

Examen FINAL Estructura de Computadores II

- La duración del examen es de 3 horas.
- Contesta en las hojas de respuestas.
- Las notas finales se publicarán el **29 de junio**.
- La revisión se realizará el **3 de julio**.

curso 2006-2007 Q2

P1 Problemas Cortos (1,5 puntos)

1) Dado el siguiente formato de punto flotante:

S		Exponente			Mantisa		
12	11		7	6			0

Donde: S es el bit de signo, el exponente se representa en exceso 16, hay bit implícito, la mantisa está normalizada y, en caso necesario, se redondea por truncamiento.

Indica la secuencia de bits (en binario y en hexadecimal) que representa el número **decimal -13,1875**

2) **Escribe UNA ÚNICA** instrucción IA32 que realice la misma tarea que la secuencia de microoperaciones siguiente (%EIP es el registro que contiene el contador de programa y %EPSW es la palabra de estado):

```
%EIP <- M[%esp]
%esp <- %esp + 4
%EPSW <- M[%esp]
%esp <- %esp + 4
```

3) **Escribe** una secuencia de instrucciones que permita ejecutar la operación $a = (b+c)/(b-c)$ en una arquitectura de tipo pila en la que las operaciones se describen de la siguiente forma:

```
add/sub/mul/div #pila[sp+1]=pila[sp] op pila[sp+1]; sp=sp+1
push @          #sp=sp-1; pila[sp]=M[@]
pop @           #M[@]=pila[sp]; sp=sp+1
```

4) Disponemos de 10 discos físicos de 250 **Gbytes** cada uno. Con estos 10 discos montamos un RAID 5. **Indica** la capacidad útil (en **Tbytes**) del RAID.

5) Disponemos de un bus con las siguientes características:

- Frecuencia: 500 MHz
- líneas de datos: 64 bits
- líneas de direcciones: 36 bits

Calcula el ancho de banda del bus (da el resultado en **Gbytes/s**).

6) En un DIMM SDRAM con 8 módulos (chips), 4 bancos en cada módulo, 1024 filas y 512 columnas por módulo, **calcula** la dirección del byte que se encuentra en (todos los datos están en hexadecimal):

- módulo 3
- banco 2
- fila 01F
- columna 023

Da el resultado en **hexadecimal**.

P2 Teoría (1 punto)

a) **Dibuja** el esquema de diseño interno de una memoria cache totalmente asociativa de 6 líneas y 4 bytes por línea para un procesador de 8 bits que lanza direcciones de 12 bits. **Indica** claramente las conexiones entre los diferentes bloques de la memoria y con los bits del bus de direcciones del procesador. **Indica** la anchura de cada bus. **Indica** el tamaño TOTAL de la memoria de etiquetas.

b) **Explica** brevemente (**máximo 6 líneas**) el funcionamiento del diseño que has dibujado en el punto anterior, detallando la función que realiza cada uno de los bloques.

P3 LM+MC+MV (1.5 puntos)

Dado el siguiente código escrito en ensamblador del IA32:

```

movl $0, %ebx
movl $0, %esi
for: cmpl $256*1000, %esi
    jge end
(a) movl (%ebx, %esi, 4), %eax
(b) movl %eax, 4*1024(%ebx, %esi, 4)
(c) addl %eax, 8*1024(%ebx, %esi, 4)
    addl $VAL, %esi
    jmp for
end:

```

Suponiendo una memoria cache de datos **2-asociativa**, escritura **write through + write NO allocate** de **4 Kbytes** y **líneas de 16 bytes**, responde a las siguientes preguntas:

1) Suponiendo que **\$VAL vale 1**, para cada uno de los accesos (etiquetas a, b, c), **indica** a qué conjunto de la memoria cache se accede en cada una de las 17 primeras iteraciones.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a																	
b																	
c																	

	referencia a	referencia b	referencia c
2) Calcula la cantidad de aciertos de cache , en todo el bucle, para la			
3) Calcula la cantidad total de fallos de cache , en todo el bucle			

