**Hands On Lab: Memoria**

1. Avaluant matrius…

* **Fes una taula amb les mides de la matriu i els temps d’execució pels dos recorreguts.**

Matriu de 64x64

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 6 µsegons | Temps 1: 10  Temps 2: 6  Temps 3: 7 | Mitja de temps: 7,67 | Temps 1: 8  Temps 2: 12  Temps 3: 12 | Mitja de temps: 10,67 |
| 2 | 7 µsegons | Temps 1: 7  Temps 2: 10  Temps 3: 10 | Mitja de temps: 9,00 | Temps 1: 6  Temps 2: 11  Temps 3: 6 | Mitja de temps: 7,67 |
| 3 | 5 µsegons | Temps 1: 10  Temps 2: 10  Temps 3: 5 | Mitja de temps: 8,33 | Temps 1: 11  Temps 2: 11  Temps 3: 6 | Mitja de temps: 9,33 |
| 4 | 10 µsegons | Temps 1: 10  Temps 2: 9  Temps 3: 6 | Mitja de temps: 8,33 | Temps 1: 11  Temps 2: 11  Temps 3: 11 | Mitja de temps: 11,00 |
| 5 | 10 µsegons | Temps 1: 10  Temps 2: 5  Temps 3: 5 | Mitja de temps: 6,67 | Temps 1: 6  Temps 2: 11  Temps 3: 6 | Mitja de temps: 7,67 |

Matriu de 128x128

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 35 µsegons | Temps 1: 35  Temps 2: 40  Temps 3: 37 | Mitja de temps: 37,33 | Temps 1: 29  Temps 2: 28  Temps 3: 29 | Mitja de temps: 28,67 |
| 2 | 37 µsegons | Temps 1: 37  Temps 2: 40  Temps 3: 37 | Mitja de temps: 38,00 | Temps 1: 28  Temps 2: 28  Temps 3: 28 | Mitja de temps: 28,00 |
| 3 | 32 µsegons | Temps 1: 37  Temps 2: 39  Temps 3: 40 | Mitja de temps: 38,67 | Temps 1: 37  Temps 2: 28  Temps 3: 29 | Mitja de temps: 31,33 |
| 4 | 37 µsegons | Temps 1: 37  Temps 2: 41  Temps 3: 37 | Mitja de temps: 38,33 | Temps 1: 45  Temps 2: 28  Temps 3: 46 | Mitja de temps: 39,67 |
| 5 | 37 µsegons | Temps 1: 40  Temps 2: 43  Temps 3: 38 | Mitja de temps: 40,33 | Temps 1: 28  Temps 2: 30  Temps 3: 28 | Mitja de temps: 28,67 |

Matriu de 256x256

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 134 µsegons | Temps 1: 129  Temps 2: 128  Temps 3: 123 | Mitja de temps: 126,67 | Temps 1: 165  Temps 2: 162  Temps 3: 165 | Mitja de temps: 164,00 |
| 2 | 154 µsegons | Temps 1: 140  Temps 2: 144  Temps 3: 138 | Mitja de temps: 140,67 | Temps 1: 192  Temps 2: 188  Temps 3: 185 | Mitja de temps: 188,33 |
| 3 | 166 µsegons | Temps 1: 150  Temps 2: 138  Temps 3: 140 | Mitja de temps: 142,67 | Temps 1: 176  Temps 2: 175  Temps 3: 187 | Mitja de temps: 179,33 |
| 4 | 139 µsegons | Temps 1: 139  Temps 2: 133  Temps 3: 146 | Mitja de temps: 139,33 | Temps 1: 172  Temps 2: 178  Temps 3: 175 | Mitja de temps: 175,00 |
| 5 | 141 µsegons | Temps 1: 134  Temps 2: 144  Temps 3: 140 | Mitja de temps: 139,33 | Temps 1: 183  Temps 2: 172  Temps 3: 182 | Mitja de temps: 179,00 |

Matriu de 512x512

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 453 µsegons | Temps 1: 455  Temps 2: 453  Temps 3: 451 | Mitja de temps: 453,00 | Temps 1: 713  Temps 2: 710  Temps 3: 694 | Mitja de temps: 705,67 |
| 2 | 472 µsegons | Temps 1: 481  Temps 2: 541  Temps 3: 625 | Mitja de temps: 549,00 | Temps 1: 774  Temps 2: 745  Temps 3: 746 | Mitja de temps: 755,00 |
| 3 | 498 µsegons | Temps 1: 482  Temps 2: 489  Temps 3: 490 | Mitja de temps: 487,00 | Temps 1: 758  Temps 2: 753  Temps 3: 781 | Mitja de temps: 764,00 |
| 4 | 492 µsegons | Temps 1: 519  Temps 2: 489  Temps 3: 549 | Mitja de temps: 519,00 | Temps 1: 788  Temps 2: 781  Temps 3: 762 | Mitja de temps: 777,00 |
| 5 | 512 µsegons | Temps 1: 507  Temps 2: 531  Temps 3: 513 | Mitja de temps: 517,00 | Temps 1: 803  Temps 2: 803  Temps 3: 770 | Mitja de temps: 792,00 |

