

Examen FINAL Estructura de Computadores II

curso 2005-2006 Q1

- La duración del examen es de 3 horas.
- Contestad las preguntas en las hojas de respuestas.
- Las notas finales se publicarán el **19 de enero**.
- La revisión se realizará el **24 de enero**.

Problema 1 (1 punto)

Traduce **literalmente** a ensamblador del IA32 la siguiente función escrita en C:

```
int CCE(int v[], int n) {
    int tmp1, tmp2;
    if (n==1)
        return v[0];
    else {
        tmp1 = CCE(&v[0], n>>1);
        tmp2 = CCE(&v[n>>1], n>>1);
        return tmp1 ^ tmp2;
    }
}
```

Problema 2 (1 punto)

Traduce a ensamblador del IA32, **de la forma más eficiente posible**, la siguiente función escrita en C:

```
int EXA(int M[25][25], int v[25]) {
    int i, sum;
    for (i=0; i<25; i++)
        v[i] = M[i][i];
    sum = 0;
    for(i=0; i<25; i++)
        sum = sum + v[i];
}
```

Problema 3 (1 punto)

Dado el siguiente código escrito en C:

```
typedef struct {
    char c1;
    char *p;
    unsigned short M[5][3];
    int a;
    char c2;
} XX;

int RutXX (int v[4], XX *pXX, char c)
{
    XX vxx[5];
    int i;
    ...
}
```

- Dibuja** cómo quedaría almacenada la estructura de datos "xx" en linux, indicando claramente los desplazamientos respecto al inicio de la estructura y el tamaño de ésta.
- Dibuja** el bloque de activación de la rutina "RutXX", indicando claramente los desplazamientos necesarios para acceder a los parámetros y variables locales.

Problema 4 (2 puntos)

Dado el siguiente código escrito en ensamblador del IA32:

```
    movl $0x80000000, %ebx
    xorl %esi, %esi
    xorl %edi, %edi
for: cmpl $1024, %esi
    jge end
    movl 32*1024(%ebx, %edi, 4), %eax
    addl %eax, (%ebx, %esi, 4)
    addl $1024, %edi
    addl $1, %esi
    jmp for
end:
```

Teniendo en cuenta exclusivamente los accesos a datos (sin contar los accesos a instrucciones), **calcula** los siguientes parámetros:

- 1) Número de operaciones de Escritura en Memoria.
- 2) Número de operaciones de Lectura en Memoria.
- 3) Número de fallos en una Memoria Cache directa de 4 Kbytes con líneas de 16 bytes.
- 4) Tamaño mínimo que tendría que tener esta cache para minimizar el número de fallos.
- 5) Número de fallos en un TLB con 8 entradas, completamente asociativo, con reemplazo LRU y páginas de 8 Kbytes.

Problema 5 (2 puntos)

Un procesador con 16 bits de direcciones y 32 bits de datos dispone de un sistema de memoria con 1 nivel de cache unificada con las siguientes características:

- Tamaño de línea = 64 bytes
- Política de escritura = WRITE THROUGH + WRITE NO ALLOCATE
- Tiempo de servicio en caso de acierto (Tsa) = 1 ciclo
- MP consta de 1 módulo DIMM de 8 bytes
- Tiempo de fila de los módulos de MP = 3 ciclos
- Tiempo de columna de los módulos de MP = 3 ciclos
- Ancho de banda del bus MP↔ MC: 8 bytes por ciclo

- 1) Indica cómo 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**)

- a) Acierto en lectura.

Cache																			
Memoria																			

- b) Fallo en lectura.

Cache																			
Memoria																			

- c) Acierto en escritura.

Cache																			
Memoria																			

d) Fallo en escritura.

Cache																				
Memoria																				

2) **Calcula** el tiempo medio de acceso a memoria (**T_{ma}**) a partir de los datos anteriores, sabiendo que:

- El 30% de los accesos son escrituras.
- El 10% de las lecturas son fallo.
- El 20% de las escrituras son fallo.

Pregunta 6 (0.5 puntos)

Escribe **UNA ÚNICA** instrucción IA32 que realice la misma tarea que la secuencia de microoperaciones siguiente (las variables tmp1, tmp2 y tmp3 son operandos de 32 bits):

```
tmp1 <- M[%esp]
%esp <- %esp + 4
tmp2 <- %ebp << 3
tmp3 <- %ebx + tmp2
tmp3 <- tmp3 - 1
M[tmp3] <- tmp1
```

Pregunta 7 (0.5 puntos)

A partir de la posición de memoria 0xABCDEF00 se encuentra almacenada la secuencia (hexadecimal) 0123456789ABCDEFEDCBA9876543210. ¿Cuál es el valor del operando de tamaño longword que se encuentra en la posición de memoria 0xABCDEF08 si el almacenamiento se hace de forma little endian?

Pregunta 8 (1 punto)

Para cada una de las siguientes afirmaciones, indica si son ciertas (C) o falsas (F). Hay que contestarlas todas. 1 Fallo: 0,5 puntos. 2 ó más fallos: 0 puntos.

- ☐ Una celda de SRAM tiene más transistores que una celda de DRAM
- ☐ Una memoria DDR es una RAM síncrona que suministra un dato por ciclo
- ☐ En una cache asociativa por conjuntos se busca simultáneamente en la memoria de etiquetas y en la de datos
- ☐ La tasa de fallos de la cache de segundo nivel es similar a la de primer nivel
- ☐ Los algoritmos de reemplazo de página de la memoria virtual son similares en complejidad a los de reemplazo de línea de la memoria cache

Pregunta 9 (1 punto)

Para cada una de las siguientes afirmaciones, indica si son ciertas (C) o falsas (F). Hay que contestarlas todas. 1 Fallo: 0,5 puntos. 2 ó más fallos: 0 puntos.

- ☐ La E/S del Pentium 4 está mapeada en memoria
- ☐ El bus SCSI es un bus de expansión
- ☐ El árbitro del bus, si existe, siempre es el dueño del bus
- ☐ Un RAID es visto por el SO como una sola unidad lógica
- ☐ Los CD-ROM giran a velocidad angular constante para poder ser leídos correctamente