Examen FINAL Estructura de Computadores II

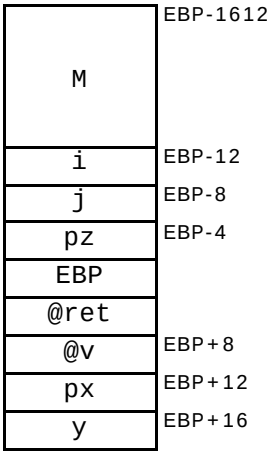
curso 2004-2005 Q1

- Los problemas 1, 2 y 3 se ha de entregar en una hojas separadas.
- El resto de problemas se han de entregar en las hojas de respuestas.
- Las notas finales se publicarán el **25 de enero**.
- La revisión se realizará el **27 de enero**.

Problema 1 (1.5 puntos)

A continuación se muestra la definición de la subrutina "`final1`" con su correspondiente bloque de activación:

```
int final1(int v[20], int *px, int y)
{ int M[20][20];
  int i, j, *pz;
  . . .
}
```



Se pide lo siguiente:

a) **Traducid** a ensamblador del IA32 la siguiente sentencia escrita en C:

```
M[i][j] = *px + y;
```

suponiendo que está dentro de la rutina "`final1`".

b) **Traducid** a ensamblador del IA32 la siguiente sentencia escrita en C:

```
*pz = final1(&M[i][0], pz, j);
```

suponiendo que está dentro de la rutina "`final1`".

c) **Traducid** de forma **óptima** a ensamblador del IA32 la siguiente sentencia escrita en C:

```
for (i=0; i<20; i++)
  M[i][i] = v[i];
```

suponiendo que está dentro de la rutina "`final1`".

Problema 2 (1.5 puntos)

Dada la definición de 2 "struct's" escritas en C:

```
typedef struct {
  char nif[9];
  char pass[9];
  int tot;
  char c
} S1;
```

```
typedef struct {
  char M[100][9];
  S1 v[100];
  S1 *ps1;
  int n;
} S2;
```

Se pide lo siguiente:

a) **Dibujad** como quedarían almacenadas en memoria las struct's S1 y S2 (indicando claramente el tamaño de cada campo y los desplazamientos necesarios para acceder a cada uno de ellos). Suponed que las direcciones iniciales de S1 y S2 son múltiplos de 4.

b) **Traducid** a ensamblador del IA32 la siguiente sentencia escrita en C:

```
S2 SS;
int i;
for (i=0; i<9; i++)
  SS.M[SS.n][i] = SS.v[SS.n].nif[i];
```

suponiendo que el registro %ebx contiene la dirección de inicio de SS y que podéis utilizar un registro cualquiera para la variable `i`.

c) **Traducid** a ensamblador del IA32 la siguiente sentencia escrita en C:

```
S1 XX;
XX.c = XX.nif[8];
```

suponiendo que el registro %ebx contiene la dirección de inicio de XX.

Problema 3 (1.5 puntos)

Tenemos un procesador de 32 bits con cache de instrucciones y datos separadas. Las características de las dos caches son las siguientes:

	cache de datos	**cache de instrucciones**
Tamaño de línea	32 bytes	16 bytes
Tasa de fallos	15%	5%
Tiempo de servicio en caso de acierto (tsa)	1 ciclo	1 ciclo
Número de referencias por instrucción	0.5	1
Política de escritura	copy back + write allocate	-
% lineas modificadas	40%	-
% escrituras	25%	-

A este procesador se le ha conectado una Memoria Principal con las siguientes características:

- MP organizada en 16 módulos entrelazados de 1 byte cada uno
- Tiempo de acceso a los módulos de MP = 5 ciclos
- Ancho de banda del bus MP↔ MC: 8 bytes por ciclo

Calculad los siguientes valores:

a) Tiempo medio de acceso (Tma_I) de la cache de instrucciones.

b) Tiempo medio de acceso (Tma_D) de la cache de datos.

c) Tiempo medio de acceso a memoria (Tma).

Problema 4 (1.5 punts)

Es disposa d'un sistema de memòria compost per una memòria cache (**C**) i el sistema de memòria principal (**M**). Per simplificar es considera que M engloba tot el subsistema format pel bus entre memòria cache i memòria principal i els mòduls de memòria principal. Les característiques rellevants del sistema de memòria son:

- Política d'escriptura: WRITE THROUGH + WRITE NO ALLOCATE
- Temps llegir / escriure 1 paraula a C: 1 cicle
- Temps de llegir / escriure una línia a M: 7 cicles
- Temps de llegir / escriure una paraula a M: 3 cicles
- Quan hi ha un fallo, si cal carregar una nova línia a C, aquesta s'ha de carregar completament a C abans de poder servir la dada.
- Si el processador te que escriure a C i a M, ho fa simultàniament.
- Quan el processador te que escriure directament a M es te que completar totalment l'escriptura abans de poder iniciar un nou accés.