Matriu de 1024x1024

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 1797 µsegons | Temps 1: 1988  Temps 2: 1923  Temps 3: 1963 | Mitja de temps: 1958,00 | Temps 1: 4279  Temps 2: 3702  Temps 3: 3306 | Mitja de temps: 3762,33 |
| 2 | 1831 µsegons | Temps 1: 1872  Temps 2: 1922  Temps 3: 1980 | Mitja de temps: 1924,67 | Temps 1: 4566  Temps 2: 4139  Temps 3: 3618 | Mitja de temps: 4107,67 |
| 3 | 2042 µsegons | Temps 1: 2231  Temps 2: 1893  Temps 3: 1940 | Mitja de temps: 2021,33 | Temps 1: 3737  Temps 2: 4246  Temps 3: 4041 | Mitja de temps: 4008,00 |
| 4 | 2017 µsegons | Temps 1: 2012  Temps 2: 1879  Temps 3: 1887 | Mitja de temps: 1926,00 | Temps 1: 3792  Temps 2: 3675  Temps 3: 2941 | Mitja de temps: 3469,33 |
| 5 | 1729 µsegons | Temps 1: 1798  Temps 2: 1698  Temps 3: 1685 | Mitja de temps: 1727,00 | Temps 1: 3611  Temps 2: 3427  Temps 3: 4929 | Mitja de temps: 3989,00 |

Matriu de 2048x2048

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 5165 µsegons | Temps 1: 5368  Temps 2: 5472  Temps 3: 5361 | Mitja de temps: 5400,33 | Temps 1: 33512  Temps 2: 33177  Temps 3: 29950 | Mitja de temps: 32213,00 |
| 2 | 4813 µsegons | Temps 1: 5446  Temps 2: 5266  Temps 3: 5088 | Mitja de temps: 5266,67 | Temps 1: 30354  Temps 2: 31300  Temps 3: 31798 | Mitja de temps: 31150,67 |
| 3 | 5299 µsegons | Temps 1: 5046  Temps 2: 5232  Temps 3: 5329 | Mitja de temps: 5202,33 | Temps 1: 31961  Temps 2: 29804  Temps 3: 31604 | Mitja de temps: 31123,00 |
| 4 | 5423 µsegons | Temps 1: 5423  Temps 2: 5154  Temps 3: 5094 | Mitja de temps: 5223,67 | Temps 1: 32285  Temps 2: 30948  Temps 3: 29276 | Mitja de temps: 30836,33 |
| 5 | 5549 µsegons | Temps 1: 5325  Temps 2: 5435  Temps 3: 5262 | Mitja de temps: 5340,67 | Temps 1: 31884  Temps 2: 31367  Temps 3: 32021 | Mitja de temps: 31757,33 |

Matriu de 4096x4096

| **Nº de vegades executat** | **Inicialització** | **Fila (µsegons)** | **Mitjana per fila (µsegons)** | **Columa (µsegons)** | **Mitjana per columna (µsegons)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 27660 µsegons | Temps 1: 29334  Temps 2: 27476  Temps 3: 25440 | Mitja de temps: 27416,67 | Temps 1: 202253  Temps 2: 210451  Temps 3: 200421 | Mitja de temps: 204375,00 |
| 2 | 26408 µsegons | Temps 1: 24724  Temps 2: 23795  Temps 3: 26100 | Mitja de temps: 24873,00 | Temps 1: 206816  Temps 2: 191936  Temps 3: 206150 | Mitja de temps: 201634,00 |
| 3 | 25076 µsegons | Temps 1: 26301  Temps 2: 24732  Temps 3: 29378 | Mitja de temps: 26803,67 | Temps 1: 243667  Temps 2: 248421  Temps 3: 231174 | Mitja de temps: 241087,33 |
| 4 | 24548 µsegons | Temps 1: 23268  Temps 2: 24413  Temps 3: 25418 | Mitja de temps: 24366,33 | Temps 1: 202175  Temps 2: 205512  Temps 3: 204223 | Mitja de temps: 203970,00 |
| 5 | 25296 µsegons | Temps 1: 25754  Temps 2: 28058  Temps 3: 27581 | Mitja de temps: 27131,00 | Temps 1: 226349  Temps 2: 216389  Temps 3: 215395 | Mitja de temps: 219377,67 |

* **Es similar el temps d’execució quan es recorre per files de quan es recorre per columnes? (Assegurat de fer un parell d’execucions per estar-ne segur)**

Al principi, en matrius de menor escala, sí que són similars, però a mesura que creix la dimensió de la matriu, el temps d’execució de les columnes és molt superior al de les files.

