

Laboratori Sessió 08: Polítiques d'escriptura

Objectiu de la Sessió

L'objectiu de la sessió és consolidar les polítiques d'escriptura. Per fer-ho, programareu un simulador de cache bàsic que simuli una cache de lectura/escriptura.

Característiques de la cache

En aquesta sessió farem servir una memòria cache amb les següents característiques:

- Les adreces són de 32 bits (per simplificar, assumirem que tots els accessos són a byte)
- Algorisme d'emplaçament: Emplaçament directe
- Mida Cache: 4 Kbytes
- Mida Línia: 32 bytes
- Política d'escriptura: Write Through
- Política d'al·locació: NO Write Allocate

Presa de contacte amb l'entorn del simulador

El simulador és composta de 3 fitxers: CacheSimWT.o, CacheSim.h i MiSimulador.c

El programa principal i alguns components del simulador ja estan programats i és troben al fitxer CacheSimWT.o. Aquest fitxer s'encarrega de generar seqüències de test, d'imprimir els resultats de la simulació per pantalla amb un format agradable i de comprovar el correcte funcionament del vostre simulador.

Abans de que comenceu a programar el simulador, és interessant fer algunes proves amb aquest entorn. Per començar, compileu el simulador (MiSimulador.c no funciona correctament, però compila)

```
$> gcc CacheSimWT.o MiSimulador.c -o sim
```

El programa disposa de 3 tests:

- Test 0: Genera la seqüència de 20 referències de la Taula del treball previ.
- Test 1: Genera accessos seqüencials a un vector d'enters (1000 referències)
- Test 2: Genera els accessos d'un producte de matrius de 25x25 (62500 referències)

Per passar qualsevol dels tests, només cal posar el nº de test com a paràmetre del simulador. Per exemple, per passar el test 0 faríem:

```
$> sim 0
Test 0 FAIL :-(
$>
```

Evidentment el test ha fallat, ja que encara no hem programat el simulador. En cas de que el simulador falli, ens interessarà veure que esta passant. Per això, podem utilitzar la opció "v" (de verbose) al simulador (la v cal que aparegui com el primer paràmetre):

```
$> sim v 0
eca130 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
eca131 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec2172 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
eca133 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec3175 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec3175 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ecb136 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
eca137 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec2178 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ecb139 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
10eca230 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
eca131 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec2172 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
10eca233 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec3175 E -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec3175 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ecb136 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
10eca237 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ec2278 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
ecb139 L -> 1 MP:4212dfa0 1 MC:bffffe7a0 TAG:804a000 byte:40007dcf MISS L:134513233 E:1073774031 -> fd0
Test 0 FAIL :-(
```

Aquesta opció ens donarà una sortida semblant a l'anterior. Com podem veure, les columnes és corresponen bàsicament a la taula del treball previ. D'aquesta manera, comparant la sortida i la taula podem veure on és el problema. Donat que en els tests 1 i 2 el nº de referències és molt alt, us recomanem que no els proveu fins que us funcioni perfectament el test 0. Amb la opció v, els tests 1 i 2 es paren tan bon punt apareix el primer error per tal de facilitar la seva identificació.

Programació del mòdul "MiSimulador.c"

Per programar el vostre simulador de cache, heu de programar 3 seccions del fitxer "MiSimulador.c":

Estructures globals

En aquesta secció heu de declarar les estructures de dades globals necessàries per mantenir l'estat de la cache. Es necessari que siguin globals, ja que la part principal del simulador es la rutina "reference", que s'executa un cop per referència i, com ja sabeu molt be del tema de subrutines, el seu estat desapareix un cop executada.

Inicialització de la cache

La rutina "init_cache" es cridada abans de passar cada test per inicialitzar les estructures de dades globals necessàries. L'objectiu és deixar la cache en un estat inicial correcte (cache buida).

Simulació de referències

La simulació de les referències cal fer-la a la rutina "reference". Aquesta rutina és cridada un cop per cada referència a simular. Només cal que genereu el valor correcte de les 11 variables locals que ja teniu declarades a l'inici de la rutina, i que és corresponen bàsicament columnes de l'exercici previ.

```
void reference (unsigned int address, unsigned int LE)
{
    unsigned int byte;
    unsigned int linea_mp;
    unsigned int linea_mc;
    unsigned int tag;
    unsigned int miss;
    unsigned int lec_mp;          // booleà que indica si es llegeix de MP
    unsigned int mida_lec_mp;
    unsigned int esc_mp;         // booleà que indica si s'escriu a MP
    unsigned int mida_esc_mp;
    unsigned int replacement;    // booleà que indica si es reeplàça una línia vàlida
    unsigned int tag_out;        // TAG de la línia reemplaçada
```

En altres paraules, el que heu de fer es implementar l'algorisme que, de forma intuïtiva, heu fet servir manualment per emplenar la taula prèvia.

