

Fonaments dels Computadors

Arquitectura del Processador: Jerarquia de Memoria

Exercicis

Grau en Intel·ligència Artificial

Xavier Martorell

Xavi Verdú

Facultat d'Informàtica de Barcelona (FIB)
Universitat Politècnica de Catalunya (UPC)

Creative Commons License

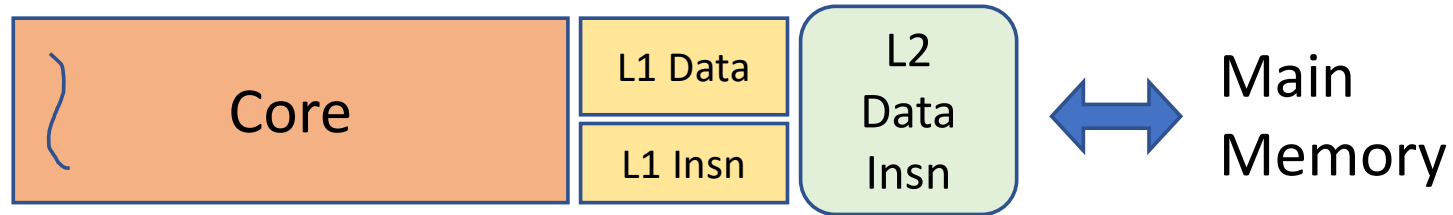
Aquest document utilitza Creative Commons Attribution 3.0 Unported License



Els detalls d'aquesta licencia es poden trobar a <https://creativecommons.org/licenses/by-nc-nd/4.0>

Exercici 1

- Tenim un processador amb 1 core i 1 hw thread (línia blava serpentejant). Una jerarquia de memòria amb una L1 d'instruccions, una L1 de dades, una L2 unificada i memòria RAM.



- La jerarquia de memòria, té:
 - una L1 d'instruccions de 16Bytes de mida de línia, un total de 4 sets, d'1 línia cadascun
 - una L1 de dades de 16Bytes de mida de línia, un total de 8 sets, d'1 línia cadascun
 - una L2 de dades i instruccions compartida de 32Bytes de mida de línia, un total de 16 sets, d'1 línia cadascun
 - memòria RAM amb una capacitat de 2 GB (Gigabyte)

Exercici 1

- quina és la mida (en Bytes o Kbytes) de cada unitat de la jerarquia de memòria cache?
 - L1 Instruccions:
 - L1 Dades:
 - L2:
- Assumint que l'arquitectura és de 32bits, quina distribució de bits de l'adreça s'utilitzarà pel tag, set i block offset, en base a les especificacions indicades en l'enunciat?
 - L1 Instruccions:
 - L1 Dades:
 - L2:
- Assumint que l'arquitectura és de 32bits, quina és la quantitat màxima de memòria que pot adreçar aquesta arquitectura?

Exercici 1

En la taula de la propera transparència, indica UNICAMENT els accessos generats per accedir a la matriu “A”.
Omple les adreces que s’accedirien al executar aquest codi:

```
#define MAX_ROWS 3
#define MAX_COLS 3
int A[MAX_ROWS][MAX_COLS], B[MAX_ROWS][MAX_COLS];
main(int argc, char **argv){
    int i, j, k, res;

    for(i=0; i<2; i++){
        for(j=0; j<MAX_COLS; j++){
            for(k=0; k<MAX_COLS; k++){
                res = A[j][k] + B[j][k];
            }
        }
    }
}
```

Exercici 1

La primera adreça correspon a la primera posició de la matriu i primera iteració del bucle

Ordre	@	L1 Dades	L2	Justificació (a més, si és miss, indica el tipus de miss)
1	0x01A000			
2				
3				
4				
5				
6				
7				
8				
9				
10				

Exercici 2

- Determina com el programa accedeix a la memòria RAM i com la jerarquia de memòria porta les dades a la cache del nivell L1
- RAM**

Cache

8 línies
16 bytes per línia
total: 128 bytes

Assumint

&v[0] == 0x0100

Programa

```
int v[80];
int r = 0;
int i;
for (i = 0; i < 80; i++) {
    r += v[i];
}
```

L1 data cache

tags

data

[illegible]

RAM

0x0000

0x0100 v[0]

0x0120

0x0140

0x0160

0x0180

0x01A0

0x01C0

0x01E0

0x0200

0x0220

0x0240

Exercici 3

- Determina com la variable “matrix” es mapeja en la memòria RAM usant una distribució per files - **row-major** - i sabent que l’arquitectura és **little-endian**
- RAM**

Cache

8 lines

16 bytes per line

total: 128 bytes

Assume

```
&matrix[0][0] == 0x0100
```

Data

```
int matrix [3][6];
```

4	6	8	10	12	14
13	15	17	19	21	23
22	24	26	28	30	32
31	33	35	37	39	41
40	42	44	46	48	50
39	41	43	45	47	49

tags

L1 data cache

data

0x0100

[illegible]

RAM

0x0000

0x0100

0x0120

0x0140

0x0160

0x0180

0x01A0

0x01C0

0x01E0

0x0200

0x0220

0x0240

Exercici 4

- Determina com la variable “matrix” es mapeja en la memòria RAM usant una distribució per columnes - **column-major** - i sabent que l’arquitectura és **little-endian**
- RAM**

Cache

8 lines

16 bytes per line

total: 128 bytes

Assume

```
&matrix[0][0] == 0x0100
```

Data

```
int matrix [3][6];
```

4	6	8	10	12	14
13	15	17	19	21	23
22	24	26	28	30	32
31	33	35	37	39	41
40	42	44	46	48	50
39	41	43	45	47	49

0x0000

RAM

0x0100

0x0120

0x0140

0x0160

0x0180

0x01A0

0x01C0

0x01E0

0x0200

0x0220

0x0240

tags

L1 data cache

data

0x0100

Exercici 5

- Indica “Hit”, “Miss” or “---” segons els accessos fets a la jerarquia de memòria

Jerarquia de cache

L1-Ins.: 8 línies; 64B/línia
L1-Data: 8 línies; 64B/línia
L2: 16 línies; 256B/línia

- Identifiqueu alguna classe de localitat en l'accés a les dades?

Order	Instr/ data	@	L1 Ins.	L1 Data	L2 (Ins. & Data)
1	Instr.	0x001000			
2	Instr.	0x001004			
3	Dada	0x00B000			
4	Dada	0x00B004			
5	Instr.	0x001000			
6	Instr.	0x001004			
7	Dada	0x00B008			
8	Dada	0x00B00C			
9	Instr.	0x001000			
10	Instr.	0x001004			
11	Dada	0x00B010			
12	Dada	0x00B014			

Exercici 6

- Indica “Hit”, “Miss” or “---” segons els accessos fets a la jerarquia de memòria

Jerarquia de cache

L1-Ins.: 8 línies; 64B/línia
L1-Data: 8 línies; 64B/línia
L2: 16 línies; 256B/línia

- Identifiques alguna classe de localitat?**
- Compara la locality de dades amb la que has vist en l’Exercici 5**

Order	Instr/ data	@	L1 Ins.	L1 Data	L2 (Ins. & Data)
1	Instr.	0x002000			
2	Instr.	0x002004			
3	Dada	0x010000			
4	Dada	0x010084			
5	Instr.	0x002008			
6	Instr.	0x00200C			
7	Dada	0x010108			
8	Dada	0x010118			
9	Instr.	0x002200			
10	Instr.	0x002204			
11	Dada	0x0100F0			
12	Dada	0x010200			