro, Pentium 4, Pentium II.
  - 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.
- 

Un RISC con ventanas de registros solapadas permite pasar parámetros a subrutinas rápidamente.

Usuaria Profesores

V

---

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

Profesores

V

---

Si el contenido del registro %rax es 0x10 antes de ejecutar la instrucción shl \$0xc,%rax, ¿cuánto es su contenido tras ejecutarla?

Usaria Profesores

- ✓ • a) **0x10000**  
sí, pasa de bit 4 a bit 16 activado  
0x1000  
b) no, sería shl \$8,  
0x4000  
c) no, sería shl \$10  
0x800  
d) no, sería shl \$7

---

¿De qué depende el tamaño del contador de programa?

Usaria Profesores

- ✓ • a) el tamaño no importa  
b) del ancho del bus de datos  
c) **ninguna de las otras respuestas es cierta**  
de la longitud del código de operación  
d)

---

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

Profesores

- a) El bucle while  
b) El bucle for  
c) **El bucle do-while**  
La selección switch-case  
d)

---

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

Profesores

- a) SF = 1
- b) CF = 1
- SF ≠ OF**
- c) XOR(SF,OF) es cierto cuando sólo uno de ellos es cierto, de forma que deben ser distintos
- d) SF = CF

---

Una arquitectura de registros de propósito general puede tener instrucciones máquina de la ALU con un único operando explícito.

Usaria Profesores

**V**      **V**

---

¿Cómo se devuelve en ensamblador x86-64 Linux gcc el valor de retorno de una función long int al terminar ésta?

Usaria Profesores

- 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) **Ninguna de esas formas es la correcta.**

---

El lenguaje máquina es...

Profesores

- a) **difícil de codificar manualmente.**
- b) portable entre arquitecturas.
- c) fácilmente legible por el programador.
- d) una alternativa razonable al uso del lenguaje ensamblador.

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:

Usuaria Profesores

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

¿Cuál de los siguientes microprocesadores no es de 64 bits?

Profesores

- a) Itanium  
b) **Pentium III**  
c) Core 2  
d) Core i7
- 

Un overflow nunca puede ocurrir cuando:

Usuaria Profesores

- ✓ • a) se resta un número positivo de un número negativo  
0x80000000 - 0x00000001  
b) se suman dos números negativos  
0x80000000 + 0xffffffff  
c) se suma un número positivo a un número negativo  
resultado queda entre ambos => se puede representar  
d) se suman dos números positivos  
0x7fffffff + 0x1

---

El direccionamiento inmediato no es más lento que el direccionamiento directo.

Profesores

V

---

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

Usuario Profesores

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

---

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

**Verdadero**

---

¿Qué tipo de direccionamiento se usa para el registro destino en la instrucción x86-64 add array(%rbx,4), %edx?

Usuario Profesores

- a) Direccionamiento inmediato
- b) **Direccionamiento a registro**
- c) Direccionamiento indexado
- d) Direccionamiento relativo a registro base

---

Una doble palabra está formada por 32 bits en cualquier ordenador.

**Falso**

---

Un procedimiento llamado por una instrucción call debe guardar y restaurar los registros siguientes siempre que los altere:

Usuario Profesores

- a) %rax, %rbx, %rcx, %rdx  
sólo RBX s-invocado
- b) %rax, %rdx, %rcx  
s-invocante
- c) %rsi, %rdi  
s-invocante
- d) **%rbx, %rbp**  
**ambos s-invocado**

La instrucción CMP realiza la operación XOR sin almacenar el resultado.

**Falso**

---

¿Qué modo de direccionamiento usa el operando fuente en la instrucción mov (%rcx), %al?

Usuario Profesores

- a) Registro
  - b) **Indirecto a memoria a través de registro**
  - c) Inmediato
  - d) Directo a memoria
- 

Si desplazamos rbx mediante salq \$4, %rbx:

Usuario Profesores

- a) Lo multiplicamos por 4
  - b) Lo dividimos por 4
  - c) **Lo multiplicamos por 16**
  - d) Lo dividimos por 16
- 

La primera letra (l) de la instrucción lea:

Usuario Profesores

- a) indica que la instrucción trabaja con un operando destino de 32 bits (long word)
- b) **forma parte del nemotécnico de la instrucción**
- c) indica que la instrucción afecta a los 16 bits menos significativos del operando destino (low word)
- d) indica que la instrucción usa ordenación de bytes little-endian

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

Usuario Profesores

- a) -3
- b) -2

- c) -4
  - d) -1
- 

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)

Usuaria Profesores

- a) push arg1, push arg2, call foo, push %ebp
  - b) push arg2, push arg1, push %ebp, call foo
  - c) push arg1, push arg2, push %ebp, call foo
  - d) **push arg2, push arg1, call foo, push %ebp**
- 

Un sistema de memoria utiliza una caché con bloques de 8 palabras y memorias DRAM trabajando en modo página. ¿Cuántos ciclos se necesitan para copiar un bloque de memoria en la caché?

Usuaria Profesores

- a) 4 ciclos completos RAS-CAS y 4 ciclos CAS
  - b) 7 ciclos completos RAS-CAS y 1 ciclo CAS
  - c) 8 ciclos completos RAS-CAS
  - d) **1 ciclo completo RAS-CAS y 7 ciclos CAS**
- 

Respecto al sistema de Entrada / Salida, ¿cuál de las siguientes afirmaciones es incorrecta?

Usuaria Profesores

- a) Un protocolo sirve para “ponerse de acuerdo” en cosas como velocidad, paridad, nº de bits, etc.
  - b) Un controlador se encarga de la comunicación con la CPU.
  - c) **La mayoría de los periféricos trabajan a velocidad muy superior a la CPU; por eso es necesario sincronizar.**
  - d) La CPU se comunica con el periférico por medio del controlador y de software de E/S.
-

La caché es gestionada por:

Usaria Profesores

- a) el programador
  - b) el sistema operativo
  - algoritmos hardware**
  - 
  - c) **o unidades de "manejo" (gestión)**
  - hardware**
  - d) ninguna es cierta
- 

La "postescritura (write-back) marcada"

Usaria Profesores

- a) requiere más bits de modificación ("bits sucios") cuando aumenta el número de vías
  - b) requiere menos hardware que la "postescritura siempre"
  - c) provoca una menor tasa de faltas que la "postescritura siempre"
  - 
  - d) **es más eficiente que la "postescritura siempre"**
- 

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

Usaria Profesores

- a) Soporte, número de líneas, tipo de conector, dimensión de las tarjetas,...
  - 
  - b) **Alimentación, impedancia, nivel de señal,...**
  - c) Tipos de transferencias, temporización de las señales, arbitraje,...
  - d) Ninguna de las anteriores respuestas es cierta
- 

¿Cuál de las siguientes definiciones de modos de direccionamiento es \*incorrecta\*?

Usaria Profesores

- a) Registro: el dato se encuentra en un registro de propósito general.  
**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.**
- 
- b) **no, eso sería relativo a base (o indexado)**
- c) Inmediato: el dato está codificado dentro de la propia instrucción,

- en uno de los campos en los que se divide el formato de instrucción.
- d) Indirecto: el dato está contenido en una posición de memoria que es apuntada por un registro de propósito general.
- 

¿Cuál de las siguientes afirmaciones acerca de una jerarquía de memoria es cierta?

Usuaria Profesores

- a) **Para aumentar la eficiencia se transfieren bloques completos**
  - b) Toda la información que la CPU 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) Todas las afirmaciones anteriores son falsas
- 

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

Usuaria Profesores

- 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.
- 

Para un mismo procesador puede ocurrir que existan dos instrucciones cuyos campos de código de operación tengan diferente longitud.

**Verdadero**

---

La instrucción JGE / JNL provoca un salto si...

Usuaria Profesores

- a) SF = 1  
sería js
- b) CF = 1  
sería jc/jb
- c) SF = 0  
sería jns

### OF = SF

- d) basta recordar que "Less" no era un flag solo (es OF^SF)  
recordar también que "Below" comprueba CF
- 

El cuerpo del siguiente código C:

```
unsigned copy(unsigned u) {return u;}
```

puede traducirse a ensamblador como:

Usuaria Profesores

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

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

Usuaria Profesores

- a) val[5] es de tipo int y vale 3
- b) val[4]==3, val[5] apunta fuera del array  
**&val[2] es de tipo int\* y vale lo mismo que (void\*)val+8**
- c) val+4 es de tipo int\* y se cumple que \*(val+4)==5  
Por aritmética de punteros, \*(val+4)==val[4]==3
- d) val+1 es de tipo int y vale 2  
val+1==&val[1] es int\*

En cdecl/x86, ¿cuál de los siguientes registros tiene que ser guardado por la función llamada si es alterado por ésta?

Usuario Profesores

- a) ecx
  - b) edx
  - c) eax
  - d) ebx
- 

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

Usuario Profesores

- a) addl \$1, (%eax,%edx,4)
- b) addl \$4, (%eax,%edx)
- c) **addl \$1, (%edx,%eax,4)**
- d) addl \$1, (%edx,%eax)

---

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 micropogramación?

Usuario Profesores

- a) **3744250**
  - b) 259206
  - c) 287935
  - d) ninguno de los resultados anteriores es exacto
- 

Para realizar la microoperación MAR <- PC, habrá que activar:

Usuario Profesores

- a) LdPC y EnMAR
  - b) LdPC y LdMAR
  - ✓     • c) **EnPC y LdMAR**
  - EnPC y EnMAR
  - d)
- 

Dado un camino de datos concreto, un posible formato de micropogramación se caracteriza como horizontal o vertical según tenga más o menos (señalar la respuesta falsa)

Usuario Profesores

- a) longitud relativa de microinstrucción
  - b) codificación
  - c) solapamiento
  - d) **microbifurcaciones**
- 

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

Usuario Profesores

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

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

Usuario Profesores

F

F

---

El control residual se utiliza para:

Usuaria Profesores

- 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
- 

Un diseño vertical de una unidad de control...

Usuaria Profesores

- a) siempre supone un ahorro considerable de bits respecto a uno horizontal
  - b) siempre es más rápido que uno horizontal
  - c) **en general es más lento que uno horizontal**
  - d) en general desperdicia bits respecto a uno horizontal
- 

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?

Usuaria Profesores

- a) 4
  - b) 9
  - c) 1
  - ✓ • d) 8
- 

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

Usuaria Profesores

- a) se puede usar un demultiplexor.
  - b) no se puede usar un multiplexor.
  - c) se puede realizar una conexión directa.
  - d) **se pueden usar dos buffers triestado.**
- 

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?

Usaria Profesores

- a) **66,67 MHz**
  - b) 500 MHz
  - c) 200 MHz
  - d) 40 MHz
- 

Respecto a las unidades de control nanoprogramadas:

Usaria Profesores

- a) La realización nanoprogramada de una unidad de control es más rápida que la microprogramada.
  - b) El diseño de las unidades de control nanoprogramadas debe ser vertical.  
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  $(n \cdot m + 2^m \cdot w) - n \cdot w$ .
  - c) **La anchura de la memoria de nanoprograma es la misma**
  - d) **que la de memoria de microprograma en un diseño de la misma unidad de control que no usara nanoprogramación.**
- 

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:

Usaria Profesores

- a) 2,5
  - b) **4**
  - c) 80
  - d) 5
- 

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

Usaria Profesores

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

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

Usaria Profesores

- a) 34 bits
  - b) 25 bits
  - c) 35 bits
  - d) **14 bits**
- 

Respecto a los términos microinstrucción y microcódigo:

Usaria Profesores

**Microcódigo es el contenido de la memoria de control, y una microinstrucción es una palabra de dicha memoria**

- ✓ • a) tr.42 define literalmente ucód. como conjunto de microprogs. que a su vez son el conjunto de uinstr. de una instr.
  - b) Son equivalentes, llamamos microcódigo o microinstrucción a una palabra de la memoria de control
  - c) Una microinstrucción está programada en microcódigo, que es un lenguaje para programar señales de control
  - d) 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
- 

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 micropogramación?

Usaria Profesores

- ✓ • a) **19440**
  - b) 42280
  - c) 9840
  - d) ninguno de los anteriores resultados es exacto.
- 

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

Usaria Profesores

- 
- - a) en un banco de registros.
    - b) **en una ROM.**
    - c) en una PLA.
    - d) en una RAM.
- 

¿Cómo actúa el indicador de signo?

Usaria Profesores

- a) Se pone a 1 cuando el resultado es positivo
  - b) Se pone a 0 cuando el resultado es negativo
  - c) Se pone a 1 cuando el resultado es distinto
  - ✓      • **Se pone a 1 cuando el resultado es negativo**
  - d)
- 

Las instrucciones de salto...

Usaria Profesores

- a) son uno de los tipos de instrucciones máquina con menor frecuencia dinámica de uso.
  - **b) complican el diseño eficiente de los procesadores segmentados.**
  - c) siempre utilizan direccionamiento absoluto.
  - d) Todas las afirmaciones anteriores son ciertas.
- 

¿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

Usaria Profesores

- a) load (R1,R2), R2
  - b) add R1, R2
  - c) **store R1+R2, (R2)**
  - d) move Y(R1,R2), Z(R2)
- 

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

Usaria Profesores

- a) Microinstrucciones cortas.
- b) Escasa capacidad para expresar paralelismo entre microoperaciones.
- ✓      • **c) Ninguna o escasa codificación.**
- T3 tr.49

- 
- d) Muchos campos solapados.
- 

¿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?

Usaria Profesores

- a) Un multiplexor
  - b) Un contador
  - c) Un demultiplexor
  - d) **Una memoria**
- 

La microprogramación vertical se caracteriza por tener:

Usaria Profesores

- ✓ • a) microinstrucciones largas
  - b) **muchas codificaciones**
  - c) escaso o ningún solapamiento entre campos  
capacidad para expresar un alto grado de paralelismo en las
  - d) microoperaciones a ejecutar
- 

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:

Usaria Profesores

- a)  $2^{10}$
  - b) 20
  - c) 10
  - d)  $2^{20}$
- 

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

Usaria Profesores

- a) 3928652
- b) 259206
- c) 287935
- ✓ • d) **ninguno de los resultados anteriores es exacto**

---

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

Usaria Profesores

- a)  $SP=SP-1 ; m[SP]=PC ; PC=X$   
     b)  $PC=X ; SP=SP-1 ; m[SP]=PC$   
     c)  $SP=SP-1 ; m[SP]=PC ; PC=PC+1$   
     d)  $SP=PC-1 ; m[SP]=PC ; PC=X$
- 

Alguna de las siguientes señales NO es salida de la unidad de control. ¿Cuál?

Usaria Profesores

- a) Códigos de selección en multiplexores, decodificadores, ALU, etc (00,01,10,11...)  
     b) Señales de carga, habilitación y/o desplazamiento de registros (Load, Enable, ShiftL, ShiftR)  
     c) Señales de lectura y escritura en memoria (RD, WR)
  - d) **Dirección de la siguiente microinstrucción (bits del campo DIR o Memoria B de Wilkes)**
- 

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

- a) **se pueden usar dos buffers triestado.**
  - b) se puede usar un demultiplexor.
  - c) se puede realizar una conexión directa.
  - d) no se puede usar un multiplexor.
- 

¿Cuál de las siguientes afirmaciones es verdadera?

Usaria Profesores

- a) El registro puntero de pila es un registro de propósito general que suele contener tanto direcciones como datos.  
     b) Las únicas instrucciones en las que algunas de sus fases de ejecución llevan un acceso a memoria son las instrucciones load y store.
- c) **La unidad de control necesita como entrada el registro de estado para poder controlar la ejecución de las instrucciones de salto condicional.**

### Tema 3 tr.20

El registro de instrucción es un registro de propósito específico

- d) que contiene la dirección de la siguiente instrucción a ejecutar.
-

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?

Usuaria Profesores

- a) No se produce ahorro
  - b) 2560 bits
  - c) 19440 bits
  - d) 5260 bits
- 

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?

Usuaria Profesores

- a)  $k + n \cdot 2^m$
  - b)  $k + n^m$
  - c)  $k + n \cdot (2^m - 1)$
  - d) Ninguno de los anteriores
- 

¿Cuál de las siguientes características sobre RISC es \*FALSA\*?

Usuaria Profesores

- 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.**  
La decodificación de las instrucciones debe ser simple: un
  - c) procesador RISC debería emplear pocos formatos de instrucción.
  - d) La unidad de control de un procesador RISC debería ser cableada, no micropogramada.
- 

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

Usuaria Profesores

- a) realizar operación ALU y guardar resultado en registro
- b) (leer o escribir) un registro (de / a) memoria
- c) transferir un registro a otro

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

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...

Usaria Profesores

**codificarlas con 4 bits, y sobraría un código que quedaría**

- a) sin uso  
tr.47 dibujo superior
  - 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  
solaparlas en un solo campo de 7 bits, ahorrando por tanto 6 bits
  - d) bits
- 

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

Usaria Profesores

- a) 17 bits
  - b) **5 bits**  
T3 tr.47
  - c) 16 bits
  - d) 4 bits
- 

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?

Usaria Profesores

- a) 3840 bits
  - b) **256 bits**
  - c) 4096 bits
  - d) No se produce ahorro
- 

En una unidad de control microprogramada se tiene un campo de 14 señales de control

de las cuales se activarían dos o menos en un ciclo de reloj, nunca tres o más en el mismo ciclo de reloj, y cuando se activan dos una es del grupo 1-7 y otra es del grupo 8-14, nunca las dos del mismo grupo. Sería entonces posible...

Usaria Profesores

- a) codificarlas con 4 bits, y sobraría un código que quedaría sin uso
  - codificarlas con 6 bits (2 códigos de 3 bits), y no sobraría**
  - b) ningún código sin uso  
**tr.47 dibujo inferior**
  - 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
- 

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

Usaria Profesores

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

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

Usaria Profesores

- a) 4 bits
  - b) 17 bits
  - c) 16 bits
  - d) **5 bits**
- 

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

Usaria Profesores

- a) dos buffers triestado
- b) **dos multiplexores de 2 a 1**
- c) tres conexiones directas al bus común
- d) tres demultiplexores

---

Si rax contiene x, ¿cuál de las siguientes instrucciones calcula  $x^9$ ?

Usaria Profesores

- 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
- 

A medida que nos acercamos a la CPU en una memoria organizada en forma jerárquica, ¿qué suele ocurrir con respecto al tamaño de las unidades de transferencia entre niveles?

Usaria Profesores

- a) Aumenta
  - b) **Disminuye**
  - c) Permanece constante
  - d) Todas las posibilidades anteriores suelen darse
- 

Se ha declarado en un programa C la variable int val[5]={1,5,2,1,3}; ¿Cuál de las siguientes afirmaciones es cierta?

Usaria Profesores

- 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
- 

Se ha declarado en un programa C la variable int val[5]={1,5,2,1,3}; ¿Cuál de las siguientes afirmaciones es FALSA?

Usaria Profesores

- a) **&val[2] es de tipo int \* y vale lo mismo que (void \*) val+8**
  - b) **val[1] ==1**
  - c) sizeof(val) == 20 // 5 elem del array x 4B cada uno
  - d) &val[3] == val + 3
- 

¿Cuál de las siguientes afirmaciones sobre memorias es correcta?

Usuario Profesores

- a) La memoria cache se construye con tecnología electrónica de tipo DRAM.

no, cache es SRAM

**Los chips de memoria DRAM se conectan entre sí en un**

- **b) circuito impreso constituyendo lo que se denomina DIMM.**

Las memorias SRAM no son volátiles; es decir, cuando no

- c) están alimentadas eléctricamente siguen guardando toda la información.

no, claro que no guardan info apagadas

La memoria principal se construye con tecnología electrónica

- d) de tipo SRAM.

no, MP es DRAM

---

¿A qué tipo de memoria caché corresponde la siguiente afirmación: "permite que cualquier dirección se pueda almacenar en cualquier marco de bloque de caché"?

- a) Con correspondencia directa
  - b) **Totalmente asociativa**
  - c) Asociativa por conjuntos
  - d) Ninguna de las otras respuestas es correcta
- 

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)?

Usaria Profesores

- a) 28000 MIPS
  - b) 16000 MIPS**
  - c) 4000 MIPS
  - d) 80000 MIPS
- 

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

Usaria Profesores

- a) Adaptar el formato de las señales

- - b) **Ejecutar el programa de transferencia de información entre el procesador y los dispositivos de E/S**
    - c) Recibir señales de control desde el procesador
    - d) Ajustar la temporización entre el procesador y los dispositivos de E/S
- 

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:

Usuaria Profesores

- a) base con desplazamiento
  - b) directo o absoluto
  - c) indirecto a registro través de memoria  
**ninguno de los anteriores**
  - d)
- 

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?

Usuaria Profesores

¿Qué modo de direccionamiento  
usa el operando fuente en la  
instrucción mov (%rcx), %al?

- a) 12,5%
  - b) 100%
  - c) 25%
  - 50%**
  - d) 16bits => 64Kpal  
 $16+16 \text{ Kpal} = 32\text{Kpal} = 50\% \text{ 64Kpal}$
- 

¿Cuál de los siguientes modos de direccionamiento es \*menos\* preferible para un procesador con segmentación de cauce?

Usuaria Profesores

- a) Indexado (o relativo a base, o base+índice)
- b) Registro

- c) Indirecto a través de memoria
  - d) Indirecto a través de registro
- 

¿Qué conjunto de componentes permite construir una memoria 256Mx32? (sin que sobren componentes)

Usaria Profesores

- a) 16 chips 64Mx4
  - b) **32 chips 64Mx4**
  - c) 16 chips 64Mx16
  - d) Ninguna de las anteriores
- 

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;
};
```

Usuario Profesores

- a) 24
  - b) 58
  - c) 60
  - d) **64**
- 

Si el registro r12b contiene la variable booleana cond, y rax la variable valor, la secuencia de instrucciones:

```
testb %r12b, %r12b
movq $13, %rax
cmovc $17, %rax
```

realiza la operación:

Usaria Profesores

- a) valor = cond ? 17 : 13;
- b) valor = 17;

- 
- c) valor = 13;
  - d) **valor = cond ? 13 : 17;**
- 

¿Cuál de los siguientes grupos de señales no se usa en un chip de memoria SRAM?

(  $\_x$  simboliza subíndice)

Usuaria Profesores

- a) Direcciones A $_{\{n-1\}}\text{-A}_0$ .
  - b) Datos D $_{\{n-1\}}\text{-D}_0$ .
  - c) **Selección de filas RAS# y de columnas CAS#.**  
**no, eso es para DRAM**
  - d) Selección de chip CS# y habilitación de escritura WE#.
- 

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

Usuaria Profesores

- a) 3
  - b) 0
  - c) 1
  - d) 2
- 

[T6.2]

¿En qué tipo de memorias coincide el tiempo de acceso y el tiempo de ciclo?

Usuaria Profesores

- a) **SRAM**
  - b) DRAM
  - c) Tanto en a) como en b)
  - d) Ninguna de las anteriores
- 

Un Pentium funcionando en modo protegido...

Usuario Profesores

- a) **siempre tiene activa la unidad de segmentación**
- b) siempre tiene activa la unidad de paginación
- c) siempre tiene activas la segmentación y la paginación
- d) puede tener desactivadas la segmentación o la paginación

---

¿Cuál de las siguientes secuencias de tipos de memoria está ordenada de menores a mayores prestaciones?

Usuario Profesores

- a) DDR, SDRAM, FPM
- b) EDO, SRAM, FPM
- c) SDRAM, DDR, EDO
- **FPM, EDO, RDRAM**
- d)

---

En un sistema de interrupciones vectorizado y en daisy-chain, ¿cuál de las siguientes afirmaciones es cierta?

Usaria Profesores

- La gestión de prioridades queda establecida por el orden en que
- a) los dispositivos reciben la señal INTA y la identificación de los dispositivos se realiza leyendo sus registros de estado
  - b) la identificación de los dispositivos se realiza leyendo sus registros de estado
  - **El daisy-chain asigna a todos los dispositivos la misma prioridad y**
  - c) **que los dispositivos reciben la señal INTA y el dispositivo se identifica por un dato que deposita en el bus**
  - d) 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

---

De las siguientes instrucciones, ¿cuál utiliza únicamente direccionamiento implícito?

Usaria Profesores

- a) push
- b) mul
- **c) popf**
- d) cmp

---

¿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?

Profesores

Usaria

- a) 6 bits
  - b) 7 bits
  - c) 10 bits
  - d) **8 bits**
- 

¿En qué tipo de ciclo de refresco se hace RAS# = 0?

Usaria Profesores

- a) Sólo RAS#
  - b) CAS# antes de RAS#
  - c) Refresco transparente
  - d) **En todos los anteriores**
- 

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?

Usaria Profesores

- a) 00000000 00000000 00000000 00000000
  - b) **11111111 10101010 01010101 00000000**
  - c) 11111111 10101010 01010101 11110000
  - d) 00000000 10101010 01010101 11110000
- 

Suponer un procesador tipo MIPS segmentado en etapas IF, ID, EX, MEM, WB, como el presentado de ejemplo en clase. ¿Cuál afirmación es correcta?

Usaria Profesores

- 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

---

¿En qué orden debería ejecutarse en una máquina de tipo pila la operación aritmética ( $a+b/c-d$ )?

Usuario Profesores

- a)  $a + b / c - d$
  - b)  **$a b c / + d -$**
  - c)  $a b + / c d -$
  - d)  $a b + c d - /$
- 

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:

Usuario Profesores

- a) **polling**
  - b) daisy-chain
  - c) interrupción
  - d) DMA
- 

Las arquitecturas de acumulador se caracterizan por:

Usuario Profesores

- a) instrucciones muy largas.
  - b) facilidad de generación de código eficiente para los compiladores.
  - c) **diseño simple del procesador.**
  - d) tráfico reducido entre procesador y memoria.
- 

[P6.1]

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 += 2)
 v[i] = 9;
```

¿Cuál será la tasa de fallos aproximada que se obtiene en la primera ejecución del bucle anterior?

Usuario Profesores

- a) 0 (ningún fallo)
- b) 1/2 (mitad aciertos, mitad fallos)

### **1/8 (un fallo por cada 8 accesos)**

- c) es un array de ints (4B) y se salta  $i+=2$ , los accesos están separados 8B y las líneas son de 64B, los accesos con índice  $i$  múltiplo  $8x$  son fallos, el resto son aciertos.
  - d) 1 (todo son fallos)
- 

[T5.1]

¿En qué tipos de técnicas de E/S la transferencia de información está bajo el control directo de la CPU?

Usuario Profesores

- 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
- 

[P3T]

En la convención cdecl estándar para arquitecturas x86 de 32 bits, el resultado de una función se devuelve usualmente en el registro:

Usuario Profesores

- a) EBP
  - b) ESI
  - c) EBX
  - d) **EAX**
- 

[T6.5]

¿Cuál de las siguientes políticas está \*menos\* relacionada con la jerarquía memoria?

Usuario Profesores

- a) **Política de planificación: en qué orden se ejecutarán los procesos pendientes.**
  - b) Política de reemplazo: qué bloque se tiene que sustituir (reemplazar) cuando se trae un bloque desde otro nivel.
  - c) Política de escritura: determina cómo se actualiza el nivel de la memoria  $i+1$  cuando se ejecutan instrucciones de almacenamiento en el nivel  $i$ .
  - d) Política de colocación: dónde se almacena un bloque de datos dentro de la memoria.
-

[P6.1]

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:

Usuario Profesores

- 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
  - **el vector se indexa con la variable de control del bucle, con un incremento o paso de line**
  - d) el bucle es for (unsigned i = 0; i < bytes.size(); i += line), si line se duplica se realizan la mitad de accesos
- 

[T2.3.1]

La dirección efectiva del primer parámetro de llamada a una función suele calcularse desde el código de la función como:

Usuario Profesores

- a) EBP+4
  - b) EBP-4
  - c) EBP-8
  - **EBP+8**
  - d)
- 

Un "buffer de traducción anticipada" (TLB) almacena información relativa a las últimas direcciones de memoria a las que ha accedido la CPU.

Usuario Profesores

V

---

[T2.3.1]

¿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?

Usuario Profesores

- a) %ebp + 0x4
- b) %ebp - 0x4
- **%esp + 0x4**
- d) %esp - 0x4

---

El ancho de banda de memoria es:

Usaria Profesores

- - 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

---

EEEn las

políticas anticipativas de extracción de caché, ¿cuál de ellas se caracteriza por preextraer el bloque  $i+1$  si se referencia al bloque  $i$  y se produce falta de bloque?

Usuario Profesores

- - a) Preextracción siempre
  - b) Preextracción marcada
  - c) **Preextracción por falta**
  - d) Preextracción indexada

---

El ancho de banda de memoria es:

Usuario Profesores

- - 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 tiempo que se tarda en transferir una palabra entre memoria y CPU
  - c) el intervalo de frecuencias de reloj permitidas entre memoria y CPU
  - d) **ninguna de las anteriores es cierta**

---

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  $**x$ .

Usaria Profesores

- $*((x + 16)) + 28)$
- a) cierto si se hubiese hecho typecast (void\*)x para evitar aritmética punteros

- - b)  $\ast(\ast(x + 2) + 4) + 3$   
sería  $x[2][4+3]$ ,  $+4+3 == +7$  sí sería aritmética punteros  
 $\ast(\ast(x + 2) + 4) + 3$
    - c) sería  $x[2][4]+3$ , la última suma sería suma entera, no aritmética punteros  
 $\ast((\ast x) + 2) + 7$
    - d)  $(\ast x)+2 == x[0]+2 != x[2]$
- 

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

Usuaria Profesores

- a) 12
  - b) 11
  - c) **6**
  - 10
  - d)
- 

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

Usuaria Profesores

- - a) el salto incondicional hacia adelante
    - b) goto test , según las transparencias
    - c) el salto incondicional hacia atrás
    - d) el salto condicional hacia adelante
- 

En una memoria DRAM que permite el acceso en modo página se accede a la palabra 0x1234. Si emplea páginas de 256 palabras, ¿Cuál será la menor dirección a la que podremos acceder rápidamente?

Usuaria Profesores

- - a) 0x1000
    - b) 0x1200
    - c) 0x1230
    - d) Otra
-

Una memoria que está estructurada en palabras de 16 bits tiene una capacidad de 64 Kbits.  
¿Cuántas palabras tiene?

Usuario Profesores

- a) 64000
  - b) 4000
  - c) 4096
  - d) 65536
- 

¿Cuál de las siguientes instrucciones máquina copia en el registro RDX la dirección efectiva resultante de la operación RAX\*8 + RBX?

Usuario Profesores

- a) movq (%rbx, %rax, 8), %rdx
  - b) movq 8(%rdx, %rax), %rdx
  - ✓ • c) leaq (%rbx, %rax, 8), %rdx
  - d) leaq 8(%rdx, %rax), %rdx
- 

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.

Usuario Profesores

F F

---

El tamaño del registro indicador de una memoria asociativa de n palabras x m bits/palabra es:

Usuario Profesores

- a) m bits
  - b) un bit
  - c) n/m bits
  - d) n bits
- 

La postescritura ("write-back") es más eficiente que la escritura directa ("write-through").

Usuario Profesores

V V

---

En una jerarquía de memoria, a medida que nos alejamos del procesador:

Usuario Profesores

- ✓ • a) **el tamaño de la unidad de transferencia entre dos niveles aumenta**  
b) el tiempo de transferencia disminuye  
c) el coste por byte aumenta  
    el tamaño de la memoria disminuye
  
  - d)
- 

La política de correspondencia de una memoria cache con 1 único conjunto es:

Usuario Profesores

**Totalmente asociativa**

- ✓ • a) todas las líneas son distintas vías del único conjunto ->  
    cualquier bloque de memoria puede ir a cualquier marco de  
    cache. Ver Tema 6 tr.137  
b) Directa  
c) Asociativa por conjuntos con una única línea  
d) Asociativa por conjuntos de una única vía
- 

En una jerarquía de memoria, a medida que nos alejamos de la CPU:

Usuario Profesores

- a) el tiempo de transferencia disminuye  
b) el coste por byte aumenta
  - c) **el tamaño de la unidad de transferencia entre dos niveles aumenta**  
d) el tamaño de la memoria disminuye
- 

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?

Usuaria Profesores

- a) 31,25 microsegundos
  - b) 7,8125 microsegundos
  - c) 32,768 segundos
  - d) 8192 milisegundos
- 

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?

Usuario Profesores

- a) **0xAC6**
  - b) 0xAC6E
  - c) 0xAC70
  - d) 0xAC72
- 

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?

Usuaria Profesores

- a) 8 ns
  - b) 10 ns
  - c) 12 ns
  - d)  $0.92 \cdot 4\text{ns} + 0.08 \cdot (100+4)\text{ns}$
- 

Sea una cache asociativa por conjuntos de 4-vías. ¿Cuál de las siguientes afirmaciones es cierta?