Després del vostre codi, la rutina acaba amb una crida a la rutina "test_and_print" per comprovar si els valors de les variables son correctes i treure'ls per pantalla en cas de tenir la opció "v" activada.

Resultats de la sessió

- Abans d'acabar la sessió, heu d'haver programat el simulador de cache de lectura/escriptura amb polítiques Write Through + NO Write Allocate i comprovat el seu funcionament correcte.
- Si teniu temps, implementeu també un simulador amb les mateixes característiques, però amb polítiques d'escriptura Copy Back + Write Allocate. Per comprovar el seu funcionament compileu amb el fitxer CacheSimCB.o

Notes

Al fitxer "CacheSim.h" hi ha declarades les constants "true" i "false", que podeu fer servir per assignar a les variables booleans.

El parametre "unsigned int LE" pot tenir dos valors: "lectura" o "escriptura", constants que també estan declarades a "CacheSim.h".

Els fitxers per aquesta sessió els podeu trobar a la pàgina web de la assignatura: "Programas.Sesion08.tar.gz".

Treball previ a la sessió

1) Empleneu la taula de sota, indicant per cada referència, la informació següent:

- el byte de la línia que és accedit (**byte**),
- la línia de memòria principal (**línia MP**),
- la línia de memòria cache on es mapejarà la referència (**línia MC**),
- la etiqueta (**TAG**) que es guardarà d'aquesta referència,
- si el accés és **HIT** o **MISS**,
- si hi ha lectura a memòria principal indicar la mida en bytes (**lec MP**),
- si hi ha escriptura a memòria principal indicar la mida en bytes (**esc MP**),
- i en cas que es reemplaci una línia vàlida, el TAG de la línia reemplaçada (**TAG out**).

La memòria cache té les següents característiques:

- Les adreces són de 32 bits (per simplificar, assumirem que tots els accessos són a byte)
- Algorisme d'emplaçament: Emplaçament directe
- Mida Cache: 4 Kbytes
- Mida Línia: 32 bytes
- Política d'escriptura Write Through
- Política d'al·locació NO Write Allocate

adreça	L/E	byte	línia MP	línia MC	TAG	hit/miss	lec MP	esc MP	TAG out
00eca130	L								
00eca131	E								
00ec2172	E								
00eca133	L								
00ec3175	L								
00ec3175	L								
00ecb136	L								
00eca137	E								
00ec2178	L								
00ecb139	E								
10eca230	L								
00eca131	E								
00ec2172	L								
10eca233	E								
00ec3175	E								
00ec3175	L								
00ecb136	L								
10eca237	L								
00ec2278	L								
00ecb139	L								

Nota: Emplenar les columnes **byte**, **línia MP**, **línia MC**, **TAG** i **TAG out** en hexa.

2) Empleneu la taula de sota, indicant per cada referència, la informació següent:

- el byte de la línia que és accedit (**byte**),
- la línia de memòria principal (**línia MP**),
- la línia de memòria cache on es mapejarà la referència (**línia MC**),
- la etiqueta (**TAG**) que es guardarà d'aquesta referència,
- si el accés és **HIT** o **MISS**,
- si hi ha lectura a memòria principal indicar la mida en bytes (**lec MP**),
- si hi ha escriptura a memòria principal indicar la mida en bytes (**esc MP**),
- i en cas que es reemplaci una línia vàlida, el TAG de la línia reemplaçada (**TAG out**).

La memòria cache té les següents característiques:

- Les adreces són de 32 bits (per simplificar, assumirem que tots els accessos són a byte)
- Algorisme d'emplaçament: Emplaçament directe
- Mida Cache: 4 Kbytes
- Mida Línia: 32 bytes
- Política d'escriptura Copy Back
- Política d'al·locació Write Allocate

adreça	L/E	byte	línia MP	línia MC	TAG	hit/miss	lec MP	esc MP	TAG out
00eca130	L								
00eca131	E								
00ec2172	E								
00eca133	L								
00ec3175	L								
00ec3175	L								
00ecb136	L								
00eca137	E								
00ec2178	L								
00ecb139	E								
10eca230	L								
00eca131	E								
00ec2172	L								
10eca233	E								
00ec3175	E								
00ec3175	L								
00ecb136	L								
10eca237	L								
00ec2278	L								
00ecb139	L								

Nota: Emplenar les columnes **byte**, **línia MP**, **línia MC**, **TAG** i **TAG out** en hexa.