1 – pourquoi la mémoire cache permet d'améliorer la performance de l'ordinateur ?

Accéder plus rapidement à certains informations

* + celles auxquelles on accède le plus souvent
  + celles auxquelles on est susceptible

Efficacité de mémoire cache

Une instruction ou une donnée accédée a plus de chances d'êtres accédée par la suite (utilisation de variables et de boucles) : localité temporelle

* + accès à une variable
  + boucles

Une instruction voisine d'une instruction exécuté a plus de chances d'être exécutée : localité spatiale

Pour que le cache ait un intérêt, il faut que les prénoms appelés soient plus souvent dans le cache que gros répertoire.

* + parcours d'un tableau
  + boucles

Cache direct : très rapide

Je regarde la page C, si ce n'est pas le bon, je regarde dans le gros répertoire.

Si j'appelle un prénom commençant par C, si ce n'est pas le bon, je le mets dans la page C et écrase celui qui y était déjà.

Cache associatif : moins rapide

plusieurs emplacements pour chaque lettre

besoin de regarder plusieurs entrées dans un ensemble

si l'ensemble est plein, enlever un prénom au hasard/enlever le moins récemment utilisé

2 – fonctionnement de l'algorithme LRU

LRU = Last Recently Used

Si toutes les entrées sont pleines, le bloc est chargé à la place de celui le moins récemment utilisé.

Exemple d'algorithme :

chaque entrée possède une valeur LRU correspondant à son ancienneté (entre 1et le nombre d'entrées)

chaque utilisation d'une entrée remet à 1 son LRU et màj les autres entrées

3 – inconvénient d'un cache associatif totale

cache = 1 ligne avec n entrées

un bloc peut aller dans n'importe quelle entrée

pour savoir si un bloc est dans le cache : comparaison des étiquettes de toutes les entrées

Algorithme LRU : gérer le champs LRU de toutes les entrées

4 – pourquoi on stocke l’étiquette plutôt que le numéro de bloc

??????

5 – fonctionnement général de cache

Lecture par le processeur d'une instruction (ou d'une donnée)

Recherche dans le cache

Si succès, le cache renvoie l'instruction sur le bus de données

si échec, accès à la mémoire

* + la mémoire renvoie l'instruction sur le bus de données
  + l'instruction est automatiquement recopiée dans la cache

9 - Comment met-on en œuvre le principe du Write Back

en cas de succès :

écriture simultanée (write-through) dans le cache et en mémoire

écriture différée en (write-back) lors du remplacement du bloc par un autre

en cas d' échec :

écriture allouée : bloc copié dans le cache

écriture non alloué : écriture seulement en mémoire