Usuario Profesores

- a) La cache tiene 4 conjuntos por bloque
  - b) La cache tiene 4 conjuntos por línea
  - c) **La cache tiene 4 líneas por conjunto**
  - d) La cache tiene 4 bloques por línea
- 

La tasa de fallos en una caché disminuye al aumentar la asociatividad.

Usuario Profesores

**V**

===== ¿Qué conjunto de componentes permite construir una memoria 256Mx32? (sin que sobren componentes)

Usuario Profesores

- a) 16 chips 64Mx4
- b) **32 chips 64Mx4**
- c) 16 chips 64Mx16
- d) Ninguna de las anteriores

**x**

===== Una celda de memoria SRAM tiene menos transistores por bit que una DRAM.

Usuario Profesores

**F**

===== 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:

Usuaria Profesores

- a)  $(4+n) / 5t$
- b)  $5n / (4+n)$
- c)  $kn/(k+n-1)$ , con k=5
- d)  $4n / (5+n)$

**x**

===== Una memoria estática tiene un bus de datos de 32 bits y su bus de direcciones es de 20 bits, ¿cuál es su capacidad?

Usuario Profesores

- ✓ • a) 32 MBytes
- ✓ • b) **4 MBytes**
- ✓ • c) 80 GBytes
- ✓ • d) 1 MByte

===== La "postescritura ("write-back") marcada" es más eficiente que la "postescritura siempre".

Usuario Profesores

**V V**

=====

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.

Usuario Profesores

F

===== ¿En qué tipo de ciclo de refresco RAS# permanece a 1?

Usuario Profesores

- a) Sólo RAS#
- b) CAS# antes de RAS#
- c) Refresco transparente  
**En ninguno de los**
- d) anteriores

===== Un TLB (buffer de traducción anticipada) tiene un número de entradas o elementos mucho menor que el número de páginas de la memoria virtual.

Usuario Profesores

V

===== [T6.3]

Se desea construir una memoria de SRAM de tamaño 3G X 8 a partir de elementos de memoria SRAM más pequeños. Cuál de las siguientes soluciones sería correcta:

Usuario Profesores

- a) 256 chips de 16Mx 1 bits
- b) 16 chips de 512 M x 2 bits
- c) **12 chips de 512M x 4 bits**  
Ninguna de las anteriores es correcta
- d)

===== Un procesador de 1GHz tarda 4ns en realizar 4 instrucciones sin realizar segmentación de cauce. ¿Cuánto tardaría en realizar 9 instrucciones un procesador con segmentación de cauce de 4 etapas si no existiera ningún retraso en ninguna de las instrucciones?

Usuaria Profesores

- X a) 4.5 ns
- b) 9 ns
- c) 3 ns
- d) 2 ns

=====

En una memoria asociativa sin registro de máscara, el valor de una clave almacenada en el registro de entrada se compara simultáneamente con todas las palabras de memoria.

Usuario Profesores

V

=====

¿En qué tipo de refresco de memoria DRAM CAS# permanece a 0 después del ciclo de lectura o escritura precedente?

Usuario Profesores

- a) RAS# antes de CAS#
- b) Sólo RAS#
- c) **Refresco transparente**  
Ninguna de las anteriores respuestas es correcta
- d)

=====

¿Qué tipo de información no contiene una entrada de la tabla de páginas cuando utilizamos memoria virtual paginada?

Usuario Profesores

- a) **Longitud de la página**
- b) Tipo de acceso permitido
- c) Dirección del marco de página que cupa
- d) Bit indicador de si la página es valida

=====

Si A=FF0Fh y B=0004h, el resultado de desplazar A a la derecha aritméticamente B veces es:

Usuario Profesores

- a) **FFF0h**
- b) 0FF0h
- c) F0F0h
- d) F0FFh

=====

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:

Usuario Profesores

- a) 4
- b) **3,3**
- c) 0,3
- d) 5

=====

Con 8 circuitos de memoria RAM de 1K x 8 se puede crear un memoria de:

Usuario Profesores

- a) 1K x 64
- b) 8K x 8
- c) 2K x 32
- d) **posibles**

**Todas las combinaciones anteriores son**

=====

¿Cuántas patillas de dirección tiene una memoria DRAM de 1G palabra, siendo la longitud de palabra de 16 bits?

Usuario Profesores

- a) 16
- b) 20
- c) 30
- d) **15**

=====

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".

Usuario Profesores

F

=====

Uno de los puntos clave de la traducción que gcc hace de una construcción switch-case de lenguaje C a lenguaje ensamblador es...

Usuario Profesores

- ✓ •
    - a) el salto directo
    - b) el salto condicional hacia atrás
    - c) **el salto indirecto**
    - d) el salto relativo a contador de programa
- =====

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?

Usuario Profesores

- - a) **El tamaño de un long es 4 bytes  
(IA32: 4B) vs. (x86-64: 8B)**
    - b) El tamaño de un short es 2 bytes
    - c) El tamaño de un unsigned es 4 bytes  
El tamaño de un int es 4 bytes
    - d)
- =====

Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señalar la respuesta falsa)

Usuaria Profesores

- - a) cola de instrucciones (precaptación)  
T4 tr.19

**x**

- b) registros de acople (coste de la segmentación)
  - c) riesgos (hazards)
  - d) fragmentación desigual (duración desigual de etapas)
- 

¿Cuál de las siguientes afirmaciones acerca de las memorias RAM estáticas es falsa?

Usuario Profesores

- a) Son más veloces que las memorias RAM dinámicas
  - b) Las operaciones de lectura no son destructivas
  - El número de transistores necesario para implementar**
  - c) cada celda es menor que en las memorias RAM dinámicas**
    - Los datos almacenados se mantienen por un tiempo
  - d) indefinido
- 

El tamaño de la unidad de transferencia entre dos niveles de una jerarquía de memoria aumenta a medida que nos alejamos de la CPU.

Usuario Profesores

**V**

---

Cada bit de una RAM dinámica ha de ser refrescado cada pocos nanosegundos.

Usuario Profesores

**F**

**F**

---

Un SIMM con 8 chips de 4Mx1 contiene 1 M palabras de 32 bits.

Usuario Profesores

**F**

---

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

Usuario Profesores

**ambos se recalculan tras cada operación aritmético-lógica con ints, correspondiendo al programador**

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

=====

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?

Usuario Profesores

- - a) **8192**
  - b) 1024
  - c) 65536
  - d) 64

=====

La penalización por una falta de página suele ser de cientos de miles de ciclos de reloj de la CPU.

Usuario Profesores

V

=====

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?

- |         |                                                                                                                                                                                                                                                             |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usuario | Profesores                                                                                                                                                                                                                                                  |
|         | <ul style="list-style-type: none"><li>a) Depende del tamaño del bus de direcciones</li><li>b) El que tiene E/S mapeada en memoria</li><li>c) <b>El que tiene E/S independiente</b></li><li>d) Ambos podrán acceder a la misma cantidad de memoria</li></ul> |
- =====

¿Cuál de las siguientes afirmaciones es falsa?

- |         |                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usuario | Profesores                                                                                                                                                                                                                                                                                                                                                                                                                          |
|         | <ul style="list-style-type: none"><li>• <ul style="list-style-type: none"><li>a) <b>Las memorias DRAM son en general mucho más rápidas que las SRAM</b></li><li>b) Una celda DRAM sólo necesita un transistor y un condensador</li><li>c) Las memorias DRAM presentan generalmente una capacidad de almacenamiento mucho mayor que las SRAM</li><li>d) La operación de lectura de una celda DRAM es destructiva</li></ul></li></ul> |
- =====

En el contexto de las DRAM, RAS significa:

- |         |                                                                                                                                                                                                                                                                                                                                                                                  |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usuario | Profesores                                                                                                                                                                                                                                                                                                                                                                       |
|         | <ul style="list-style-type: none"><li>• <ul style="list-style-type: none"><li>a) <b>Row Access Strobe (muestreo de acceso a filas)</b></li><li>b) Random Access Shot (disparo de acceso aleatorio)</li><li>c) Random Access Strobe (muestreo de acceso aleatorio)<br/>Refresh After Select (refresco después de selección de la memoria)</li><li>d) memoria)</li></ul></li></ul> |
- =====

Siendo RDX=0xf000 y RCX=0x0100, ¿cuál de las siguientes instrucciones tiene como dirección efectiva 0xf400?

- |         |                                                                                                                                                                                                                                               |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Usuario | Profesores                                                                                                                                                                                                                                    |
|         | <ul style="list-style-type: none"><li><ul style="list-style-type: none"><li>a) movq 0x8(%rdx), %rax</li><li>b) leal 0x80(%rdx, 2), %eax</li><li>• c) <b>xorl (%rdx, %rcx, 4), %eax</b></li><li>d) addq (%rdx, %rcx), %rax</li></ul></li></ul> |
- =====

Una memoria SRAM tiene una capacidad de 64 Kbits y utiliza 12 líneas para

direcciónamiento. Indique cuál es el tamaño de palabra de dicha memoria:

Usuario Profesores

- a) 32 bits
- b) 8 bits
- c) 64 bits
- **16 bits**
- d)

=====Una memoria que está estructurada en palabras de 8 bits tiene una capacidad de 32 Kbits.

¿Cuántas líneas de dirección tiene dicha memoria?

Usuario Profesores

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

=====

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

Usuario Profesores

- ✓ • a) Todas requieren la misma atención
- b) **E/S mediante acceso directo a memoria**
- c) E/S mediante interrupciones
- d) E/S programada

=====

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?

Usuaria Profesores

- ✗ • a) **Entrelazado de orden superior**
- b) Entrelazado de orden inferior
- c) Entrelazado de orden intermedio
- d) Es igual de difícil en cada caso

=====

Si AX = FA50h y ejecutamos XOR AX, 00FFh

Usuaria Profesores

- ✓ • a) Se realiza el complemento a 1 de AH.
- b) El registro AH se pone a 0.
- c) **Se realiza el complemento a 1 de AL.**

- d) El registro AL se pone a 0.
- 

¿Cuál de las siguientes afirmaciones acerca de la jerarquía de memoria es \*FALSA\*?

Usuario Profesores

- Un computador puede tener una pequeña cantidad de
- a) memoria rápida además de una gran cantidad de memoria más lenta
  - 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) Acceder a los discos es órdenes de magnitud más lento que acceder a la RAM
- 

¿En qué se diferencian las estrategias de mantenimiento de coherencia en memoria escritura directa y post-escritura?

Usuario Profesores

- a) **En cuándo tiene lugar la actualización**
  - b) En cómo tiene lugar la actualización
  - c) Tanto en a) como en b)
  - d) Ni en a) ni en b
- 

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:

Usaria Profesores

- a) var es un array multi-nivel (punteros a enteros) de cuatro filas
- x b) var es un array multi-nivel pero no se pueden adivinar las dimensiones
- c) var es un array bidimensional de enteros, no se pueden adivinar dimensiones

**var es un array bidimensional de enteros, con cinco columnas**

se puede adivinar incluso el código C:

- d)

```
int get_var_digit(size_t fila, size_t columna){
 return var[fila][columna];
}
```

porque se calcula  $\%rsi = fila * 5 + columna$  para indexar en var,  
 $\%rsi, 4$ )

---

Un módulo de memoria de 16 GB está formado por varios chips DRAM de 1024Mx4.

¿Cuántos chips DRAM necesita el módulo?

Usuario Profesores

- a) 8
- b) 4
- c) 16
- 32**

$$16G \times 8 = 16 \cdot (1024M) \times 2 \cdot (4)$$

2x "a lo ancho", 16x "a lo alto"

- d)

---

Una instrucción de "salto si menor", para números sin signo, tiene que comprobar el valor de:

Usaria Profesores

- a) el bit de signo
- b) el bit de cero
- ✓     c) el bit de acarreo**
- d) los bits de signo y desbordamiento

---

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

Usaria Profesores

- a) 40 bits
- b) **35 bits**
- c) 29 bits
- d) 39 bits

---

La instrucción movzbl %al, %eax

Usuaria Profesores

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

---

En las arquitecturas RISC:

Usuaria Profesores

- a) la programación resulta más simple que en los CISC.
- b) realmente se usan pocas instrucciones de las disponibles en el conjunto de instrucciones.
- c) **se usan instrucciones muy simples que se pueden segmentar.**
- d) la UC suele ser más compleja que en CISC, al ser cableada.

---

¿En qué registro se pasa el primer argumento a una función en Linux gcc x86-64?

Usuaria Profesores

- a) edi
- b) esi
- c) ecx
- d) edx

---

En una cache asociativa por conjuntos, la vía i está constituida por:

Usuario Profesores

- a) **todos los bloques i-ésimos de cada conjunto**
- b) todos los bloques del conjunto i
- c) todos los conjuntos del bloque i

- d) ninguna de las anteriores es cierta
- 

Toda celda de memoria DRAM tiene al menos cuatro transistores.

Usaria Profesores

F

---

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

Usaria Profesores

- a) **CMP, TEST**
- b) IMUL, IDIV
- c) AND, OR, XOR
- ADC, SBB
- d)

X

---

¿Cuál de los siguientes no es un modo de direccionamiento IA32?

Usaria Profesores

- X a) Memoria
- b) Inmediato
  - c) **Cache**
  - c) no existe, ver [T2.1.3ConASM] tr.27
  - d) Registro
- 

En la secuencia de instrucciones siguiente, siendo el primer registro el destino, ¿cuántos riesgos se dan?

sub r4,r1,r3

or r8,r6,r4

Usaria Profesores

- ✓ a) Un riesgo estructural
- b) Un riesgo por dependencia de datos
  - c) Un riesgo estructural y dos por dependencia de datos
  - Dos riesgos por dependencia de datos y uno de control
  - d)
- 

¿Cuál de las siguientes afirmaciones es cierta?

Usaria Profesores

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

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

¿Cuál es el resultado de evaluar la expresión 0b1110 ^ 0b1010 en lenguaje C?

(0b indica que el número está expresado en binario)

Usaria Profesores

- a) **0b0100**
  - b) 0b1010
  - x c) 0b0110
  - d) 0b1111
- 

La primera instrucción ensamblador de una subrutina compilada con gcc en Linux/x86 cdecl suele ser:

Usaria Profesores

- a) pop %ebx
  - b) **push %ebp**
  - x c) mov %esp, %ebp
  - d) push %ebx
- 

La instrucción jbe / jna provoca un salto si...

Usaria Profesores

- ✓ • a) SF == 1 || ZF == 1
  - b) **CF == 1 || ZF == 1**
  - c) CF == 1
  - d) SF != OF
- 

Usaria Profesores

- a) pop %ebx
  - b) **push %ebp**
  - x c) mov %esp, %ebp
  - d) push %ebx
- 

=  
¿Qué valor contendrá el registro edx tras ejecutar las dos instrucciones siguientes?

movl \$-1, %edx

movb \$1, %dl

Usaria Profesores

- a) 00000000 00000000 00000000 00000001  
b) 00000001 00000000 00000000 00000000  
c) 11111111 11111111 11111111 11111111  
• d) **11111111 11111111 11111111 00000001**
- 

La idea de la arquitectura RISC se debe, entre otros, a:

Usaria Profesores

- a) Maurice V. Wilkes  
b) Andrew S. Tanenbaum  
c) Alan M. Turing  
• d) **John Cocke**
- 

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

Usaria Profesores

- a) Escasa capacidad para expresar paralelismo entre microoperaciones.  
• b) **Ninguna o escasa codificación.**  
c) Muchos campos solapados.  
d) Microinstrucciones cortas.
- 

¿Cuál es el ancho del bus de direcciones de un chip DRAM de 1G palabra, siendo la longitud de palabra de 16 bits?

Usuario Profesores

- a) 20  
b) 16  
c) 30  
• d) **15**
- 

¿Cuál de las siguientes afirmaciones es incorrecta?

Usaria Profesores

- a) En las arquitecturas CISC hay más instrucciones que en las RISC.  
b) Las arquitecturas RISC son del tipo registro-registro.  
• c) **El tamaño de una instrucción en lenguaje máquina**

**siempre ocupa dos bytes en los procesadores RISC.**

- d) Las arquitecturas RISC simplifican la decodificación.
- 

Si el registro EAX contiene X, La secuencia de instrucciones siguiente:

cmpl \$6, %eax

jae Destino

salta a la etiqueta Destino sólo si:

Usuaria Profesores

- a)  $X \geq 0 \text{ && } X \leq 6$
  - x** b)  $X \leq 6$
  - c)  $X > 6$
  - d)  $X < 0 \text{ || } X \geq 6$
- 

La memoria virtual del computador es:

Usuario Profesores

- 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.**
- 

Se ha declarado en un programa C la variable int val[5]={1,5,2,1,3} . ¿Cuál de las siguientes afirmaciones es FALSA?

Usuaria Profesores

- a)  $\text{val}[1] == 1$
  - b)  $\&\text{val}[3] == \text{val} + 3$
  - x** c)  $\text{sizeof}(\text{val}) == 20$
  - Todas son ciertas
  - d)
- 

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?

Usuario Profesores

- a) 16 B
  - b) 128 B
  - c) 64 B
  - c)  $4\text{MB} = 2^{22}\text{B} = 2^{12}\text{conj.} \cdot 2^4\text{lin./conj. (vías)} \cdot 2^6\text{B/lin.}$
  - d) 32 B
- 

La expresión  $d(t+1) = d(t)+1$ , donde  $d(x)$  es la dirección de memoria referenciada en el instante de tiempo  $x$ , corresponde al principio de localidad temporal.

Usuario Profesores

F

---

El comienzo de un procedimiento que siga la convención cdecl es:

Usuaria Profesores

- a) **mov %ebp,%esp; push %ebp**
  - X b) mov %esp,%ebp; push %ebp
  - c) push %ebp; mov %ebp,%esp  
**push %ebp; mov %esp,**
  - d) **%ebp**
- 

¿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?

Usuario Profesores

- a) 6 bits
  - b) 10bits
  - c) **8 bits**
  - d) 7 bits
- 

[P2T]

Dada la siguiente definición de datos:

lista: .int 0x10000000, 0x50000000,

0x10000000, 0x20000000

longlista: .int (.lista)/4

resultado: .quad 0x123456789ABCDEF

formato: .ascii "suma=%llu=%llx hex\n\0"

La instrucción para copiar la dirección de memoria donde comienza lista en el registro RBX es:

Usuario Profesores

- a) movq lista, %rbx
  - b) movl \$lista, (%rbx)
  - c) **movq \$lista, %rbx**
  - c) Práct.2, Tut, pág.10
  - d) movq (lista), %rbx
- 

¿Qué combinación de flags aritmético-lógicos corresponde al código de condición b (below)?

Usaria Profesores

- a) **CF**
  - b) CF xor OF
  - c) OF xor SF
  - d) OF
- 

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

Usaria Profesores

- a) **AMD64**
  - b) IA-32
  - c) x86
  - d) IA-64
- 

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

Usaria Profesores

- a) **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**
- b) Una subrutina que modifique algún registro debe restaurar su valor anterior antes de retornar
- c) Hay registros modificables, otros que deben ser restaurados, y las subrutinas anidadas deben respetar los

registros modificables que están en uso por otras subrutinas  
Todos los registros pueden ser modificados libremente por  
todas las subrutinas

d)

---

El primer microprocesador de 32 bits de la familia x86 fue el:

Usaria Profesores

- a) 80486
  - b) **80386**
  - c) 80286
  - x d) Pentium
- 

¿Cuál es el popcount (peso Hamming, nº de bits activados) del número 19?

Usaria Profesores

- a) 2
  - b) **3**  
 $19 = 0x13 = 16+3 = 0b0001\ 0011 \rightarrow \text{popcount } 3$
  - c) 4
  - d) 5
- 

¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar):

Usaria Profesores

- a) movw %dx, (%rax)
  - b) **movb \$0xFF, (%dl)**  
no, no se puede direccionar con (%dl)
  - c) movswl (%eax), %edx
  - x d) movzbl %dl, %eax
- 

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

Usaria Profesores

- a) dos saltos condicionales (uno para la parte if y otro para la parte else)
- b) dos saltos condicionales y dos saltos incondicionales

- c) un salto condicional, según la condición opuesta a la del código C, y otro salto incondicional
  - d) un salto condicional, según la condición expresada en el código C
- 

El microprocesador de la familia x86 usado en los primeros IBM PC originales fue el:

Usuaria Profesores

- a) 80486
  - b) 8085
  - c) **8088**
  - d) Pentium
- 

¿Qué tipos de instrucciones se emplean más en una arquitectura de registros de propósito general?

Usuaria Profesores

- a) De desplazamiento y rotación.
  - b) **Aritmético-lógicas.**
  - c) De transferencia de datos.  
      De transferencia de control.
  - d)
- 

Un módulo de memoria de 16 GB está formado por varios chips DRAM de 1024Mx4.

¿Cuántos chips DRAM necesita el módulo?

Usuario Profesores

**32**

- a)  $16\text{G} \times 8 = 16 \cdot (1024\text{M}) \times 2 \cdot (4)$   
      2x "a lo ancho", 16x "a lo alto"
  - b) 4
  - c) 8
  - d) 16
- 

¿Cuál de los siguientes modos de direccionamiento es menos preferible para un procesador de 32 bits y con tamaño de instrucción de 32 bits?

Usuaria Profesores

- a) indexado
- b) indirecto a través de registro
- c) registro

- d) directo (o absoluto)
- 

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:

Usaria Profesores

- a) Un mayor retraso en la ejecución de los programas debido al incremento del número de etapas
  - b) Una disminución en la posible dependencia de datos
  - c) **Un incremento de las prestaciones**
  - d) Una disminución de la máxima frecuencia de reloj a la que puede operar el cauce
- 

Respecto a la segmentación:

Usaria Profesores

- a) Cuanto más parecidos sean el tiempo de ejecución de una instrucción sin segmentar y el tiempo de una etapa en el procesador segmentado, mayor será la ganancia máxima que se puede obtener  
Cuando el número de instrucciones ejecutadas en un
  - b) procesador segmentado crece, la ganancia máxima que se puede obtener tiende a 1  
**Cuanto mayor sea la relación entre el tiempo de ejecución de una instrucción sin segmentar y el tiempo de una etapa en el procesador segmentado, mayor será la ganancia máxima que se puede obtener**  
Cuando el número de instrucciones ejecutadas tiende al
  - c) d) número de etapas de un procesador segmentado, la ganancia máxima que se puede obtener tiende a infinito
- 

Los riesgos de datos consisten en que...

Usaria Profesores

- a) dos instrucciones acceden a la vez al mismo dato  
**una instrucción necesita un dato calculado por otra anterior**
  - b) dos instrucciones necesitan leer el mismo dato
  - c) todas las respuestas anteriores son correctas
- 

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

Usaria Profesores

- ✓ •
    - a) estructurales
    - b) **(por dependencias) de datos**
    - c) organizativos
    - d) de control
- 

Respecto a la segmentación, ¿cuál de las siguientes afirmaciones es falsa?

Usaria Profesores

**Retrasar la fase de decisión saltar/no saltar de las**

- a) **instrucciones de salto condicional contribuye a mejorar el rendimiento del procesador**  
Cuantas más etapas tenga un cauce, más instrucciones se
  - b) estarán ejecutando en distintas fases y más posibilidades se presentan de que existan riesgos entre ellas  
La técnica de register forwarding habilita una serie de
  - c) caminos (buses) que se añaden al cauce para permitir que los resultados de una etapa pasen como entradas a la etapa donde son necesarias
  - x d) La reorganización del código y la introducción de instrucciones nop permite evitar dependencias de datos
- 

La segmentación de cauce...

Usaria Profesores

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

Un salto condicional del tipo "delayed branch", o salto retardado, ejecuta la(s) instrucción(es) siguiente(s)...

Usaria Profesores

- sólo si el salto se produce (las ignora si NO se produce), de
- a) manera que instrucción(es) en el destino del salto podrían adelantarse tras la propia instrucción de salto
- sólo si el salto NO se produce (las ignora si se produce), de
- b) 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
- 

En la secuencia de instrucciones siguiente, siendo el primer registro el destino, ¿cuántos riesgos se dan?

sub r2,r1,r3

or r8,r6,r2

Usaria Profesores

- a) Dos riesgos por dependencia de datos y uno de control  
**Un riesgo por dependencia de datos**

- b) la resta calculada en r2 debe estar disponible para el or con r6
  - X c) Un riesgo estructural
  - d) Un riesgo estructural y dos por dependencia de datos
- 

¿Cuál de las siguientes afirmaciones respecto a la memoria RDRAM no es cierta?

Usaria Profesores

- a) Las direcciones de fila y columna son enviadas en el mismo ciclo de reloj por buses separados
- b) La interfaz entre el procesador y la memoria es muy diferente a la de las memorias DDR
- ✓ • c) **El bus de datos suele ser muy ancho.**
- d) Funciona de forma síncrona y a muy alta velocidad
- 

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

Usaria Profesores

- X a) 64 memorias de 128 x 1 bits
- b) 8 memorias de 512 x 2 bits
- c) **8 memorias de 256 x 4 bits y un decodificador de 2 a 4**
- d) Ninguna de las anteriores respuestas es cierta
- 

Respecto al refresco de memorias DRAM, ¿cuál de las siguientes afirmaciones es falsa?

Usaria Profesores

**Una operación de refresco consiste en dar un**

- a) **impulso /CAS junto con una dirección de columna.**  
es /RAS con dirección de fila, ver Tema 6 tr.54
- X b) Se precisa una circuitería auxiliar, externa al chip DRAM o

- integrada en él, que produzca ciclos de refresco.
- c) Los chips DRAM refrescan automáticamente la fila accedida en cualquier ciclo de lectura o escritura.
- d) Los ciclos de refresco deben producirse cada pocos ms (milisegundos).
- 

¿Cuántas líneas de dirección son necesarias en un memoria RAM dinámica de 64 K palabras? ¿Y en una estática?

Usuaria Profesores

- a) 8 / 8
- b) 16 / 8
- x c) 16 /16
- d) **8 / 16**
- 

¿Cuál es el tamaño de la marca de cache en un microprocesador con 32 KB de memoria cache asociativa por conjuntos con 16 palabras de 32 bits por bloque y 8 bloques por conjunto, si el microprocesador es capaz de direccionar 1 MB de memoria principal (memoria de bytes)?

Usuaria Profesores

- a) 7 bits
- b) **8 bits**
- c) 6 bits
- x d) 10 bits
- 

Variación de los parámetros de los distintos niveles en una jerarquía de memoria (señalar la opción incorrecta)

(mediante  $_i$  y  $_i+1$  se representan los subíndices i e  $i+1$ )

Usuaria Profesores

- x a) tiempo de acceso:  $t_i \leq t_{i+1}$
- b) ancho de banda:  $b_i \geq b_{i+1}$
- c) tamaño del nivel:  $s_i \leq s_{i+1}$   
**unidad de transferencia:  $x_i$**
- d)  $>=x_{i+1}$
- 

El direccionamiento directo a memoria utiliza...

Usaria Profesores

- - a) **un desplazamiento.**
    - b) un registro.
    - c) un registro y un desplazamiento contenidos en la propia instrucción.
    - d) dos desplazamientos contenidos en la propia instrucción.
- 

¿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"?

Usaria Profesores

- a) Localidad secuencial
    - b) Localidad temporal
    - c) Ninguna de las otras respuestas es correcta
    - ✓      d) **Localidad espacial**
- 

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

Usaria Profesores

- a) 14 bits
    - b) depende del tamaño de línea (en cuanto lo revelemos se puede responder el valor concreto)  
depende del número de vías y del tamaño de línea (en cuanto revelemos ambos se puede responder el valor concreto)  
**depende del número de vías (en cuanto lo revelemos se puede responder el valor concreto)**
    - ✓      d)  $256\text{KB} = 2^{18}$ , 18 bits direccionamiento  
ej: 4 vías: 16 bits campos conjunto y palabra -> 16 bits etiqueta
- 

Un programa crea en memoria una larga secuencia de números de forma consecutiva. ¿Qué tipo de estrategia de mantenimiento de coherencia es más eficiente para ejecutar este programa en un sistema con jerarquía de memoria?

Usaria Profesores

- a) Escritura directa ("write-through")
    - ✓      b) **Post-escritura ("write-back")**
    - c) Tanto a) como b) son igual de eficientes
    - x      d) No puede saberse qué técnica es mejor
-

¿Cuál de las siguientes afirmaciones es cierta?

Usaria Profesores

- 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**
- 

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?

Usaria Profesores

- a) 0.5  
b) 0.5  
• c) **0.9**  
d) 0.8
- 

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

Usaria Profesores

- a) 64 pastillas  
 b) **16 pastillas**  
c) 8 pastillas  
 d) 32 pastillas
- 

¿Cuál de las siguientes instrucciones no modifica necesariamente la secuencia de ejecución del programa?

Usaria Profesores

- a) RET  
b) CALL dir  
c) JMP dir  
• d) **JNE dir**
- 

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:

Usaria Profesores

- a) Una rotación a la derecha a través de acarreo.  
b) Un desplazamiento lógico a la derecha.  
c) Una rotación a la derecha.

- d) Un desplazamiento aritmético a la derecha.
- 

Una máquina superescalar es aquella que:

Usuaria Profesores

- - ✓ 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.
- 

El ajuste de marco de pila que gcc (Linux/IA-32) prepara para todas las funciones consiste en las instrucciones

Usuaria Profesores

- - a) **pushl %ebp  
movl %esp, %ebp**
    - b) movl %esp, %ebp  
pushl %esp
    - c) pushl %esp  
movl %ebp, %esp  
movl %ebp, %esp
    - d) pushl %ebp
- 

¿Cuál de las siguientes afirmaciones es falsa respecto al lenguaje C?

Usuaria Profesores

- ✓ • a) **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**  
Los parámetros se introducen en la pila en el orden inverso
  - b) a como aparecen en la llamada de C, es decir, empezando por el último y acabando por el primero  
Pasar a una función un puntero a una variable se traduce en
  - c) introducir en la pila el valor de la dirección de memoria donde está almacenada la variable  
En lenguaje C, al llamar a una subrutina o función se
  - d) introducen los parámetros en la pila y después se realiza una llamada a la subrutina
-

Las señales BHE# y A0(=BLE#) son necesarias para seleccionar direcciones pares e impares en el procesador

Usuaria Profesores

- - a) 8088
  - b) **8086**
  - c) 80386
  - x d) Pentium

=====

En IA32 el puntero de pila es:

Usuaria Profesores

- x
  - a) un registro de 16 bits en el microprocesador
  - b) **una dirección de memoria de 32 bits almacenada en el contador de programa**  
un conjunto de posiciones de memoria usadas para
  - c) almacenar información temporal durante la ejecución del programa
  - d) un registro de 32 bits en el microprocesador

=====

[T2.1.2] ¿Cuál de las siguientes afirmaciones es correcta?

Usuaria Profesores

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

=====

Una caché de 64 palabras y correspondencia por sectores utiliza bloques de una palabra.

Si las direcciones de memoria principal son de 24 bits y cada sector está constituido por 32 bloques, el campo etiqueta debe tener:

Usuaria Profesores

- x
  - a) **19 bits**
  - b) 18 bits
  - c) 20 bits
  - d) 21 bits

=====

El resultado de desplazar aritméticamente dos posiciones hacia la derecha el número de 8

bits en complemento a dos -32 es:

Usuaria Profesores

- - a) 56
  - b) -128
  - c) **-8**
  - x d) Ninguno de los resultados anteriores es correcto

=====

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

Usuaria Profesores

- a) `mov (%rax,%rbx,4), %rcx`  
no, indexa 4x
- b) `lea (%rax,%rbx,8), %rcx`  
no, no mueve el valor sino la dirección
- c) `lea (%rax,%rbx,4), %rcx`  
no, indexa 4x
- ✓     • d) **`mov (%rax,%rbx,8), %rcx`**

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:

( $\wedge$  simboliza potenciación)

Usuaria Profesores

- x a) bits  $w \dots w+c-1$  (siendo  $2^c=n^o$  conjuntos)  
campo "conjunto"
- b) bits  $w \dots w+c-1$  (con  $c=b-v$ )  
campo "conjunto"
- c) últimos  $w$  bits ( $0 \dots w-1$ ) (los menos significativos)  
campo "palabra"
- d) **bits  $b \dots b+c-1$  (siendo  $2^c=n^o$  conjuntos)**  
sin sentido

En Linux IA32, si gcc usa la instrucción leave se puede asegurar que en ese punto del programa

Usuaria Profesores

- a) **ya no hay registros salva-invocado que recuperar**  
tal vez porque no había ninguno, para empezar
- b) correspondería emitir la secuencia de salida pop/ret, pero

- leave hace lo mismo y ocupa menos espacio  
leave no equivale a ret, y sólo hace pop %ebp  
ya no hay variables locales que destruir
- c) puede usarse leave para destruir rápidamente todas las variables locales, siempre que no haya que recuperar registros salva-invocado  
ya no se hacen llamadas anidadas y por tanto no hay parámetros que ocupen espacio en pila
- d) puede usarse leave habiendo espacio reservado en pila para argumentos de llamadas anidadas, siempre que no haya que recuperar registros salva-invocado
- 

GCC/Linux IA32 resuelve el ajuste de marco de pila mediante las instrucciones:

Usaria Profesores

- a) movl %ebp, %esp; popl %ebp
- b) movl %esp, %ebp; popl %esp
- c) **pushl %ebp; movl %esp,**  
**%ebp**
- d) pushl %esp; movl %ebp, %esp
- 

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

Usaria Profesores

- a) 10  
**12**
- b) 16M=  $2^{24}$ , multiplexados 12+12
- c) 24
- d) 28
- 

Usando el repertorio IA32, para intercambiar el valor de 2 variables (por ejemplo A: .int 1 y B: .int 2) se pueden usar...

