/*Temps Réel en multi-coeurs: problème de contention mémoire*/

Louisa BESSAD & Roberto MEDINA

02 Mars 2013

Plan détaillé

??? Présenter les sructures de PAPI used ???

Objectifs

1. Mesures:
   (a) Plateforme de mesures avec la librairie PAPI (sur les caches L1/L2, L3 si présent, accès mémoire)
   (b) Clouer les tâches sur les différents coeurs pour ne pas mettre d'autres tâches sur ces coeurs -> priorité maximale
2. Développer une application qui soit faussement Temps-Réel (temps d'exécution fini), paramétrables si besoin (ajout d'un temporisateur), qui accède à tous les caches et à la mémoire
   (a) prévoir le prefetch du processeur
   (b) optimisation du compilateur (accès séquentiel préféré par ex)
3. Développer une appplication attaquante qui soit la plus gourmande en accès mémoire possible
4. Mesurer le temps d'exécution de la TR seule puis quand on ajoute 1 à N attaquants

Avancée

Idée de taches attaquante occupant au maximum le bus et les caches:

- open et close successif :

  - (bloc pris en tête de liste bloc libre et puis libéré et donc mis en queue de cette même liste -> sollicite des zones mémoires différentes et donc des places dans le cache différentes).
  - pas open séquentiels car ramène plusieurs blocs lors du premier open et les autres prendraient ces blocs là plutôt que retourner sur le disque en chercher d'autres.

- *read -> pas accès méoire obg -> NON*
- write -> accès mémoire obg si on close juste après pour éviter l'utilisation du write back si il y a (et donc obliger l'accès disque).
- *déplacement pointeur -> accède uniquement à la table des fichiers ouverts pour modifier un pointeur il n'y a donc pas d'accès disque et donc d'occupation du bus -> NON*
- *dupplication -> n'alloue pas d'autres bloc mais accès mémoire obg pour rajouter un pointeur dans la table des fichiers ouverts, mais n'accède pas au disque(non swapable) -> NON*
- *copie d'un fichier vers autre???? NON car chargement séquentiel des contenus en cache.*
- open, write, dupplication, lseek, close, read sur autre desc puis close.

write >> open en temps d'accès / write ~ dupplication

Implémentations choisies:

- open, write, close en boucle
- open, read jusqu'à fin de fichier puis retour au début tant que le nombre d'itérations désirés n'est pas atteint, close

A effectuer dans cet ordre pour faire les tests.

Questions / A faire:

tester quel type de tâche attaquante est la plus gourmande.