Suponiendo que la memoria virtual utiliza **páginas de tamaño 2K bytes** y que utilizamos un **TLB de 4 entradas (reemplazo LRU)**, responde a las siguientes preguntas:

4) Suponiendo que **\$VAL vale 64**, para cada uno de los accesos (etiquetas a, b, c), **indica** a qué página de la memoria virtual se accede en cada una de las 17 primeras iteraciones.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a																	
b																	
c																	

	referencia a	referencia b	referencia c
5) Calcula la cantidad de aciertos de TLB , en todo el bucle, para la			
6) Calcula la cantidad total de fallos de TLB , en todo el bucle			
7) ¿ Cuál sería el número total de fallos de TLB con un TLB con infinitas entradas?			
8) ¿ Cuál sería el mínimo número de entradas de TLB necesarias para conseguir la tasa de fallos del apartado anterior?			

P4 Preguntas de teoría (2 puntos)

Responde a las siguientes afirmaciones poniendo una X en el recuadro correspondiente (en la columna C si la afirmación es cierta o en la columna F si la afirmación es falsa). Cada respuesta contestada correctamente SUMA 0,1 puntos. **Cada respuesta incorrecta RESTA 0,1 puntos**. Las respuestas no contestadas no se tienen en cuenta.

C	F	Afirmación a evaluar
		En C, todos los tipos de datos estructurados son pasados por el compilador por referencia cuando son parámetros de una subrutina
		La instrucción <code>movb %ah, 6(%ebx,%ecx,8)</code> produce un error de compilación
		La instrucción <code>pushl</code> resta 4 al valor del registro <code>%esp</code>
		El rango de un <code>int</code> en la arquitectura IA32 es $-2^{31} .. 2^{31} - 1$
		En una memoria cache copy back - no write allocate, en caso de fallo de escritura el dato no se trae a la cache
		El tiempo de penalización en caso de fallo de escritura de una memoria cache write through - write allocate depende de la longitud de la línea
		La memoria cache es RAM dinámica
		Una DRAM que se direcciona con n bits de fila y n bits de columna tiene n^2 bits de datos
		Una memoria DDR es una RAM síncrona que suministra un dato por ciclo
		En RAID 4 la paridad está distribuida entre varios discos, a diferencia de RAID 3
		La paginación es un esquema de traducción de direcciones que permite protección de las páginas, a diferencia de la segmentación, que es un esquema de traducción de direcciones que NO permite protección de los segmentos
		Todos los dispositivos conectados al controlador de interrupciones del PC tienen distinta prioridad
		En un PC, las interrupciones están vectorizadas
		Un computador con dos niveles de cache y caches de instrucciones y datos separadas en el primer nivel, dispone de 3 TLB, uno por cada una de las 3 caches: L1I, L1D y L2
		El árbitro de un bus decide la política de selección, pero es el procesador quien decide la política de cesión del bus
		El bus PCI es un bus multiplexado
		Un computador que no dispone de controlador de interrupciones no puede atender interrupciones de diferentes dispositivos
		El SO programa el DMA con direcciones lógicas que son traducidas posteriormente por el propio DMA con la ayuda del TLB y la tabla de páginas
		RAID 5 es similar a RAID 4, pero RAID 4 está entrelazado a nivel de byte y RAID 5 a nivel de tira
		La segmentación permite traducir direcciones físicas a direcciones lógicas

P5 IA32 Subrutinas (1,5 puntos)

Dado el siguiente código escrito en C:

```
int XXX(int m[100][25], int x);
int ec2(int v[100], int *i, int j) {
    int M[100][25];
    ...
    (1) if (*i >= 0x7fffffff)
        *i = *i + v[j];
        else
            i = &M[j][22];
    ...
    (2) v[5] = XXX(M, *i+j);
    ...
    (3) for (tmp=0; tmp <100; tmp++) //usad el registro %ecx como tmp
        v[tmp] = M[tmp][4] & (*j)
    ...
}
```

Contesta las siguientes preguntas:

- a) **Traduce literalmente** a ensamblador de IA32 la sentencia (1).
- b) **Traduce literalmente** a ensamblador de IA32 la sentencia (2).
- c) **Traduce de forma ÓPTIMA** a ensamblador de IA32 la sentencia (3).

