

EC Examen de Problemes

Exercici 1. MC (Ex. Final 2012-2013 Q1)

Un sistema disposa d'un processador de 32 bits (adreces i dades de 32 bits). La cache d'instruccions podem suposar que és ideal (sempre encerta). La cache de dades (MC) té 512 bytes i la següent organització:

- Correspondència associativa per conjunts, de grau 2 (2 blocs per conjunt)
- Blocs de 16 bytes
- Reemplaçament LRU
- Escriptura immediata sense assignació.

Un programa executa el següent bucle en C, en què les dades accedides són totes de tipus int:

```
for (i = 0; i < 3; i++)  
    res = res + vec[i];
```

Assumint que i es guarda en un registre temporal, res és una variable global a l'adreça 0x10010000, i vec és un vector global guardat en una adreça no consecutiva (hi ha altres variables globals al codi sencer), completa la seqüència de referències a dades de memòria segons s'indica a la següent taula. A la taula apareixen les adreces en hexadecimal i si són lectures o escriptures (L/E). Completa les columnes que falten indicant, per a cada referència: el número de conjunt de MC; si és encert (e) o fallada (f); i el nombre de bytes de Memòria Principal (MP) llegits i/o escrits. Podeu assumir que inicialment la MC està buida.

| L/E | adr. lògica | núm. de conjunt | encert (e)/fallada (f) | bytes MP llegits | bytes MP escrits |
|-----|-------------|-----------------|------------------------|------------------|------------------|
| L | 0x100110F8 | | | | |
| L | | | | | |
| E | | | | | |
| L | | | | | |
| L | | | | | |
| E | | | | | |
| L | | | | | |
| L | | | | | |
| E | | | | | |

Exercici 2. MC (Ex. Final 2019-2020 Q1)

Tenim un programa que, donada una matriu M, calcula la seva matriu trasposada T. El codi del programa és el següent:

```
int M[32][64], T[64][32];

for (int i=0; i<64; i++)
    for (int j=0; j<32; j++)
        T[i][j] = M[j][i];
```

Al compilar el programa, el compilador emmagatzema la matriu M a l'adreça 0x10010000 i la matriu T immediatament a continuació, mentre que les variables i i j s'emmagatzemen a registres del processador. El programa s'executa a un processador de 32 bits amb una memòria cache amb les següents característiques:

- Capacitat de 64 blocs de 32 Bytes cadascun
- Completament associativa amb algorisme de reemplaçament LRU
- Escriptura retardada amb assignació

Es demana:

- a) Quantes fallades de cache es produiran durant l'execució del programa?. Pots argumentar la resposta.
- b) Quantes fallades de cache es produiran durant l'execució del programa si només es canvia la política d'escriptura de la cache a escriptura immediata sense assignació?

Exercici 3. MV (Ex. Final 2018-2019 Q1)

La memòria virtual implementada en un sistema computador de 32 bits es caracteritza pels següents paràmetres:

- Pàgines de 4 KB de mida.
- Un màxim de 5 pàgines carregades simultàniament a memòria física per aplicació.
- Reemplaçament de pàgines a memòria física seguint l'algorisme LRU.
- TLB totalment associatiu de 8 entrades amb reemplaçament LRU.

Donat el següent codi en C:

```
int V[8192];
main() {
    int i;
    int sum = 0;
    for (i=0; i < 8192; i++) {
        sum += V[i] + V[8191 - i];
    }
}
```

Considera que les variables locals i i sum s'emmagatzemen en registres, que el vector global V s'emmagatzema a memòria a partir de l'adreça 0x00000000, i que el codi s'emmagatzema a partir de l'adreça 0x0000C000, i ocupa menys d'una pàgina. El TLB i la memòria física estan inicialment buits. Es demana:

- a) Quantes pàgines ocupa el vector V?

- b) Quantes fallades de TLB (codi i dades) es produiran en tota l'execució del programa?
- c) Quantes fallades de pàgina (codi i dades) es produiran en tota l'execució del programa?
- d) Indica els VPN (en hexadecimal) de les cinc pàgines (codi i/o dades) que hi haurà carregades a memòria física quan s'acabi l'execució d'aquest programa.

Exercici 4. MV (Ex. Final 2016-2017 Q2)

Considera un computador MIPS que gestiona memòria virtual paginada amb les pàgines de 4 KB i amb la restricció que el sistema operatiu admet fins a 6 pàgines com a màxim carregades a memòria física per programa.

L'algorisme de reemplaçament de pàgines dins de memòria física és LRU. La gestió de memòria virtual es realitza amb un TLB de 4 entrades, totalment associatiu i amb reemplaçament LRU.

```
int V[102400];
main(){
    int i, sum=0;
    for (i=0; i<102400; i++) sum += V[i];
}
```

Considera que les variables globals s'ubiquen a partir de l'adreça 0x10010000 i que el codi s'ubica a partir de l'adreça 0x00400000. El contingut del TLB i de memòria física (MF) en un moment donat de l'execució d'aquest programa, en què s'acaba de tractar una fallada de pàgina, és el següent.

| V | D | VPN | PPN |
|---|---|---------|-----|
| 1 | 0 | 0x00400 | 0 |
| 1 | 0 | 0x10024 | 5 |
| 1 | 0 | 0x10022 | 3 |
| 1 | 0 | 0x10023 | 4 |

Table 1: TLB

| PPN | VPN |
|-----|---------|
| 0 | 0x00400 |
| 1 | 0x01020 |
| 2 | 0x01021 |
| 3 | 0x01022 |
| 4 | 0x01023 |
| 5 | 0x01024 |

Table 2: MF

Es demana:

- a) Quin és el VPN (en hexadecimal) de la pàgina que s'acaba de carregar a MF?
- b) Quantes pàgines ocupa el vector V?
- c) Quantes fallades de pàgina es produiran en tota l'execució del programa?
- d) Quantes fallades de TLB es produiran en tota l'execució del programa?
- e) Quins 4 VPN (en hexadecimal) contindrà el TLB quan s'acabi d'executar tot el programa?