Usaria Profesores

- a) dos instrucciones mov
- b) 3 mov, no menos (se le llama "intercambio circular")  
si fuera arquitectura M/M se podría hacer intercambio circular:  
mov A,R1  
mov B,A; // esta instrucción no es de repertorio R/M

- mov R1,B
- ✓      •      c) una instrucción mov y una instrucción lea
  - d) **4 mov, no menos (debido a la arquitectura R/M)**  
ver la función swap()
- 

La operación aritmética calculada por el programa

```
mov r1,#5
mov r2,#3
mov r3,#7
mov r4,#8
mul r5,r2,r3
add r6,r1,r5
sub r7,r6,r4
```

es:

Usuaria Profesores

- ✗
- a)  $8 - 5 + (3 \times 7)$
  - b)  $8 - (3 \times 7) + 5$
  - c)  $(3 \times 7) + 8 - 5$
  - d) **5 + (3 × 7) - 8**
- 

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 segundo nivel es de 3 microsegundos, ¿cuál sería aproximadamente el porcentaje de fallos en los accesos al primer nivel?

Usuaria Profesores

- ✗
- a) 90%
  - b) 99%
  - c) **10%**  
 $300\text{ns} = 6\text{ns} + F \cdot 3\mu\text{s} \Rightarrow F = 0.098 \sim 0.10 = 10\%$
  - d) 1%
- 

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

Usuaria Profesores

- a) lea accede a la posición indicada, mientras que mov no lo hace
- b) **mov accede a la posición indicada, mientras que lea no lo hace**

- mov puede usarse para copiar un registro a otro, mientras
- x c) que lea no  
lea (%ebx), %eax
- lea puede usarse para copiar un registro a otro, mientras
- d) que mov no  
mov %ebx, %eax
- 

En el direccionamiento inmediato el operando reside en:

Usuaria Profesores

- a) en la pila
- b) en memoria, en la dirección indicada
- x c) en un registro del procesador  
**en la instrucción tras el código de**
- d) **operación**
- 

¿En qué tipo de memoria virtual es un problema la fragmentación externa?

Usuario Profesores

- a) Memoria paginada
- b) **Memoria segmentada**
- c) Memoria con segmentación paginada
- d) En ninguno de ellos
- 

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?

Usuaria Profesores

- x a) 0xffffffffcd
- b) 0x5fa
- c) 0x628
- d) **0x62d**
- 

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

Usaria Profesores

- - a) %rax se usa para devolver los valores de retorno de funciones
    - b) **Todos los argumentos de función se pasan a través de la pila**
    - c) %eax y %ebx pueden usarse como en un sistema IA32  
%rbp se puede usar como cualquier otro registro (no hay puntero base)
    - x d) puntero base)
- 

Precaptar instrucciones antes de que sean necesarias y almacenarlas en una cola de instrucciones, es una técnica que se usa para...

Usaria Profesores

- - x 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
- 

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

Usaria Profesores

- ✓ •
    - a) Poner a 0 los últimos 3 bits del registro EAX
    - b) **Cambiar 0<->1 (complemento a 1 de) los últimos 2 bits del registro EAX**
    - c) Poner a 1 el último bit del registro EAX
    - d) Ninguno de los anteriores resultados
- 

Si rax contiene x, ¿cuál de las siguientes instrucciones calcula x\*9?

Usaria Profesores

- ✓ •
    - 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
- 

¿Cuál expresión es cierta?

Usaria Profesores

- a)  $\text{popcount}(15) < \text{popcount}(51)$   
 $4 < 4$
- b)  $\text{popcount}(7) > \text{popcount}(60)$

3>4

**popcount( 2 ) == popcount(64)**

- c)  $1==1$   
0b0000 0010 // 0b0100 0000 // popcount 1 ambos casos
  - d)  $\text{popcount}(96) != \text{popcount}( 3 )$   
 $2!=2$
- 

Un circuito SRAM con una capacidad de 256 Kbits tiene las patillas de direcciones A14 a A0. ¿Con cuál de las siguientes expresiones indicaría las características de capacidad en direcciones y datos del circuito?

Usuaria Profesores

- a) 256 K x 1
  - b) 64 K x 4
  - c) **32 K x 8**
  - d) 8 K x 32
- 

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

Usuaria Profesores

- a) la dirección de memoria de la instrucción ret se transfiere al contador de programa
  - b) **la dirección almacenada en la cima de la pila se transfiere al contador de programa**
  - c) la dirección almacenada en la cima de la pila se transfiere al puntero de pila
  - d) la dirección de comienzo de la pila se transfiere al puntero de pila
- 

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

Usuaria Profesores

- a) pueden estar en cualquier posición de la pila.
  - b) se introducen en la pila tras realizar la operación.
  - c) **son la cima de la pila y el elemento siguiente de la cima de la pila.**
  - d) son dos registros del procesador.
- 

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?

Usuaria Profesores

- a) 1%
  - b) 50%
  - c) **10%**
  - d) 90%
- 

En una caché con bloques de 32 bytes, ¿cuál es la dirección de memoria del primer byte del marco de bloque que contiene la dirección AC72 (en hexadecimal)?

Usuaria Profesores

- a) **AC0**
  - b) ACE
  - c) AC0
  - d) AC2
- 

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:

Usuaria Profesores

- a) **5+(3×7)–8**
  - b)  $8-(3\times 7)+5$
  - c)  $8-5+(3\times 7)$
  - d)  $(3\times 7)+8-5$
- 

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

Usuaria Profesores

- a) **ESP no puede ser registro índice**
- b) ESP no puede ser registro base
- c) EBP no puede ser registro base
- d) EBP no puede ser registro índice

=====

En un sistema Linux IA32, ¿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?

Usaria Profesores

x

- a) 16
- b) 128

**64**

en realidad casi no hay que saber de cache

- c) basta entender que si hay 4 vías es como si fueran 4 caches de 1K (un conjunto son las 4 "oportunidades" de una línea) y si hay 4 conjuntos cada línea son  $2^{10}/2^2=2^8=256$  bytes =  $2^8/2^2$  B/int= $2^6 = 64$  ints

- d) 32

=====

Respecto a la interfaz de E/S, ¿cuál de las siguientes afirmaciones es \*FALSA\*?

Usaria Profesores

- a) Involucra tareas que se pueden implementar parte en hardware y parte en software.
- b) Permite configurar el funcionamiento del periférico en un momento determinado, y además conocer su estado.
- c) 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.
- d) **Una interfaz de entrada recibe los datos desde el procesador y los transforma y envía al periférico en formato digital.**

no, se dice Entrada (y Salida) desde el punto de vista del procesador

=====

Motivos que impiden que la ganancia (aceleración) de un cauce segmentado sea ideal (señalar la respuesta falsa)

Usaria Profesores

x

- a) fragmentación desigual (duración desigual de etapas)
- b) **cola de instrucciones (precaptación)**
- c) registros de acople (coste de la segmentación)
- d) riesgos (hazards)

=====

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)?

Usaria Profesores

- - a) 21000 MIPS
    - b) **9000 MIPS**
    - c) 1000 MIPS
    - d) 150 MIPS
- 

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

Usaria Profesores

- a) todas ellas impiden que haya cortocircuito en el bus de direcciones
  - b) todas ellas impiden que haya cortocircuito en el bus de datos
  - c) la selección lineal permitiría leer una palabra simultáneamente desde varios puertos de E/S  
**usando decodificación distribuida es más fácil realizar**
  - d) **expansiones al sistema de E/S**
- 

¿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.

Usaria Profesores

- a) Sí
  - b) **No**
  - c) Depende de si el número puertos de E/S es muy elevado
  - d) Ninguna de las respuestas anteriores es correcta
- 

La predicción de saltos está relacionada con...

Usaria Profesores

- 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)
-

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

Usuaria Profesores

- a) **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**
  - b) El CPI de un cauce superescalar es siempre 1 o menor que 1
  - x c) 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
  - d) La predicción de saltos es una técnica para minimizar los riesgos de datos
- 

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...

Usuaria Profesores

- x 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 **solaparlas en un solo campo de 5 bits, ahorrando por tanto 4 bits**
  - c) **tanto 4 bits**  
tr.48  
solaparlas en un solo campo de 6 bits, ahorrando por tanto 5 bits
  - d)
- 

¿Cuál de las siguientes instrucciones convierte  $%eax = 5 * %eax$ ?

- 1) mov (%eax, %eax, 4), %eax
- 2) lea (%eax, %eax, 4), %eax

Usuaria Profesores

- ✓ a) Sólo 1
  - b) Ninguna
  - c) **Sólo 2**
  - d) Ambas 1 y 2
- 

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)

Usuaria Profesores

- a)  $(**x + 2) + 7$
  - b)  $*((x + 8)) + 28$
  - c)  $*((x + 2)) + 7$
  - d)  $*((x + 2) + 7)$
- 

En el fragmento de código

400544: e8 07 00 00 00 callq 400550 <mult2>

400549: 48 89 03                mov %rax,(%rbx)

la instrucción call suma al contador de programa la cantidad:

Usuaria Profesores

- a) 0x400544  
no, esa es la dirección de la propia instrucción CALL
  - b) 0x48  
no, eso es el codop de la siguiente instrucción MOV
  - c) 0x00000007  
sí, codop e8 lleva offset 07 00 00 00 en little-endian
  - d) 0x00400549  
no, esa es la dirección de retorno
- 

¿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

Usuaria Profesores

- a) 1 y 2  
 b) 1, 2, 3 y 4  
•  c) 1  
 d) 10
- 

La precaptación (cola de instrucciones) está relacionada con...

Usaria Profesores

- 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 de control (intenta determinar de antemano el flujo de control)  
•  d) **Los riesgos estructurales (intenta evitar el efecto de un fallo de cache)**
- 

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
```

Usaria Profesores

- a) 20  
 b) 36  
•  c) 5  
 d) 6
- 

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:

Usaria Profesores

- a) 5  
 b) 20  
•  c) 4  
 d) 0,25

=====

Para el procesador con unidad de control microprogramada estudiado en clase, Tanenbaum propone codificar 16 micro-operaciones 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

Usuaria Profesores

- - a) 24 bits
  - 20 bits**
  - b) 32-12
  - c) 12 bits
  - d) 8 bits

**x**

=====

Un archivo .o que contiene código objeto reubicable:

Usuaria Profesores

- - a) **Contiene instrucciones máquina binarias.**  
Contiene instrucciones máquina y directivas en ensamblador.
  - b) Puede ejecutarse directamente.  
Contiene las direcciones definitivas de las variables
  - c) globales.

=====

¿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?

Usuaria Profesores

- x**
  - a) movl 4(%edx, %edx), %eax
  - b) leal 4(%edx, %edx), %eax
  - c) leal (%ebx, %edx, 4), %eax
  - d) **movl (%ebx, %edx, 4), %eax**

=====

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 IA32 con enteros de 4 bytes, ¿cuál sería la dirección del argumento b relativa a %ebp, en el marco de pila de fun4arg?

Usaria Profesores

- ✓ • a) %ebp + 20
- b) %ebp + 16
- c) **%ebp + 12**
- d) %ebp + 8

=====

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”?

Usaria Profesores

### SF

Normalmente jl saltaría si (SF^OF), pero si sabemos que OF==0, equivale a SF==1, y jl se podría sustituir por js

Notar que sin optimización, gcc saltaría según la condición

- a) opuesta, saltándose en este caso la parte then: un salto incondicional a label.  
El enunciado no se implica en mencionar valor concreto del flag, pero una versión optimizada saltaría directamente con la condición directa.
- b) OF
- ✗ c) CF
- d) ZF

=====

¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar):

Usaria Profesores

- ✗ a) movb %sil, (%rax)
- b) pushq \$0xFF
- c) movsbw (%rax), %dx  
**movzfq %edx, %rax**
- d) no existe movzfq, usar "mov %edx, %eax" para conseguir el mismo efecto

=====

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

Usuaria Profesores

- a) una instrucción se ejecute en T/4 segundos
  - b) 4 instrucciones se ejecuten en 4T segundos
  - c) **cada T/4 segundos se termine de ejecutar una instrucción**
  - d) cada 4T segundos se terminen de ejecutar 4 instrucciones
- 

¿Cuál de las siguientes afirmaciones es incorrecta?

Usuaria Profesores

- a) **En lenguaje ensamblador las instrucciones se escriben en binario.**
  - 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 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.
- 

Tipos de riesgos que hemos estudiado en cauces segmentados (señalar la opción incorrecta)

Usuaria Profesores

- a) riesgos de control
  - b) riesgos de (dependencia de) datos
  - c) **riesgos de transferencia**
  - d) riesgos estructurales
- 

En una suma de dos números en complemento a dos, se produce desbordamiento cuando

Usuaria Profesores

- a) Sumamos dos negativos y el resultado es negativo.
  - b) Sumamos un número positivo y uno negativo.
  - c) **Sumamos dos positivos y el resultado es negativo o bien sumamos dos negativos y el resultado es positivo.**
  - d) Sumamos dos positivos y el resultado es positivo.
- 

Un sistema tiene una cache asociativa por conjuntos de 8-vías con 64 conjuntos y líneas

de 64B. ¿A qué conjunto le corresponde el byte con dirección 0xcafec1c0?

Usaria Profesores

- a) 7  
      b) 11  
x   c) 13  
      d) 14
- 

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?

Usuario Profesores

- a) **Los 6 bits menos significativos**  
      b) Los 6 bits más significativos  
      c) Los 4 bits menos significativos  
      d) Los 4 bits más significativos
- 

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 aproxima a:

Usaria Profesores

- a) 20  
      b) 0,25  
      c) **4**  
      d) 5
- 

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

Usaria Profesores

- a) 400549  
      eso es la dirección de salto +2
- b) 400543

**X**

- eso es la dirección de la instrucción ja  
400547
- c) eso es la dirección de salto  
**400545**
- d) sí, se cumple que RIP+2=400547
- 

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:

Usuario Profesores

- a) 4
- b) 5
- c) 50
- d) **2,5**
- 

La instrucción DI (Disable Interrupts) inhabilita las interrupciones software.

Usuario Profesores

**F**

---

Un computador con 10 bits en el bus de direcciones puede direccionar como máximo:

Usuario Profesores

- a) **1024 palabras**
- b) 65535 palabras
- c) 65536 palabras
- d) 1000 palabras
- 

La posibilidad de liberar un bus por "pre-emption" consiste en que se pueda interrumpir una transferencia en curso que esté realizando un determinado maestro cuando llega una petición del bus por parte de un maestro potencial de mayor prioridad.

Usuario Profesores

**V**

---

¿Qué medida de prestaciones es la más fiable de todas las posibles?

Usuario Profesores

- **a) ninguna de las otras respuestas es correcta**
- b) MIPS equivalentes
- c) MIPS
- d) MFLOPS

---

¿Cuál de las siguientes afirmaciones acerca de las interrupciones en el PC (modo real) es cierta?

Usuario Profesores

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

---

¿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...)

Usuario Profesores

- **a) Las caches totalmente asociativas ofrecen tasas de fallo más bajas, mientras que las de correspondencia directa tienen mejor latencia**
  - b) Las caches de correspondencia directa tienen mejor latencia y tasa de fallos
  - c) Las caches totalmente asociativas ofrecen mejor latencia, mientras que las de correspondencia directa tienen tasas de fallo más bajas
  - d) Ambas tienen en general similar latencia y tasa de fallos
- 

Con respecto a la memoria virtual, ¿cómo deben ser las páginas para evitar tener tablas

de páginas de gran tamaño?

Usuario Profesores

- - a) Medianas
    - b) **Grandes**
    - c) Pequeñas
- Las tablas de páginas tiene un tamaño fijo e independiente
- d) del tamaño de las páginas

=====

El rendimiento de los procesadores crece más rápidamente que el de las memorias DRAM.

Usuario Profesores

V

=====

¿Cuántas entradas suele tener un TLB?

Usuario Profesores

- - a) **Muchas menos que el número de página de la memoria virtual**
  - b) Tantas como número de páginas tenga la memoria física
  - c) Una única entrada correspondiente a la última página accedida
  - d) Más que el número de páginas de la memoria virtual

=====

La instrucción ADD Rn,#3 utiliza dos modos de direccionamiento: directo a registro e inmediato.

Usuario Profesores

V

=====

La relación entre tiempo de ejecución de un programa escrito en ensamblador y el mismo escrito en un lenguaje de alto nivel es más alta en RISC que en CISC.

Usuario Profesores

V

=====

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.

Usuario Profesores

F

---

En un ordenador que use DMA y disponga de un único bus de datos y un único bus de direcciones, la CPU y el controlador de DMA no controlan simultáneamente (a la vez) ambos buses.

Usuario Profesores

V

---

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:

(<sup>^</sup> simboliza potenciación)

Usuario Profesores

- a) últimos w bits (0..w-1) (los menos significativos)  
campo "palabra"
- b) **bits w...w+p-1**  
sin sentido
- c) bits w...p-1  
campo "marco"
- d) primeros bits, desde el más significativo hasta el bit p  
campo "etiqueta"

-T1.5]

¿En qué generación, dentro de la historia de los computadores digitales, aparecieron la micropogramación, la segmentación de cauce, la memoria cache, los S.O. multiusuario y la memoria virtual?

Usaria Profesores

- ✓ a) 2<sup>a</sup> generación (1955-65)
  - b) **3<sup>a</sup> generación (1965-75)**  
en la 3<sup>a</sup> generación se inventó casi todo
  - c) 4<sup>a</sup> generación (1975-85)  
esas innovaciones se repartieron a lo largo de varias generaciones, no sólo una
-

Una caché totalmente asociativa puede tener dos vías.

Usuario Profesores

F

=====

El tamaño del registro de salida de una memoria asociativa de n palabras y m bits/palabra es:

Usuario Profesores

- a)  $n*m$  bits
- b) n bits
- c)  $n+m$  bits
- d) **m bits**

=====

Se dispone de un computador cuyo tiempo medio de acceso al sistema de memoria cache y memoria principal es de 18 ns. Si la tasa de fallo de la cache es de 0,2 y el tiempo de acceso a la memoria principal es 50 ns. ¿Cuál es el tiempo de acceso a la cache?

Usuario Profesores

- a) 24,4 ns
- b) 10 ns
- c) **8 ns**
- d) 6 ns

=====

Sobre la segmentación:

Usuario Profesores

Es una técnica para comenzar simultáneamente la ejecución

- a) de varias instrucciones con el fin de reducir el tiempo de ejecución.
- b) Un procesador superescalar no puede estar segmentado.
- c) **Existen limitaciones al rendimiento provocadas por las instrucciones de salto y por las dependencias de datos.**  
La frecuencia de reloj viene impuesta por la etapa más
- d) corta.

=====

¿Cuál de las siguientes afirmaciones sobre la memoria DRAM es \*incorrecta\*?

Usuario Profesores

- a) Cada celda de memoria está compuesta por un transistor y un condensador y almacena un bit de información.

- Un transistor en cada celda permite o no permite circular la corriente eléctrica a través de él. Cuando el transistor no deja
- b) 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.

**El principio de funcionamiento de los circuitos**

- **c) electrónicos de la memoria DRAM consiste en cargar o descargar un transistor.**
- no, un condensador**
- Los bits de memoria se organizan dentro del circuito
- d) integrado en forma de matriz de celdas de bit, en la que se pueden diferenciar filas y columnas.

=====

¿Cuántas líneas de dirección son necesarias en un memoria RAM dinámica de 256 K palabras? ¿Y en una estática?

Usuario Profesores

- a) 18/18
- b) 9/9
- **c) 9/18**
- d) 18/9

=====

¿Qué tipo de localidad de las referencias a memoria se define como: "si se referencia un elemento, volverá a ser referenciado pronto"?

Usuario Profesores

- a) Localidad espacial
- b) Localidad lógica
- **c) Localidad temporal**
- d) Ninguna de las respuestas anteriores es correcta

=====

La cache con correspondencia directa se puede considerar como un caso límite de la asociativa por conjuntos, en donde...

Usuario Profesores

- **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

---

[T1.2]

¿En qué pareja de registros están la dirección de memoria y el dato que se leerá o escribirá en memoria?

[T1.2ConBas]

[E20Ene] teo.16 - inspirada en [E12Feb] teo.18 - [E12Sep] teo.13

Usaria Profesores

- ✓      •
    - a) pc y mar
    - b) ir y pc
    - c) **mar y mdr/mbr**
    - d) mdr/mbr y pc
- 

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

Usuario Profesores

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

A medida que aumenta el tamaño de página en un sistema de memoria virtual, ¿qué ocurre con el tamaño de las tablas de páginas?

Usuario Profesores

- - a) Aumenta
    - b) **Disminuye**
    - c) Permanece constante
    - d) Ninguna de las anteriores
- 

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

Usuario Profesores

- - a) SLIM
    - b) SIMM
    - c) **SODIMM**
    - d) MIN
-

En una arquitectura de acumulador, la instrucción LOAD X:

Usaria Profesores

transfiere el contenido del registro X a la memoria  
no existe registro X en una máquina acumulador pura  
el argumento de LOAD es una posición de memoria X

- a) el 6502/6510 sí tenía, además de acumulador A, índices X e Y, y correspondientemente tenía instrucciones LDA, LDX, LDY

[https://es.wikipedia.org/wiki/MOS\\_6502](https://es.wikipedia.org/wiki/MOS_6502)

<http://www.6502.org/tutorials/6502opcodes.html>

**transfiere el contenido de la posición de memoria X**

- b) **al acumulador**  
c) suma M(X) al acumulador  
d) sería ADD X

transfiere el contenido del acumulador a la posición de memoria X

---

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

Usaria Profesores

- a) Los circuitos integrados  
• b) **Los transistores**  
c) La gran integración de los circuitos (VLSI)  
d) Las válvulas

---

En el direccionamiento indirecto a través de registro, la dirección efectiva...

Usaria Profesores

- a) **se encuentra en un registro general del procesador.**  
**indirecto (normal, a través de registro)**  
b) se calcula como la suma del contenido de dos registros.  
sería "indexado a base"  
se encuentra en el registro de instrucción.  
c) directo con dirección "corta" (no absoluta)? tan corta que cabe en codop...  
direcciónamiento a "página 0"?  
d) se encuentra en una dirección de memoria.  
sería indirecto "a través de memoria"

Los computadores con estructura Von Neumann utilizan el modelo de programa externo, frente a programa cableado y almacenado.

**VERDADERO**

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

Usuaria Profesores

- a) 0x00 en 0x0804913c y 0xe0 en 0x08049143
- b) 0x1f en 0x0804913c y 0xe0 en 0x08049140
- c) **0xe0 en 0x0804913c y 0x1f en 0x08049140**
- Todas las respuestas anteriores son incorrectas
- d)

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...

Usuaria Profesores

- 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
- **Todas las respuestas anteriores son incorrectas**

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 (más lejanos al procesador). Para ello hay varias políticas:

Usuario Profesores

**Post-escritura: se retrasa la actualización en los niveles**

- a) **inferiores hasta que el bloque modificado tenga que ser reemplazado**
- b) Escritura indirecta: si se modifica una palabra,

- inmediatamente se modifican los niveles superiores
- c) Las respuestas a y b son ciertas
  - d) Las respuestas a y b son falsas
- 

Elección única

[T1.1]

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

Usuaria Profesores

- ✓      •      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
- 

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:

Usuaria Profesores

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

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

```
int elem (int A[5][3], size_t, size_t j);
```

podria ser

Usuaria Profesores

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

~~~~~

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 contenido de la dirección de memoria 0x4
- b) Es el valor inmediato 1
- c) Es el valor inmediato 4**
- d) Es el contenido de la dirección de memoria 0x1

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

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

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

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

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
- 

En el fragmento de programa siguiente:

66b: e8 00 00 00 00 callq 670 <nxt>

670: 58            pop %rax

¿Qué valor termina almacenado en %rax?

- a) 0xe8000000
  - b) 0x66b
  - c) **0x670**
  - d) 0x58
- 

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, 0x00, 0x00, 0x00, 0x00, 0x02. Posteriormente se lee (carga) %rax desde (a partir de) la dirección N. ¿Cómo es %rax entonces, considerándolo con signo?

Usuaria Profesores

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

El procesador utiliza el puntero de pila...

Usuaria Profesores

- a) En todo tipo de instrucciones de saltos, incluyendo llamadas y retornos a subrutinas
  - b) En todas las instrucciones que tengan al menos dos accesos a memoria
  - c) **En las instrucciones de llamadas y retornos de subrutinas**
  - d) En todas las instrucciones
- 

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:

Usuaria Profesores

- a) 0x5fa
  - b) **-0x76**
  - c) 0xffffffff8a
  - d) 0x76
- 

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 xxxxxxx  
400431: 5d pop %rbp

Usuario Profesores

- a) 40043d
  - b) 400525
  - c) 400431
  - d) **400425**
- 

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?

Usuario Profesores

- a) 0x5fa
  - b) **0x670**
  - c) 0x66b
  - d) 0xffffffff8a
- 

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

Usuario Profesores

- a) dos direcciones (origen y destino)  
**dos tamaños (copia origen y copia destino)**
  - b) ¿dos tamaños distintos? ¿cuántas palabras se transfieren entonces?
  - c) si se desea producir una IRQ al terminar cuál de las dos direcciones es de E/S (si alguna lo es) en
  - d) lugar de Memoria
- 

Habiendo declarado int array={0,1,2,3}; y long long \*ptr=array; ¿cuánto vale ptr[1]?

Usuario Profesores

- a) 0x0000 0001 0000 0000
- b) **0x0000 0003 0000 0002**

- 
- c) 0x0000 0001 0002 0003
  - d) 0x0003 0002 0001 0000

---

Uno de los puntos clave de la traducción que gcc hace de una construcción switch-case de lenguaje C a lenguaje ensamblador es...

Usuario Profesores

- a) **el salto indirecto**
- b) el salto directo
- c) el salto condicional hacia atrás
- d) el salto relativo a contador de programa

---

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:

Usuario Profesores

- a)  $(4+k) / 5t$
- b)  **$5k / (4+k)$**
- c)  $4k / (5+k)$
- d)  $(5+k) / 4t$

---

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:

Usuario Profesores

- a) **-0x76**
- b) 0xffffffff8a
- c) 0x76
- d) 0x5fa

---

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 segundo nivel es de 3 microsegundos, ¿cuál sería aproximadamente el porcentaje de fallos en los accesos al primer nivel?

Usuario Profesores

- a) 90%
  - b) 99%
  - c) **10%**  
$$300\text{ns} = 6\text{ns} + F \cdot 3\text{us} \Rightarrow F = 0.098 \sim 0.10 = 10\%$$
  - d) 1%
- 

El procesador utiliza el puntero de pila...

Usuario Profesores

- a) En todas las instrucciones
  - 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 las instrucciones de llamadas y retornos de subrutinas**
- 

Las microoperaciones de la fase de captación de una instrucción:

Usuario Profesores

- a) **Son comunes para todas las instrucciones**  
Dependen de los indicadores de estado y del código de
  - b) operación de la instrucción que se encuentra en el registro de instrucción
  - c) Dependen del código de operación de la instrucción que se encuentra en el registro de instrucción
  - d) Dependen del valor del contador de programa
- 

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

Usuario Profesores

- RBP es un registro salva-invocado, por eso si es necesario
- a) hay que salvarlo antes de llamar a función  
RBP es s-invocado, ya lo salvará el propio invocado
  - b) **necesario hay que salvarlo antes de llamar a función**  
R10 es un registro salva-invocante, por eso si es  
sí, usualmente se protege moviéndolo a un salva-invocado

- RAX es un registro salva-invocante, por eso en cualquier función hay que salvarlo antes de modificarlo
- RAX es s-invocante pero no hay que salvar
- R11 es un registro salva-invocado, por eso en cualquier función hay que salvarlo antes de modificarlo
- R11 no es s-invocado
- 

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

Usaria Profesores

- R10 es un registro salva-invocante, por eso si es necesario**
- a) **hay que salvarlo antes de llamar a función**  
sí, usualmente se protege moviéndolo a un salva-invocado  
R11 es un registro salva-invocado, por eso en cualquier función
  - b) hay que salvarlo antes de modificarlo  
R11 no es s-invocado  
RAX es un registro salva-invocante, por eso en cualquier función
  - c) hay que salvarlo antes de modificarlo  
RAX es s-invocante pero no hay que salvar  
RBP es un registro salva-invocado, por eso si es necesario hay
  - d) que salvarlo antes de llamar a función  
RBP es s-invocado, ya lo salvará el propio invocado
- 

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:

Usaria Profesores

- a)  $4k / (5+k)$
  - b)  $(4+k) / 5t$
  - c)  $(5+k) / 4t$
  - d)  **$5k / (4+k)$**
- 

Escoger de entre las 4 operaciones la de mayor valor que pueda calcularse con enteros de 4B con signo sin problemas

Usaria Profesores

- a)  $300.000.000 + 300.000.000$   
seiscientos millones... sin problema
- b)  $100.000.000 + 100.000.000$

- doscientos millones... sin problema  
 $3.000.000.000 + 3.000.000.000$
- c) seis mil millones no cabe. Ni siquiera se puede representar correctamente tres mil millones
  - d)  **$1.000.000.000 + 1.000.000.000$**   
**dos mil millones cabe (y un poco más también)**
- 

Dada una función que devuelve la suma de 8 enteros en x86-64, ¿cuál de las siguientes instrucciones suma el 7º argumento?

Usaria Profesores

- a) add -0x4(%rsp), %eax
  - b) add -0x8(%rsp), %eax
  - c) **add 0x8(%rsp), %eax**
  - d) add 0x4(%rsp), %eax
- 

¿Cuál de las siguientes afirmaciones es \*incorrecta\*?

Usaria Profesores

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

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 salidas de la unidad de control.

Usaria Profesores

V

---

Para traducir el código de operación de las instrucciones máquina a dirección de comienzo en la memoria de control del microprograma correspondiente se puede utilizar

una PLA.

Usaria Profesores

V

---

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

Usaria Profesores

V

---

Una unidad de control microprogramada se denomina "con secuenciamiento de microinstrucciones explícito" según tenga o no tenga

Usaria Profesores

microcódigo de decodificación que analice el codop bit a bit

- a) de izquierda a derecha
    - con ROM traductora (goto f(IR)) no haría falta ese análisis
    - ROM/PLA para traducir el codop en dirección de inicio de microprograma (goto f(IR))
  - b) todos los diseños implícitos en las transparencias usan ROM traducción, pero también se podría añadir ROM al explícito
    - un multiplexor para seleccionar la fuente de la dirección de la memoria de control
  - c) todos los diseños en las transparencias tienen un MUX en la dirección de la memoria de control (o en el micro-PC)
    - micro-contador de programa atacando a las líneas de dirección de la memoria de control**
  - d) **dirección de la memoria de control**
    - si y sólo si tiene micro-PC, sería implícito
- 

El secuenciamiento de una unidad de control microprogramada debe ser implícito.

Usaria Profesores

F

---

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:

Usaria Profesores

- a) 20
  - b) 1,2
  - c) **16,67**
  - d) 1,667
- 

¿Cuál de las siguientes funciones es una tarea propia de la unidad de control en la CPU?

Usaria Profesores

- a) realizar operaciones lógicas
  - b) almacenar datos del programa
  - c) almacenar instrucciones del programa  
**decodificar las instrucciones del**
  - d) **programa**
- 

El área de chip utilizada por la unidad de control es menor en un CISC que en un RISC, al ser microprogramada en el primero.

