

## 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 **22 de enero**.
- La revisión se realizará el **24 enero**.

**curso 2007-2008 Q1**

### P1 Problemas Cortos (1,5 puntos)

- 1) **Escribid** una secuencia de microoperaciones que se comporte de la misma forma que la siguiente instrucción IA32: **xorl \$732, -35(, %eax,8)**. Las microoperaciones que podéis utilizar son las siguientes:

$R_i = R_j \text{ op } R_k$	$R_i = \text{GetDesplazamiento}(\text{IR})$
$R_i = M[R_j]$	$R_i = \text{GetFactorEscala}(\text{IR})$
$M[R_j] = R_i$	$R_i = \text{Inmediato}(\text{IR})$

; Se pueden utilizar registros temporales: T1, T2, T3, ...

- 2) **Escribid** una secuencia de instrucciones para ejecutar la operación  $a = (a-d)/((a*b)-c)$  en una arquitectura de tipo **acumulador** en la que las operaciones se describen de la siguiente forma:

```
add/sub/mul/div @ #ACC = ACC op M[@]
load @          #ACC = M[@]
store @         #M[@] = ACC
; Se pueden utilizar variables temporales: V1, V2, V3, ...
```

- 3) Disponemos de un sistema de memoria principal con las siguientes características:

- 1 DIMM con 16 chips de memoria DDR.
- Cada chip dispone de 8 bancos de memoria.
- Cada banco dispone de 2048 filas y 1024 columnas de 1 byte cada una.
- La datos están entrelazados a nivel de chip. Posiciones consecutivas de memoria están en chips diferentes. Dentro del chip los datos NO están entrelazados.

Responded a las siguientes preguntas:

- **Dibujad** una dirección de memoria principal indicando claramente los campos usados para seleccionar Chip, banco, fila y columna y el tamaño de cada uno de estos campos.
  - **¿Qué dirección** (en hexa) de memoria principal tiene el byte que se encuentra en el chip A, banco 3, fila 0xC73 y columna 0x1B2?
- 4) Disponemos de 20 discos físicos de 400 Gbytes cada uno. Con estos discos montamos un RAID 5+1. **Indicad** la capacidad útil (en Gbytes y en Tbytes) del RAID.
- 5) Disponemos de un bus con las siguientes características:
- Frecuencia: 2.5 GHz
  - líneas de control: 20 bits
  - líneas de dirección: 32 bits
  - líneas de datos: 64 bits.

**Calculad** el ancho de banda del bus (da el resultado en **Gbits/s** y en **Mbytes/s**).

- 6) **Escribid** la secuencia de comandos Linux necesaria para extraer los ficheros contenidos en "Programas.Sesion05.tar.gz".

### P2 Teoría ( 1 punto)

- a) **Dibuja** el esquema de diseño interno de una memoria cache totalmente 2-asociativa de 32 líneas y 16 bytes por línea para un procesador que lanza direcciones de 20 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 cuando se accede en acierto a la dirección 0x00666.

### P3 LM + MC (1 punto)

Dado el siguiente código escrito en ensamblador del IA32:

```
movl $0, %ebx
movl $0, %esi
for: cmpl $1024, %esi
     jge end
(a) movl (%ebx, %esi,8), %eax
(b) movl %eax, 64*1024(%ebx, %esi,8)
(c) addl %eax, 32(%ebx, %esi,8)
     addl $1, %esi
     jmp for
```

end:

Suponiendo una memoria cache de datos con **mapeo directo**, escritura **copy back + write allocate** de **8 Kbytes** y **líneas de 16 bytes**, responde a las siguientes preguntas:

- 1) Para cada uno de los accesos (etiquetas a, b, c), indica a qué línea de la memoria cache se accede en cada una de las 16 primeras iteraciones.

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

- 2) Calcula la cantidad de **aciertos y fallos de cache**, en todo el bucle, para la

	aciertos	fallos
referencia (a)		
referencia (b)		
referencia (c)		

### Problema P4 LM+MV (1 punto)

Dado el siguiente código escrito en ensamblador del IA32:

```

movl $0, %esi
for: cmpl $128*1024, %esi
    jge end
(a) movl (,%esi,8), %eax
(b) movl %eax, 4*1024(,%esi,8)
(c) addl %eax, 16*1024(,%esi,8)
    addl $128,%esi
    jmp for
end:

```

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

- 1) Para cada uno de los accesos (etiquetas a, b, c), indica a qué página de memoria se accede en cada una de las 16 primeras iteraciones.

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

- 2) Calcula la cantidad de **aciertos y fallos de TLB**, en todo el bucle, para la

	aciertos	fallos
referencia (a)		
referencia (b)		
referencia (c)		

- 3) Suponiendo un TLB con infinitas entradas, ¿cuál sería el número **mínimo** de fallos de TLB?

