grapa

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

75.573 14 01 12 EX

Espacio para la etiqueta identificativa con el código personal del **estudiante**.
Examen

Ficha técnica del examen

- **Comprueba que el código y el nombre de la asignatura corresponden a la asignatura de la cual estás matriculado.**
- **Debes pegar una sola etiqueta de estudiante en el espacio de esta hoja destinado a ello.**
- **No se puede añadir hojas adicionales.**
- **No se puede realizar las pruebas a lápiz o rotulador.**
- **Tiempo total** 2 **horas**
  - **En el caso de que los estudiantes puedan consultar algún material durante el examen, ¿cuál o cuáles pueden consultar?:** No se puede utilizar calculadora ni consultar material auxiliar.
  - **Valor de cada pregunta:** Pregunta 1 (20%); Pregunta 2 (40%); Pregunta 3 (40%).
- **En el caso de que haya preguntas tipo test: ¿descuentan las respuestas erróneas?** NO **¿Cuánto?**
- **Indicaciones específicas para la realización de este examen**

Enunciados

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Pregunta 1: (50%: 1.1: 10%, 1.2: 10%)

Solamente se deben completar las instrucciones marcadas.
Los puntos suspensivos indican que hay más código pero no se tiene que completar.

NOTA: Si el código propuesto en cada pregunta no se corresponde con la forma en que vosotros plantearíais la respuesta, podéis rescribir el código o parte del código según vuestro planteamiento.

Pregunta 1.1

Completa el código correspondiente a la subrutina m_add (suma de matrices, mr=m1+m2).

```
m_add:
    push rbp
    mov rbp, rsp

    mov esi, 0
    m_add_bucle:
      mov eax, [m1+esi*4]
      add eax, [m2+esi*4]
      mov [mr+esi*4], eax
      inc esi
      cmp esi, 8
      jle m_add_bucle

    m_add_end:
    mov rsp, rbp
    pop rbp
    ret
```

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Pregunta 1.2

Completa el código correspondiente a la función de C m_cramer (resolver el sistema de ecuaciones Ax=b por Cramer).

```
void m_cramer(){
  int detA; //determinante de la matriz A
  int i;

  m_copy(A, mr);
  m_det_mr();
   detA=det;
  if (detA==0) {
    printf("El sistema no tiene solución\n");
  } else {
    printf("Solución del sistema\n");
    for (i=0;i<3;i++){
      col=i;
      m_copy_col();
      m_det_mr();
       //Mostrar el valor de las variables x1, x2, i x3
      printf("x%d=%d/%d = %2.2f\t\n",i,det,detA,(float)det/detA);
      m_copy(A, mr);
    }
    printf("\n");
  }
}
```

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Pregunta 2 (40%: 2.1: 15%, 2.2: 15%, 2.3: 10%)

Apartado 2.1 (15%)

El estado inicial del computador CISCA justo antes de comenzar la ejecución de cada fragmento de código (de cada apartado) es el siguiente:

R0 = 00000000h R1 = 00001000h R2 = 00002000h	M(00000000h) = 00001000h M(00001000h) = 00002000h M(00002000h) = 00000000h	Z = 0 C = 0 S = 0 V = 0

¿Cuál será el estado del computador después de ejecutar cada fragmento de código? (sólo modificaciones, excluyendo el PC).

a)

```
MOV   R2, 00001000h
ADD   R2, [00001000h+R2]
XOR  R1, R2
```

R2 = 00001000h
R1 = 00000000h
Z = 1, S = 0
C = 0, V = 0

b)

```
MOV R1, 00001000h
SUB R1, [00000000h]
ADD [R1], R1
```

R1 = 0
M(00000000h) = 00001000h
Z = 0, S = 0
C = 0, V = 0

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Apartado 2.2 (15%)

En la memoria de un computador CISCA tenemos almacenada una matriz de 10 x 100 elementos (10 filas por 100 columnas) a partir de la dirección simbólica M. Cada elemento es un número entero codificado en complemento a 2 con 32 bits.

Completad los huecos del fragmento de código CISCA para poner a 0 el elemento M[i][j], lo que en C se especificaría con la sentencia:

M[i][j] = 0;

La matriz está almacenada por filas en posiciones consecutivas de memoria, como es habitual cuando se traduce código en C. Por ejemplo, los elementos M[0][0], M[0][1], M[1][0] y M[7][40] se encuentran almacenados en las direcciones de memoria M, M+4, M+400 y M+2960 respectivamente.