Usaria Profesores

F

---

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

Usaria Profesores

- a) **Salta a una dirección de memoria de control que depende de la instrucción máquina actual.**
- b) Permite saltar a la dirección de memoria de control del principio de un microbucle.
- c) Realiza una llamada a una microsubrutina.
- d) Se utiliza para realizar un microsalto condicional en función del registro de estado.

---

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.

Usaria Profesores

V

---

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...

Usaria Profesores

**codificarlas con 4 bits, y sobraría un código que**

- a) **quedaría sin uso**  
tr.47 dibujo superior
  - 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
- 

Al diseñar el formato de instrucción:

Usaria Profesores

- a) hay que indicar explícitamente todos los operandos y destinos.  
**se suele omitir el campo que indica la siguiente instrucción (la siguiente a ejecutar es la siguiente en memoria, salvo en caso de salto).**
  - b) **instrucción (la siguiente a ejecutar es la siguiente en memoria, salvo en caso de salto).**
  - c) sólo hay que saber el tipo de operación a realizar y los operandos necesarios.
  - d) el número de formatos de instrucción diferentes no afecta a la complejidad de la UC.
-

¿Cuál de las siguientes afirmaciones sobre compilación C->ASM es falsa?

Usaria Profesores

- Puede que el compilador elimine por optimización
- a) construcciones C enteras (como un bucle for completo), si se conoce el resultado en tiempo de compilación
  - b) Puede suceder que varias sentencias C se traduzcan por una única instrucción ASM
- Puede que el compilador altere por optimización el orden**
- c) **de los parámetros pasados en una llamada a función, si el marco de pila resultante es más eficiente**
  - d) Puede que el compilador altere el orden del código C, apareciendo antes la traducción de sentencias C posteriores
- 

Los procesadores SPARC utilizan ventanas de registros solapadas.

Usaria Profesores

V

---

El sufijo l de la instrucción movl significa:

Usaria Profesores

- a) Que la instrucción afecta a la parte de 16 bits más a la izquierda de los operandos (left word).
  - b) **Que la instrucción trabaja con operandos de 32 bits (long word).**
  - c) Que la instrucción usa ordenación de bytes little-endian en lugar de big-endian.
  - d) Que la instrucción afecta a los 16 bits menos significativos de los operandos (low word).
- 

La instrucción movz1q %eax, %rax

Usaria Profesores

- a) **no existe, se debe usar mov %eax, %eax**  
ver pie de página tr.44 y T2.2.1 tr.11
- b) copia en %rax el valor de %eax si el indicador de cero está activado

- c) pone a 0 el registro %rax  
copia en %rax el valor sin signo almacenado en %eax,
  - d) rellenando con ceros
- 

Sobre el direccionamiento relativo al contador de programa:

Usaria Profesores

- a) Favorece la implementación de código reubicable.
  - b) Su uso en los saltos 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.**
- 

Los modos de direccionamiento de una instrucción se especifican en ensamblador mediante las directivas.

Usaria Profesores

**F**

---

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

Usaria Profesores

- a) registros PC, SP e IR.
  - b) registros de estado.
  - **c) registros de propósito general.**
  - **d) Todas las respuestas son ciertas.**
- 

En una resta de dos números en complemento a dos, se produce desbordamiento cuando...

Usaria Profesores

- a) los dos operandos son negativos y el resultado es positivo.
  - b) los dos operandos son positivos y el resultado es negativo.
  - c) Las dos respuestas a y b son correctas.
  - **d) Ninguna de las anteriores es correcta.**
-

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;
};
```

Usaria Profesores

- **a) 60 bytes**  
 $28 + 4 + 28 = 60$
  - b) 64 bytes
  - c) 24 bytes
  - d) 84 bytes
- 

Con el repertorio IA32, para sumar %eax y %ebx dejando el resultado en %ecx se podría hacer lo siguiente:

Usaria Profesores

- a) lea %eax, %ebx, %ecx
  - b) lea %ecx, [%eax, %ebx]
  - **c) lea (%eax, %ebx, 1), %ecx**
  - d) lea %ecx, %ebx, %eax
- 

El 8086 puede funcionar en modo supervisor y en modo usuario.

Usaria Profesores

**F**

---

Una instrucción de salto si igual tiene que comprobar el valor de:

Usaria Profesores

- a) los bits de signo y desbordamiento
- b) el bit de acarreo
- **c) el bit de cero**

- d) el bit de signo
- 

De las siguientes parejas de instrucciones, ¿cuál utiliza únicamente direccionamiento implícito?

Usuaria Profesores

- - a) mul, div
    - b) sti, popf**
    - c) push, call
    - d) jnz, cmp
- 

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; };
```

Usuaria Profesores

- a) 18 B  
         sería 16 B en IA32, sólo 1 B relleno entre c y s
    - b) 20 B
    - c) 22 B
    - 24 B**
  - **d) 4 B relleno entre i y d, 1B entre c y s, 4B relleno tras s.**
- 

El direccionamiento relativo a registro base utiliza dos registros, uno sirve de base y el otro de desplazamiento.

Usuaria Profesores

**F**

---

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

Usuaria Profesores

- a) a través de variables globales
  - b) a través de los registros
  - c) a través de la pila
  - todas las anteriores son**
  - **d) ciertas**
- 

Las directivas de ensamblador controlan el listado del programa, la base de numeración, permiten definir símbolos, etc, pero en ningún caso influyen sobre el código objeto generado.

Usuaria Profesores

**F**

---

Según Ashley y Flynn, las instrucciones de bifurcación son las más frecuentemente utilizadas en un programa, debido al gran número de decisiones necesarias.

Usuaria Profesores

**F**

Uno de los objetivos del uso de direccionamiento relativo al contador de programa en los saltos es posibilitar la escritura de programas relocalizables.

Usuaria Profesores

**V**

---

GCC/Linux IA32 resuelve el ajuste de marco de pila al comenzar un procedimiento mediante las instrucciones:

Usuaria Profesores

- a) movl %ebp, %esp popl %ebp
- b) movl %esp, %ebp popl %esp

- c) **pushl %ebp movl %esp,%ebp**  
d) pushl %esp movl %ebp, %esp
- 

Las arquitecturas RISC son del tipo registro-registro.

Usaria Profesores

**V**

---

El resultado de evaluar la expresión en notación polaca inversa "8 2 5 \* +" es 21.

Usaria Profesores

**F**

---

Para desplazar %eax a la derecha un número variable de posiciones  $\leq 32$ , indicado en %ebx, se puede hacer

Usaria Profesores

- a) sar %bl, %eax  
no existe
  - b) No se puede, sar sólo admite un número fijo de posiciones  
(debe ser un valor inmediato)
  - c) sar %ebx, %eax  
no existe  
**mov %ebx, %ecx**
  - d) **sar %cl, %eax**  
ver problema Hallaron 3.8 y sección 3.5.3
- 

Respecto a los registros enteros en arquitectura IA32 de 32bits (x86)

Usaria Profesores

- 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) No hay distintos tamaños, son sólo registros de 32 bits, como

- corresponde a dicha arquitectura
- 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)
- 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) sí, los últimos 4 no tienen versión de 8 bits, p.ej. sil o spl**
- 

La codificación Huffman utiliza menos bits para las instrucciones menos frecuentes y más bits para las instrucciones más utilizadas.

Usuaria Profesores

**F**

---

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)?

Usuaria Profesores

- a) 4
  - b) 3
  - **c) 2**
  - d) 1
- 

En las arquitecturas del tipo registro-registro se usan instrucciones LOAD y STORE para transferir información entre registros y memoria principal.

Usuaria Profesores

**V**

---

Si queremos almacenar la palabra de 16 bits 8965h en memoria según "big-endian", quedará almacenada a partir de la posición 1000h como:

Usuaria Profesores

- - a) en el byte 1000h se guarda A6h y en el 1001h 91h
    - b) en el byte 1000h se guarda 89h y en el 1001h 65h**
    - c) en el byte 1000h se guarda 91h y en el 1001h A6h
    - d) en el byte 1000h se guarda 65h y en el 1001h 89h
- 

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.

Usuaria Profesores

V

---

En una matriz declarada como "int a[n][n];" en lenguaje C...

Usuaria Profesores

los n elementos de una columna se almacenan en memoria

- a) de manera contigua
  - row-major
  - **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
- 

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

Usuaria Profesores

una zona bajo (RSP) (adonde apunta RSP) que una

- **a) función puede usar sin reservarla, pero sólo si no llama a ninguna otra función mientras la usa**
- una zona bajo (RBP) (adonde apunta RBP) que una función
- b) puede usar sin reservarla, pero sólo si no llama a ninguna otra función mientras la usa
- una zona de pila en donde pueden escribir las funciones
- c) invocadas, pero teniendo en cuenta que los valores escritos pueden verse alterados si se produce una interrupción

- una zona de pila en donde no deben escribir las funciones invocadas (porque si se produjera una interrupción, el manejador de interrupción sobreescribiría los valores escritos en pila)
- 

La instrucción INC [3] no tiene sentido.

Usaria Profesores

F

---

Si el registro EAX contiene X, la sentencia en C

x &= 0x1;

se traducirá a ensamblador como:

Usaria Profesores

- a) sarl %eax
  - b) andl \$1, %eax
  - c) shr %eax
  - d) orl \$0x1, %eax
- 

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

Usaria Profesores

F

---

¿Cuál de las siguientes instrucciones \*NO\* es errónea?

Usaria Profesores

- movb \$0xF, (%rax)**
- a) se puede mover un byte adonde apunte RAX
  - b) movl %rcx, (%rsp)  
RCX no es long 4B
  - c) movq %rdx, \$0x123  
no se puede mover a inmediato

- d) movw (%rbx), 4(%rsp)  
no se puede mover M-M
- 

La directiva de ensamblador "DW 0 DUP(100)" reserva 100 palabras inicializadas a cero.

Usaria Profesores

F

---

¿Cuál de las siguientes parejas de nemotécnicos de ensamblador corresponden a la misma instrucción máquina?

Usaria Profesores

- a) CMP (comparar), SUB (restar).
  - b) JC (saltar si acarreo), JL (saltar si menor, para números con signo).
  - c) SAR (desplazamiento aritmético a la derecha) / SHR (desplazamiento lógico a la derecha).
  - d) **JZ (saltar si cero), JE (saltar si igual).**
- 

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.

Usaria Profesores

V

En las arquitecturas de registros de propósito general suele ser menor el tráfico entre procesador y memoria que en las de acumulador.

Usaria Profesores

V

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.

Usaria Profesores

V

---

Las instrucciones que asignan la dirección 0x78e00 a un puntero situado en la posición 0xff00 son:

Usaria Profesores

- a) mov 0x78e00,%eax
- b) mov %eax,0xff00
- c) mov \$0x78e00,%eax
- c) **mov \$0x78e00,%eax**
- d) lea %eax,0xff00

---

Una CPU cuya instrucción CALL salva la dirección de retorno al principio de la subrutina permitiría llamadas anidadas, siempre que ninguna fuera reentrante, pero no permitiría llamadas recursivas.

Usaria Profesores

V

---

¿Cuál de las siguientes instrucciones convierte %rax = 5 \* %rax?

1) mov 4(%rax, %rax), %rax

2) lea 4(%rax, %rax), %rax

Usaria Profesores

- a) Sólo la 1
  - b) Sólo la 2
  - c) Ambas, la 1 y la 2
  - **Ninguna de las dos**
  - d)
- 

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

Usaria Profesores

F

---

La codificación "Huffman" del código de operación...

Usaria Profesores

- a) es la más utilizada.
  - b) emplea un campo de tamaño fijo para el código de operación de todas las instrucciones.
  - c) permite una decodificación muy sencilla de la instrucción.
  - **d) permite obtener un tamaño promedio del código de operación mínimo.**
- 

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++)

Usaria Profesores

- a) 0x80
  - b) 0x13080
  - **c) 0x12F80**
  - d) 0x13000
- 

¿Cuál de las siguientes instrucciones x86 se puede usar para sumar dos registros y

guardar el resultado sin sobrescribir ninguno de los registros originales?

Usaria Profesores

- a) mov
  - b) **lea**
  - c) add
  - d) Ninguna de ellas
- 

Sobre el direccionamiento relativo a contador de programa:

Usaria Profesores

- 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.**
- 

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?

Usaria Profesores

- a) 28

- b) 32
  - c) 40
  - d) Ninguno de los anteriores
- 

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.

Usaria Profesores

V

---

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:

Usaria Profesores

- a) var (el nombre de la variable representa su posición de memoria)
  - b) k(%ebp), siendo k un número constante positivo relativamente pequeño
  - c) k(%esp), siendo k un número constante positivo relativamente pequeño
  - d) **-k(%ebp), siendo k un número constante positivo relativamente pequeño**
- 

Las instrucciones JB y JNAE del 8086 provocan un salto si CF = 1.

Usaria Profesores

V

---

Un ordenador que únicamente emplee direccionamiento directo o absoluto no permitirá escribir programas relocalizables.

Usaria Profesores

V

---

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

Usaria Profesores

- - a) Algunos registros para pasar argumentos, otros salva-invocante, otros salva-invocado, dos especiales
    - b) 3 registros salva-invocante, 3 registros salva-invocado, y 2 especiales**
    - c) Algunos registros salva-invocante, otros salva-invocado, uno especial
    - 8 registros salva-invocante, 6 registros salva-invocado, y 2
    - d) especiales
- 

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

Usaria Profesores

- ```
loop:  
a)    body;  
      if (test) goto loop;  
      if (test) goto true;  
      goto done;  
b)  true:  
      body;  
      done:  
      if (!test) goto done;  
      loop:  
c)        body;  
        if (test) goto  
        loop;  
      done:  
d)  loop:  
      if (test) goto done;  
      body;
```
-

```
    goto loop;  
done:
```

Una máquina que almacene el número 4321h a partir de la dirección 0 como 21h, 43h utiliza el sistema big-endian.

Usaria Profesores

F

Si AX = 0xFA50 y ejecutamos AND \$0xFF, %AX

Usaria Profesores

- a) El registro AH se pone a FF
 - b) El registro AL se pone a 0
 - c) **El registro AH se pone a 0**
 - d) El registro AL se pone a FF
-

La instrucción CMP tiene que realizar una resta.

Usaria Profesores

V

En las arquitecturas RISC hay pocas instrucciones, pero con muchos modos de direccionamiento.

Usaria Profesores

F

Un objeto (instrucción o dato) de tamaño s bytes está alineado (en desplazamiento de s bytes) si su dirección A cumple que $A \bmod s = 0$, y requiere un solo acceso a memoria para leerlo o escribirlo.

Usaria Profesores

V

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.

Usuaria Profesores

V

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 != 0.

Usuaria Profesores

V

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:

Usuaria Profesores

- a) xorl %ebx,%eax
 - b) testl %eax,%ebx
 - c) andl %ebx,%eax
 - d) **xorl %eax,%ebx**
-

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

Usuaria Profesores

- a) mov puede usarse para copiar un registro a otro, mientras que lea no
 - b) **mov referencia (accede) la posición indicada, mientras que lea no lo hace**
 - c) lea puede usarse para copiar un registro a otro, mientras que mov no
 - d) lea referencia (accede) la posición indicada, mientras que mov no lo hace
-

Una instrucción de llamada a

subrutina es más compleja en general que una de salto.

Usaria Profesores

V

El programa encuadernador (LINK) resuelve todas las referencias declaradas globales en los ficheros fuente.

Usaria Profesores

V

En IA32, ¿cuál de los siguientes fragmentos de programa tiene un efecto sobre los flags distinto al resto?

Usaria Profesores

 mov \$-1, %edi

- a) add \$0, %edi

 add afecta

 sub %edi, %edi

- b) adc \$0xFFFFFFFF, %edi

 adc afecta

mov \$-1, %edi

- c) **mov \$-1, %edi**

mov no afecta a los flags

 mov \$0, %edi

- d) sub \$1, %edi

 sub afecta

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

Usaria Profesores

F

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.

Usuaria Profesores

F

//KKKKKKK

Consideré la siguiente declaración de una estructura

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

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

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

¿Cuál es el código C de la función f?

1. r->p = (int *)(long)(r->a[r->i] + r->a[r->j]);

2. $r \rightarrow a[r \rightarrow i] = r \rightarrow a[r \rightarrow j];$
 3. $r \rightarrow p = \&(r \rightarrow a[r \rightarrow i + r \rightarrow j]);$
 4. $r \rightarrow a[r \rightarrow i] = r \rightarrow j;$
-

En x86-64 es responsabilidad del procedimiento llamada (callee) salvaguardar, entre otros, los registro:

Usuaria Profesores

- a) %rax, %rbx, %rcx, %rdx
 - b) %rax, %rcx, %rdx
 - c) **%rbx, %rbp**
%rbx, %rsi, %rdi
 - d)
-

¿Cuál de las siguientes instrucciones es incorrecta en ensamblador GNU/as x86-64?

Usuaria Profesores

- a) lea 0x10(%rsp); %rbp
 - b) pop %rbp
 - c) **pop %rip**
mov (%rsp), %rbp
 - d)
-

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

Usuaria Profesores

- a) mov (%rbp), %rax
retq
 - b) leal 8(%rdi), %eax
retq
 - c) **mov %edi, %eax**
retq
movq 8(%rsp), %rax
 - d) retq
-

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

1. Debido al pequeño número de operaciones simples, la sección de control de un procesador RISC puede ser cableada en lugar de microprogramada.
 2. Es posible realizar el diseño físico de una unidad de control cableada de manera semiautomática
 3. Cuanto más horizontal es la microprogramación, más largas son las microinstrucciones.
 4. **El programador de lenguaje ensamblador necesita conocer la microarquitectura del ordenador**
-

Si la estructura struct a ocupa un espacio de 26 bytes en memoria, ¿ cuántos bytes ocupa la siguiente estructura struct b cuando se construye en 64 bits?

```
struct b{  
    struct a a1;  
    int i;  
    struct a a2;  
};
```

Usuaria Profesores

- a) 58
- b) 24
- c) **60**
- 64
- d)

/* 60 porq 26(+2){struct a1} +4{int} +26 (+2){struct a2} seria 58, hay que rellenar con 2 mas, para que sea múltiplo de 4

los +2 son de relleno para conseguir múltiplos de 4 */

¿Cuál de las siguientes afirmaciones sobre size.cc es cierta?

1. La diferencia de velocidades entre L2 y L3 es mayor que la diferencia de velocidades entre L1 y L2.
2. **La gráfica tiene escalones hacia arriba porque en cada punto del eje \(\mathbf{X}\) 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.**
3. 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 tramos

horizontal.

4. Si continuáramos multiplicando por 2 el tamaño del vector en el eje $\backslash(X)\backslash$ obteniendo más puntos de la gráfica, está continuaría horizontal para cualquier valor mas alla de 64MB.
-

En el programa size.cc, 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:

1. La cache L1 es como muchas veces más rápida que la cache L2.
 - 2. La cache L1, es al menos tres veces más rápida que la cache L2.**
 3. La cache L2 es al menos el doble de rápida que la memoria pral.
 4. La cache L2 es como mucho el doble de rápida que la memoria pral.
-

Respecto a salvaguardar los registros de la CPU al inicio de una rutina de servicios de interruptor (ISR)

- 1. Se deben guardar los registros que se modifiquen en la propia ISR. Esto es posible hacerlo porque el propio programador de la ISR conoce que registros va a modificar.**
 2. Se deben guardar los registros salva-invocado, los registros salva-invocante ya los guarda el programa interrumpido.
 3. Se deben guardar todos los registros, para restaurarlos a la salida y asi garantizar que el programa interrumpido no sufre ninguna modificación (salvo el inevitable retraso temporal) debido a la interrupción.
 4. No es necesario salvar ninguno más, si el contador de programa y los flags ya los salva la propia CPU como parte del mecanismo de interrupción
-

=====

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

Usaria Profesores

- - a) 12
 - b) **6**
 - c) 11
 - 10
 - d)
-

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

Usaria Profesores

- - a) 64
 - b) **4**
 - c) 16
 - d) Ninguno
-

¿Cuál de los siguientes es un registro de un controlador de DMA?

Usuario Profesores

- - a) IR (Instruction Register)
 - b) PC (Program Counter)
 - c) **WC (Word Count)**
 - d) SP (Stack Pointer)

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

Usuario Profesores

- - a) **sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo**
 - no cubre el caso de E/S programada para dispositivos "sin estado" (ej: display 7 segmentos en las diapositivas)
 - b) sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S
 - 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
 - c) cubre los casos de IRQ de dispositivo e IRQ de DMA
 - 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) originalmente redactado "se suele/puede hacer con E/S progr y con E/S IRQ" - redacción actual más clara, cubre los casos de bucle de espera ocupada y varios dispositivos conectados a la misma IRQ
-

Mediante el software adecuado se puede hacer que las salidas a una impresora se escriban en un buffer en disco. Tendríamos entonces un dispositivo de salida lógico.

Verdadero

De todas las siguientes estructuras de bus, ¿cuál es la más propensa a quedar inutilizada si falla un componente cualquiera del sistema?

Usuario Profesores

- - a) Bus tipo anillo
 - b) Bus único avanzado
 - c) Bus único
 - d) Bus tipo estrella
-

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. **Verdadero**

Se llama "Broadcast" a la escritura simultánea en varios esclavos.

FALSO

Con cuatro controladores de interrupciones 8259 se pueden manejar 32 niveles de prioridad.

Falso

En el 8086 varias patillas son comunes al bus de direcciones y al de datos.

Verdadero

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

¿De cuántos canales de E/S independientes dispone el controlador de acceso directo a memoria 8237?

Usuario Profesores

- a) 2
 - b) 8
 - c) 16
 - d) 4
-

Uno de los casos más comunes en los que suele utilizar DMA es para leer datos del teclado de forma rápida.

Falso

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?

Usuario Profesores

- a) 50%
 - b) 90%
 - c) 10%
 - d) 1%
-

Se desea utilizar interrupciones vectorizadas para gestionar las interrupciones procedentes de 16 dispositivos. Si tenemos una CPU con un bus de datos de 8 bits, ¿es posible?

Usuario Profesores

- a) Sí, empleando un decodificador
 - b) **Sí, empleando un codificador**
 - c) Sí, empleando un multiplexor
 - d) Ninguna de las respuestas anteriores es cierta
-

Para determinar la causa de una interrupción se pueden usar las siguientes técnicas: (señalar la respuesta falsa)

Usuario Profesores

- a) consulta de estado, o polling
 - b) línea de reconocimiento INTA#
 - c) interrupciones vectorizadas
 - d) múltiples líneas de interrupción INT1#, INT2#...
-

¿A qué tipo de interrupciones pertenecen las condiciones de tiempo real y los fallos hardware?

Usuario Profesores

- - a) Enmascarables
 - b) No enmascarables**
 - c) Ninguna de las otras respuestas es correcta
 - d) Puede ser tanto enmascarables como no enmascarables
-

La CPU guarda automáticamente el contexto del programa que ejecuta cuando le llega y atiende una petición de DMA.

Falso

[T5.3]

Con una línea de interrupción organizada en colector abierto:

Usuario Profesores

- a) se pueden conectar varios dispositivos de manera sencilla sin necesidad de circuitos combinacionales
 - b) a la patilla del circuito integrado que genera esa línea hay que conectarle un transistor
 - c) un dispositivo conectado a ella puede solicitar la interrupción poniendo a nivel bajo la línea
 - d) a) y c) son ciertas**
-

¿Cuántos puertos de E/S permite manejar la interfaz de periféricos programable 8255?

Usuario Profesores

- - a) 3 puertos de 12 bits
 - b) 2 puertos de 16 bits y 1 puerto de 8 bits
 - c) 2 puertos de 8 bits y 2 puertos de 4 bits**
 - d) Todas las respuestas anteriores son falsas
-

¿Cuál de las siguientes afirmaciones es cierta?

Usuario Profesores

- a) Ninguna de las otras respuestas es cierta
 - Los estándares ATA/IDE, SCSI y Firewire (IEEE 1394) definen
 - b) 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**
-

La interfaz de periféricos programable 8255 tiene tres modos de funcionamiento: programado, por interrupciones, y por DMA.

Falso

La red Ethernet es un ejemplo de conexión multipunto.

Verdadero

Respecto a la E/S programada...

Usuario Profesores

- 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**
-

La red Ethernet es un ejemplo de conexión multipunto en anillo (cada ordenador recibe el testigo y mensajes del ordenador precedente y los manda al siguiente).

Falso

Cuando una CPU divide por cero se genera una interrupción, encargada de dejar "colgada" la máquina.

Falso

El ancho de banda máximo del bus MCA no llega a 1 MByte/s.

Falso

La E/S programada:

Usuario Profesores

- a) Mejora las prestaciones globales del sistema respecto a la E/S por interrupciones porque la CPU tiene el control de toda la operación.

- Empeora las prestaciones globales del sistema respecto a la E/S por interrupciones porque una
- b) 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.
 - c) 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.
- **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.**
-

La consulta de estado que se puede llevar a cabo en una operación de salida mediante E/S programada sirve para...

Usuario Profesores

- 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
-

Una interrupción enmascarable no puede ser deshabilitada con una instrucción de salida sobre el registro de máscara de interrupciones del controlador de interrupciones.

Falso

Un computador que utilice únicamente controladores programables 8237 para realizar el DMA no puede realizar más de cuatro transferencias por DMA concurrentes.

Falso

El ancho de banda máximo de los buses VESA Local Bus y PCI es mayor de 1 GB/s.

Falso

Una conexión serie punto a punto "full-duplex" permite comunicación en los dos sentidos, pero no simultáneamente.

Falso

No tiene sentido utilizar "polling" para identificar la fuente de una interrupción en un sistema con "daisy-chain".

Falso

¿Cuál de las siguientes afirmaciones sobre la E/S programada con consulta de estado es cierta?

Usuario Profesores

- a) Un programa que realice salida programada con consulta de estado no ejecutará ninguna instrucción de entrada o carga
 - b) Sólo la E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo de E/S
 - c) **Si se emplea E/S programada puede hacerse con consulta de estado o sin consulta de estado**
 - d) La escritura de un led requiere consulta de estado
-

En un sistema que utilice DMA transparente la CPU genera alguna señal de control externa que indica al controlador de DMA cuándo realizar el acceso directo a memoria.

Verdadero

¿Cuál de las siguientes funciones no corresponde a un módulo de E/S?

Usuario Profesores

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

En un controlador de DMA, el registro WC es un registro para almacenar los datos que no se utilizan.

Falso

¿Cuál de las siguientes afirmaciones es cierta?

Usuario Profesores

- 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) Ninguna de las anteriores es cierta
-

¿Cuál de las siguientes afirmaciones es *falsa*?

Usuario Profesores

- **a)** **Sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo de E/S**
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)
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)
Sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S
 - b)
 - c)
 - d)
-

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

Respecto a las políticas de arbitraje de buses...

Usuario Profesores

- a) pueden ser dinámicas o estáticas
 - b) las políticas estáticas reparten el uso del bus por turnos de forma fija entre los dispositivos activos
 - c) las políticas dinámicas permiten cambiar el maestro del bus en función de la situación en cada instante
 - **d) Todas las respuestas anteriores son ciertas**
-

Cuando a una CPU microprogramada llega una interrupción, no se interrumpe el microprograma al finalizar la ejecución de la microinstrucción en curso.

Usuario Profesores

V

[T1.1]

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

Usuario Profesores

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

El registro base de la tabla de páginas contiene la dirección de memoria a partir de la cual se encuentra la tabla de páginas de la memoria virtual.

Usuario Profesores

V

[T6.2]

¿Cuál de las siguientes afirmaciones acerca de la memoria es *FALSA*?

Usuario Profesores

- 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
-

El "handshaking" se utiliza en transferencias de datos "síncronas" entre CPU y periféricos.

Usuario Profesores

F

Alguna de las siguientes NO es una técnica de E/S de las estudiadas en clase:

Usuario Profesores

- a) E/S controlada por interrupciones
 - b) E/S programada
 - c) **E/S asíncrona**
 - d) E/S mediante Acceso Directo a Memoria
-

El entrelazado incrementa el ancho de banda ensanchando la palabra de memoria y el bus de interconexión con el procesador.

Usuario Profesores

F

El sistema de paginación del Pentium utiliza dos niveles de tablas para traducir una dirección lineal a dirección física.