## P5 Preguntas de teoría (1.5 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
		Dados dos números enteros sin signo en binario de 4 bits 1110 y 0111, si los sumamos se produce error por desbordamiento.
		La operación "%ebx <- 9*%eax -37" se puede hacer en una sola instrucción de IA32
		Los structs de tamaño superior a 32 bytes se pasan por referencia cuando son parámetros de una subrutina
		El orden de los campos de un struct influye en el tamaño total del mismo
		Una celda de una memoria DDR es idéntica a una celda de una memoria SDRAM.
		En una memoria cache 2-asociativa se accede simultáneamente a la memoria de etiquetas y a la de datos.
		El número de bancos de una DRAM siempre ha de ser potencia de 2.
		El tiempo de penalización en caso de fallo en lectura de una memoria cache write through y write NO allocate no depende de la longitud de la línea.
		Cuando el DMA accede a Memoria Principal lo ha de hacer utilizando direcciones físicas.
		Siempre que se produce un fallo de página, se ha producido antes un fallo de TLB.
		En un computador con DMA no se puede tener 2 niveles de cache porque se provocan problemas de coherencia.
		Un bus PCI de 64 bits a 66.6 MHz tiene un ancho de banda de 533 MB/s.
		Un RAID 4+1 tiene cuatro discos de datos y uno con información redundante.
		En un RAID, la probabilidad de que falle algún disco aumenta cuando incrementamos el número de discos.
		El bus que conecta el procesador con memoria es un bus de nivel 2.

## P6 IA32 Subrutinas ( 1 punto)

Dada la siguiente rutina en C:

```
int rutina(int *i, int c) {
    int k;
    k = *i-c;
    if (k>64)
        k = rutina(&k, *i-5);
    return (k-c);
}
```

Traducid literalmente el código a ensamblador.

## P7 IA32 struct's (1'5 puntos)

Dadas las siguientes estructuras en C:

```
typedef struct {
    char c1;
    int a;
    char c2;
    char *p;
    short x;
} ss;

typedef struct {
    char c3;
    ss m[5][3];
    ss *v[20];
} big;

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

Suponiendo que `ps` se encuentra en `%ebx`, `i` en `%esi`, y `j` en `%edi`, responde a las siguientes preguntas:

- a) **Dibujad** como quedarían almacenadas las estructuras de datos en linux, indicando claramente los desplazamientos respecto al inicio de la estructura y el tamaño de éstas.
- b) **Escribid** una secuencia de 2 instrucciones que realice la siguiente asignación:
- ```
ps->v[3]->x = ps->v[3]->x + 3;
```
- c) **Escribid** una secuencia de instrucciones que realice la siguiente asignación (se puede hacer en 5):
- ```
ps->m[i][j].c1 = ps->m[i][j].c2;
```
- d) **Escribid** 1 instrucción que realice la siguiente asignación:
- ```
ps->m[3][2].p = &ps->c3;
```
- e) **Reescribid** en C la struct `ss` para que ocupe el mínimo espacio posible. **Dibujad** como quedaría la nueva estructura.

## P8 MC (1,5 puntos)

Disponemos de un sistema formado por:

Un procesador de 32 bits con las siguientes características:

- CPI ideal: 1.3 ciclos por instrucción
- Frecuencia del procesador: 2GHz
- Número de referencias por instrucción (nr): 1.5 (1 a instrucciones y 0,5 a datos)
- Caches de instrucciones y datos separadas
- Las caches no tienen ninguna optimización

Cache de Instrucciones:

- Tiempo de servicio en caso de acierto ( $T_{sa}$ ) = 1 ciclo
- Tamaño de línea 64 bytes
- Tasa de fallos 5%

Cache de Datos:

- Tiempo de servicio en caso de acierto ( $T_{sa}$ ) = 1 ciclo
- Tamaño de línea 32 bytes
- Tasa de fallos 10%
- Política de escritura = Write Through + Write NO Allocate
- 75% de lecturas

Memoria principal:

- Organizada en DIMMs con 8 chips DDR de 1 byte cada uno
- Latencia de fila de los módulos de MP = 3 ciclos
- Latencia de columna de los módulos de MP = 2 ciclos
- Ancho de banda del bus MP↔ MC: 8 bytes por ciclo

**Indicad** como se ocupan los recursos del sistema en las siguientes situaciones. Para ello usad las letras que se indican a continuación para mostrar la ocupación de cache y memoria en las situaciones correspondientes.

- Cache: acceso en acierto (H), acceso en fallo (M)
- Memoria: acceso a columna (C), acceso a fila (F), transferencia de datos (D)

- 1) Fallo en la cache de Instrucciones
- 2) Fallo en lectura en la cache de datos
- 3) Acierto en escritura en la cache de datos

**Calculad** las siguientes medidas de rendimiento:

- 4) Tiempo medio de acceso de la cache de instrucciones en ciclos
- 5) Tiempo medio de acceso de la cache de datos en ciclos
- 6) Tiempo de ejecución de  $2 \times 10^9$  instrucciones en segundos