Donada la seqüència d'accessos següent que es realitzen de forma consecutiva:

1) lectura i encert
2) escriptura i encert
3) lectura i encert
4) escriptura i fallo
5) lectura i fallo
6) escriptura i encert
7) escriptura i fallo
8) lectura i encert

a) **Emplena** la taula 4.a al full de respostes indicant a cada cicle la ocupació del sistema M i de C. Per això has d'escriure el numero d'accés al quadre corresponent. Com a exemple ja em emplenat l'accés 1.

cicle	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	31	32
C	1																															
M																																

Per millorar el rendiment del sistema s'ha incorporat un buffer d'escriptura (**B**) de 2 posicions:

- El temps d'escriure una paraula al buffer es 1 cicle.
- La CPU escriu simultàniament a C i B.
- El contingut de B es pot escriure a M al cicle següent.
- Es pot escriure una paraula de B a M sempre que el sistema M no estigui ocupat per servir una lectura. En cas que es pugin iniciar 2 acessos (lectura de línia i escriptura de capçalera del buffer) sempre es dona prioritat a llegir la línia.
- La capçalera de B esta ocupada fins que s'ha completat la seva escriptura a M
- Si B esta ple i la CPU vol fer una escriptura es bloqueja, i el accés no comença fins que hi una posició lliure a B

b) **Emplena** la taula 4.b al full de respostes indicant de la mateixa forma la ocupació de C M i B. Emplena també a la fila **Num** el nombre de posicions que hi ha ocupades de B.

cicle	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	31	32
C	1																															
M																																
B																																
Num	0																															

c) **Quin** es el temps mig d'accés a memòria que veu el processador per als 8 accessos.

Pregunta 5 (1 punto)

Explicad qué implicaciones tiene la transferencia via DMA en la jerarquía de memoria. ¡**NO expliquéis** el funcionamiento del DMA!

Pregunta 6 (3 puntos)

a) Para cada una de las siguientes afirmaciones, **indicad** si es cierta (C) o falsa (F):

- ☐ La instrucción *movw (%eax),6(%eip,%eax,2)* puede producir un error de "segmentation fault" en tiempo de ejecución.
- ☐ La instrucción *pushl %esp* suma 4 al valor del registro %esp antes de empilarlo.
- ☐ Los bits de condición no se modifican al ejecutar la instrucción *leal.*
- ☐ Sea %eax=0x12348765 y %ebx=0x9ABCDEF. La instrucción *movswl %ax,%ebx* deja en %ebx el valor 0x00008765.
- ☐ El registro %ecx no se debe modificar en el interior de una subrutina si no es previamente salvado.
- ☐ El comando de linux que permite cambiar los niveles de protección de un fichero es *cprot.*
- ☐ La instrucción *leal 0x30(%eax,%ebx,16),%ebp* pone en el registro %ebp el número 0x30+%eax+%ebx*16.

b) Para cada una de las siguientes afirmaciones, **indicad** si es cierta (C) o falsa (F):

- ☐ La memoria Cache es memoria RAM estática, mientras que la Memoria Principal es RAM dinámica.
- ☐ La DDR SDRAM es una memoria estática síncrona que transmite una ráfaga de datos cuando recibe una dirección.
- ☐ La jerarquía de memoria funciona gracias a que las memorias actuales transmiten a ráfagas, y por lo tanto traer muchos datos almacenados consecutivamente es poco más costoso que traer un sólo dato.
- ☐ En una memoria cache Write Through + Write No Allocate se pueden producir problemas de coherencia de memoria porque el contenido de la MC y de la MP puede no ser el mismo.
- ☐ El algoritmo de reemplazo LRU siempre produce una tasa de fallos menor que el algoritmo de reemplazo aleatorio.
- ☐ Una Victim Cache permite reducir los fallos debidos a carga.
- ☐ En Memoria Virtual, el espacio lógico siempre es mayor que el espacio físico.

c) Para cada una de las siguientes afirmaciones, **indicad** si es cierta (C) o falsa (F):

- ☐ El objetivo de un RAID es aumentar la capacidad de almacenamiento de un sistema informático.
- ☐ Cada disco de un RAID es una unidad lógica distinta para el SO.
- ☐ En una unidad CD RW se necesita un laser de más potencia para escribir un 0 que para escribir un 1.
- ☐ El bus PCI es un bus local.
- ☐ El Bus PCI tiene arbitraje centralizado.
- ☐ Un dispositivo que consuma 25 watios y tenga conexión USB puede conectarse a un bus USB y conseguir la alimentación eléctrica directamente del bus.
- ☐ Un bus síncrono requiere que todos los dispositivos conectados a él transmitan a la misma frecuencia.