***Delahaye Chloé Master 1 IRCOMS***

***Teboul Nathan***

Compte rendu TP

-

Conception d’un simulateur de mémoire cache

***Sommaire***

Table des matières

[Introduction : 3](#_Toc528521225)

[Explication de l’algorithme 4](#_Toc528521226)

[1) Structure et tableau 4](#_Toc528521227)

[2) Initialisation 4](#_Toc528521228)

[3) Traitement 5](#_Toc528521229)

[4) Affichage 7](#_Toc528521230)

[Résultats Expérimentaux 8](#_Toc528521231)

[1) mergesort2000Trace 8](#_Toc528521232)

[2) multTrace 8](#_Toc528521233)

[3) multTrace10x10 9](#_Toc528521234)

# Introduction :

Le but de ce TP est d’écrire un programme capable de simuler le fonctionnement d’une mémoire cache.

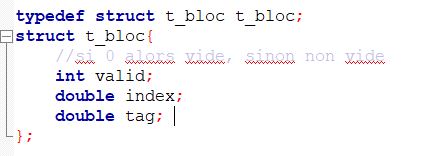
Pour ce TP, la valeur ASCII de la lettre D étant 68 et celle de la lettre T 84, le code ASCII du binôme est 152.

Les méthodes utilisées pour ce TP seront donc :

* Pour la technique d’écriture, l’écriture direct (ou Write Through (WT)) 🡪 on écrit en même temps en mémoire et en mémoire cache. Il n’est pas nécessaire de faire es sauvegardes.
* Pour le remplacement des blocs, la méthodes FIFO 🡪 le premier bloc entré, premier sorti.

# Explication de l’algorithme

## Structure et tableau



cache

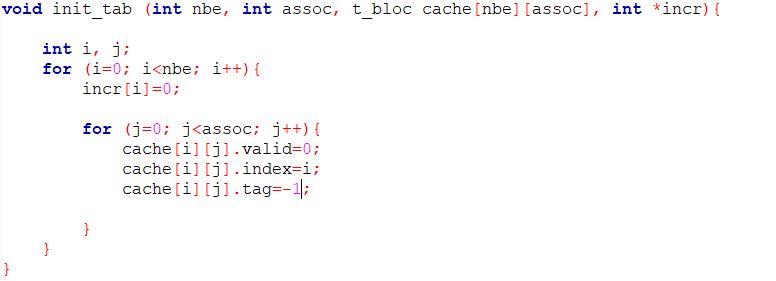
compteur_FIFO

Nous avons tout d’abord créé un structure t\_bloc représentant un bloc de la mémoire cache. Chaque bloc est composé d’un valid, d’un index, et d’un tag.

Le cache est représenté sous forme de tableau de bloc (t\_bloc), de dimension nombre d’ensemble et associativité.

Enfin, nous avons créé un tableau incr, à une dimension, de même longueur que le cache (nombre d’ensemble) qui nous servira de compteur pour les différents blocs.

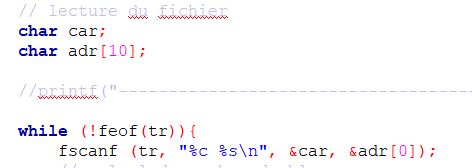
## Initialisation



Lors de l’initialisation, on initialise les valid à 0 (car aucun bloc n’est occupé au départ), on initialise le tag à -1, pour marquer le fait qu’aucun tag n’est encore entré dans les blocs.

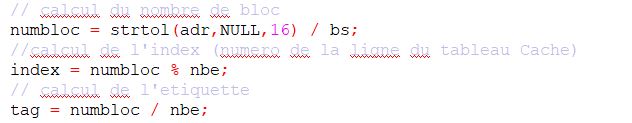
## Traitement

1. Lecture du fichier

******

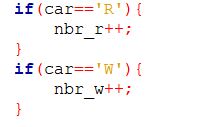
On initialise deux variables : car qui va permettre de lire le premier caractère de chaque ligne (R ou W), et une variable adr qui prendra les caractères restant dans chaque ligne, correspondant aux adresses. La lecture se fait ligne par ligne jusqu’à la fin du fichier entré en paramètre.