Usuario Profesores

V

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

Usuario Profesores

- a) Está mal, porque EAX no se carga con ningún valor
 - b) Está mal, porque EAX no se puede usar en modo 64bit, debería ser RAX
 - c) Está mal, porque el primer argumento de una función C no se pasa en RDI
 - d) **Está bien, y pone a cero los 32 bits más significativos de RAX**
-

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

Usuario Profesores

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

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

Usuario Profesores

- a) Registro de dirección
- b) **Registro de prioridades**

-
- c) Registro de órdenes
 - d) Registro contador
-

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

Usuario Profesores

2 a) 2

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]){\n/* Se asume valor inicial C = {0,0...} */\nint i,j,k;\nfor (i=0; i<N; i++)\n    for (j=0; j<N; j++)\n        for (k=0; k<N; k++)\n            C[i][j] += A[i][k] * B[k][j];\n}
```

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 registr

Aproximadamente, ¿qué tasa de fallos se podría esperar de esta función para valores grandes de N?

- a) $\frac{1}{4}$
- b) $\frac{1}{8}$
- c) $\frac{1}{2}$**
- d) $\frac{1}{16}$

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

La diferencia de velocidades entre L2 y L3 es mayor que la diferencia de velocidades entre L1 y L2.

- a) puede que haya sistemas con esa característica, pero desde luego los que usamos nosotros en prácticas no.

La gráfica tiene escalones hacia arriba porque en cada punto del eje X

- b) 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.**

si cabe el vector en cache, los millones de accesos (siempre el mismo número de accesos) son todo aciertos por localidad temporal

Si continuáramos multiplicando por 2 el tamaño del vector en el eje X obteniendo

- c) más puntos de la gráfica, esta continuaría horizontal para cualquier valor más allá de 64 MB.

hay quien ha probado a superar la cantidad de memoria física (RAM, para lo cual hay que tener una partición de swap de tamaño conveniente) y ha observado la lentitud del disco duro, comparado con la RAM.

La gráfica tiene tramos horizontales porque el hecho de realizar la mitad de

- d) 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.

siempre se hacen los mismos millones de accesos

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) sacar un uno (1) por el stream line
- b) salir del bucle si el tamaño de línea se volviera menor o igual que 1 para algún elemento del vector
- c) volver al principio del vector cuando el índice exceda la longitud del vector
- d) **duplicar el tamaño del salto en los accesos al vector respecto a la iteración**

anterior

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...

de ambas formas sale gráfica

- a) **creciente**
- b) recorriendo una vez sale gráfica creciente
- c) con núm. fijo accesos sale gráfica creciente
- d) de ninguna de las dos formas sale gráfica creciente

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**

¿Cuál de las siguientes afirmaciones sobre las caches es *FALSA*?

- a) Un procesador actual tiene varias caches de nivel 1
- b) La cache de nivel 3 no contiene toda la memoria que maneja el programa
- c) Las direcciones a las que accede un programa no son completamente aleatorias, sino que se rigen por ciertos patrones de localidad
- d) **Casi ningún procesador actual tiene memoria cache L2**

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 observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

- a) 16/32/64 estudiantes
- b) 4/32/512 estudiantes
- c) 32/256/8192 estudiantes
- d) 8/64/2048 estudiantes**

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?

- 1. 4B
- 2. 8B**
- 3. 16B
- 4. ninguno de las anteriores

El código del programa "size" de la práctica de la cache accede al vector saltando...

- 1. byte en bytes
- 2. de 64 en 64 bytes**
- 3. de 1kb en 1 kb
- 4. de 64 kb en 64 kb

En el programa line.cc de la práctica de cache, para cada tamaño de línea (line) recorremos 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

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
-

Suponga que una memoria de acceso aleatorio (RAM) tiene 2^k palabras de 2^p bits cada una. En tal caso sería común que el registro de dirección de memoria tuviera k bits y el registro intermedio de memoria tuviera p bits.

Usuario Profesores

F

[T6.2]

Una SRAM de 1Mx4bit (4Mbit) puede venir organizada en 2048 filas, dedicando por tanto al decodificador de columnas...

Usuario Profesores

- a) 7 bits
 - b) 8 bits
 - c) 6 bits
 - 9 bits**
 - d) $1M = 2^{20} = 2^{11} \cdot 2^9 = 2048 \text{ filas} \cdot 512 \text{ columnas}$
observar que no es 1Mbit organizado en 2048 filas (en cuyo caso la respuesta sería 7bits)
-

Una operación de E/S por DMA (transferencia de un bloque o conjunto de datos) es inicializada automáticamente por el controlador de DMA, es decir, la CPU no tiene que ejecutar ninguna instrucción de inicialización.

Usuario Profesores

F

Para direccionar una memoria de 16K x 16 necesitamos un bus de direcciones de:

Usuario Profesores

- a) **14 bits**
 b) 16 bits
 c) 4 bits
 Otro valor
 d)
-

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...

Usuaria Profesores

- a) **se ahorran bits pero el funcionamiento es más lento.**
 b) no se ahorran bits y además el funcionamiento es más lento.
 c) no se ahorran bits pero el funcionamiento es más rápido.
 d) se ahorran bits y el funcionamiento es más rápido.
-

¿Cuál de las siguientes afirmaciones es verdadera?

Usuaria Profesores

- a) El registro puntero de pila es un registro de propósito general que suele contener tanto direcciones como datos.
- b) El registro de instrucción es un registro de propósito específico que contiene la dirección de la siguiente instrucción a ejecutar.
La unidad de control necesita como entrada el registro de estado para poder controlar la ejecución de las instrucciones de salto condicional.
- c)

Tema 3 tr.20

Las únicas instrucciones en las que algunas de sus fases de

- d) ejecución conllevan un acceso a memoria son las instrucciones load y store.
-

Las excepciones provocadas por errores de programas son interrupciones software.

Usuario Profesores

F

3,2 Gbits/s equivale aproximadamente a 400 MB/s.

Usuario Profesores

V

La prioridad de la línea de interrupción de un reloj de tiempo real ha de ser mayor que la prioridad de la línea de interrupción de una unidad de disquete.

Usuario Profesores

V

Suponga que una memoria de acceso aleatorio (RAM) tiene 2^k palabras de 2^p bits cada una. En tal caso sería común que el registro de dirección de memoria tuviera k bits y el registro intermedio de memoria tuviera 2^p bits.

Usuario Profesores

V

No tendrá sentido usar dos memorias cachés separadas (arquitectura Harvard) de 8 KB cada una y tasa de fallos de 5 cada una, en lugar de una caché de 16 KB unificada con una tasa de fallos del 5

Usuario Profesores

F

La codificación en bloque del código de operación utiliza menos bits para las instrucciones más frecuentes y más bits para las instrucciones menos utilizadas.

FALSO

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

Usuaria Profesores

Una instrucción de movimiento incondicional, pero sólo

- a) si el procesador es Pentium Pro/II o superior
¿movimiento INcondicional?
- b) el S.O. es de 32 bits
¿cmov con cualquier 32 bits, como por ejemplo un 386?
- c) **Un salto condicional, según la condición expresada en el código C, y otro salto incondicional podría hacerlo, si le conviene poner el cuerpo del else a continuación y el then en otra parte**
Un salto incondicional, según la condición opuesta a la
- d) del código C, y otro salto condicional
¿salto INcondicional según cuál condición?

[T2.2.2]

Al ejecutar el fragmento de código:

```
# leal válida, dir.64b truncada a 32b
leal -1(%rax), %edx
cmpl $9, %edx
ja .L2
```

se salta a .L2 si el contenido del registro %eax:

Usuaria Profesores

- a) **está fuera del intervalo [1,10]**
- x b) es menor o igual que 1
- c) está dentro del intervalo [1,10]
- d) es mayor o igual que 10

[T2.2.1]

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

Usuaria Profesores

- ✓ • a) leal 0x80(, %edx, 2), %eax
- b) movl 0x8(%edx), %eax
- c) **xorl (%edx, %ecx, 4), %eax**
- d) addl (%edx, %ecx), %eax

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

FALSO

La BIU del 8086 es la unidad de control de la EU ("datapath").

FALSO

[T2.1.4]

En un sistema Linux x86-64, ¿cuál de las siguientes variables ocupa más bytes en memoria?

Usuaria Profesores

- - a) char a[7]
 - b) int *c
 - c) short b[3]
 - d) float d

-----La instrucción

TEST_AND_SET se utiliza para implementar exclusión mutua en el acceso a un recurso compartido.

VERDADERO

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

FALSO

[T2.2.4]

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:

Usuaria Profesores

- ✓ •
 - a) está fuera del intervalo [48,57]
 - b) es mayor o igual que 48
 - c) está dentro del intervalo [48,57]

- d) es mayor o igual que 57
-

La instrucción movl %esp,%ebp

Usuaria Profesores

- a) Introduce en la pila el contenido del registro EBP.
 - b) Intercambia los contenidos de los registros ESP y EBP.
 - c) Mueve el contenido del registro ESP al registro EBP, poniendo a 0 el registro ESP.
 - d) Copia el contenido del registro ESP en el registro EBP.**
-

¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar)

Usuaria Profesores

- **a) pushb \$0xFF**
 - b) movzbl %dl, %eax
 - c) movw %dx, (%eax)
 - d) movswl (%eax), %edx
-

En las arquitecturas RISC:

Usuaria Profesores

- a) la programación resulta más simple que en los CISC.
 - b) la UC suele ser más compleja que en CISC, al ser cableada.
 - c) se usan instrucciones muy simples que se pueden segmentar.**
 - d) realmente se usan pocas instrucciones de las disponibles en el conjunto de instrucciones.
-

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

VERDADERO

ADD R0,R0,R0 es una instrucción de no-operación.

VERDADERO

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

El modo de direccionamiento a pila es un caso particular del direccionamiento inmediato.

FALSO

PREGUNTAS EC PRÁCTICAS

Normas:

- Arial 11 y la correcta se señala en negrita.
- No valen fotos, ponerla en texto.
- Comprobad que la pregunta ya no esté puesta.
- No entrar con la cuenta go.ugr y si se entra que al cambiar de cuenta se quite también el archivo de la zona de compartidos.
- Cuidado con pisar preguntas de otra gente, escribid y ya voy entrando y ordeno.

1. [P3.2] 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:

- a) produce el resultado correcto
- b) 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.

- c) no es correcta; fallaría por ejemplo 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

2. [P2T] ¿Cuál de los siguientes contenidos no está incluido en un fichero ELF ejecutable?

- a) tabla de símbolos
 - incluso símbolos de depuración con -g
- b) código máquina
 - sección .text
- c) pila del usuario**
 - por exclusión... y por lógica, para qué almacenarla si está vacía
- d) variables globales
 - secciones .data/.bss

3. Compilar .s → ejecutable, usando sólo as y ld, sin gcc...

- a) Se puede, usando en as y ld los modificadores (switches) que corresponda
- b) No se puede
- c) Basta usar ld, con los modificadores que corresponda
- d) Basta usar as, con los modificadores que corresponda

Test Teoría 2020 Febrero

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, 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 EAX=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
```

```
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. 0xfffffffcd
- 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. Alguno de los siguientes no es un nombre de registro en una máquina IA32 en modo 32 bits

- a) sil (Sí lo sería en modo 64 bits)**
- b) ax
- c) ebp
- d) dh

24. Los switches `-m elf_i386` y `-m elf_x86_64` para trabajar en 32 bits / 64 bits corresponden a la herramienta...

- a) nm
- b) as
- c) gcc
- d) ld

25. 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?

- a) 0x0400 0000
n*32 == n << 5 y el bit 26 (0x04...) pasaría a 31 (0x80...) y el resultado llegaría *justo* a convertirse en negativo. Algo más pequeño que 0x04... no llegaría.
- b) 0x4000 0000
- c) 0x0800 0000
- d) 0x8000 0000

26. En x86_64 se pueden referenciar los registros

- a) %rax, %eax, %ax, %ah, %al
- b) %r12q, %r12d, %r12w, %r12l
- c) %rsi, %esi, %si, %sih, %sil
- d) %r8, %r8d, %r8w, %r8l

27. 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
push resultado+4
push resultado
push resultado+4
push \$formato
call printf
add \$20, %esp
- b) push resultado+4

```

push resultado
push resultado+4
push resultado
push $formato
call printf
add $20, %esp

(Sí.  $20 = 8 (\%llx) + 8 (\%llu) + 4 (\$formato)$ )

```

- c) **push resultado+4**
push resultado
push \$formato
call printf
add \$12, %esp
- d) **push resultado**
push resultado+4
push \$formato
call printf
add \$12, %esp

28. 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+4**
movl %edx, resultado
- b) **movl (%eax), resultado+4**
movl (%edx), resultado
- c) **movl (%eax), resultado**
movl (%edx), resultado+4
- d) **movl %eax, resultado**
movl %edx, resultado+4

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

- a) section .text
.start _global
_start:
 - b) .text:
_start:
añadido (:) a (.text) para evitar subterfugios:
a) puede que .global _start se añada más tarde (y funciona)
b) aunque no se ponga .global, ld supone dirección por defecto (y funciona)
Añadiendo (:) es error seguro (.text already defined)
 - c) .section .text
.local _start
_start:
 -
 - d) .section .text
.global _start
_start:
-

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

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

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.**
-

En la práctica "media" se pide sumar una lista de 32 enteros SIN signo de 32bits en una plataforma de 32bits sin perder precisión, esto es, evitando perder acarreos. ¿Cuál es el mínimo valor entero que repetido en toda la lista causaría acarreo con 32bits (sin signo)?

- a) 0xfbff ffff
 - b) 0x07ff ffff
 - c) **0x0800 0000**
 - d) 0xfc00 0000
-

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\n0"
```

la instrucción movl longlista, %ecx copia el siguiente valor:

- a) 32
- b) 16

- - c) 4
Práct.2, Tut, pág.7
 - d) 8
-

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) 0x8000 0000
es negativo, el enunciado pide menor valor positivo
 - b) 0x0800 0000
suma 0x1000 0000 sin problema
 - c) **0x4000 0000**
0x8000 0000 sería negativo en 32bit
 - d) 0x0400 0000
suma 0x0800 0000 sin problema
-

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?

- a) en uno se interpreta como negativo, en otro como positivo
ambos programas interpretan 0x0000 0000 8000 0000 como positivo
- b) en uno ocupa 32 bits, en otro 64 bits
ambos resultados son de 64 bits
- c) **no se diferencian**
0x0000 0000 8000 0000
el resultado cabría en 32bits
- d) en uno los 32 bits superiores son 0xFFFF FFFF, en el otro no
en ambos los 32bits superiores son 0x0000 0000

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

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

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**
Práct.2, Tut, pág.8
 - d) movl (lista), %ebx
-

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
No llega, $\text{abs}(-10 \text{ millones}) \leq 64M$
 - b) **-100 000 000**
Se pasa, 100 millones >> 64M
 - c) -1 000 000 000
 - d) -10 000 000 000
-

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

¿Qué hace gcc -S?

- a) Compilar optimizando tamaño (size), no tiempo

- b) Compilar borrando del ejecutable la tabla de símbolos (strip)
 - c) Compilar .s → .o (fuente ASM a objeto)
 - d) **Compilar .c → .s (fuente C a fuente ASM)**
-

¿Cuál de las siguientes no es una sección de un fichero ELF?

- a) .bss
para datos sin inicialización, mencionada en P2 Apéndice 2 Tabla 9
 - b) .data
para datos inicializados como en la Práctica "media"
 - c) .text
para el código
 - d) **.static inventada**
-

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
 little-endian => primero el menos significativo**
- b) movl %eax, resultado+4
 movl %edx, resultado
 no es little-endian (tampoco es big-)
- c) movl (%eax), resultado
 movl (%edx), resultado+4
 mov M-M imposible

- d) movl (%eax), resultado+4
 movl (%edx), resultado
 mov M-M imposible
-

En X86-64, el registro contador de programa se denomina:

- a) **RIP**
 b) IP
 c) EIP
 d) R15
-

¿Qué modificador (switch) de gcc hace falta para compilar .s → .o sin llamar al enlazador?

- a) **gcc -c**
 b) Eso no se puede hacer con gcc
 c) gcc -S
 d) gcc -s
-

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:

- a) 0x0008d03c y 0x0008d03c
 b) 0x0008d040 y 0x0008d040
 c) 0x0008d044 y 0x00000000
 d) **0x0008d040 y 0x00000000**
-

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?

- - a) **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
 - b) 0xAAAA AAAA
0x0000 0015 5555 5540 != 0xffff fff5 5555 5540
 - c) 0x9999 9999
0x0000 0013 3333 3320 != 0xffff fff3 3333 3320
 - d) 0xFFFF FFFF
0x0000 001f ffff ffe0 != 0xffff ffff ffff ffe0
-

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 \$0xa30bf18a,%eax
cltd
 - b) movl \$0xf18a,%eax
movl \$0xa30b,%edx
 - c) movq \$0xa30bf18a,%rax
 - d) **movl \$0xa30bf18a,%eax**
xorl %edx,%edx
-

¿De qué tipo son los procesadores Intel que usamos en los laboratorios?

Usuario Profesor

a es

- - a) big-endian
 - b) el concepto de endian no es aplicable a estas máquinas, ya que un registro del procesador no cabe en una posición de memoria
 - c) **little-endian**

-
- d) puede ajustarse mediante un bit de control en el registro CR0 que funcionen como little- o big-endian

El primer parámetro de printf:

Usuari Profesor

a es

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

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:

Usuari Profesor

a es

- a) int var 1
- b) .int: var 1
- c) **var: .int 1**
- d) .int var 1

Sobre el programa ensamblador:

Usuari Profesor

a es

- 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.**

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

Usuari Profesor

a es

- a) main
 - b) _init
 - c) begin
 - ✓ • d) **_start**
-

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

Usuari Profesor

a es

- a) .L0
 - b) _init
 - ✓ • c) **_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
 - d) _main
-

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

Usuari Profesor

a es

- a) _int tam = 0
 - b) var tam : integer;
 - c) int tam;
El enunciado original decía "gcc/as" e introducía una ambigüedad indeseada con esta respuesta
 - d) tam: .int .-msg
-

Pregunta:

Los switches --32 y --64 para trabajar en 32bit/64bit corresponden a la herramienta...

Respuesta:

as

Pregunta:

¿Qué hace gcc -O0?

Respuesta:

Compilar sin optimización

Pregunta:

Si ECX vale 0, la instrucción adc \$-1,%ecx

Respuesta:

No cambia CF

Pregunta:

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...

Respuesta:

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

Pregunta:

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

Respuesta:

-m32

Pregunta:

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

Respuesta:

0xC8 en EAX y 0x0 en EDX

Pregunta:

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)?

Respuesta:

0xfbff ffff

Pregunta:

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?

Respuesta:

- 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) El resto siempre tiene el mismo signo que la suma
- d) La media se redondea al entero más próximo**
(división truncada IDIV - no se redondea, se trunca)

Pregunta:

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

Respuesta:

- a) ADC, SBB
 - b) IMUL, IDIV
 - c) **CMP, TEST**
 - d) AND, OR, XOR
-

Pregunta:

¿Qué hace gcc -O1?

Respuesta:

Compilar con optimización

Pregunta:

¿Qué hace gcc -O?

Respuesta:

Compilar con optimización suave (*igual que -O1*)

Pregunta:

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

Respuesta:

No se puede

Pregunta:

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?

Respuesta:

(-1 :-30)

Pregunta:

¿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?

Respuesta:

-32

Pregunta:

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

Respuesta:

gcc

Pregunta:

El switch -l para indicar librerías *NO* funciona con la herramienta...

Respuesta:

as

Pregunta:

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.

Respuesta:

1 000 000 000

(Se pasa, 1000 millones >> 128M)

(32bit sin signo llega a $2^{32}-1 = 4G-1 \sim 4\ 000\ 000\ 000$)

(Habría acarreo si llegáramos a 4G. Con 32 elementos iguales, tendrían que valer $4G/32 = 1G/8 = 128M$)

Pregunta:

¿Qué modificador (switch) de gcc hace falta para compilar una aplicación de 32bits (fuente C) en un sistema de 64bits en el que se ha instalado también el compilador de 32bits?

Respuesta:

-m32

Pregunta:

¿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):

Respuesta:

file.c → file.s → file.o → file

Pregunta:

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,

Respuesta:

es ventajoso para enlazarlo usar gcc, y entonces es conveniente que contenga main (aunque ambas cosas son opcionales)

Pregunta:

Tras ejecutar las tres instrucciones que se muestran desensambladas a continuación, el registro EAX toma el valor

```
08048074 <_start>:  
8048074: be 74 80 04 08 mov $_start, %esi  
8048079: 46           inc %esi  
804807a: 8b 06         mov (%esi), %eax
```

Respuesta:

0x08048074

Pregunta:

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?

Respuesta:

0ffbff ffff

(en cuanto sea algo más grande que 0xfc00 0000 sale overflow)

Pregunta:

En contraposición a un ejecutable Linux ELF, un fichero objeto (obtenido con gcc -c)

Respuesta:

ubica el código (y los datos) a partir de la posición 0x0, las direcciones definitivas sólo se calculan tras enlazar

Pregunta:

Alguno de los siguientes no es un nombre de registro en una máquina x86-64 en modo 64 bits

Respuesta:

- a) sih (%sil en todo caso, no %sih)
 - b) r8d
 - c) r12w
 - d) spl
-

Pregunta:

¿Qué modificador (switch) de gcc hace falta para compilar .c → .o (de fuente C a código objeto)?

Respuesta:

```
gcc -c
```

Pregunta:

El switch de gcc para que únicamente compile de lenguaje C a ensamblador, y no realice ningún paso adicional (ensamblar, enlazar, etc), es...

Respuesta:

```
-S
```

Pregunta:

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:

Respuesta:

Fallaría con lista: .int 1,2,3,4, 1,2,3,4, ...

(Debería salir suma=80 y sale suma=32 porque suma 32 veces lista[0]=1. $32 << 4G - 1$, así que no hay acarreos ni por asomo.)

Pregunta:

¿Qué valor contendrá %edx tras ejecutar las siguientes instrucciones?

```
xor %eax, %eax  
sub $1, %eax  
cltd  
idiv %eax
```

Respuesta:

0

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 incorrecta?

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

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
No. El ejemplo grande falla (`NBITS=20`), igual que fallarían otros ejemplos grandes más pequeños (desde `NBITS=7` sale que $7 \cdot 2^6 = 448 > 255$).

Fallaría con cualquier ejemplo en que el resultado fuera superior a `0xFF = 255`, porque entonces `val>>8` no sería cero y se sumaría erróneamente.
- b) fallaría con `array={0,1,2,3}`
No, $4 < 255$, siempre `val>>8 == 0`, no se suma nada erróneamente
- c) fallaría con `array={1,2,4,8}`
No, $4 < 255$, siempre `val>>8 == 0`, no se suma nada erróneamente
- d) **no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos**

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`:

- a) 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.
 - 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
-

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

- a) arquitectura de 32 bits
 - b) un registro y dos sobrescritos (clobber)
 - c) **dos entradas y una salida**
[v] cuenta como salida y entrada, [x] como entrada
 - d) ninguna
-

En la práctica "parity" 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

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:

- a) produce el resultado correcto
- b) **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.**
- c) no es correcta; fallaría por ejemplo 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

La práctica “parity” debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. La siguiente función contiene un único error (en realidad se han editado 2 líneas de código sobre la versión correcta) pero produce resultado correcto cuando se usa como test el array...

```
int parity4(unsigned* array, int len){
    int i;
    unsigned x;
    int val=0, result=0;

    for (i=0; i<len; i++){
        x = array[i];
        asm("\n"
"ini:           \n\t"
        "xor %[x], %[v]  \n\t"
        "shr %[x]         \n\t"
        "jnz ini          \n\t"
        : [v]"+r" (val)
        : [x] "r" (x)
    );
        result += val & 0x1;
    }
    return result;
}
```

- a) array={1, 16, 256, 1024}
debería salir 4 (todos impares)
sale 2 (1,0,1,0, cada otro impar deshace el anterior)
- b) array={1, 2, 4, 8}
debería salir 4 (1,1,1,1, todos impares)
sale 2 (1,0,1,0, cada otro impar deshace el anterior)
- c) array={0, 1, 2, 3}
debería salir 2 (0,1,1,0, impares el 1 y el 2)
sale 1 (0,1,0,0, el 2 deshace el impar del 1)
-
- d) array={5, 4, 3, 2}
debería salir 2 (impares el 4 y el 2)
sale 2 (0,1,1,0, el 3 cuenta como impar, el 2 deshace el impar)

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;  
}
```

Esta función parity6:

- - a) produce siempre el resultado correcto
bastaba con un sobrescrito "edx", y con 6
instrucciones máquina (no 8)
en concreto, la 3^a instrucción podría haber sido xor
%[x],%%edx y ya sólo quedaría operar con %dh y %dl,
y no usar %cl sino el propio %dl
 - 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

En qué registro se pasa el primer argumento a una función según el estándar _cdecl en una arquitectura IA32?

- a) edi
 - b) esi
 - c) eax
 - d) **Las anteriores respuestas son erróneas**
-

En qué registro se pasa el primer argumento a una función según el estándar _cdecl en una arquitectura IA32?

- a) edi
 - b) esi
 - c) eax
 - d) **Las anteriores respuestas son erróneas**
-

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 popcorn4:

- a) produce siempre el resultado correcto
No. Las sumas paralelas pueden llevar acarreos al siguiente byte. Y la suma total puede ser superior a 255, cosa que no permite el código mostrado.
- b) fallaría con array={0,1,2,3}
No. La suma total es 4, no hay acarreo al byte superior ($4 \ll 256$), quedan sin efecto las instrucciones erróneas o cambiadas de sitio ($>>16$, $>>8$, $\&0xff$)
- c) fallaría con array={1,2,4,8}
No. La suma total es 4, no hay acarreo al byte superior ($4 \ll 256$), quedan sin efecto las instrucciones erróneas o cambiadas de sitio ($>>16$, $>>8$, $\&0xff$)
- d) **no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos Efectivamente.**
El ejemplo más pequeño para llegar al error (suma paralela menos significativa ≥ 256) sería lista[32]={0xff, 0xff, ... 0xff}
En ese caso, calcula result=256, le suma 1 (result=257) y lo enmascara: result=1

¿Cuál es el popcorn (peso Hamming, nº de bits activados) del número 19?

- a) 2
- b) 3
19 = 0x13 = 16+3 = 0b0001 0011 -> popcorn 3
- c) 4
- d) 5

En la convención cdecl estándar para arquitecturas x86 de 32 bits, el resultado de una función se devuelve usualmente en el registro:

- a) EBX
- b) **EAX**
- c) EBP
- d) ESI

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 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
solemos escribir res+=val&0x1, en lugar de ponerlo en
2 sentencias C**
- 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
-

Para averiguar la paridad de un número se puede usar la operación:

- a) AND
 - b) OR
 - c) NOT
 - d) XOR
-

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
Efectivamente, con este for también se cumple que el LSB (bit menos significativo) sigue siendo el XOR lateral de todos los bits.
 - 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
-

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
-

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 popcorn3:

- - a) 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.
 - 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
-

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;  
}
```

Esta función parity4:

- - a) produce siempre el resultado correcto
solemos escribir `res+=val&0x1`, en lugar de ponerlo en 2 sentencias C
adicionalmente, `shr` afecta al flag ZF de manera que sobra `test %[x],%[x]`
 - 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
-

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) la salida de la función se fuerza a que esté en la variable result
 - c) r es una posición de memoria de entrada/salida
 - d) i es un registro de entrada
-

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

- - a) 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

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

¿En qué registro se pasa el primer argumento a una función en Linux gcc x86-64?

- - a) edi
 - b) esi
 - c) edx
 - d) ecx
-

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
-

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 mediante scanf(). Por "cifrar" podemos entender la cifra del César, por ejemplo. "Invertir" es darle la vuelta al string de manera que la primera letra se convierta en la última y viceversa.

- - a) **1 string del usuario se cifra, y se compara con el string del fuente**
 - b) 2 strings del fuente se invierten, se concatenan, se cifra el resultado, y se compara con el string del usuario
 - c) 2 strings del fuente se concatenan, se invierte el resultado, se cifra, y se compara con el string del usuario
 - d) Las 2 (o 3) opciones más difíciles son de la misma dificultad, así que no se puede marcar ninguna como la más difícil
-

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)
-

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) escribir info line main en el panel de línea de comandos gdb
 - b) **recompilar con información de depuración, por si se nos había olvidado, ya que sin –g el ejecutable no contiene información de depuración**
sí, claro, como que nos van a pasar el código fuente... :-)
 - c) asegurarse de que se ha escrito correctamente el nombre del ejecutable
 - d) comprobar que está activado el panel View → Machine Code Window
-

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>:
```

...el código numérico (pin) es...

- a) el entero almacenado a partir de la posición de memoria 0x24(%esp)

- - b) el entero almacenado a partir de la posición de memoria 0x804a044
 - c) el entero cuya dirección está almacenada en la posición de memoria 0x804a044
 - d) el entero 0x804a044

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 pc1(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`:

Usuario Profesores

- 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**
en ambos, el bucle interno empieza perdiendo el LSB, al ejemplo b le pierde 2 (por el 1 y el 3) y al ejemplo c le pierde 1 (por el elemento 1)

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.

- - a) 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
 - b) 2 enteros del fuente se suman, se procesa la suma, y se compara el resultado con el entero del usuario
 - c) 2 enteros del fuente se procesan, se suman los resultados, y se compara la suma con el entero del usuario
 - d) Las opciones más difíciles son de la misma dificultad, así que no se puede marcar ninguna como la más difícil

En la práctica "media" se pide usar `cltd/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 ffffffffffffff0 (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:

Usuario Profesores

- a) al inicio de main EAX vale 0 y R8 contiene un valor inferior a 0x7fffffff00000000
 - b) al llamar a suma RBX contiene un valor inferior a 0x600000 y RCX vale 16
 - c) **al llamar a printf, ECX vale 16 y R8 contiene un valor superior a 0x80000000**
ECX se corresponde con el 0x00000010 de la última línea y R8 con el 0x9f816d80
 - d) tras volver de printf RAX contiene un valor superior a 60 y RDI superior a 0x600000
-

¿Qué sentencia usamos en el programa blink (led intermitente) para encender el led integrado en la placa Elegoo Mega2560?

Usuario Profesores

- a) pulseIn (LED_BUILTIN, HIGH);
- b) analogWrite (LED_BUILTIN, HIGH);
- c) pinMode (LED_BUILTIN, OUTPUT);
- d) **digitalWrite (LED_BUILTIN, HIGH);**

¿Cuál expresión es cierta?

Usuario Profesores

- a) **popcount(2) == popcount(64)**
 $1==1$
0b0000 0010 // 0b0100 0000 // popcount 1 ambos casos
- b) $\text{popcount}(15) < \text{popcount}(51)$
 $4<4$
- c) $\text{popcount}(7) > \text{popcount}(60)$
 $3>4$

d) `popcount(96) != popcount(3)`

`2!=2`

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).

- - a) **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
 - b) 2 strings del usuario se concatenan, se cifra el resultado y se compara con el string del fuente
 - c) 2 strings del usuario se cifran, se concatenan los resultados, y se compara con el string del fuente
 - d) Las opciones más fáciles son de la misma dificultad, así que no se puede marcar ninguna como la más fácil

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.

Usuario Profesores

- a) 1 string del fuente se cifra, se invierte y se compara con el string del usuario
en la comparación pillamos la contraseña
- b) **el string del usuario se cifra y se compara con 1 string del fuente**
para que salga bien la comparación tendremos que entender la cifra y aplicarla inversamente para teclear la contraseña correcta
- c) 2 strings del fuente se invierten, se concatenan, se cifra el resultado, y se compara con el string del usuario
en la comparación pillamos la contraseña

- 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
basta con reconocer nuestro string al inicio de lo que se compara, y teclear el inicio de lo que se espera en la comparación
-

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"  
            "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:

Usuario Profesores

- a) **produce siempre el resultado correcto**
shr afecta al flag de cero, el test previo no hacía falta
- 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

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

- a) 1 entero del fuente se procesa, y se compara con el entero del usuario
- b) 2 enteros del usuario se suman, se procesa la suma, y se compara con el entero del fuente
- c) **2 enteros del usuario se procesan, se suman los resultados, y se compara con el entero del fuente**
- d) Las 2 (o 3) opciones más difíciles son de la misma dificultad, así que no se puede marcar ninguna como la más difícil

¿Para qué se utiliza la función gettimeofday en la práctica de la "bomba digital"?

- a) Para imprimir la hora en la pantalla
- b) Para cifrar la contraseña en función de la hora actual
- c) Para cronometrar y poder comparar las duraciones de las distintas soluciones del programa
- d) **Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la contraseña o el PIN**

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

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) 0x1009 4F97 (269 045 655)
 - b) 0xffff (4095)
 - c) 0x7ff (2047)
 - d) 1
-

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 se hace mov 0x0804a044, %eax justo antes de cmp %eax, %edx, en donde %edx está relacionado con scanf**
 - d) el entero 0x804a044
-

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**
-

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

- c) gdb
 - d) objdump
-

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 intrucción 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\x72\x62\x61\x64\x61\x63\x61\x72\x62\x61" ("abadacarba")
 - b) "\x61\x62\x72\x61\x61\x62\x72\x61\x63\x61\x64" ("abraabracad")
 - c) "\x63\x61\x64\x61\x62\x72\x61\x61\x62\x72\x61" ("cadabraabra")
 -
 - d) "\x61\x62\x72\x61\x63\x61\x64\x61\x62\x72\x61" ("abracadabra")
-

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) ddd
- b) hexedit
-
- c) **objdump**
- d) gdb

En la práctica de la bomba, el primer ejercicio consistía en “saltarse” las “explosiones”, para lo cual se puede utilizar...

- a) hexedit u objdump
 - b) **gdb o ddd**
un depurador cualquiera sirve para forzar que se cumpla la condición para "saltarse" la bomba
 - c) objdump o gdb
 - d) ddd o hexedit
-

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

- a) Si continuáramos multiplicando por 2 el tamaño del vector en el eje X obteniendo más puntos de la gráfica, esta continuaría horizontal para cualquier valor más allá de 64 MB.
hay quien ha probado a superar la cantidad de memoria física (RAM, para lo cual hay que tener una partición de swap de tamaño conveniente) y ha observado la lentitud del disco duro, comparado con la RAM.
- b) **La gráfica tiene escalones hacia arriba porque en cada punto del eje X accedemos al mismo número de elementos del vector y el número de aciertos por localidad temporal disminuye bruscamente en ciertos puntos al aumentar el tamaño del vector.**
si cabe el vector en cache, los millones de accesos (siempre el mismo número de accesos) son todo aciertos por localidad temporal
- c) La gráfica tiene tramos horizontales porque el hecho de realizar la mitad de accesos al vector en cada punto de un tramo horizontal respecto al anterior punto de ese mismo tramo horizontal es compensado por el número de fallos creciente en ese mismo tramo horizontal.
siempre se hacen los mismos millones de accesos

-
- d) La diferencia de velocidades entre L2 y L3 es mayor que la diferencia de velocidades entre L1 y L2.
-

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) sacar un uno (1) por el stream line
 - b) salir del bucle si el tamaño de línea se volviera menor o igual que 1 para algún elemento del vector
 - c) volver al principio del vector cuando el índice exceda la longitud del vector
 - d) **duplicar el tamaño del salto en los accesos al vector respecto a la iteración anterior**
-

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
-

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 observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

- - a) 8 / 64 / 2048 estudiantes
 - b) 16 / 32 / 64 estudiantes
 - c) 4 / 32 / 512 estudiantes
 - d) 32 / 256 / 8192 estudiantes

¿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

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.

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) 1/4
 - b) 1/8
 - c) **1/16**
uno de cada 16 $A[i][k]$ y otro de cada 16 $B[i][k]$
 - d) 1/2
-

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:

- a) La cache L1 es como mucho cuatro veces más rápida que la cache L2
- b) **La cache L1 es al menos cuatro veces más de rápida que la cache L2**
sólo sería exactamente 4 si el tiempo de cómputo adicional a los accesos de memoria fuera despreciable. Como en general no lo es, los accesos deben ser aún más rápidos para que el incremento sea 4x.
Matemáticamente, el enunciado dice que $c+m_1=2$, $c+m_2=8$, entonces $(c+m_2)/(c+m_1)=4$, y despejando sale $m_1=(m_2)/4-3c/4$, es decir, sólo si $c=0$ entonces $m_1=(m_2)/4$, y si $c>0$ entonces $m_1<(m_2)/4$
- c) La cache L1 es seis veces más rápida que la cache L2

- d) La cache L1 es cuatro veces más rápida que la cache L2
-

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**
-

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 += 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)
-

Abajo se ofrece el listado de una función para multiplicar matrices C = A x 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) 1/4
 - b) **1/2**
1/2 por cada B[k][j] + 1/32 por cada 16-ésimo A[i][k]
 - c) 1/16
 - d) 1/8
-

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 espacial aumenta, porque ya habíamos accedido a cada posición i-1 del vector cuando lo recorrimos en el punto anterior del eje X.
- b) 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.

- c) 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.
 - d) **Cada vez que line aumenta al doble, realizamos la mitad de accesos al vector que para el valor anterior.**
-

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

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
es el tamaño de línea deducido en la práctica "line"**
- c) de 1 KB en 1 KB
- d) de 64 KB en 64 KB

¿Cuál de las siguientes instrucciones es *errónea*?

Usuari Profesore

a s

- a) mov (%r9,%r10,8), %r11
- b) lea (%rbp,%rsp,4), %r8
RSP no puede ser índice
- c) lea (%rdx,%rsi,2), %rdi
- d) mov (%rax,%rbx,1), %rcx

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:

Usuari Profesore

a s

- a) ambos nos muestran la longitud del string (que es/vale/equivale a 28)
- b) **el primero (x) nos muestra la longitud, y el segundo (print) nos muestra otro valor distinto**
el otro valor es la dirección de longsaludo, quitar & si se desea ver el valor con print, con x siempre se usan direcciones (volcado de memoria)
- c) el segundo (print) nos muestra la longitud, y el primero (x) nos muestra otro valor distinto
- d) alguno o ambos contienen algún error gramatical (falta o sobra algún &, algún typecast (char*) o (long), etc.)

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:

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/idivq %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 0x7fffffff)

Usuari Profesore

a s

- **a) media = 0x0fffffff, resto = 0**
RAX acumula correctamente 0x7 ffff fff0 y RDX se extiende a 0, pero al dividir por ECX en lugar de RDX sólo se divide EDX:EAX = 0x0000...: ffff fff0 dejando media EAX 0x0fff ffff resto EDX 0x0
 - b) media = 0x7fffffff, resto = 0
 - c) media = 0xffffffff, resto = 0
 - d) resto distinto de cero
-

¿Qué modificador (switch) de gcc hace falta para compilar .c→.s (de fuente C a fuente ASM)?

Usuari Profesore

a s

- a) gcc -c
- **b) gcc -S**
- c) Eso no se puede hacer con gcc

d) gcc -s

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
```

Estando bien programado todo lo demás, este código

Usuari Profesore

a s

- a) fallaría con lista: .int -1,-2,-4,-8
 - b) fallaría con lista: .int 0,1,2,3
 - c) no siempre produce el resultado correcto, pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos
 - d) **produce siempre el resultado correcto**
inc %edx se hace después de adc, cuando ya se ha usado el valor de %edx
Recordar que usamos cltd para extender a %edx el elemento leido en %eax
Igual podríamos haber hecho xor %edx,%edx y tampoco pasaría nada
-

¿Cuál de las siguientes sumas con signo produce desbordamiento con 32 bits?