* **A la vista de la taula, què en pots extreure? Hi ha algun canvi en els temps d’execucció entre els 2 recorreguts? Quant ocupa 1 fila? Per quècreus que passa?**

Amb la taula (concretament la de 512x512), podem extreure que comença a haver-hi un canvi en el temps d’execució d’uns 250-280 μs. Les files, en aquest cas, ocupen 512 Bytes, mentre que la matriu ocupa 262144 Bytes. Aquest canvi en el temps d’execució és degut a, possiblement, a què les dades es guardin a la memòria segons la fila, per tant, és més eficient llegir la informació d’aquesta manera.

* **Comprova els nivells de cache del teu processador [CPUDB] amb els resultats que t’han sortit a la taula.**

Els nivells de cache de l'ordinador amb el qual s’ha executat el programa són els següents:

Nivell 1: 8x32 KB en cache d’instruccions i 8x48 KB en caches de dades.

Nivell 2: 8x1,25 MB.

Nivell 3: 24 MB.

1. Patrons d’accés i impacte de la cache

* **Apunta el temps d’execució per la versió triada:**

Versió JKI

Dimensió = 256: t= 57,778 milisegons.

Dimensió = 512: t= 849,848 milisegons

Dimensió = 1024: t= 9683,562 milisegons

* **Supossant que cada element de la matriu ocupa 4 bytes, i que la mida de pàgina del nostre sistema és de 4Kbytes calcula pel cas triat el nombre de pàgines de memòria virtual que s’usen a l’executar completament el bucle més intern 1 cop. Per exemple, en el cas mm-ijk, per cada execució de tot el bucle ‘k’ la matriu C només accedeix a 1 element (4bytes), mentres que la matriu A accedeix a tota 1 fila (elements consecutius) i la matriu B a tota 1 columna (elements no consecutius).**

Versió JKI

Dimensió = 256: A = 256 || B= 1 || C= 256

Dimensió = 512: A = 512 || B= 1 || C= 512

Dimensió = 1024: A = 1024 || B= 1 || C=1024

1. Memòria dinàmica
2. Motxil.la bruixot

* Crea un programa que mostri un menu similar a: Current Bag has a total of %d elements What do you want to do? 1) Add a new element 2) List elements Enter command or Ctrl-C to exit: Que es repeteixi continuament fins que es premi Ctrl-C. Aquest menu mostra el nombre d’objectes total dins la motxil.la, permet afegir-hi 1 objecte i llistar els objectes continguts. Completa el codi dins ‘geraldBag.c’ que ja conté la major part del codi.

***Arxiu geraldBag.c afegit a la tasca.***

1. Limitacions [Opcional]

* **Executa el codi *mem1*, observa el resultat i intenta deduir què està pasant**

Els primers 10 accessos a la memòria són legals, per tal deixa escriure sense cap problema (definits a char \* x = malloc(10)).

Els próxims 10, encara que son ilegals, poden caure dins d'una zona de memòria que el procés pot accedir, encara que no es troben dins del rang assignat específicament a “char \* x = malloc(10)”.

Després, el programa executa un bucle “infinit”, que s’acaba forçosament quan s’intenta escriure a una adreça de memòria limitada pel sistema operatiu. En el nostre cas, arriba fins a 28310 accessos il·legals abans de tancar-se el programa.

* **Executa el codi *mem2*, observa el resultat i intenta deduir què està pasant**

En aquest programa, al igual que el primer, reserva X amb malloc(10), llavors escriu de manera legal sobre 10 direccions de memoria. Posteriorment, aquest espai s’allibera amb free(x).

A continuació, es reserva Y de la mateixa manera que es reserva X, però en escriure més enllà de la desena direcció (espai que ja no està reservat), concretament fins a la posició 133000 (designat al bucle for). En el nostre cas, el programa s’atura abans d’arribar a aquesta direcció, però en el codi, després d’arribar, es vol alliberar l’espai d’Y amb free(y), però en aquest punt la memòria es troba totalment corrompuda. Finalment, el programa vol reservar X un altre cop, però a causa de la corrupció anterior, aquest pot fallar o no realitzar l’ordre correctament.