Se sabe que en R1 se encuentra almacenado el valor de la variable i, y en R2 el de la j y que después de ejecutarse el fragmento de código todos los registros deben mantener los valores originales. El código es correcto pero no es todo lo eficiente que podría ser.

```
PUSH  R1
PUSH  R2
MUL   R1, 100
ADD   R1, R2
MUL   R1, 4; o també SLH R1, 2
MOV   [M+R1], 0
POP   R2
POP   R1
```

Examen 2011/12-1

Asignatura	**Código**	**Fecha**	**Hora inicio**
Estructura de computadores	75.573	14/01/2012	12:00

Apartado 2.3 (10%)

Dado el siguiente fragmento de código de un programa en lenguaje ensamblador del CISCA:

```
MOV  R3,  A
MOV  R3, [R3]
CMP  R3, [B]
```

Traducidlo a lenguaje máquina y expresadlo en la siguiente tabla. Suponed que la primera instrucción del código se ensambla a partir de la dirección 00FF0000h (que es el valor del PC antes de empezar la ejecución del fragmento de código). Suponed también que las direcciones simbólicas A y B valen 0000E000h y 0000F000h respectivamente. En la tabla de resultados usad una fila para codificar cada instrucción. Si suponemos que la instrucción comienza en la dirección @, el valor Bk de cada uno de los bytes de la instrucción con direcciones @+k para k=0, 1,… se debe indicar en la tabla en hexadecimal en la columna correspondiente (recordad que los campos que codifican un desplazamiento en 2 bytes o un inmediato o una dirección en 4 bytes lo hacen en formato little endian, esto hay que tenerlo en cuenta escribiendo los bytes de menor peso, de dirección más pequeña, a la izquierda y los de mayor peso, dirección mayor, a la derecha). Completad también la columna @ que indica para cada fila la dirección de memoria del byte B0 de la instrucción que se codifica en esa fila de la tabla.

A continuación os damos como ayuda las tablas de códigos:

Tabla de códigos de instrucción

Instrucción	B0
MOV	10h
CMP	26h

Tabla de modos de direccionamiento (Bk<7..4>)

Campo modo Bk<7..4>	Modo
0h	Inmediato
1h	Registro
2h	Memoria
3h	Indirecto
4h	Relativo
5h	Indexado
6h	A PC

Tabla de modos de direccionamiento (Bk<3..0>)

Examen 2011/12-1

Asignatura	**Código**	**Fecha**	**Hora inicio**
Estructura de computadores	75.573	14/01/2012	12:00

Campo modo Bk<3..0>	Significado
Nº registro	Si el modo debe especificar un registro
0	No se especifica registro.

Tabla para contestar a la pregunta

		Bk para k=0..10										
@	**Ensamblador**	0	1	2	3	4	5	6	7	8	9	10
00FF0000	MOV R3, A	10	13	00	00	E0	00	00				
00FF0007	MOV R3, [R3]	10	13	33								
00FF000A	CMP R3, [B]	26	13	20	00	F0	00	00				
00FF0011												

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Pregunta 3 (40%: 3.1: 15%, 3.2: 15%, 3.3: 10%)

Apartado 3.1 (15%)

Tenemos un sistema de memoria en el que todos los accesos se realizan a palabra (no nos importa de qué tamaño es una palabra). Supondremos que el espacio de direcciones de memoria se descompone en bloques de 4 palabras. Cada bloque comienza en una dirección múltiplo de 4. Así, el bloque 0 contiene las direcciones 0, 1, 2 i 3, el bloque 1, las direcciones 4, 5, 6 i 7, y el bloque N las direcciones 4*N, 4*N+1, 4*N+2 i 4*N+3. Una fórmula para calcular el identificador numérico del bloque es la siguiente:

Bloque = Dirección de memoria (dirección a palabra) DIV 4 (tamaño del bloque en palabras)

Suponemos que el sistema también dispone de una memoria cache de 4 líneas (donde cada línea tiene el tamaño de un bloque, es decir, 4 palabras). Estas líneas se identifican como líneas 0, 1, 2 i 3. Cuando se hace referencia a una dirección de memoria principal, si esta dirección no se encuentra en la memoria cache, se lleva todo el bloque correspondiente desde la memoria principal a una línea de la memoria cache (así si hademos referencia a la dirección 2 de memoria principal traeremos el bloque formado por las palabras 0, 1, 2 i 3).