1. Calculs

******

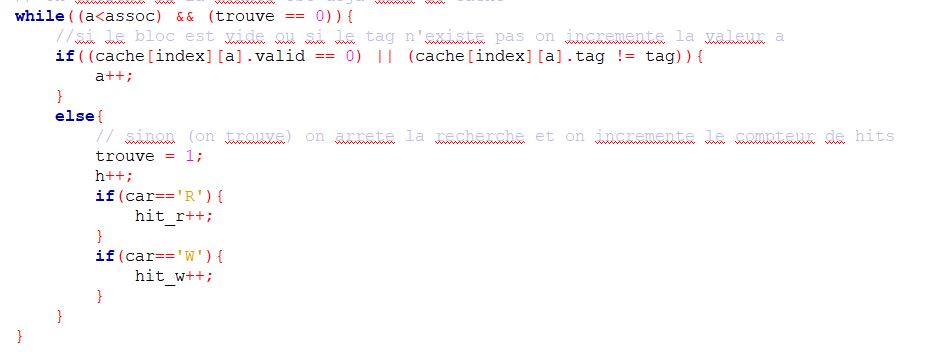
Cette partie permet de réaliser les calculs utiles à la détermination de la ligne à laquelle il faut ajouter l’adresse (index) et au tag à entrer en mémoire (tag).

1. Nombre lectures/écritures

******

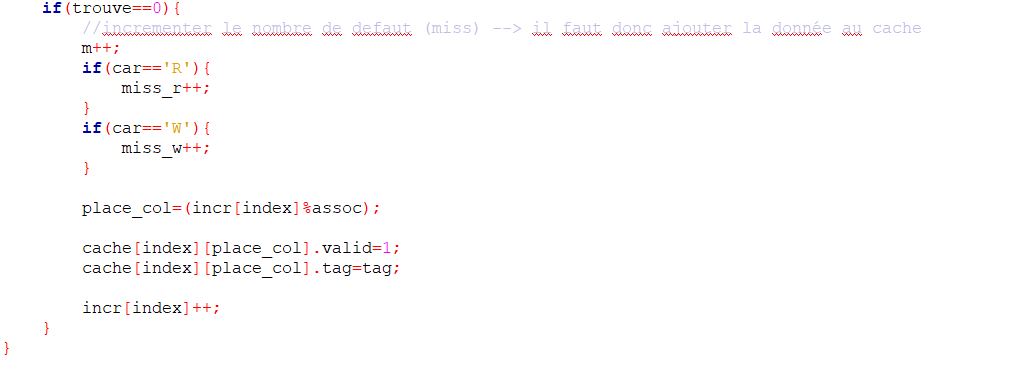
Cette partie nous permet de compter le nombre total de lectures et d’écritures.

1. Hits

******

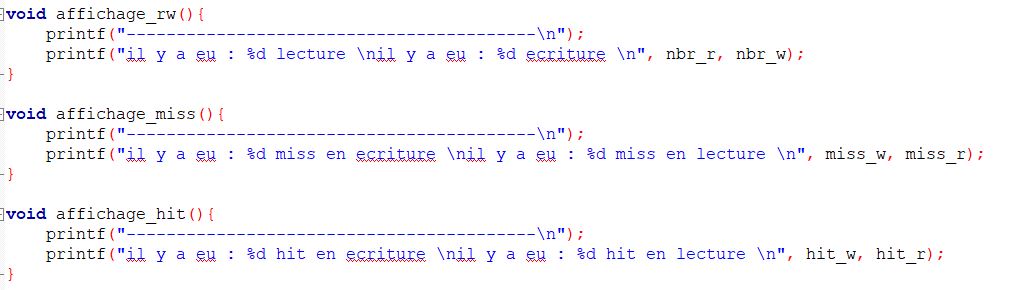
On parcourt la cache à la recherche de l’adresse courante. Si l’adresse est trouvée dans le cache, on la marque comme trouvée, il n’y a donc pas besoin de l’ajouter. On incrémente donc le compteur de hits.

1. Miss

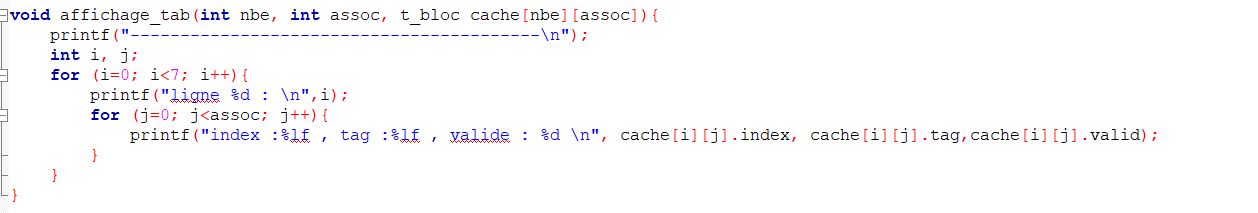
******

Si la donnée n’a pas été trouvée dans le cache, il faut l’ajouter. On incrémente donc le compteur de miss, puis on place l’adresse dans le cache en fonction de son index et des blocs déjà occupés dans la ligne (symbolisé par le tableau incr). Si la ligne est déjà pleine, grâce au tableau incr, le premier bloc rentré sera remplacé par le nouveau bloc.