P6 IA32 Struct's (1,5 puntos)

Dado el siguiente código escrito en C:

```
typedef struct {
    char a;
    unsigned short us[4];
} small;

typedef struct {
    char b;
    small sv[3];
    int j
    char *pc
} big;

big *ps; /* almacenado en %ebx */
int i; /* almacenado en %esi */
```

Suponiendo que *ps* se encuentra en %ebx y que *i* está en %esi, responde a las siguientes preguntas:

- a) **Dibuja** cómo quedarían almacenadas las estructuras de datos (*small* y *big*) en linux, indicando claramente los desplazamientos respecto el inicio de la estructura y el tamaño de ésta.
- b) **Escribe** una instrucción IA32 que realice la siguiente asignación:
`ps->j = i;`
- c) **Escribe** una secuencia de 2 instrucciones IA32 que realicen la siguiente asignación:
`ps->pc = &ps->sv[2].a;`
- d) **Escribe** una instrucción IA32 que realice la siguiente asignación:
`(*ps).sv[0].us[i] = (*ps).sv[0].us[i] - 47;`
- e) **Escribe** una secuencia de 2 instrucciones IA32 que realicen la siguiente asignación:
`ps->sv[1].us[2] = ps->sv[2].us[3];`

P7 Memoria Cache con prefetch (1 punto)

Disponemos de un procesador de 16 bits con direcciones de 16 bits que tiene una memoria cache de datos con las siguientes características:

- 3-asociativa, con algoritmo de reemplazo LRU
- 64 bytes por línea
- 12 líneas
- política de escritura: copy back + write allocate

El contenido inicial de la memoria de etiquetas (tags) es el siguiente:

conjunto 0	DB	conjunto 1	DB	conjunto 2	DB	conjunto 3	DB
13	1	13	1	13	0	13	0
43	1	43	1	43	0	43	0
AC	0	AC	0	AC	1	AC	1

El DB=1 indica que la línea correspondiente ha sido modificada. La información de reemplazo está implícita en la posición. Las posiciones inferiores corresponden a las líneas que llevan más tiempo sin utilizarse. Las posiciones superiores corresponden a las últimas líneas utilizadas. Por ejemplo, en el conjunto 3, la línea con tag 13 es la última utilizada, y la línea con tag AC la que lleva más tiempo sin ser utilizada.

A esta cache le hemos añadido un **buffer de prefetch** de una entrada. En este buffer se hace prebúsqueda hardware de la línea **i+1** cuando se accede (tanto en acierto como en fallo) a la línea **i**, siempre que la **i+1** no esté ya en la cache o en el buffer. En este último caso, no se realiza prefetch.

Partiendo de los datos iniciales, **rellena** la siguiente tabla indicando, para cada referencia, el número de línea de MP que le corresponde, la etiqueta (TAG), a qué conjunto de MC va a parar, si se produce acierto o fallo en la cache (A/F), el TAG de la línea reemplazada, el número de bytes totales leídos de MP (si se lee de MP), el número de bytes totales escritos en MP (si se escribe en MP), la línea de MP que se encuentra en el buffer (si procede), si se produce acierto o fallo (A/F) en el buffer y la línea que se prebusca de MP (si procede).

tipo	dirección (hex)	línea MP (hex)	TAG (hex)	conjunto MC	Cache ¿acierto o fallo?	TAG línea reemplazada	bytes escritura MP	bytes lectura MP	línea actual buffer	Buffer ¿acierto o fallo?	línea prefetch buffer
LECT	B12B								---		
LECT	B145										
LECT	B1AF										
LECT	B1C4										
ESCR	4387										
LECT	1108										
ESCR	1199										
LECT	11AA										