

Previo Sesión 07

1. Suponiendo que cada elemento de $v[i]$ ocupa 1 byte, calculad cuántos fallos de cache provoca el acceso $v[i]$ en los siguientes casos:

Código	Memoria Cache	stepA	stepB	stepC	stepD
<pre>for (j=0, i=0; j<10000; j++) { sum = sum + v[i]; i = i + step; }</pre>	Cache Directa Tamaño: 4KB Tamaño línea: 8B	step = 1	step = 4	step = 8	step = 16
<pre>for (j=0, i=0; j<10000; j++) { sum = sum + v[i]; i = i + step; }</pre>	Cache 2-asociativa Tamaño: 4KB Tamaño línea: 16B	step = 1	step = 4	step = 8	step = 16

Cache Directa:

stepA = 1250 fallos

stepB = 5000 fallos

stepC = 10000 fallos

stepD = 10000 fallos

Cache 2-asociativa

stepA = 625 fallos

stepB = 2500 fallos

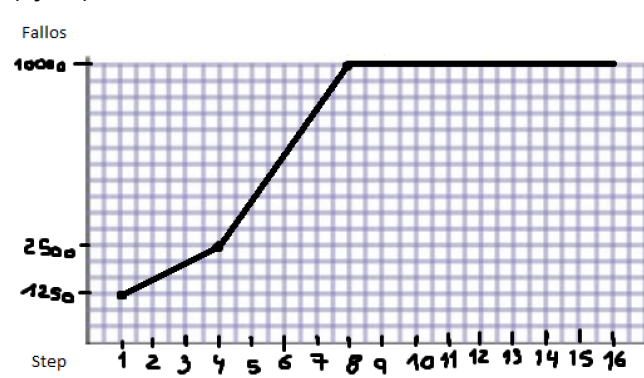
stepC = 5000 fallos

stepD = 10000 fallos

2. Dado el siguiente código y suponiendo que cada elemento de $v[i]$ ocupa 1 byte:

```
for (j=0, i=0; j<10000; j++) {
    sum = sum + v[i]; // cada elemento de v[i] ocupa 1 byte
    i = i + step;
}
```

Suponiendo que la cache es directa (16KB) con líneas de 8B, dibujad una gráfica donde se represente el número de fallos que se producen (eje y) variando la variable step de 1 a 16 (eje x)



3. Suponiendo que cada elemento de $v[i]$ ocupa 1 byte, calculad cuántos fallos de cache provoca el acceso $v[i]$ en los siguientes casos:

Código	Memoria Cache	Valores de limite
<pre>for (i=0, j=0; j<32; j++) { sum = sum + v[i]; i = i + 8; if (i >= limite) i = 0; }</pre>	Cache Directa Tamaño: 4 líneas Tamaño línea: 8B	limite= 16, 32, 40, 48, 64 y 128 o lo que es lo mismo limite=16B, 32B, 40B, 48B, 64B y 128B
<pre>for (i=0, j=0; j<32; j++) { sum = sum + v[i]; i = i + 8; if (i >= limite) i = 0; }</pre>	Cache 2-asociativa Tamaño: 4 líneas Tamaño línea: 8B	limite=16B, 32B, 40B, 48B, 64B y 128B
<pre>for (i=0, j=0; j<32; j++) { sum = sum + v[i]; i = i + 8; if (i >= limite) i = 0; }</pre>	Cache 4-asociativa Tamaño: 4 líneas Tamaño línea: 8B	limite=16B, 32B, 40B, 48B, 64B y 128B

Código	Memoria Cache	Valores de limite					
		16B	32B	40B	48B	64B	128B
<pre>for (i=0, j=0; j<32; j++) { sum = sum + v[i]; i = i + 8; if (i >= limite) i = 0; }</pre>	Cache Directa Tamaño: 4 líneas Tamaño línea: 8B	2	4	16	24	32	32
<pre>for (i=0, j=0; j<32; j++) { sum = sum + v[i]; i = i + 8; if (i >= limite) i = 0; }</pre>	Cache 2-asociativa Tamaño: 4 líneas Tamaño línea: 8B	2	4	24	32	32	32
<pre>for (i=0, j=0; j<32; j++) { sum = sum + v[i]; i = i + 8; if (i >= limite) i = 0; }</pre>	Cache 4-asociativa Tamaño: 4 líneas Tamaño línea: 8B	2	4	32	32	32	32