Usuari Profesore

a s

- a) 0xFFFFFFFF + 0xFFFFFFFF
-1-1 ... sin problema
- b) 0x7FFFFFFF + 0xFFFFFFFF
positivo mayor -1 ... sin

problema

- c) **0x7FFFFFFF + 0x00000001**
positivo mayor +1 ... problema
 - d) **0xFFFFFFFF + 0x00000001**
-1+1 ... sin problema
-

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?

Usuari Profesore

a s

- a) 0xffff ffff
 - b) 0xfc00 0000
 - c) **0xf7ff ffff**
 - d) 0xf000 0000
-

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 CON signo, y usando como contraejemplo las siguientes listas:

Usuari Profesore

a s

- 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
-

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)

Usuari Profesore

a s

- ✓ • 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
-

La línea de código ensamblador: mov \$msg, %rsi

Usuari Profesore

a s

- 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.**
-

[P4T]

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 <main+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 <main+247>
0x40084d <main+242> call 0x400727 <boom>
0x400852 <main+247> lea 0x10(%rsp),%rdi
```

...el código numérico (pin) es...

Usuario Profesores

- a) el entero 0x601060
no, 0x601060 es la dirección del pin
 - b) el entero cuya dirección está almacenada en la posición de memoria 0x4009f8
no, 0x4009f8 es el 1er arg del 1er scanf, un formato ("%i" para el pin)
 - c) el entero almacenado a partir de la posición de memoria 0x4009fb
no, 0x4009fb es el 1er arg del 2º scanf, un formato ("%*s" para consumir)
 - ✓ • d) **el entero almacenado a partir de la posición de memoria 0x200819+0x400847**
sí, gdb lo anota como #0x601060, es 0x200819(%rip)
cuando RIP vale 0x400847. Se carga esa variable en EAX
y luego se compara con 0xc(%rsp) que fue el 2º arg del 1er scanf, el pin tecleado
-

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)...

Usuari Profesore

a s

- a) ADC \$0, ...
 - b) **CLTD, ADD y ADC**
 - c) JNC y INC
 - d) CLTQ, CQTO y IDIV
-

¿Qué es el lenguaje máquina?

- a) Conjunto de datos binarios que representan señales eléctricas internas de la unidad de control de un microprocesador.
no, representan el programa a ejecutar
- 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.**
-

Invocando a printf de libC (SystemV AMD64) desde ensamblador...

Usuario Profesores

- a) el primer argumento debe ponerse en %rax
en %rdi
 - b) el segundo argumento es el formato
el primero
 - c) si se desean imprimir tres long, el tercero debe ponerse en
%rdx
en %rcx
 - d) **si se desean imprimir cuatro int, el cuarto debe
ponerse en %r8d**
diane's silk dress costs 89
formato en rdi, ints en esi, edx, ecx, r8d
-

En la práctica de la bomba, el primer ejercicio consistía en saltarse las explosiones, para lo cual se puede utilizar... (marcar opción *falsa*)

Usuari Profesore

a s

- a) ddd
 - b) **objdump**
un debugger cualquiera serviría, objdump no ejecuta
nada
 - c) gdb
 - d) eclipse
-

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<strcmp>
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...

Usuari Profesore

a s

- 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

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...

Usuari Profesore

a s

- 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**

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.

Usuari Profesore

a s

- a) 1 string del fuente se cifra, se invierte y se compara con el string del usuario
en la comparación pillamos la contraseña
- ✓ • b) **el string del usuario se cifra y se compara con 1 string del fuente**
para que salga bien la comparación tendremos que entender la cifra y aplicarla inversamente para teclear la contraseña correcta
- c) 2 strings del fuente se invierten, se concatenan, se cifra el resultado, y se compara con el string del usuario
en la comparación pillamos la contraseña
- 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
basta con reconocer nuestro string al inicio de lo que se compara, y teclear el inicio de lo que se espera en la comparación

¿Para qué se utiliza la función scanf() en la práctica de la "bomba digital"?

Usuari Profesore

a s

- a) Para leer la contraseña (clave alfanumérica)
- ✓ • b) **Para leer el PIN (clave numérica)**
- c) Para escanear el fichero ejecutable "bomba" y asegurarse de que no contenga virus
- d) Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la clave

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...

Usuari Profesore

a s

- 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

Un estudiante entrega la solución de una bomba y en ella aparece el comando de gdb “p(char*)\$rdi”. Este comando:

Usuari Profesore

a s

- 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

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<strcmp>
```

```
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...

Usuari Profesore

a s

- a) el string almacenado a partir de 0x601068 cierto, esa dirección es la que calcula db
- b) el string alm. a partir de 0x2008a1+0x4007c7 cierto, 0x2008a1(%rip) se calcula cuando %rip apunta a la siguiente instrucción
- c) el string almacenado a partir de 0x30(%rsp) eso sería el string tecleado
- d) **no se puede marcar una y sólo una de las respuestas anteriores**

```
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 (pin) es...

Usuari Profesore

a s

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

En la práctica de la bomba, el primer ejercicio consistía en “saltarse” las “explosiones”, para lo cual se puede utilizar...

Usuari Profesore

a s

- a) objdump o gdb
 - b) **gdb o eclipse**
 - un depurador cualquiera sirve para forzar que se cumpla la condición para "saltarse" la bomba
 - c) eclipse o ghex
 - d) ghex u objdump
-

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 <main+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 <main+247>
0x40084d <main+242> call 0x400727 <boom>
0x400852 <main+247> lea 0x10(%rsp),%rdi
```

...el código numérico (pin) es...

Usuari Profesore

a s

- a) el entero almacenado a partir de 0xc(%rsp)
no, 0x601060 es la dirección del pin
- b) el entero alm. a partir de 0x1ae+0x40074b
no, 0x4009f8 es el 1er arg del 1er scanf, un formato ("%i" para el pin)
- c) el entero almacenado a partir de 0x6010560
no, 0x4009fb es el 1er arg del 2º scanf, un formato ("%*s" para consumir)

- **d) el entero alm. a partir de 0x2008ee+0x400762**
sí, gdb lo anota como #0x601060, es 0x200819(%rip)
cuando RIP vale 0x400847. Se carga esa variable en EAX y
luego se compara con 0xc(%rsp) que fue el 2º arg del 1er
scanf, el pin tecleado
-

La función gettimeofday() en la práctica de la "bomba digital" se utiliza para

Usuari Profesore

a s

- 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**
-

En la práctica "media" se pide como ejercicio previo sumar una lista de N enteros SIN signo produciendo un resultado .quad (doble precisión). El programa esqueleto ofrecido (suma.s, que suma la conocida lista de 9 elementos en simple precisión) no es válido. Se podría comprobar imprimiendo el resultado SIN signo y usando como contraejemplo las siguientes listas:

Usuario Profesores

- a) 0x4000 0000 y 0x8000 0000 (el usuario piensa que sus datos NO tienen signo)
 - **b) 0x8000 0000 y 0xC000 0000**
 - c) ninguna de las dos
 - d) ambas
-

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

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:

Usuario Profesores

- a) está correctamente diseñado, la instrucción ret final es optativa, y no es concebible que quitar el ret cause algún error
 - b) produce un error "Segmentation fault" cuando empieza a acceder a memoria que no le corresponde (EIP)
 - c) **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"**
 - d) no se puede marcar ninguna de las opciones anteriores
-

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)...

Usuario Profesores

- a) ADC \$0, ...
 - b) CLTD, ADD y ADC
 - **c) JNC y INC**
 - d) CLTQ, CQTO y
IDIV
-

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 (%rbx,%rsi,4), %eax
cltd
add %eax, %edi
adc %eax, %ebp
inc %esi
cmp %esi, %ecx
jne bucle
```

Esta versión de la suma CON signo

Usuario Profesores

- 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
-

¿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?

Usuario Profesores

- a) -32
 - b) -m32
 - c) No hace falta modificador, Id lo deduce del tipo de objeto a enlazar
 - d) **-m elf_i386**
-

¿Qué hace gcc -O?

Usuario Profesores

- a) **Compilar con optimización, igual que -O1**
 - b) Compilar .c → .o
 - x c) Compilar sin optimización, igual que -O0
 - d) Compilar .s → .o
-

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"
.ascii "lt = 0x %08x lt = 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 0x7fffffff)

Usuario Profesores

- a) **media = 2147483647 resto = 0
= 0x 00000010 = 0x ffffff0**

la media correctamente calculada 2G-1, pero ECX contiene aún 16, por eso se imprime 0x00000010. Se puede adivinar que R8d contiene 0xffffffff0 (ya lo sabíamos, desde antes de idiv)

- b) media = 16 resto = -16
 = 0x 7fffffff = 0x 00000000
 - c) media = 16 resto = -16
 = 0x 00000010 = 0x ffffff0
 - d) media = 2147483647 resto = 0
 = 0x 7fffffff = 0x 00000000
-

La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de `popcount7`:

```
int popcount7(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 = 0x0f0f0f0f0f0f0f0f;
    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 `popcount7` da resultado correcto:

Usuari Profesore

0 s

- **a) con array={3,2,1,0}**

 - b) con array={8,4,2,1}
en este caso sí afecta porque
 $x1=0x0000\ 0004\ 0000\ 0008$ manipulado hasta $x1=2$
correctamente
 $x2=0x0000\ 0001\ 0000\ 0002$ manipulado a $0x0000\ 0001\ 0000\ 0001$

 - la suma es una cantidad enorme $0x0000\ 0001\ 0000\ 0003=4G+3$ en lugar de 4
 - c) con ambos ejemplos
 - d) con ninguno de los dos ejemplos
-

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;
```

Usuari Profesore

0 s

- 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.
-

En la convención cdecl estándar para arquitecturas x86 de 32 bits, cuál de las siguientes afirmaciones es cierta:

Usuari Profesore

0 S

- **a) Los parámetros se pasan en pila, de derecha a izquierda; es decir, primero se pasa el último parámetro, después el penúltimo... y por fin el primero**
 - b) Solamente es necesario salvar el registro EAX
 - c) Los registros EBX, ESI y EDI son salva-invocante
 - d) Ninguna de las anteriores es cierta
-

¿Cuál es el popcount (peso Hamming, nº de bits activados) de una lista de N números inicializada con los valores 0..N-1?

Usuari Profesore

0 S

- a) $(N-1)*N/2$
no, eso es la fórmula de Gauss para la suma de la progresión aritmética
 - b) $N*(N+1)/2$
no, eso sería la suma de la progresión aritmética de 1..N
 - c) si N es par, $N*(N/2)$
no, en todo caso $N/2$ sería la suma de paridades
 - **d) si N es potencia de 2, $(\log_2(N)*N)/2$**
todas las combinaciones de $\log_2(N)$ bits, en cada posición de bit la mitad son 0 y la otra mitad son 1
p.ej.: LSB mitad par mitad impar, MSB mitad empieza por 0 mitad por 1, etc...
-

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, 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`:

Usuari Profesore

0 s

- **a) produce siempre el resultado correcto**
afortunadamente el tipo de array es `unsigned*`, y entonces `x=array[i]` toma sólo el elemento indicado, los 4 MSBs de x y val quedan a 0, `val>>32` es cero, y se acumula en res la cantidad correcta
 - 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
-

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...

Usuari Profesore

0 s

- a) **test %[x], %[x]**
así viene en `pcount_r` (Tema 2.3 tr.38)
 - b) **cmp %[x], %[r]**
no sirve
 - c) **sbb \$0, %[r]**
¿se pretende sustituir la segunda instrucción?
redacción original `adc $0, %[x]` hubiera servido para sustituir la 3^a instrucción, así que cambiamos por esta otra que no sirve (usa `%[r]`).
 - d) **sar %[x]**
-

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

Usuari Profesore

0 s

- a) a través de variables globales
 - b) a través de los registros
 - c) a través de la pila
 - d) **todas las anteriores son ciertas**
-

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"
        "    adc $0, %[r]  \n"
        "fin4:         \n\t"
        "    shr %[x]      \n\t"
        "    jne ini3      \n\t"
        : [r]"+r" (res)
        : [x] "r" (x) );
}
return res;
}

```

En la práctica “media” se pide sumar una lista de 16 enteros *sin* signo de 32 bits (unsigned en C) evitando desbordamiento (sin signo), esto es, acarreo. ¿Cuál es el mayor

valor que repetido en toda la lista *no* causaría acarreo en 32 bits (unsigned en C)?

- a. 0xFFFF FFFF mayor que b.
- b. 0x7FFF FFFF mayor que c.
- c. 0x1000 0000 al desplazar 4 bits a la izquierda sale el 1 hacia el CF
- > d. 0xFFFF FFFF**

En la práctica “media” se pide sumar una lista de 16 enteros *sin* signo de 32 bits (unsigned en C) evitando desbordamiento (sin signo), esto es, acarreo. ¿Cuál es el menor

valor que repetido en toda la lista causaría acarreo en 32 bits (unsigned en C)?

- a. 0xFFFF FFFF
- b. 0x7FFF FFFF
- > c. 0x1000 0000**
- //al desplazar 4 bits a la izquierda sale el 1 hacia el CF
- d. 0x0FFF FFFF

Habiendo definido en código fuente ensamblador saludo: .ascii "Hola a todos!\nHello World!\n"), para conseguir desde gdb cambiar el string para que la llamada al sistema WRITE

escriba solo una línea en lugar de dos, se debería hacer...

- a. nada, porque ya de por sí WRITE escribe saludo como una sola línea
- b. nada, desde gdb no se puede conseguir que WRITE escribe saludo como una sola línea
- c. p (char*) &saludo+13 y también p (char*) &saludo+14_

-> d. set var *((char*)&saludo+13)='.'

Esta función es muy diferente a la versión "oficial". Notar el salto condicional a "ini3" en la función popcorn3 (en donde sí se hizo bien el ini3:/shr/adc/test/jne recomendado).

Esta función popcorn4:

Usuari Profesore

0 s

- 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**
-

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          "  
        :                  // output  
        :                  // input  
        : "cc",              // clobber  
        : "rax", "rdx"  
    );  
}
```

no incluye la instrucción ret dentro de la sentencia asm porque:

Usuari Profesore

0 s

- 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
-

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`:

Usuari Profesore

0 s

- 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**
-

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;
    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 se diferencia de la versión "oficial" en los tipos de array y x.

Esta función popcount5:

Usuari Profesore

0 s

- **a) produce siempre el resultado correcto**
 - b) fallaría con array={0,1,2,3}
 - c) fallaría con array={0,-1,-2,-3}
 - d) no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos
-

¿Cuál es el popcorn (peso Hamming, no de bits activados) del número 42?

- a. 2
- > b. 3 //42 = 0x2A = 0b10 1010 ⇒ 3 bits (32+8+2)**
- c. 4
- d. 5

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 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, y el profesor estaría quedando en ridículo (por injusto) si intenta entorpecer o dificultar de cualquier forma que el estudiante se lleve la merecida nota
- > b) el profesor puede pedirle que vuelva a descargar la bomba original e intente repetir con ese ejecutable las claves que acaba de teclear: en realidad eso es sólo para asegurarse completamente de que no se trata de una errata en la lista, aunque también es por la curiosidad de ver si el estudiante se da cuenta de que el profesor sabe lo que ha hecho
- c) hay que denunciar al profesor a los representantes de estudiantes
- d) el enunciado es capcioso: si la lista no tiene erratas y el pin se escribe mal la bomba tiene que explotar, y si la bomba no explota es porque las claves se escribieron bien y la lista está mal. En el primer caso no debe puntuar, en el segundo caso sí debe puntuar, y en ningún caso el profesor tiene derecho a hacer comprobaciones distintas a las dos mencionadas

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, el estudiante debería...

- 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”

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

Usuari Profesore

o s

- a) Pasar a una función un puntero a una variable se traduce en introducir en la pila el valor de la variable
 - b) 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
 - 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**
 - d) 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
-

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:

Usuari Profesore

0 s

- a) con array={0x80000000, 0x00400000, 0x00000200, 0x00000001}
sale bien (4), sólo corría riesgo 0x1 y contabiliza 1, así que no hay problema
- b) **con array={0x10000000, 0x00200000, 0x00000400, 0x00000008}**
sale mal (3), corría riesgo 0x8 y efectivamente contabiliza 0, se pierde el bit
- c) con ambos ejemplos
- d) con ninguno de los dos ejemplos

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:

Usuario Profesores

- a)
produce siempre el resultado correcto
solemos escribir res+=x&0x1, en lugar de ponerlo en 2 sentencias C
- 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

La práctica "popcount" debía calcular la suma de bits (peso Hamming) de los elementos de un array. Un estudiante entrega la siguiente versión de popcorn4:

```
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 popcorn3 y popcorn4, con alguna instrucción cambiada. Esta función popcorn4:

Usuari Profesore

0 s

- a) **produce siempre el resultado correcto**

básicamente es popcorn3 con dos clc al inicio y al final que no hacen daño (para asegurarse habría que indicar en las restricciones que se han modificado los flags)

- 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
-

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:

Usuari Profesore

0 S

- a) con array={0xA0000000, 0x00B00000, 0x00000C00, 0x0000000D}
sale bien (10), ABC cuentan 2+3+2=7, sólo corría riesgo 0xD, debe contar 3, y contabiliza 0b01+0b10=0b11=3, así que no hay problema
 - b) con array={0x10000000, 0x00200000, 0x00000400, 0x00000008}
sale mal (3), corría riesgo 0x8 y efectivamente contabiliza 0, se pierde el bit
 - c) con ambos ejemplos
 - d) con ninguno de los dos ejemplos
-

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;
    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:

Usuari Profesore

0 S

- a) con array={1, 16, 256, 4096}
256 es 0x100, ese 1 se cuenta 2 veces
 - b) con array={1, 32, 1024, 32768}
 - c) con ambos ejemplos
 - d) con ninguno de los dos ejemplos
-

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:

Usuari Profesore

0 S

-
- 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
-

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?

Usuari Profesore

o s

- a) un error de gdb
 - b) una variable de entorno
 - c) **0x0**
ver Fig.3 pág.4: (%rsp) es argc=4, (%rsp+8,16,24,32) son los 4 strings (suma_01 uno dos tres) y (%rsp+40) es el NULL separando los argumentos de las variables de entorno
 - d) tres
-

En la práctica “popcount”, 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:

Usuari Profesore

o s

- a) void crono(int * func , char* msg);
no compila, 1er argumento es puntero a entero
 - b) void crono((int *)func (), char* msg);
no compila, sobran paréntesis, sería
prototip.func.devolv.ptr.int
 - c) **void crono(int (* func)(), char* msg);**
ok, 1er argumento es el puntero a función devolviendo entero
 - d) void crono(int * func (), char* msg);
-

¿Cuál es el popcorn (peso Hamming, nº de bits activados) del número 29?

Usuari Profesore

0 s

- a) 2
 - b) 3
 - c) **4**
 - 29 = 0x1D = 16+8+4+1 = 0b0001 1101 -> popcorn
4
 - d) 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 popcorn2:

```
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 popcorn2:

Usuari Profesore

0 s

- 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**
-

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

Usuari Profesore

0 S

- a) ninguna
- b) arquitectura de 32 bits
- c) dos entradas y una salida
- **d) un registro y dos sobrescritos
(clobber)**

Utilizando la sentencia `asm()`, las denominadas restricciones que se indican al final de dicha sentencia, involucran a:

Usuari Profesore

0 S

- a) Solamente las entradas

- b) Solamente las salidas
 - c) Solamente los sobrescritos
 - d) **Ninguna de las anteriores es cierta**
-

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 <main+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 <main+247>
0x40084d <main+242> call 0x400727 <boom>
0x400852 <main+247> lea 0x10(%rsp),%rdi
```

...el código numérico (pin) es...

Usuari Profesore

a s

- a) el entero almacenado a partir de la posición de memoria
0x4009fb
no, 0x4009fb es el 1er arg del 2º scanf, un formato ("%*s" para consumir)
- b) el entero 0x601060
no, 0x601060 es la dirección del pin
- c) **el entero almacenado a partir de la posición de memoria
0x200819+0x400847**
sí, gdb lo anota como #0x601060, es 0x200819(%rip)
cuando RIP vale 0x400847. Se carga esa variable en EAX y luego se compara con 0xc(%rsp) que fue el 2º arg del 1er scanf, el pin tecleado

- d) el entero cuya dirección está almacenada en la posición de memoria 0x4009f8
no, 0x4009f8 es el 1er arg del 1er scanf, un formato ("%i" para el pin)
-

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<strcmp>
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...

Usuari Profesore

a s

- a) el string almacenado a partir de 0x601068
cierto, esa dirección es la que calcula db
 - b) el string alm. a partir de 0x2008a1+0x4007c7
cierto, 0x2008a1(%rip) se calcula cuando %rip apunta a la siguiente instrucción
 - c) el string almacenado a partir de 0x30(%rsp)
eso sería el string tecleado
 - d) no se puede marcar una y sólo una de las respuestas anteriores
-

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...

Usuari Profesore

0 s

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

¿Qué sentencia usamos en el programa blink (led intermitente) para encender el led integrado en la placa Elegoo Mega2560?

Usuari Profesore

a s

- a) pinMode (LED_BUILTIN,
 OUTPUT);
- ✓ • b) digitalWrite (LED_BUILTIN,
 HIGH);

- c) analogWrite (LED_BUILTIN,
HIGH);
 - d) pulseIn (LED_BUILTIN, HIGH);
-

En la práctica de E/S en Arduino, la instrucción CBI del repertorio del microcontrolador realiza...

Usuari Profesore

a s

- a) una comparación
 - b) un complemento a uno
 - c) **una operación de bits**
el guión dice "hay que usar la instrucción máquina CBI P,b (Clear Bit in I/O Register), que pone a 0 el bit b del registro P [1]:"
 - d) una llamada a subrutina
-

En la práctica de E/S en Arduino, DDRB es

Usuari Profesore

a s

- a) el segundo canal de memoria DDR
 - ✓ • b) **el registro de dirección de datos del puerto B**
el guión dice "Cada uno de los 8 bits del registro Port B Data Direction Register (DDRB) indica si el bit correspondiente del puerto B está configurado como entrada (0) o salida (1)."
 - c) el registro de datos y direcciones B
 - d) el registro buffer de datos D
-

¿Para qué se utiliza la función gettimeofday() en la práctica de la "bomba digital"?

Usuari Profesore

o s

- a) **Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la clave**
 - ✗ b) Para cronometrar y poder comparar lo que tardan las distintas versiones del programa
 - c) Para imprimir la hora en la pantalla
 - d) Para cifrar la clave en función de la hora actual
-

Las resistencias utilizadas en la práctica de Arduino

Usuari Profesore

a s

- - a) Son de color azul claro y tienen 5 bandas de color: las 3 primeras indican un valor, la 4^a banda es un multiplicador y la 5^a banda es la tolerancia
 - b) Son de color beige y tienen 4 bandas de color: las 2 primeras indican un valor, la 3^a banda es un multiplicador y la 4^a 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.

Sobre el resultado devuelto por la función de Arduino map(sensorValue, sensorLow, sensorHigh, 50, 4000); usada en el programa del Theremín de luz:

Usuari Profesore

a s

- a) Si sensorValue vale 50, devuelve sensorLow
- b) Si sensorValue vale 50, devuelve sensorHigh
- c) Si sensorValue vale 2025, devuelve la mitad entre sensorLow y sensorHigh
- d) Si sensorValue vale la mitad entre sensorLow y sensorHigh, devuelve 2025
sí, map traslada rango 2º-3er args al rango 4º-5º args

En la práctica de E/S en Arduino, la instrucción SBIW del repertorio del microcontrolador realiza...

Usuari Profesore

a s

- a) una suma de un valor inmediato
- ✓ • b) una resta de un valor inmediato
el guión dice "La instrucción SBIW RdI,K (Subtract Immediate from Word) resta la constante K del registro de 16 bits Rdh:Rdl, es decir, hace Rdh:Rdl ← Rdh:Rdl – K."
- c) una operación de bits

d) un salto incondicional

Al colocar un led en la placa de prototipado de Arduino, ¿cómo se sabe cuáles son el ánodo y el cátodo?

Usuari Profesore

a s

- 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**
-

¿Cuál de las siguientes patillas (pin) no se menciona (no se pide usar/no se debe usar) en la 1^a práctica del Theremín de luz de Arduino? (Proyecto p06 / sólo photocélula y zumbador)

Usuari Profesore

a s

- a) Analog In A0
 - ✓ • b) **AREF**
 - c) ledPin = 13
 - d) GND
-

En la placa del kit de Arduino, las patillas de tierra vienen etiquetadas con la leyenda:

Usuari Profesore

a s

- a) A0
 - b) 5V
 - ✓ • c) **GND**
 - d) 3.3V
-

En la práctica de E/S en Arduino, la instrucción SBI del repertorio del microcontrolador realiza...

Usuari Profesore

a s

- a) una suma de un valor inmediato
 - b) una resta de un valor inmediato
 - c) **una operación de bits**
el guión dice "SBI P,b (Set Bit in I/O Register) pone a 1 el bit b del registro P [1]"
 - d) un salto incondicional
-

La función setup() de Arduino es llamada:

Usuari Profesore

a s

- a) Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, pero no cuando se sube desde el entorno de desarrollo
 - b) **Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, o se sube desde el entorno de desarrollo**
 - c) Al principio de cada iteración de la función loop()
 - d) Cuando se sube desde el entorno de desarrollo o se pulsa el botón de reset, pero no cuando se conecta la alimentación
-

La función setup() de Arduino es llamada:

Usuari Profesore

a s

- a) Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, pero no cuando se sube desde el entorno de desarrollo
 - b) **Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, o se sube desde el entorno de desarrollo**
 - c) Al principio de cada iteración de la función loop()
 - d) Cuando se sube desde el entorno de desarrollo o se pulsa el botón de reset, pero no cuando se conecta la alimentación
-

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:

Usuari Profesore

a s

- a) Exactamente esa solución viene en el guión, no hace falta aprenderla fuera de clase
- b) **Nosotros conectamos los led en serie con una resistencia como dice el guión**
- c) Hay que revisar ese equipo en concreto, debe tener algún defecto, ese led no debe parpadear
- d) Es cierto, es un error de diseño del guión, es más sencillo conectarlo así

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)...

Usuario Profesores

- a) **ADC \$0, ...**
- b) CLTD, ADD y ADC
- c) JNC y INC
- d) CLTQ, CQTO y
IDIV

El circuito que hay que montar en la práctica del zumbador pasivo con Arduino es un:

Usuari Profesore

a s

- a) Buffer triestado
- ✓ • b) **Divisor de tensión**
- c) Seguidor de voltaje

d) Comparador

¿En qué registro devuelven el resultado las funciones que siguen la convención de llamada a función “System V AMD64 ABI”?

c) RAX

¿Qué registros podemos modificar a nuestro antojo en el interior de una función que sigua el protocolo “System V AMD64 ABI”, también llamados salva invocante?

a) RAX, RCX, RDX, RDI, RSI, R8, R9, R10 y R11.

¿Cuál será el código de una función cuya definición es int f(int a, int b, int c) y que se implementa así?

```
    mov %esi , %eax  
    ret
```

a) return b;

¿Cuál sería el código de una función cuya definición es int f(int a, int b, int c) y que se implementa así?

```
    mov %edx , %eax  
    ret
```

d) return c;

¿Cuál sería el código de una función cuya definición es int f(int a, int b, int c) y que se implementa así?

```
    mov %edi , %eax  
    ret
```

a) return c;

¿Dónde se pasan los primeros 6 parámetros a una función mediante el protocolo “System V AMD64 ABI”?

b) En registros.

¿A qué función puede corresponder el siguiente código ensamblador?

```
    cmp %esi , %edi  
    mov %esi , %eax  
    cmovle %edi , %eax  
    ret
```

b) int f (int a , int b)

```
{  
    if ( a > b ) return a ;  
    else return b ;  
}
```

¿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 ∩, con un tramo ascendente y otro descendente.
 - c) Una escalera decreciente con varios tramos horizontales.
 - d) Una escalera creciente con varios tramos horizontales.**
-

¿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 MB, L2 en GB, L3 en GB o TB.
 - b) L1 en KB, L2 en MB, L3 en GB.
 - c) L1 en KB, L2 en KB o MB, L3 en MB.**
 - d) L1 en MB, L2 en MB, L3 en GB.
-

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**
-

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) 32 / 256 / 8192 usuarios
 - d) 48 / 384 / 23040
usuarios
-

En el programa line.cc de la práctica "cache", 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, se accede con éxito a más posiciones del vector en niveles de jerarquía de memoria más rápidos.
 - 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, 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.
 - d) Cada vez que line aumenta al doble, realizamos la mitad de accesos al vector que para el valor anterior.**
-

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

- a) Si continuáramos multiplicando por 2 el tamaño del vector en el eje X obteniendo más puntos de la gráfica, esta continuaría horizontal para cualquier valor más allá de 64 MB.
- b) La diferencia de velocidades entre L2 y L3 es mayor que la diferencia de velocidades entre L1 y L2.

- 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.
-

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) sacar un uno (1) por el stream line
 - 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 excede la longitud del vector
 - d) salir del bucle si el tamaño de línea se volviera menor o igual que 1 para algún elemento del vector
-

En el programa line.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**
-

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
-

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 (todo son fallos)
 - c) **1/2 (mitad aciertos, mitad fallos)**
 - d) 1/8 (un fallo por cada 8 accesos)
-

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 /, una gráfica siempre creciente y sin escalones
 - b) Una escalera con varios tramos horizontales**
 - c) Forma de U (o V) invertida, con un tramo ascendente y otro descendente
 - d) Forma de U (o V) con un tramo descendente y otro ascendente
-

En la práctica de cache hemos hecho una gráfica con el código line.cc ¿Qué forma tiene la gráfica que se puede obtener?

- a) Forma de U (o V), con un tramo descendente y otro ascendente
- b) Forma de /, una gráfica siempre creciente
- 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

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, 10, 16, 20, 30, 32, 40, 50, 60, 64, la tasa de acierto es 0,50. ¿Cuál es el tamaño de línea de la cache?

- a) 4 bytes
 - b) 8 bytes
 - c) **16 bytes**
 - d) Ninguno de los anteriores
-

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
-

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:

- a) La cache L1 es cuatro veces más rápida que la cache L2
 - b) **La cache L1 es al menos cuatro veces más de 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 seis veces más rápida que la cache L2
-

En el programa line.cc de la práctica de cache, para cada tamaño de línea (line) recorremos 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.
-

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

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>:
```

...el código numérico (pin) es...