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Apartado 3.1.1
Suponemos que el sistema utiliza una **política de asignación directa**, de manera que cada bloque de la memoria principal sólo se puede llevar a una línea determinada de la memoria cache. En este caso, el identificador del bloque determina la línea específica donde se puede guardar utilizando la siguiente fórmula (similar a la fórmula para determinar el bloque):

Línea = identificador de bloque MOD 4 (tamaño de la cache en líneas)

La ejecución de un programa genera la siguiente lista de lecturas a memoria:

1, 2, 3, 8, 9, 10, 11, 12, 30, 31, 32, 1, 2, 3, 8, 9, 10, 11, 12, 30

3.1.1.a)
La siguiente tabla muestra el estado inicial de la cache, que contiene las primeras 16 palabras de la memoria (organizadas en 4 bloques). Completar la tabla para mostrar la evolución de la cache durante la ejecución del programa. Para cada fallo en la cache hay que rellenar una nueva columna indicando que referencia a memoria ha provocado el fallo y el cambio que se produce en el estado de la memoria cache (la línea que se modifica).

	Estado Inicial	Fallo:30	Fallo:32	Fallo:1	Fallo:12
Línea 0	0, 1, 2, 3		32, 33, 34, 35	0, 1, 2, 3	
Línea 1	4, 5, 6, 7				
Línea 2	8, 9, 10, 11				
Línea 3	12, 13, 14, 15	28, 29, 30, 31			12, 13, 14, 15

	Fallo:30	Fallo:	Fallo:	Fallo:	Fallo:
Línea 0					
Línea 1					
Línea 2					
Línea 3	28, 29, 30, 31				

3.1.1.b) ¿Cuál es la tasa de fallos (T_f) ?

T_f = 5 fallos / 20 accesos = 0,25

3.1.1.c) Suponemos que el tiempo de acceso a la memoria cache, o tiempo de acceso en caso de acierto (t_e), es de 2 ns y el tiempo total de acceso en caso de fallo (t_f) es de 24 ns. Considerando la tasa de fallos obtenida en la pregunta anterior, ¿cuál es el tiempo promedio de acceso a memoria (t_m) ?

$t_m = T_f \times t_f + (1-T_f) \times t_e$ = 0,25 * 24 ns + 0,75 * 2 ns = 6 ns + 1,5 ns = 7,5 ns

Examen 2011/12-1

Asignatura	**Código**	**Fecha**	**Hora inicio**
Estructura de computadores	75.573	14/01/2012	12:00

Apartado 3.1.2

Ahora suponemos que el mismo sistema utiliza una **política de asignación completamente asociativa,** de manera que cualquier bloque de la memoria principal se puede llevar a cualquier bloque de la memoria cache.

Si encontramos que la cache ya está llena, se utiliza un algoritmo de reemplazo LRU, de manera que sacaremos de la memoria cache aquel bloque que hace más tiempo que no se ha referenciado.

Consideramos la misma lista de lecturas a memoria:

1, 2, 3, 8, 9, 10, 11, 12, 30, 31, 32, 1, 2, 3, 8, 9, 10, 11, 12, 30

3.1.2.a)
La siguiente tabla muestra el estado inicial de la cache, que contiene las primeras 16 palabras de la memoria (organizadas en 4 bloques). Completar la tabla para mostrar la evolución de la cache durante la ejecución del programa. Para cada fallo en la cache hay que rellenar una nueva columna indicando que referencia a memoria ha provocado el fallo y el cambio que se produce en el estado de la memoria cache (la línea que se modifica).

	Estado Inicial	Fallo:30	Fallo:32	Fallo:1	Fallo:8
Línea 0	0, 1, 2, 3		32, 33, 34, 35		
Línea 1	4, 5, 6, 7	28, 29, 30, 31			
Línea 2	8, 9, 10, 11			0, 1, 2, 3	
Línea 3	12, 13, 14, 15				8, 9, 10, 11

	Fallo:12	Fallo:30	Fallo:	Fallo:	Fallo:
Línea 0		28, 29, 30, 31			
Línea 1	12, 13, 14, 15				
Línea 2					
Línea 3					

3.1.2.b) ¿Cuál es la tasa de fallos (T_f) ?

T_f = 6 fallos / 20 accesos = 0,3

3.1.2.c) Suponemos que el tiempo de acceso a la memoria cache, o tiempo de acceso en caso de acierto (t_e), es de 2 ns y el tiempo total de acceso en caso de fallo (t_f) es de 24 ns. Considerando la tasa de fallos obtenida en la pregunta anterior, ¿cuál es el tiempo promedio de acceso a memoria (t_m) ?