## Affichage



Cela permet l’affichage des données principales recherchées.



Cet affichage permet de voir comment le tableau se remplit, on n’affiche que les premières lignes, car en cas de fichier volumineux, l’affichage de l’intégralité du tableau serait trop long.

# Résultats Expérimentaux

## mergesort2000Trace

Nombre de lectures : 85240

Nombre d’écritures : 65856

1. Variation du cache size

Bs = 64, assoc = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **cs** | **8192 (nbe=32)** | **4096 (nbe=16)** | **2048 (nbe=8)** | **1024 (nbe=4)** |
| **Défauts en lecture** | | 519 | 959 | 1456 | 2051 |
| **Défaut en écriture** | | 271 | 430 | 641 | 1011 |

1. Variation du bloc size

Cs = 4096, assoc = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **bs** | **64 (nbe=16)** | **32 (nbe=32)** | **16 (nbe=64)** | **8 (nbe=128)** |
| **Défauts en lecture** | | 959 | 1847 | 3593 | 7186 |
| **Défaut en écriture** | | 430 | 813 | 1560 | 3120 |

1. Variation de l’associativité

Cs = 4096, bs = 64

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **assoc** | **16 (nbe=4)** | **8 (nbe=8)** | **4 (nbe=16)** | **2 (nbe=32)** |
| **Défauts en lecture** | | 1057 | 1040 | 959 | 1495 |
| **Défaut en écriture** | | 449 | 428 | 430 | 1166 |

## multTrace

Nombre de lectures : 5020000

Nombre d’écritures : 1020000

1. Variation du cache size

Bs = 64, assoc = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **cs** | **8192 (nbe=32)** | **4096 (nbe=16)** | **2048 (nbe=8)** | **1024 (nbe=4)** |
| **Défauts en lecture** | | 154510 | 1086908 | 1106244 | 1198092 |
| **Défaut en écriture** | | 6042 | 18297 | 35503 | 76912 |

1. Variation du bloc size

Cs = 4096, assoc = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **bs** | **64 (nbe=16)** | **32 (nbe=32)** | **16 (nbe=64)** | **8 (nbe=128)** |
| **Défauts en lecture** | | 1086908 | 258542 | 276069 | 552702 |
| **Défaut en écriture** | | 18297 | 7364 | 2546 | 5049 |

1. Variation de l’associativité

Cs = 4096, bs = 64

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **assoc** | **16 (nbe=4)** | **8 (nbe=8)** | **4 (nbe=16)** | **2 (nbe=32)** |
| **Défauts en lecture** | | 1020000 | 1086781 | 1086908 | 990699 |
| **Défaut en écriture** | | 17330 | 17365 | 18297 | 23799 |

## multTrace10x10

Nombre de lectures : 5200

Nombre d’écriture : 1200

1. Variation du cache size

Bs = 64, assoc = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ***cs*** | **8192 (nbe=32)** | **4096 (nbe=16)** | **2048 (nbe=8)** | **1024 (nbe=4)** |
| **Défauts en lecture** | | 13 | 13 | 13 | 48 |
| **Défaut en écriture** | | 6 | 6 | 6 | 15 |

1. Variation du bloc size

Cs = 4096, assoc = 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **bs** | **64 (nbe=16)** | **32 (nbe=32)** | **16 (nbe=64)** | **8 (nbe=128)** |
| **Défauts en lecture** | | 13 | 26 | 51 | 101 |
| **Défaut en écriture** | | 6 | 12 | 25 | 51 |

1. Variation de l’associativité

Cs = 4096, bs = 64

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **assoc** | **16 (nbe=4)** | **8 (nbe=8)** | **4 (nbe=16)** | **2 (nbe=32)** |
| **Défauts en lecture** | | 13 | 13 | 13 | 13 |
| **Défaut en écriture** | | 6 | 6 | 6 | 6 |