20/12/2020

Tout d’abord avant de commencer mon code, j’ai essayé de comprendre le fonctionnement du cache. J’ai vu que quand le processeur avait besoin d’une information il allait d’abord voir si cette information était dans le cache, si elle n’y est pas il va alors la chercher dans la mémoire principale.

Ensuite, comme la mémoire du cache est très limitée il faudra sacrifier une case dans le cache. Le choix de la victime se fera de manière optimale de façon à sacrifier la case dont le processeur a utilisé le moins récemment.

J’ai donc décidé de couper mon programme en 3 fonctions :

1. Lecture : Fonction qui permet de lire une valeur à une certaine adresse dans la mémoire mais avant cela il vérifie si cette adresse est dans la cache.
2. Écriture : Fonction qui permet d'écrire une valeur v à une certaine adresse dans la mémoire. Si celle-ci n'a pas été modifiée précédemment . L'on choisit une case à sacrifier dans la cache, on prend la case qui doit être sacrifiée et on stock la valeur de cette case dans la mémoire si elle n'a pas été modifiée précédemment.
3. Main : Fonction qui rappelle les autres fonctions et aussi qui ouvre un certain fichier.

Pour vérifier si mon programme fonctionnait correctement je me suis aidé du fichier donné(code\_expl.txt) et des affichages données dans l’énoncé.

J’ai détaillé mon code avec certains commentaire, voir cacheSimulator.py

En exécutant la commande sur mon terminal :

python3 cacheSimulator.py code\_expl.txt

Voici ce que j’obtiens :

Une image contenant texte

Description générée automatiquement