$t_m = T_f \times t_f + (1-T_f) \times t_e$ = 0,3 * 24 ns + 0,7 * 2 ns = 7,2 ns + 1,4 ns = 8,6 ns

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Apartado 3.2 (15%)

Una RSI de repertorio CISCA transfiere datos desde el disco a la memoria del procesador:

- Tiempo medio de latencia del disco de 3 ms ($t_{latencia}$)
- Velocidad de transferencia del disco de 20 MB/s (v_{transf})
- Procesador con una frecuencia de reloj de 2 GHz
- La RSI ejecuta 10 instrucciones, cada una en un tiempo de 3 ciclos de reloj
- La CPU requiere 8 ciclos de reloj adicionales desde que se detecta la interrupción del disco hasta que se transfiere el control a la RSI, (t_{rec_int}).
- Tiempo de programación y finalización de la transferencia de 20 ns (t_{prog} + t_{final})
- Transferencia de N_{datos}= 4.000 datos de 4 Bytes

¿Cuál es el tiempo total que dedica la CPU a la tarea de Entrada/Salida, t_{cpu} ?

Tiempo de ciclo de reloj, t_{clock}, 1 / 2 GHz = 0,5 nanosegundos

Tiempo para ejecutar una instrucción, t_{instr}, 3 × 0,5 = 1,5 nanosegundos

Tiempo para atender la interrupción, t_{rec_int}: 8 ciclos × 0,5 ns = 4 ns

Tiempo de ejecución RSI, t_{rsi}: N_{rsi} × t_{instr} = 10 instr. × 1,5 ns = 15 ns

Tiempo consumido por la CPU en cada interrupción, t_{transf_dato}:

$$t_{transf\_dato} = t_{rec\_int} + t_{rsi} = 4 + 15 = 19 \text{ ns}$$

Número de interrupciones producidas (o número total de datos, N_{datos}):
4.000 interrupciones

Tiempo consumido en total en TODAS las interrupciones:
$t_{transf\_bloque} = t_{transf\_dato} \times N_{datos}$ = 19 ns × 4.000 interrupciones = 76 us (microsegundos)

t_{cpu}= (t_{prog} + t_{final}) + t_{transf_bloque} = 20 ns + 76 us = 76,02 us.

Examen 2011/12-1

Asignatura	Código	Fecha	Hora inicio
Estructura de computadores	75.573	14/01/2012	12:00

Apartado 3.3 (10%)

3.3.1 (5%)

El siguiente código CISCA transfiere datos entre memoria y disco utilizando la técnica de E/S programada.

```
1.              MOV   R1, 200
2.              MOV   R2, DADES
3.              MOV   R3, 10000
4. Bucle:       IN    R4,[R3]
5.              AND   R4, 4
6.              JE    Bucle
7.              MOV   R0,[R2]
8.              OUT   R0,[R3+4]
9.              ADD   R2,4
10.             DEC   R1
11.             JNE   Bucle
```

Indicar si la transferencia es de salida al disco o de entrada desde el disco, cuántos datos se transfieren en total entre la memoria y el disco, y qué bit del registro de estado del controlador de disco verifica que se puede realizar la transferencia de un dato

Transferencia de entrada de 200 datos (200×4 = 800 Bytes).

El bit 2 (o el tercer bit menos significativo).

3.3.2 (5%)

El sistema de E/S por DMA de un computador tiene las siguientes características:

- El **tiempo de cesión** y el **tiempo de recuperación** del bus ($t_{cesión}$ + t_{recup}) son 4 ns.
- El **tiempo de la transferencia** por el bus (t_{mem}) es de 2 ns.
- Se envían por el bus 2.000 datos de 32 bits cada uno
- La memoria intermedia utiliza ráfagas de 8 datos.

Calcular el tiempo total de ocupación del bus por parte del controlador de disco/DMA para llevar a término la transferencia utilizando ráfagas.

Tiempo de ocupación del Bus, t_{transf_dato}: $t_{cesión} + 8 \times t_{mem} + t_{recup} = 4 + 8 \times 2 = 20$ ns

Número de peticiones del Bus, N_{pet_bus}: $N_{datos} / N_{ráfaga} = 2.000 / 8 = 250$

Tiempo total de ocupación del Bus t_{transf_bloque}: $t_{transf\_dato} \times N_{pet\_bus} = 20 \text{ ns} \times 250 = 5$ us