- a) el entero 0x804a044
 - b) el entero 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 cuya dirección está almacenada en la posición de memoria 0x804a044
-

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 <main+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 <main+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 0x200819+0x400847**
 - d) el entero almacenado a partir de la posición de memoria 0x4009fb
-

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...

Usuaria Profesores

- a) el valor que tenga %eax
 - X** 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**
-

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 valor que tenga %eax
 - b) el string almacenado a partir de 0x80486f6
 - c) el entero almacenado a partir de donde apunta %eax
 - d) **ninguna de las anteriores**
-

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...

Usuaria Profesores

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

¿Para qué se utiliza la función scanf() en la práctica de la "bomba digital"?

Usuaria Profesores

- a) Para leer la contraseña (clave alfanumérica)
 - ✓ • b) **Para leer el PIN (clave numérica)**
 - c) Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la clave
 - d) Para escanear el fichero ejecutable “bomba” y asegurarse de que no contenga virus
-

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:

Usuaria Profesores

- X a) 0x080485e
 - b) 0x0804853
 - c) **0x000006d**
 - d) 0x50
-

Un estudiante entrega la solución de una bomba y en ella aparece el comando de gdb “p(char*)\$rdi”. Este comando:

Usuaria Profesores

- X 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
-

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...

Usuaria Profesores

- X**
- 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
-

En la placa del kit de Arduino, las patillas de tierra vienen etiquetadas con la leyenda:

Usuaria Profesores

- a) A0
 - b) 5V
 - c) **GN**
D
 - d) 3.3V
-

¿Cuál de las siguientes instrucciones máquina copia en el registro RDX la dirección efectiva resultante de la operación RAX*8 + RBX?

Usuaria Profesores

- X**
- a) movq (%rbx, %rax, 8), %rdx

-
- b) movq 8(%rdx, %rax), %rdx
 - c) **leaq (%rbx, %rax, 8), %rdx**
 - d) leaq 8(%rdx, %rax), %rdx

Suponga la expresión: unsigned int val = x & 0x01010101; donde x es un unsigned int. ¿Cuál de los siguientes valores de x da como resultado val = 0x01010101?

Usuaria Profesores

- X**
- a) 0x1010101
0
 - b) 0aaaaaaaa
a
 - c) 0eeeeeeee
 - d) **0x13579bd**
f

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?

Usuaria Profesores

- a) 0xFFFF FFFF
- b) 0x7FFF FFFF
- c) **0x1000 0000**
desplazado 4 bits a la izquierda se sale un bit
- d) 0xFFFF FFFF

La función setup() de Arduino es llamada:

Usuaria Profesores

- a) Cuando se sube desde el entorno de desarrollo o se pulsa el botón de reset, pero no cuando se conecta la alimentación
- b) Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, pero no cuando se sube desde el entorno de desarrollo

- c) Cuando se conecta la alimentación a la placa, se pulsa el botón de reset, o se sube desde el entorno de desarrollo
 - ✓ X d) Al principio de cada iteración de la función loop()
-

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

Usuaria Profesores

- 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
-

En la práctica de la cache, el código de size.cc accede al vector saltando de 64 en 64. ¿Por qué?

Usuaria Profesores

- a) Para recorrer el vector más rápidamente
 - b) Para anular los aciertos por localidad espacial, esto es, que sólo pueda haber aciertos por localidad temporal
 - c) Porque el tamaño de cache L1 de todos los procesadores actuales es de 64KB
 - X d) Porque cada elemento del vector ocupa 64 bytes
-

Si rax contiene x, ¿cuál de las siguientes instrucciones calcula x^9 ?

Usuaria Profesores

- a) leaq 8(%rax,%rax),%rdx
 - b) leaq (%rax,%rax,8),%rdx
 - c) leaq 3(%rax,%rax,2),%rdx
 - X d) leaq 5(%rax,%rax,4),%rdx
-

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 ...
sumas:
    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:

Usuaria Profesores

- a) con lista: .int 0x10000000, ... (16 elementos)
materialmente imposible, salva sólo mov %eax,
resultado
 - b) **con lista: .int 200000000, ... (16 elementos)**
3.2G cabe en 32 bits, al no haber acarreo no importa
adc
 - c) con ambos ejemplos
 - d) con ninguno de los dos ejemplos
-

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:    .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:

Usuaria Profesores

- a) para lista: .int 1, 2, 1, 2, ... (16 elementos)
 - b) para lista: .int -1,-2,-1,-2, ... (16 elementos)
 - c) para ambos ejemplos
 - d) para ninguno de los dos ejemplos
-

Suponga la expresión: unsigned int val = x & 0x01010101; donde x es un unsigned int. ¿Cuál de los siguientes valores de x da como resultado val = 0x01010101?

Usuaria Profesores

- a) 0x10101010
- b) 0aaaaaaaaa
- c) 0eeeeeeee
- d) **0x13579bdf**

//kkkkkkk

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
```

- a) objdump -t p2.o
- b) **objdump -d p2.o**
- d porque es un desensamblado (no cabeceras -h ni tablas -t)

- p2.0 porque las direcciones empiezan en 0 (un ejecutable x86 empezaría en 0x080480xx)
- c) objdump -d p
 - d) objdump -h p

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:

Usuario Profesores

- 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
-

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:

- > a. ambos nos muestran el string (el mensaje “Hola mundo”) Práct.2, pág.7, Fig.11
- b. el primero (x) nos muestra el string, y el segundo (print) nos muestra otro valor distinto
- c. el segundo (print) nos muestra el string, y el primero (x) nos muestra otro valor distinto
- d. alguno o ambos contienen algún error gramatical (falta o sobra algún &, algún typecast (char*) ó (long), etc)

¿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 tecleada por el usuario
 - > c. Para leer el PIN o código numérico //P4 pág.3 Fig.1**
 - d. Para lanzar un error cuando el usuario tarde demasiado tiempo en introducir la contraseña o el PIN
-

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 %0x080486ef=0xeb
- **set
(char)0x080486ef=0xeb**
- d) set *(char*)0x080486f6=jmp

PREGUNTAS EC TESTS

SOLO TEMA 1

No en todas las instrucciones máquina hay una fase de

- captura de operandos

La exo-arquitectura de una máquina especifica el conjunto de instrucciones, los modos de direccionamiento, la longitud de palabra, el nº de palabras de memoria, etc.

-Verdadero

El objetivo de un diseño CISC es...

- disminuir el número de instrucciones a ejecutar por un programa.

Una memoria que está estructurada en palabras de 8 bits tiene una capacidad de 64 Kbits.
¿Cuántas líneas de dirección tiene dicha memoria?

-13

¿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.

¿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

¿En qué registro está contenido el último dato (o instrucción) leído de memoria, o el dato que se va a escribir en memoria?

-MBR

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

El direccionamiento directo a memoria utiliza

- un desplazamiento

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

- 0xe0 en 0x0804913c y 0x1f en 0x08049140

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

Son funciones de la unidad de control:

- el secuenciamiento de las instrucciones máquina

¿Cuál de las siguientes afirmaciones es incorrecta?

- El tamaño de una instrucción en lenguaje máquina siempre ocupa dos bytes en los procesadores RISC.

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

¿De qué depende el tamaño del contador de programa?

- Del número de direcciones de memoria.

Una doble palabra está formada por 32 bits en cualquier ordenador.

-Falso

SP y PC no son registros de uso general (GPR).

-Verdadero

Si queremos almacenar la palabra de 16 bits 0x9660 en una memoria de bytes según "little-endian", quedará almacenada a partir de la posición 0x1000 como:

- $M[0x1000]=0x60$ y $M[0x1001]=0x96$

¿Cuál de las siguientes afirmaciones es incorrecta?

- Los operandos siempre están almacenados en memoria.

En una arquitectura RISC típica:

- se usan muchas instrucciones de las disponibles en el conjunto de instrucciones.

El direccionamiento directo a memoria utiliza...

- un desplazamiento.

¿Cuál de las siguientes afirmaciones es cierta?

- 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)

En una arquitectura RISC típica:

- suele usarse segmentación

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.

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

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) directo o absoluto
- c) indirecto a registro través de memoria
- d) ninguno de los anteriores

En el direccionamiento inmediato, tras captarse completamente la instrucción:

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

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 0xCAFEBAB0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al consultar la media palabra de la posición 0xCAFEBABE?

- -1

(los contenidos son CAFEBA0: FF FF FF FF

CAFEBAB4: FE FF FF FF

CAFEBAB8: FD FF FF FF

CAFEBABC: FC FF FF FF

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

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}

SOLO TEMA 2

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

- memoria-memoria

¿En qué orden debería ejecutarse en una máquina de tipo pila la operación aritmética ($a+b/c-d$)?

- a b c / + d -

Un overflow nunca puede ocurrir cuando:

- se suma un número positivo a un número negativo (resultado queda entre ambos => se puede representar)

¿De qué depende el tamaño del contador de programa?

- a) de la longitud del código de operación
- b) ninguna de las otras respuestas es cierta
- c) el tamaño no importa
- d) del ancho del bus de datos

¿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?

- %esp + 0x4

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

- Falso

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

- WORD

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

- `movl 8(%ebp), %eax`

Un archivo .o que contiene código objeto:

- Contiene instrucciones máquina binarias.

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:`

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

- Falso

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

- CF == 1

Diferencias gcc Linux IA32/x86-64: marcar la respuesta falsa

- el tipo double pasa de 4 B a 8 B

Cuando un operando se encuentra almacenado en un registro, se trata de un direccionamiento...

- directo a registro

El análisis dinámico de la frecuencia de utilización de instrucciones se realiza mediante el estudio de un gran número de listados de programas.

-Falso

Tras comparar números con signo, para saltar si menor se utilizan:

- El overflow y el signo

En las arquitecturas RISC hay...

- muchos registros y pocos modos de direccionamiento.

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 ...})
```

El resultado de desplazar aritméticamente dos posiciones hacia la derecha el número de 8 bits en complemento a dos -32 es:

- -8

¿Cuál de las siguientes listas está correctamente ordenada temporalmente?

- 8088, 80486, Pentium, Pentium Pro, Pentium II Pentium 4.

¿Cuál sería el "equivalente x86-64" del "pseudo-código C" rcx = ((int*)rax)[rcx]?

- mov (%rax,%rcx,4),%rcx

La instrucción NOP no se usa nunca, ya que no hace nada.

-Falso

Sean un int*a y un 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

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.

-Falso

En un diseño de un repertorio de instrucciones máquina suele suceder que no todas las combinaciones de bits corresponden a instrucciones válidas.

-Verdadero

En los casos concretos indicados para las siguientes instrucciones IA32, ¿cuál no funciona como instrucción de transferencia?

- cmpl 0x08048040, %eax

En IA32, ¿cuál de las siguientes afirmaciones es incorrecta?

- CALL ETIQUETA guarda en la pila la dirección de retorno y el registro de indicadores

Si d es un desplazamiento, r un registro índice e i una constante apropiada, el modo de direccionamiento indexado con postautoincremento realiza...

- dirección efectiva = r + d ; r = r + i

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

En el modo de direccionamiento implícito, el objeto direccionado es una constante contenida en la propia instrucción.

-Falso

SOLO TEMA 3

Usando chips bit-slice de 4 bits se pueden construir procesadores 4, 8 ó 12 bits, pero no más.

-Falso

La microprogramación vertical se caracteriza por tener:

-muchas codificaciones

Si se emplea una memoria de control del tipo RAM, es posible cambiar el repertorio de instrucciones de un ordenador sin necesidad de volver a diseñar y fabricar su hardware.

-Verdadero

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?

-7

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

En una arquitectura RISC típica:

- se usan muchas instrucciones de las disponibles en el conjunto de instrucciones.

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

En un procesador RISC, las bifurcaciones no degradan las prestaciones del "pipeline".

-Falso

Todo cauce ("pipeline") de instrucciones con 5 etapas tarda 7 ciclos de reloj en ejecutar 3 instrucciones si éstas utilizan las cinco etapas.

-Falso

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

-Verdadero

En la captación de la instrucción:

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

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

En general, una 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.

-Verdadero

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

Son funciones de la unidad de control:

- el secuenciamiento de las instrucciones máquina

Una máquina superescalar es aquella que:

- emite simultáneamente múltiples instrucciones por ciclo de reloj, por ejemplo, una entera y otra de coma flotante.

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

Para realizar la microoperación MAR <- PC, habrá que activar:

- EnPC y LdMAR

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

Se suelen utilizar PLA en las unidades de control cableadas.

-Verdadero

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

Cuanto más horizontal es la microprogramación más largas son las microinstrucciones.

-Verdadero

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?

-No se produce ahorro

PC y SP son dos registros de uso general (GPR).

-Falso

La siguiente notación, usada para algunos de los registros más frecuentes del procesador, es correcta: PC = contador de programa, SP = puntero de pila, IR = registro índice, MBR = registro de dirección de memoria, MAR = registro intermedio de memoria o registro de datos de memoria.

-Falso

Al finalizar la fase de captación del código de operación, si el modo de direccionamiento es indexado, EA = [PC] + X, siendo X el contenido del registro índice y PC el contenido del contador de programa.

-Verdadero

La microprogramación vertical se caracteriza por tener:

- mucha codificación

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

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

En una arquitectura RISC típica:

- se usa un porcentaje elevado de las instrucciones del repertorio.

SOLO TEMA 4

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

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.

-Verdadero

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

Los riesgos de datos consisten en que...

- una instrucción necesita un dato calculado por otra anterior

Las instrucciones de salto...

- complican el diseño eficiente de los procesadores segmentados.

Sólo los microprocesadores RISC utilizan segmentación en la ejecución de sus instrucciones.

-Falso

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

¿Cuál de los siguientes modos de direccionamiento es menos preferible para un procesador de 32 bits y con tamaño de instrucción de 32 bits?

- directo (o absoluto)

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

El Pentium III es superescalar porque permite "emitir" varias instrucciones por ciclo de reloj.

- Verdadero

No es posible diseñar un microprocesador de 1 GHz que ejecute 2000 MIPS.

- Falso

¿Cuál de los siguientes modos de direccionamiento es *menos* preferible para un procesador con segmentación de cauce?

- Indirecto a través de memoria

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

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

Respecto a la segmentación, ¿cuál de las siguientes afirmaciones es falsa?

- Retrasar la fase de decisión saltar/no saltar de las instrucciones de salto condicional contribuye a mejorar el rendimiento del procesador

Es posible diseñar un microprocesador que ejecute más de 100 MIPS, trabajando a una frecuencia de reloj de 100 MHz.

- Verdadero

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:

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

-Verdadero

Un microprocesador es superencauzado ("superpipelined") si permite la emisión de dos o más instrucciones en un mismo ciclo de reloj.

-Falso

En una CPU con segmentación de cauce, una instrucción tarda un ciclo de reloj desde que se va a captar de memoria hasta que terminan todas sus fases de ejecución y almacenamiento de resultados.

-Falso

A fin de acelerar la ejecución de instrucciones, algunas veces resulta posible iniciar la operación de captación de la siguiente instrucción antes de completar la ejecución de la instrucción presente.

-Verdadero

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

-Verdadero

Un procesador de 1GHz tarda 4ns en realizar 4 instrucciones sin realizar segmentación de cauce. ¿Cuanto tardaría en realizar 9 instrucciones una versión de dicho procesador con segmentación de cauce de 4 etapas si no existiera ningún retraso en ninguna de las instrucciones?

-3 ns

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:

-4ns

En un procesador RISC, las bifurcaciones no degradan las prestaciones del "pipeline".

-Falso

Los riesgos de datos consisten en que...

- una instrucción necesita un dato calculado por otra anterior

La precaptación (cola de instrucciones) está relacionada con...

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

En la secuencia de instrucciones siguiente, siendo el primer registro el destino, ¿cuántos riesgos se dan? sub r2,r1,r3 or r8,r6,r2

- Un riesgo por dependencia de datos

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

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:

-4

TESTS DE TODO MEZCLADO

¿Cuál de las siguientes no es una característica de los procesadores RISC?

- Las funciones que realizan los computadores RISC deben ser lo más complejas y potentes que sea posible.

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

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

-Push

Una ventaja de la realización de una unidad de control cableada mediante una PLA en lugar de utilizar componentes LSI es su mayor facilidad de diseño y fiabilidad.

-Verdadero

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

-ebx

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

-0xABFC

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

- el bit de cero

Un microprocesador es superencauzado ("superpipelined") si permite la emisión de dos o más instrucciones en un mismo ciclo de reloj.

-Falso

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

-Word

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.
- c) Las dos respuestas a y b son correctas.
- d) Ninguna de las anteriores es correcta.

En la captación de la instrucción:

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

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 aproxima a:

-4

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)

La primera generación de computadores se caracteriza por el uso de:

-Tubos de vacío

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

- Ninguna o escasa codificación.

¿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

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.

-Verdadero

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.

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.

-Falso

La codificación en bloque del código de operación utiliza menos bits para las instrucciones más frecuentes y más bits para las instrucciones menos utilizadas.

-Falso

No en todas las instrucciones máquina

- hay una fase de captura de operandos

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).

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

-Estructural

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

- 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

Suponga que la micropalabra de una máquina microprogramada tiene 16 bits de ancho y se usan 256 micropalabras diferentes en un microprograma de 512 micropalabras. No se ahorran bits usando nanoprogramación.

-Verdadero

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

- se pueden usar dos buffers triestado.

¿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

En IA32, el registro contador de programa se denomina:

-EIP

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:

- no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

Un archivo .o que contiene código objeto:

- Contiene instrucciones máquina binarias.

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 producir 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) $x \leq -1 \text{ || } x \geq 11$
- d) $x \geq 0 \text{ && } x \leq 10$

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 micropogramada.
- 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) Sigue 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 $RDX*8 + RBX$?

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

`lea` calcula la dirección efectiva $[RBX+RDX*8]$ 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 $8+2*RDX$

- d) `movq 8(%rdx, %rdx), %rax`

`mov` mueve desde memoria, y además la dirección efectiva sería $8+2*RDX$

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 NTeo>=2.5p y NPrá>=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?
- Un protocolo sirve para “ponerse de acuerdo” en cosas como velocidad, paridad, nº de bits, etc.
 - La mayoría de los periféricos trabajan a velocidad muy superior a la CPU; por eso es necesario sincronizar.
 - Un controlador se encarga de la comunicación con la CPU.
 - 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?
- Sí, permite averiguar el estado de un periférico y también alterar la prioridad de los periféricos
 - No, porque no permite ni averiguar el estado de un periférico ni alterar la prioridad de los periféricos
 - No, porque no permite averiguar el estado de un periférico
 - 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?
- 23 ns
 - 32 ns
 - 57.5 ns
 - 92 ns
67. Cuando dos o más instrucciones necesitan un recurso hardware en el mismo ciclo, se trata de un riesgo:
- estructural
 - de control
 - de salto
 - por dependencia de datos
68. En el arbitraje de un bus...
- los dispositivos pasivos pueden requerir el uso del bus para iniciar una transferencia
 - si hay un único dispositivo pasivo, siempre funciona como esclavo
 - si hay varios dispositivos activos, siempre funcionan como maestros
 - 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?
(Ob 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) 0aaaaaaaaa
 - c) 0eeeeeeeee
 - 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,Indice,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 CR0 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 únicamente 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 acoplamiento, 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) movzlq %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 ∩, 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



RullenCastro

www.wuolah.com/student/RullenCastro



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 <strcmp>

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

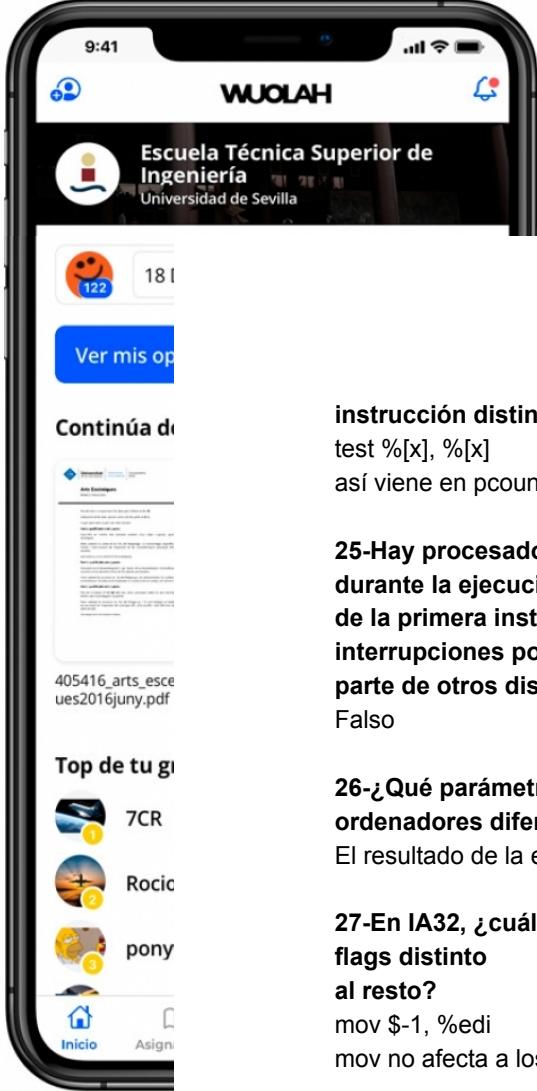
23-En la práctica "media" se suma una lista de 32 enteros de 4 B con signo para producir una media y un resto usando la instrucción IDIV. ¿Cuál de las siguientes afirmaciones es falsa?

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.

Available on the
App Store

GET IT ON
Google Play

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 -

"I da 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

direcciónamiento 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 direcciónamiento 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

**Ids IOBuf ; Apuntar puntero pila a
; ...área mem.intermedia
Idx 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·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

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 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.

Available on the
App Store

GET IT ON
Google Play

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 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.

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 int*a y un 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

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

1^a 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 llenar 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 envia 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)  
);
```

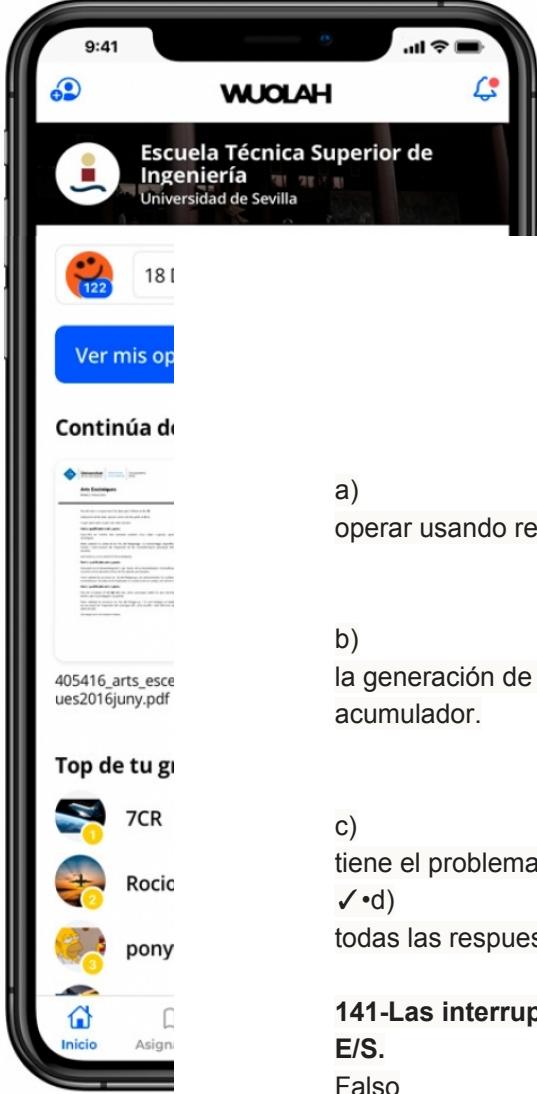
¿Cuál de las siguientes afirmaciones es correcta?

- 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.

Available on the
App Store

GET IT ON
Google Play

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.

Verdadero

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

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 0xCAFEBAB0 cuatro palabras con valores -1, -2, -3, -4. ¿Qué se obtendría al consultar la media palabra de la posición 0xCAFEBABE?

-1

los contenidos son

CAFEBAB0: FF FF FF FF

CAFEBAB4: FE FF FF FF

CAFEBAB8: FD FF FF FF

CAFEBABC: FC FF FF FF

las últimas dos posiciones, a partir de CAFEBABC, 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

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 micropograma 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^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

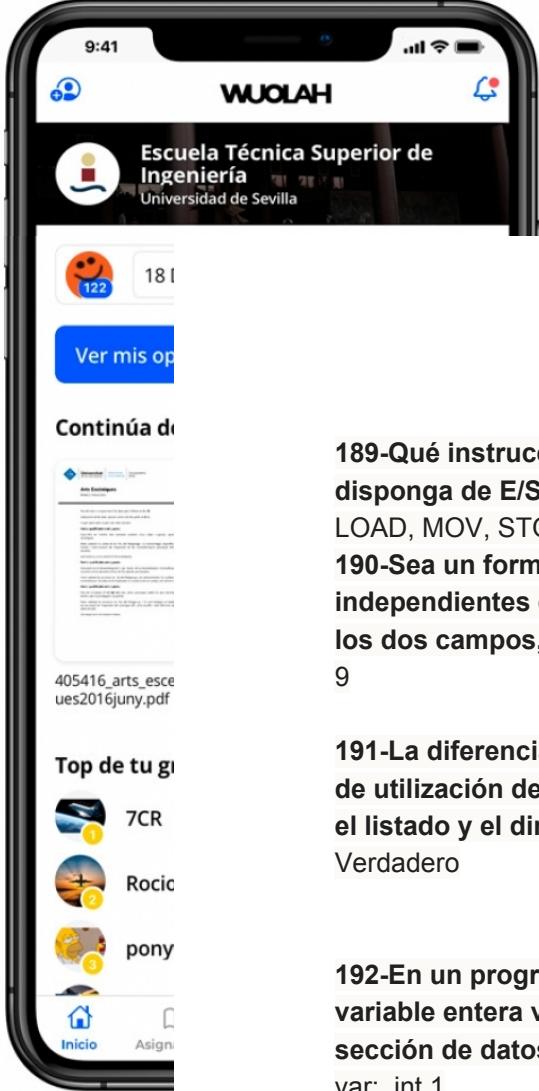
opuestos a programas secuenciales son los programas paralelos

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.

Available on the
App Store

GET IT ON
Google Play

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:

0xCAFEBAC0

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
- 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

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

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 micropograma.

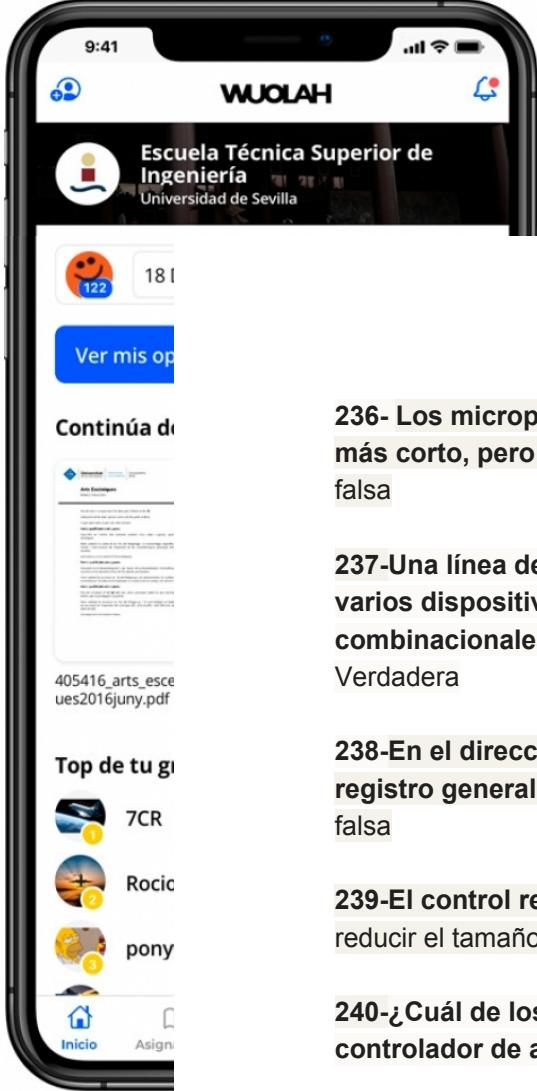
Verdadero

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.

Available on the
App Store

GET IT ON
Google Play

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.
veradero

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

veradero

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

veradero

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



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 ffffffe0 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/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 5^a 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)

$\ast(\ast(x + 2)) + 7$

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)

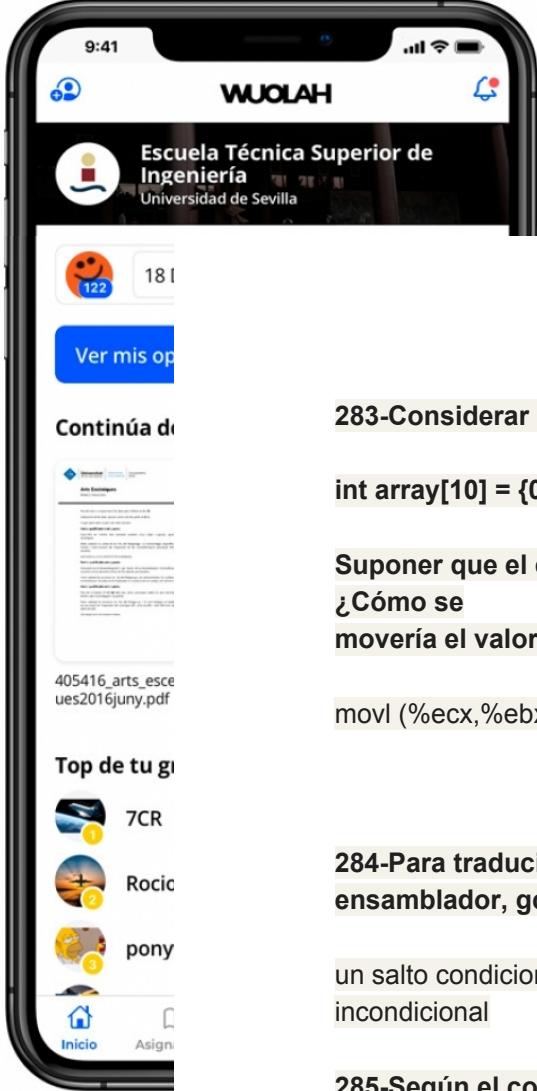
se evita el cuello de botella (por ejemplo, pila, o acumulador) que otras arquitecturas presentan al evaluar expresiones aritméticas complejas

✓ 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.

Available on the
App Store

GET IT ON
Google Play

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-EI 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 del 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



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 $n \cdot w - (n \cdot m + 2^m \cdot w)$.

✓

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

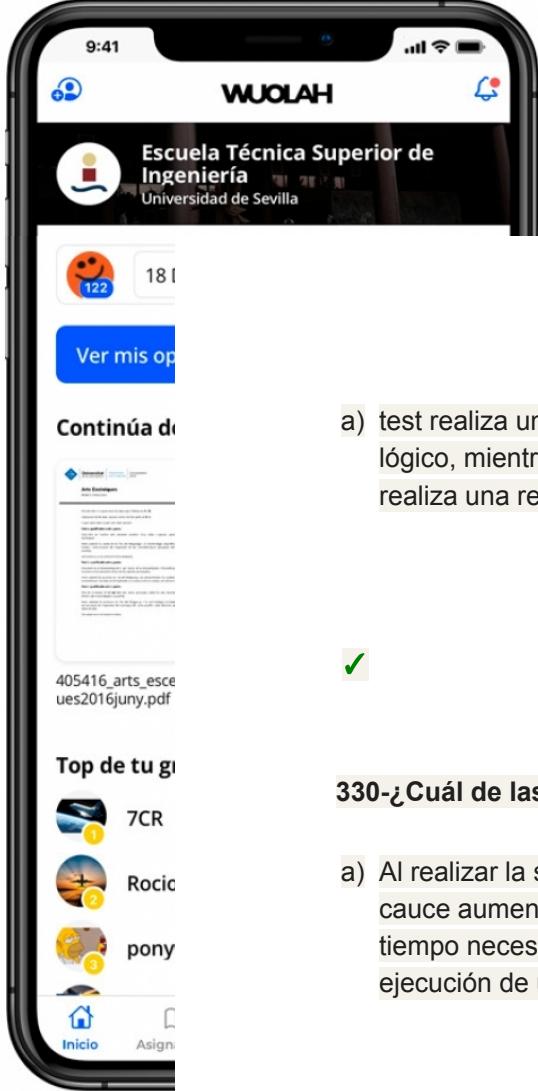
327-Más de la cuarta parte de las instrucciones máquina que se ejecutan en un programa típico son de movimiento o transferencia de datos.

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.

Available on the
App Store

GET IT ON
Google Play

- 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



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.

✓

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.

✓

376-Dada la siguiente definición de datos:

```
lista: .int 0x10000000, 0x50000000,  
        0x10000000, 0x20000000
```

```
longlista: .int (.lista)/4
```

```
resultado: .quad 0x123456789ABCDEF
```

```
formato: .ascii "suma=%lu=%lx 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.