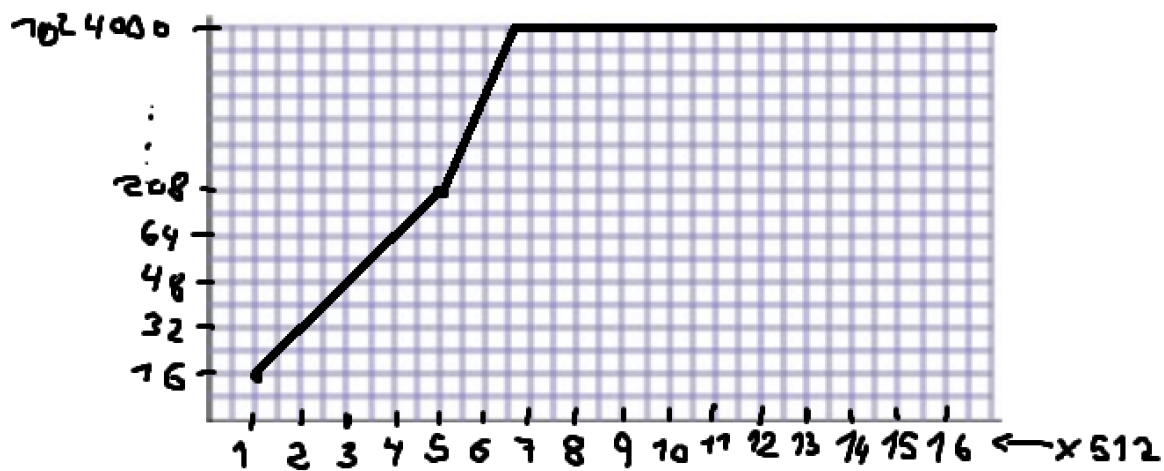
4. Dado el siguiente código:

```

for (i=0, j=0; j<M; j++) { // M vale 1024*1000
    sum = sum+ v[i];        // cada elemento de v[i] ocupa 1 byte
    i = i + 32;
    if (i >= limite) i = 0;
}

```

Suponiendo que la caché es directa (2KB) con líneas de 32B, dibujad una gráfica con los fallos que se producen (eje y) variando la variable límite de 512 a 8192 con incrementos de 512 (eje x).



5. Estudiaremos ahora la influencia de la asociatividad en el número de fallos. Dado el siguiente código:

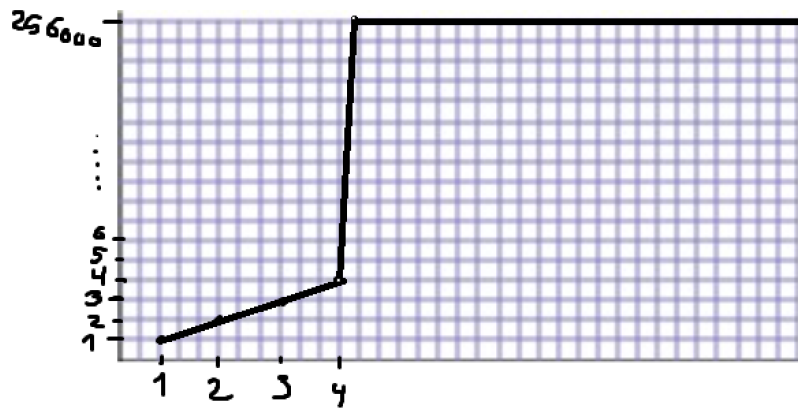
```

for (limite = 1; limite <= 32; limite++) {
    for (i=0, j=0; j<M; j++) { // M vale 256*1000
        if ((j % limite)==0) i=0;
        sum = sum+ v[i];        // cada elemento de v[i] ocupa 1 byte
        i=i+cache_size;
    }
}

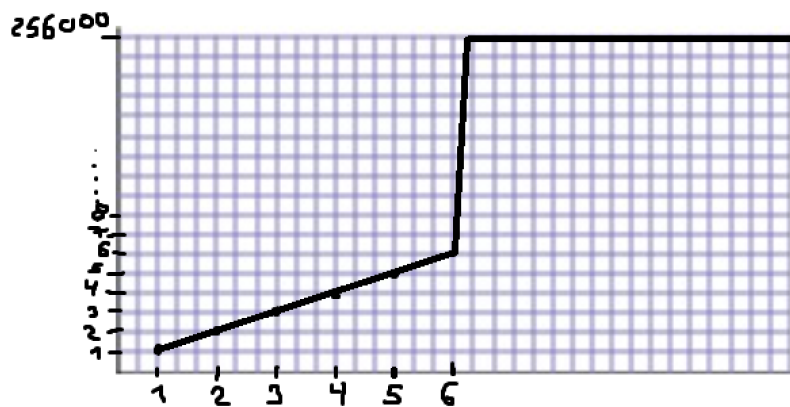
```

Suponiendo que la cache es de 3KB con líneas de 8 bytes, dibujad una gráfica con los fallos que se producen (eje y) respecto a la variable límite (eje x) en cada uno de los siguientes casos (no es necesario calcular todos los puntos):

a) asociatividad = 4



b) asociatividad = 6



¿Cuál es la relación entre el número de fallos, la variable límite y la asociatividad de la caché?

Si el límite supera el grado de asociatividad el número de fallos es muy alto (incluso llega al máximo), mientras que si el límite es inferior al grado de asociatividad el número de fallos es pequeño.