

TP6

Architecture :

- config.h : contient le nombre magique 10
- diskJoin : (.h et .c) : permet de faire le nestedLoop dans chaque dossier
- generate : (.h et .c) : permet de générer les fichiers pour le nestedLoop et pour le hash
- graceHashJoin : (.h et .c) : permet de faire les diskJoin pour le grace Hash Join
- inode : (.h et .c) : permet de gérer les fichiers (savoir combien il y en a dans un répertoire, concatener le nom du fichier en fonction des dossiers, ...)
- inOut : (.h et .c) : permet de gérer l'écriture et la lecture dans les fichiers
- nestedLoop : (.h et .c) : permet de faire un nestedLoop entre deux blocks
- tabGenerator : (.h et .c) : permet de générer les tableaux de 256 shorts avec 32 occurrences identiques
- main.c : permettant de faire les appels des autres fonctions en fonction du tp.

Le découpage permet de réutiliser le même algorithme pour faire le nestedLoop pour le cas normal et pour faire le nestedLoop dans chaque dossier.

Le script compare.sh permet de comparer les deux fichiers obtenus pour avoir leur différence. Pour l'exécution `./compare.sh RS/RSN0.txt RSH/RSH0.txt`.

Nous obtenons les résultats suivants :

Grace Hash : Read : 103 et Write : 1
Nested Loop : Read : 677 et Write : 4

Avec une différence entre les output et un modulo de 10 dans la fonction de hash

Le nombre de lecture n'est pas optimal, et mettant un modulo de 40, le nombre de lecture descend à 20 !!