

Estructura de Computadores

7.3. Considerem un computador amb processador MIPS com l'estudiat a classe, que té una memòria cache de dades amb les següents característiques:

- associativa per conjunts de 4 vies (és a dir, de 4 blocs per conjunt)
- 64 conjunts
- 32 bytes per bloc
- algorisme de reemplaçament LRU

Considerem també que té un TLB de dades amb les següents característiques:

- completament associatiu
- 32 entrades
- mida de pàgina: 8 KB
- algorisme de reemplaçament LRU

Sobre aquest sistema s'executen 2 versions diferents d'una mateixa aplicació:

```
int A[128][1024]; /* emmagatzemada a partir de l'adreça 0 */

/* versió A */
sumA = 0;
for (i=0; i<128; i++)
    for (j=0; j<1024; j++)
        sumA = sumA + A[i][j];

/* versió B */
sumA = 0;
for (j=0; j<1024; j++)
    for (i=0; i<128; i++)
        sumA = sumA + A[i][j];
```

Suposant que les variables *i*, *j*, *sumA* s'emmagatzemen en registres, indica quantes fallades hi ha a la cache i al TLB, per a cada versió.

Problema - Parte 1

Disposen d'un processador amb sistema de memòria virtual paginada que té les següents característiques:

- Pàgines de 256 bytes
- Mida total de la taula de pàgines (TP) = 256 bytes
- Cada entrada de la TP consta dels bits P i D i del PPN.
- Espai de memòria física (MF) = 16 Kbytes
- Reemplaçament de pàgines físiques: LRU
- TLB totalment associatiu de 4 entrades

a) Omple les següents dades:

Nombre de pàgines físiques =

Nombre de bits d'una entrada de la TP =

Nombre de pàgines lògiques =

Nombre de bits de l'adreça lògica =

- b) Considerem els següents continguts inicials del TLB i d'una part de la taula de pàgines (TP). Suposarem que la resta d'entrades de la TP estan buides ($P=0$ i $D=0$) i que les pàgines físiques que no apareixen a la taula ($PPN = 5, 6, 7$, etc.) estan lliures. Per a noves pàgines a carregar a MF, s'assignaran números de pàgina física (PPN) correlatius, a partir del número de pàgina física lliure major (és a dir, $PPN = 5, 6, 7$, etc.). Per al funcionament de l'algorisme LRU del TLB suposarem que les entrades han estat accedides en l'ordre que apareixen a la taula.

VPN	TP		
	P	D	PPN
0	1	0	4
1	1	0	3
2	1	0	2
3	1	1	1
4	1	0	0
5	0	0	-
6	0	0	-
...	0	0	-

TLB			
V	D	VPN	PPN
1	0	1	3
1	0	2	2
1	1	3	1
1	0	4	0

(entrada més antiga)

La següent taula mostra una seqüència de referències a memòria (E: escriptura/ L: lectura). Emplena la taula fent servir la informació del TLB i la taula de pàgines.

adr. lògica (hex)		VPN	TLB		fallada de pàg? (SI/NO)	es llegirà del disc? (SI/NO)	s'escriurà al disc? (SI/NO)	PPN resultant de la traducció
			fallada TLB? (SI/NO)	pàg. reemplaçada (VPN)				
L	0x04F2							
E	0x00F1							
L	0x0132							
L	0x0541							
E	0x065A							

¿Verdadero o Falso?

1. Si l'accés a dades d'una instrucció produeix un encert al TLB, però el bit V val 0, llavors la instrucció causarà una excepció de fallada de pàgina.

¿Verdadero o Falso?

1. Si l'accés a dades d'una instrucció produeix un encert al TLB, però el bit V val 0, llavors la instrucció causarà una excepció de fallada de pàgina.
2. Una mateixa instrucció pot causar durant la seva execució 2 fallades de pàgina.

¿Verdadero o Falso?

1. Si l'accés a dades d'una instrucció produeix un encert al TLB, però el bit V val 0, llavors la instrucció causarà una excepció de fallada de pàgina.
2. Una mateixa instrucció pot causar durant la seva execució 2 fallades de pàgina.
3. Una fallada al TLB no implica que hi hagi una fallada de pàgina.

¿Verdadero o Falso?

1. Si l'accés a dades d'una instrucció produeix un encert al TLB, però el bit V val 0, llavors la instrucció causarà una excepció de fallada de pàgina.
2. Una mateixa instrucció pot causar durant la seva execució 2 fallades de pàgina.
3. Una fallada al TLB no implica que hi hagi una fallada de pàgina.
4. En memòria virtual paginada, sempre que reemplacem de la memòria física una pàgina cal escriure-la en disc.