✓

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

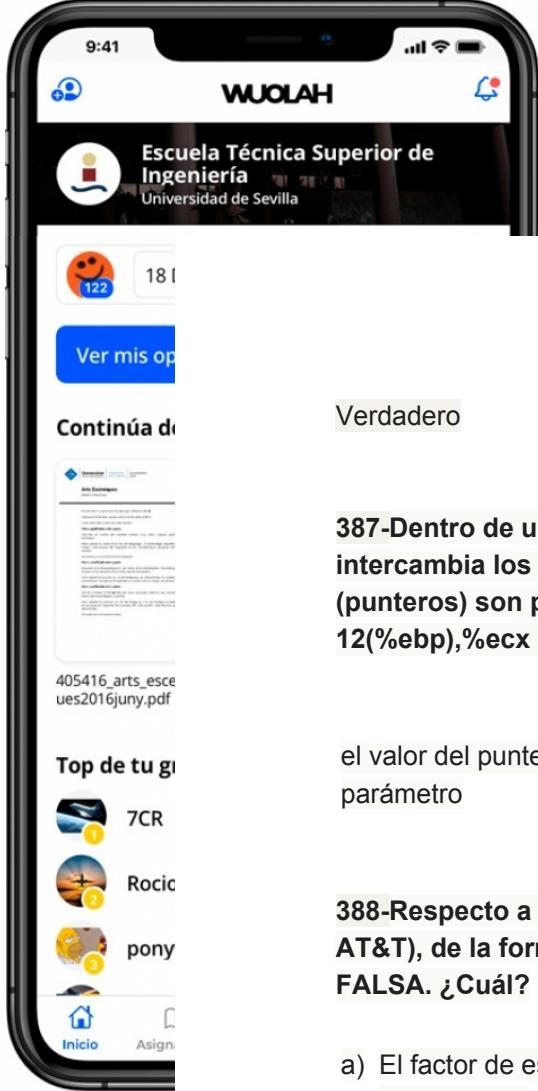
384-¿Cuántos controladores de interrupciones 8259 hacen falta como mínimo para manejar 25 líneas de interrupción?

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.

Available on the
App Store

GET IT ON
Google Play

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) geométric
- a

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, una 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

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

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 Id 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

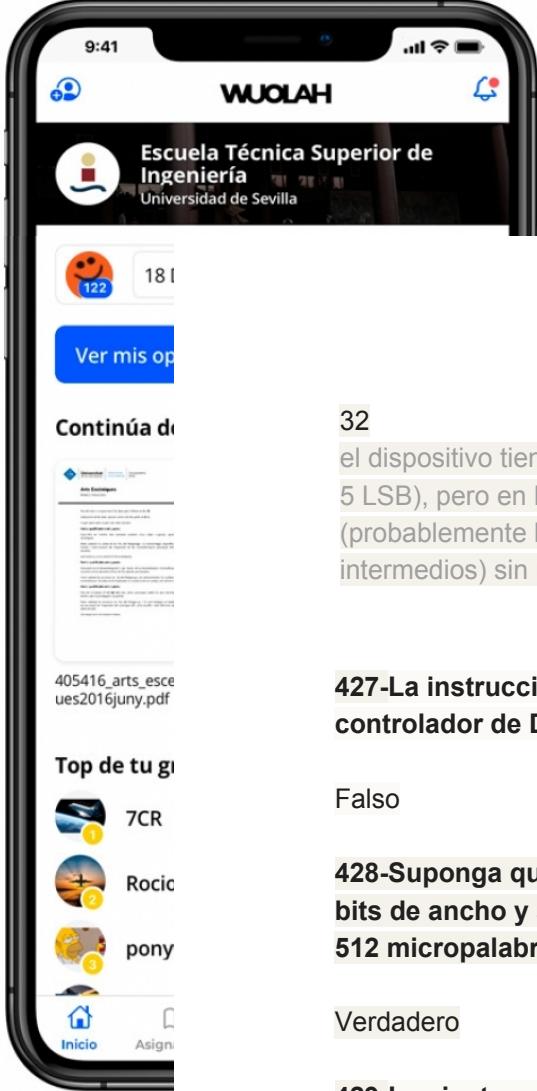
VERDADERO

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.

Available on the
App Store

GET IT ON
Google Play

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"
        );
    }
    return res;
}
```

```
"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

ringteacher



447-Para desplazar %eax a la derecha un número variable de posiciones <= 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 ^ 0b1010 en lenguaje C?

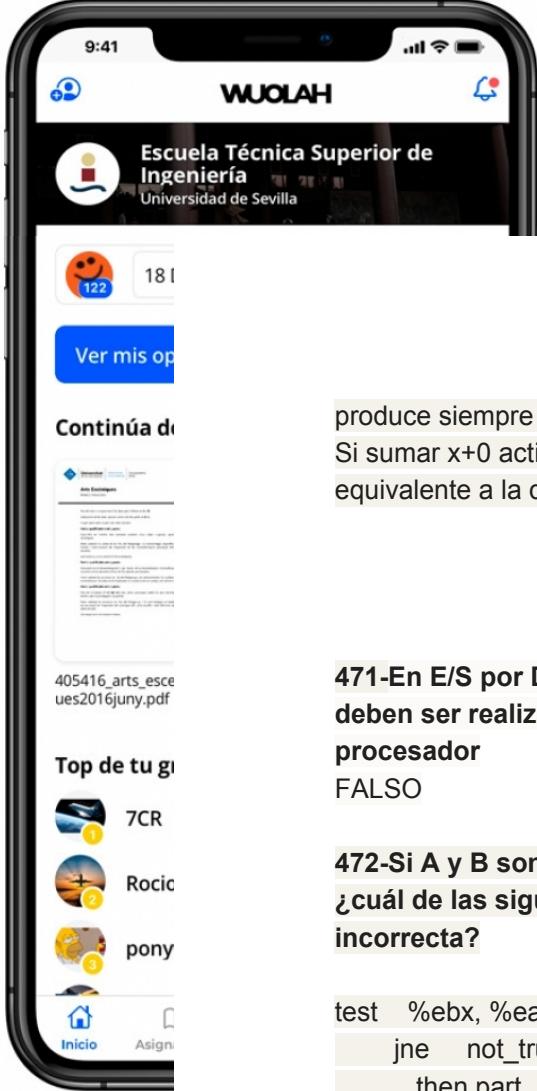
0b0100 (hace el AND y pone 1 cuando no es AND, 1 y 1 = 0, 0 y 1 = 1)

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.

Available on the
App Store

GET IT ON
Google Play

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.

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

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.

Available on the
App Store

GET IT ON
Google Play

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  
cltd  
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

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.

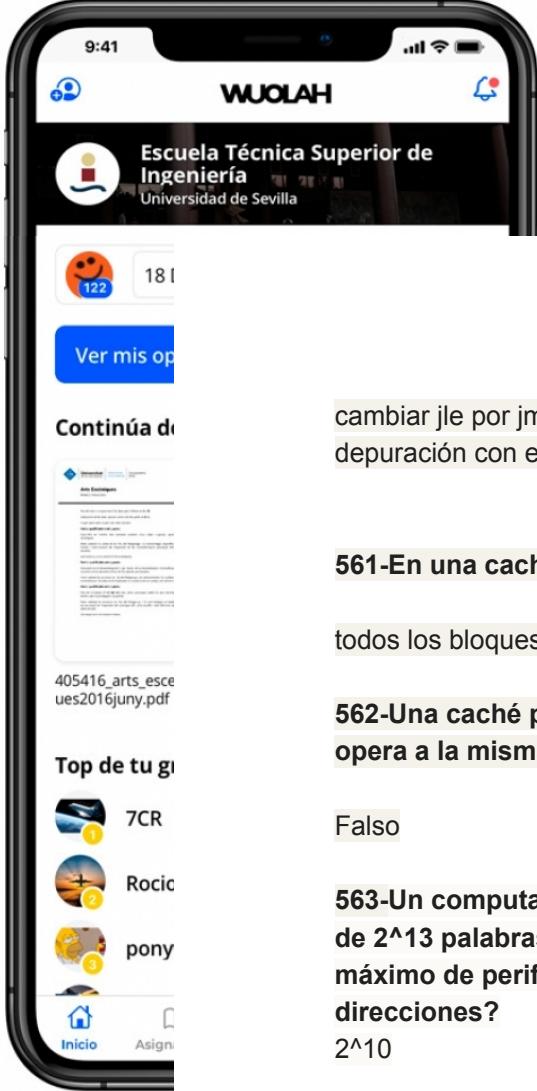
559-Es más caro emplear buses dedicados que no dedicados, pero ello permite establecer más comunicaciones simultáneamente, aumentando las prestaciones de un computador.

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.

Available on the
App Store

GET IT ON
Google Play

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=%lu=%lx 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

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)

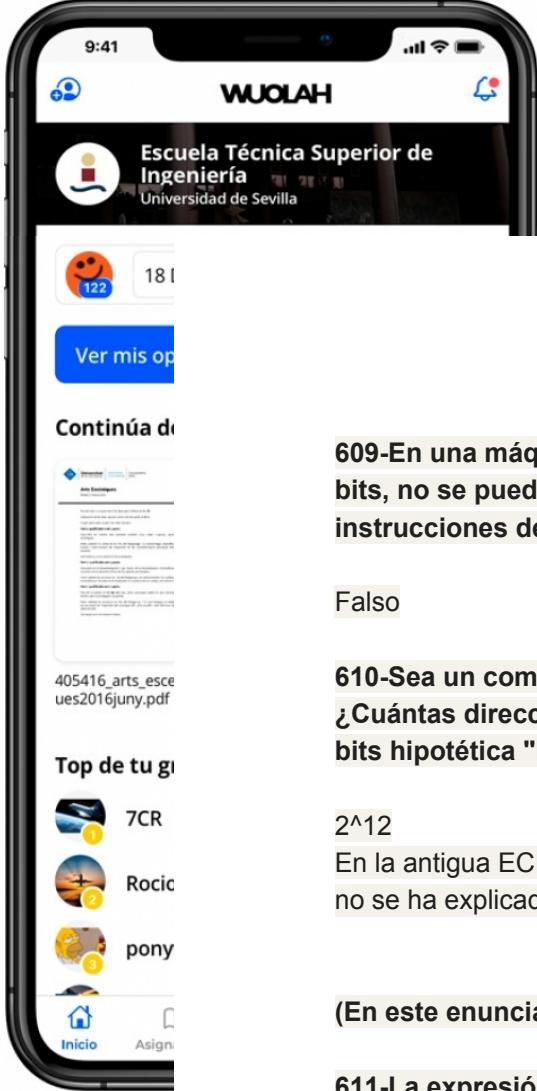
redacción original era "x86-64 Linux long double 8x, Windows no", motivo por el cual notamos errata transparencias (no dicen nada de x86-64 Windows, y dicen que x86-64 Linux long double 8x, cuando libro pág.325 dice que 16x). Posible corrección: "alinea long double a 12x (Windows no)". Redacción final 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.

Available on the
App Store

GET IT ON
Google Play

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 != 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:
muchas codificaciones

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

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?

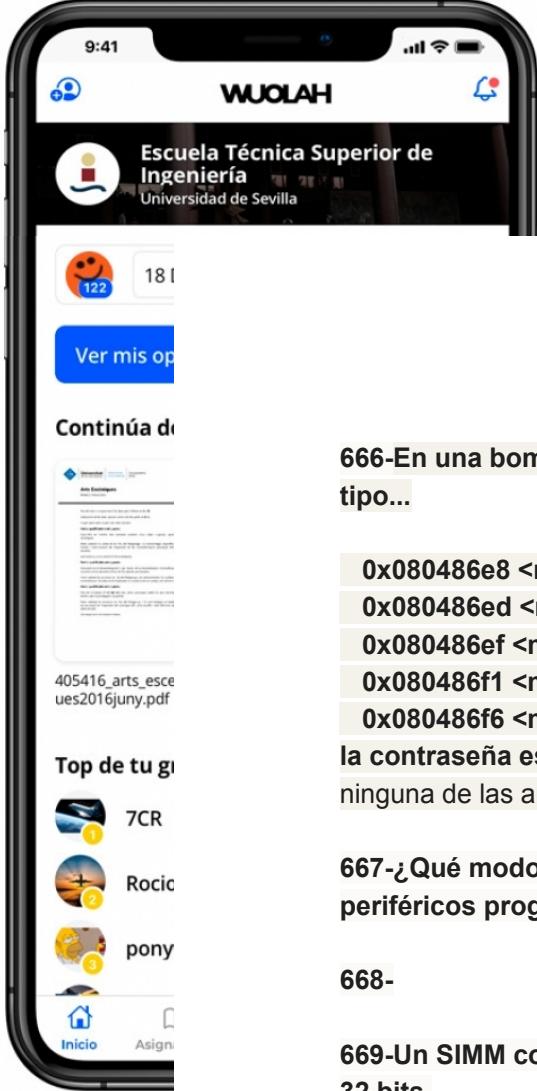
662-Una cache de 256 B asociativa por conjuntos de 4-vías con líneas de 16 B tendría

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.

Available on the
App Store

GET IT ON
Google Play

666-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...

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

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 <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

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-El 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".

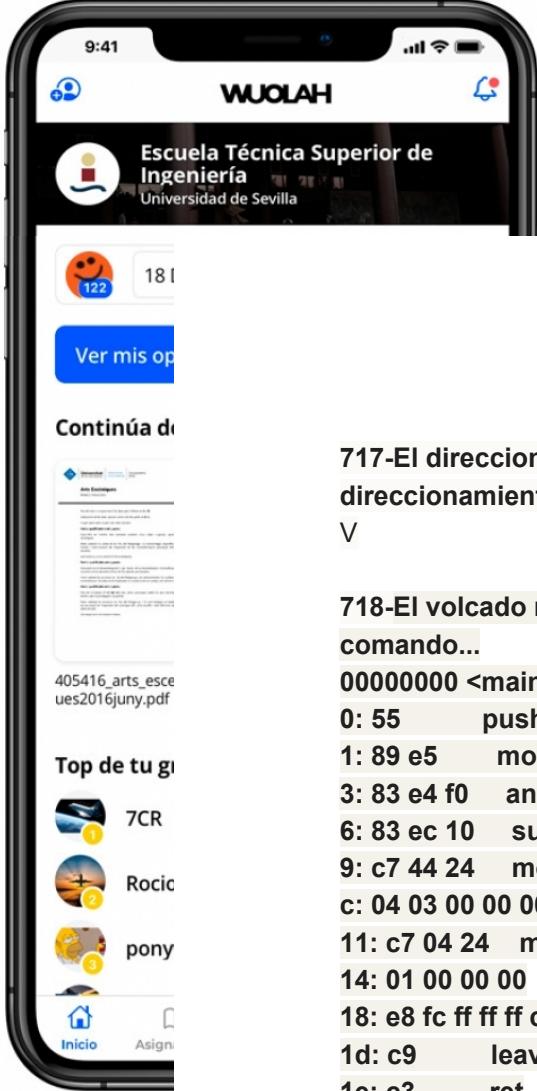
FALSO

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.

Available on the
App Store

GET IT ON
Google Play

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\n0"
```

```
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 <- 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
sal \$2, %eax
sería x5x4=x20

b)

imull \$0x10, %eax
sería x16

c)

addl %eax, %eax
shll \$5, %eax
sería x2x32=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=%lu=%lx 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 $0, 1, \dots, 2M-1$ 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

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?

3^a generación (1965-75)

en la 3^a 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?

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

PREGUNTAS EC TESTS

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

¿A qué tipo de memoria cache corresponde la siguiente afirmación: "permite que cualquier dirección se pueda almacenar en cualquier marco de bloque de cache"?

- Totalmente asociativa

RAS significa impulso de acceso a filas o selección de acceso a filas.

- Verdadero

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)solapamiento
- b)microbifurcaciones
- c) codificación
- d) longitud relativa de microinstrucción

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?

- 0xBEE3DE40

La instrucción setg %al:

- Pone AL a 1 en algunos casos.

¿Cuál de las siguientes afirmaciones es incorrecta?

- Todas las instrucciones deben tener operando fuente y operando destino

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?

- $k + n \cdot (2^m - 1)$

¿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 las instrucciones se escriben en binario.
- c) En lenguaje ensamblador cada instrucción se corresponde con una instrucción máquina.
- d) El lenguaje de alto nivel es más portable que el lenguaje máquina.

Suponga que una memoria de acceso aleatorio (RAM) tiene 2^k palabras de 2^p bits cada una. En tal caso sería común que el registro de dirección de memoria tuviera k bits y el registro intermedio de memoria tuviera 2^p bits.

-Verdadero

¿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

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

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

En una arquitectura Von Neumann existe un mismo espacio de memoria para datos y para programa.

-Verdadero

La instrucción ADD Rn,#3 suma el contenido del registro Rn con el de la posición de memoria 3.

-Falso

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

En arbitraje por "daisy-chain" la señal BUS_REQUEST es compartida por los maestros potenciales.

-Verdadero

Respecto al refresco de memorias DRAM, ¿cuál de las siguientes afirmaciones es falsa?

- a) Los chips DRAM refrescan automáticamente la fila accedida en cualquier ciclo de lectura o escritura.
- b) Los ciclos de refresco deben producirse cada pocos ms (milisegundos).
- c) Se precisa una circuitería auxiliar, externa al chip DRAM o integrada en él, que produzca ciclos de refresco.
- d) Una operación de refresco consiste en dar un impulso /CAS junto con una dirección de columna.

En un sistema x86-64, si %rsp tiene el valor 0x7fffff0000 inmediatamente antes de ejecutar una instrucción retq, ¿cuál es su valor inmediatamente después?

-0x7fffff0008

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

-0xC8 en EAX y 0x0 en EDX

El número de páginas direccionables con la dirección virtual puede ser superior al número de páginas direccionables con la dirección física.

-Verdadero

Las instrucciones de salto...

-complican el diseño eficiente de los procesadores segmentados.

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?

-400 millones de bytes por segundo

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

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

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

-Pone CF=0

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

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

El TLB tiene tantas entradas como número de páginas la memoria virtual.

-Falso

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 observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

-8 / 64 / 2048 estudiantes

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)?

-0xffff ffff

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:

-movl \$lista, %ebx

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

-Verdadero

En la arquitectura Von Neumann...

-el programa se encuentra residente en memoria.

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

-Falso

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

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.

¿Cuál de las siguientes expresiones toma el valor 0x01 si x es múltiplo de 32 y 0x0 en caso contrario? Asumir que x es unsigned int.

-!(x & 0x1f)

Se dice que las máquinas con arquitectura Von Neumann siguen un modelo de programa...

-Almacenado

Un computador con 10 bits en el bus de direcciones puede direccionar como máximo:

-1024 palabras

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:

-produce siempre el resultado correcto

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?

-0xAC60

¿Qué modelo de programa se ejecuta en las arquitecturas Von Neumann?

- Almacenado.

El sistema de paginación del Pentium utiliza dos niveles de tablas para traducir una dirección lineal a dirección física.

-Verdadero

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?

-1/2 (mitad aciertos, mitad fallos)

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

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

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

En las transferencias asíncronas no existe un reloj común a maestro y esclavo.

-Verdadero

Algunas de las ventajas de la E/S aislada frente a la E/S mapeada en memoria son:

-Los programas pueden ser más rápidos, y las instrucciones de E/S son fácilmente reconocibles

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 salidas de la unidad de control.

-Verdadero

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

El programa encuadernador (LINK) resuelve todas las referencias declaradas globales en los ficheros fuente.

-Verdadero

Algunas de las ventajas de la E/S aislada frente a la E/S mapeada en memoria son:

-Los programas pueden ser más rápidos, y las instrucciones de E/S son fácilmente reconocibles

En una operación de lectura en bus de ciclo completo se libera el bus entre el envío de la dirección y la recepción del dato.

-Falso

La práctica “parity” debía calcular la suma de paridades impar (XOR de todos los bits) de los elementos de un array. La siguiente función contiene un único error (en realidad se han editado 2 líneas de código sobre la versión correcta) pero produce resultado correcto cuando se usa como test el array...

```
int parity4(unsigned* array, int len){  
    int i;  
    unsigned x;  
    int val=0, result=0;  
  
    for (i=0; i<len; i++){  
        x = array[i];  
        asm("\n"  
            "ini:          \n\t"  
            "xor %[x], %[v]  \n\t"  
            "shr %[x]      \n\t"  
            "jnz ini      \n\t"  
            : [v]"+r" (val)  
            : [x] "r" (x)  
        );  
        result += val & 0x1;  
    }  
    return result;  
}
```

-array={5, 4, 3, 2}

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

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

¿Qué tipo de instrucciones se emplean más en una arquitectura de acumulador?

-de transferencia de datos con memoria

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

-Verdadero

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

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]

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:

-polling

¿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

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 += 2)
    v[i] = 9;
```

¿Cuál será la tasa de fallos aproximada que se obtiene en la primera ejecución del bucle anterior?

-1/8 (un fallo por cada 8 accesos)

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))))

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

-un registro y dos sobrescritos (clobber) (((el registro es [x] "+r" y los sobrescritos son "edx", "ecx"))))

¿Cuál de las siguientes afirmaciones acerca de la memoria es *FALSA*?

-Las celdas de memoria estática tienen que ser constantemente refrescadas

¿Cuál de los siguientes lenguajes no permite el paso de parámetros por referencia?

-C

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·C/F

El instrumento GIADA de la sonda espacial ROSETTA (diseñado en Granada) está basado en un microprocesador 8086 y el siguiente mapa de memoria:

RAM volátil: 00000 - OFFFF

RAM no volátil: 10000 - 1FFFF

ROM: F0000 - FFFFF

¿Cuál es el tamaño total de la memoria?

-192KB

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

-0x1e180

¿Cuál de las siguientes funciones es una tarea propia de la unidad de control en la CPU?

-decodificar las instrucciones del programa

Indicar 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

-0804837f

En la práctica de la cache, el código de size.cc accede al vector saltando de 64 en 64. ¿Por qué?

-Para anular los aciertos por localidad espacial, esto es, que sólo pueda haber aciertos por localidad temporal

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*)

-objdump

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

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

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

-coherencia de cache

¿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

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,

-es ventajoso para enlazarlo usar gcc, y entonces es conveniente que contenga main (aunque ambas cosas son opcionales)

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?

- a) La cache de postescritura no puede ser más eficiente que la de escritura inmediata
- b) La cache de postescritura siempre será más eficiente que la de escritura inmediata
- c) Más de 8 veces
 - d) No se puede responder con los datos proporcionados

La política de correspondencia de una memoria cache con 1 único conjunto es:

-Totalmente asociativa (((((todas las líneas son distintas vías del único conjunto -> cualquier bloque de memoria puede ir a cualquier marco de cache.))))

El switch de gcc para que únicamente compile de lenguaje C a ensamblador, y no realice ningún paso adicional (ensamblar, enlazar, etc), es...

- -S

¿Cuál de las siguientes instrucciones es incorrecta en ensamblador GNU/as IA32?

-pop %eip

En cdecl/x86, ¿cuál de los siguientes registros tiene que ser guardado por la función llamada si es alterado por ésta?

-ebx

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?

-60 bytes

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?

-duplicar el tamaño del salto en los accesos al vector respecto a la iteración anterior

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 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 observarán saltos en las prestaciones debidos a accesos a la jerarquía de memoria?

-8 / 64 / 2048 estudiantes

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

Si ECX vale 0, la instrucción adc \$-1,%ecx

-No cambia CF

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

-ebx

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

-Línea de reconocimiento INTA#

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;
```

¿Cuál es el tamaño de my_node en bytes?

-32

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\t 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:

-produce siempre el resultado correcto

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?

-34ns

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 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 valor que tenga %eax
- b) el string almacenado a partir de 0x80486f6
- c) el entero almacenado a partir de donde apunta %eax
- d) ninguna de las anteriores

¿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

¿Cuál de las siguientes afirmaciones es falsa?

- a) Los programas necesitan una forma de pasar parámetros a las subrutinas y de recibir las salidas de vuelta
- b) Las subrutinas necesitan recibir parámetros desde el programa que hace la llamada que indiquen qué registros pueden alterar y cuáles no
- c) 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
- d) 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

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;
}
```

Esta función parity4:

- produce siempre el resultado correcto

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

-0x1e180

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
```

En una arquitectura de acumulador, la instrucción LOAD X:

-transfiere el contenido de la posición de memoria X al acumulador

¿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?

```
-%esp + 0x4
```

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:

-Cada vez que line aumenta al doble, realizamos la mitad de accesos al vector que para el valor anterior.

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?

-8Bytes

La instrucción test es...

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

¿Cuántos operandos acompañan a la instrucción PUSH en IA32?

-1

Si queremos almacenar la palabra de 64bits 0x0000001f ffffffe0 en una memoria de bytes según la convención little-endian a partir de la posición 0x0804913c, quedará

-0xe0 en 0x0804913c y 0x1f en 0x08049140

En la práctica de la bomba, el primer ejercicio consistía en “saltarse” las “explosiones”, para lo cual se puede utilizar...

-gdb o ddd

¿Cuál de las siguientes funciones no corresponde a la unidad de control de un procesador?

-Calculo de operaciones de coma flotante

La conexión entre un dispositivo de E/S y el procesador mediante bus:

-Permite conectar en paralelo varios dispositivos

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

-LOAD, MOV, STORE

El switch -l para indicar librerías *NO* funciona con la herramienta...

-as

¿Cuál de las siguientes no es una característica de los procesadores RISC?

- a) Las funciones que realizan los computadores RISC deben ser lo más complejas y potentes que sea posible.
- b) La decodificación de las instrucciones debe ser simple: un computador RISC debería emplear un único formato de instrucción
- c) Para acelerar el computador RISC se emplean técnicas de pipelining.
- d) Un computador RISC no debe emplear microprogramación.

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

-Compilador

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

Alguno de los siguientes no es un nombre de registro en una máquina x86-64 en modo 64 bits

...-sih ((((%sil en todo caso, no %sih

- b) r8d
- c) spl
- d) r12w

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

-Los transistores

En la práctica "media" se pide usar cltq/cdq y cqto/cq 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:

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/idivq %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 0x7fffffff)

-media = 0x0fffffff, resto = 0

((RAX acumula correctamente 0x7 ffff fff0 y RDx se extiende a 0, pero al dividir por ECX en lugar de RDx sólo se divide EDX:EAX = 0x0000...: ffff fff0 dejando media EAX 0x0fff ffff resto EDX 0x0))))

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)...

-JNC y INC

Compilar .s → ejecutable, usando sólo as y ld, sin gcc...

-Se puede, usando en as y ld los modificadores (switches) que corresponda

¿Cuál de las siguientes no es una sección de un fichero ELF?

-static

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

- -m32

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

((n*32 == n << 5 y el bit 26 (0x04...) pasaría a 31 (0x80...) y el resultado llegaría *justo* a convertirse en negativo. Algo más pequeño que 0x04... no llegaría.)))

¿Qué valor contendrá %edx tras ejecutar las siguientes instrucciones?

```
xor %eax, %eax  
sub $1, %eax  
cltd  
idiv %eax  
-0
```

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:

-con ambos ejemplos

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:

-Fallaría con lista: .int 1,2,3,4, 1,2,3,4, ...
(((Debería salir suma=80 y sale suma=32 porque suma 32 veces lista[0]=1. $32 << 4G - 1$, así que no hay acarreos ni por asomo.))

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

¿Qué hace gcc -S?

-Compilar .c → .s (fuente C a fuente ASM)

En la práctica "media" se pide sumar una lista de 16 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.

-1 000 000 000

((((Se pasa, 1000 millones >> 256M 32bit sin signo llega a $2^{32}-1 = 4G-1 \sim 4000000000$. Habría acarreo si llegáramos a 4G. Con 16 elementos iguales, tendrían que valer $4G/16 = 1G/4 = 256M$.

En X86-64, el registro contador de programa se denomina:

-RIP

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...

-cambiar jle por jmp usando ddd o un editor hex, salvar el programa y reiniciar la depuración con el nuevo ejecutable

Abajo se ofrece el listado de una función para multiplicar matrices C = A x 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?

-1/2 (((1/2 por cada B[k][j] + 1/32 por cada 16-ésimo A[i][k]

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; 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 se diferencia de la versión "oficial" en los tipos de array y x.

Esta función `popcount5`:

-produce siempre el resultado correcto

((los 8 bits de signo añadidos no llegan al MSB de la máscara 0x01010101, justamente entonces termina el bucle for(j)

Una de las “bombas” utiliza el siguiente código para cifrar la clave numérica introducida por el usuario y ahora almacenada en eax:

804870d: xor \$0fff,%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:

-1 ((Tanto 0 como 1 servirían de contraseña. El enunciado de la bomba prohibía explícitamente casos como éste. La bomba debe ser un valor fijo. Uno, no dos.

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.

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?

-1/16

uno de cada 16 $A[i][k]$ y otro de cada 16 $B[i][k]$

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

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

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)?

-0xfffb ffff

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: .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 = -2, resto = 8
(((si se hubiera usado cqto se hubiera extendido correctamente RAX->RDX antes de dividir por RCX
al hacer mov \$0, %edx se considera RDX:RAX un número positivo grande (en lugar de -16E-8=-24 sale 16E-24=0x0000....: ffff ffff ffff ffe8)
al dividirlo con signo sale media RAX 0xffff ffff fffe resto RDX 0x8
al guardar RAX little-endian sale media -2 resto 8)))

El código del programa "size" de la práctica de la cache accede al vector saltando...

-de 64 en 64 bytes

((es el tamaño de línea deducido en la práctica "line"

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

Si val es una variable de tipo unsigned long, entonces la sentencia: val += (val >> 32);

-Su traducción incluye una instrucción shr seguida de una suma.

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
```

Estando bien programado todo lo demás, este código

-produce siempre el resultado correcto

((((inc %edx se hace después de adc, cuando ya se ha usado el valor de %edx

Recordar que usamos cltd para extender a %edx el elemento leído en %eax

Igual podríamos haber hecho xor %edx,%edx y tampoco pasaría nada

¿Cuál de las siguientes instrucciones es errónea? (sale mensaje de error al intentar ensamblar):

-movb \$0xFF, (%dl)

no, no se puede direccionar con (%dl)

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`.

-no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

En la práctica de E/S en Arduino, la instrucción SBIW del repertorio del microcontrolador realiza...

-una resta de un valor inmediato

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

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

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

-estructural

¿Qué tipo de localidad de las referencias a memoria se define como: "si se referencia un elemento, los elementos cercanos a él serán referenciados pronto"?

-Localidad espacial

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

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

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

-pila del usuario

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

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

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

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

-long int pasa de 4 B a 8 B

¿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

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

Comparando las convenciones de llamada de gcc Linux IA32 con x86-64 respecto a registros

- En IA32 %ecx es salva-invocante, y en x86-64 %rcx es salva-invocante también
%rcx es 4º argumento - salva-invocante, ok

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.

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:    .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)

¿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):

-file.c → file.s → file.o → file

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:
    -push resultado+4
    push resultado
    push resultado+4
    push resultado
    push $formato
    call printf
    add $20, %esp

```

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

-E/S mediante acceso directo a memoria

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;  
    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 ambos ejemplos

En x86-64, el registro contador de programa se denomina:

-RIP

En un sistema Linux x86-64, ¿cuál de las siguientes variables ocupa más bytes en memoria?

-int *c

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

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

-0804837f

¿Cuáles de las siguientes señales son entradas a la unidad de control?

-El contenido del registro de instrucción

¿Cuál de los siguientes lenguajes no permite el paso de parámetros por referencia?

-C

Para averiguar la paridad de un número se puede usar la operación:

-XOR

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?

-0x400549

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:

-La cache L1 es al menos cuatro veces más de rápida que la cache L2

Para comprobar si el entero almacenado en EAX es cero (y posiblemente saltar a continuación usando JZ/JNZ), gcc genera el código

-test %eax, %eax

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:

-2

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

-gcc

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

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:

-no es correcta pero el error no se manifiesta en los ejemplos propuestos, o se manifiesta en ambos

