

*RIVAL Gaëtan
GUYOT Sacha
SAJIDE Idriss*

Compte Rendu Architecture des Systèmes

TP3 – Pagination

Lien Git : <https://github.com/GaetanRival/AS.git>

Dans ce TP, notre principal objectif est de remplacer l'algorithme de remplacement de pages dans le système de Nanvix. Pour cela nous avons dans un premier temps analysé en profondeur les fichiers "mm.c", "paging.c" et "paging.h" afin de comprendre l'ensemble du fonctionnement de la pagination de Nanvix.

Par la suite, nous avons eu l'idée d'un algorithme qui pourrait éventuellement améliorer les performances du remplacement de la page. Nous avons donc travaillé sur la fréquence de consultation des pages en mémoire. Autrement dit, nous avons décidé de favoriser les pages fréquemment accédées, l'idée étant de ne pas les sortir de la mémoire même si elles étaient les plus anciennes afin de ne pas avoir à les recharger tout de suite après. (cf Algo de vieillissement "aging")

Afin de mettre cet algorithme en place, nous avons décidé de fixer un score pour chaque page, que nous avons initialisé à 1000. Plus cette dernière est lue ou écrite, plus son score augmente (dans notre cas le score augmente de 100 pour une lecture, 100 pour une écriture et 200 si elle a été lue ET écrite depuis la dernière actualisation du score). Dans le cas inverse, elle voit son score diminuer suivant une fonction exponentielle (son score est divisé par 1.1).

Ensuite, lorsque qu'une page doit être chargée par un accès mémoire, tout d'abord l'algo vérifie si une page est libre. Dans ce cas, on charge à cet endroit. Sinon il cherche la page avec le plus petit score et la remplace (avoir le plus petit score indique une fréquence d'utilisation moindre ou aucun accès depuis longtemps). Les pages utilisées par plusieurs processus sont bien sûr ignorées. Si plusieurs pages ont le même plus petit score, l'arbitrage se fait selon l'âge des pages ex-aequo : la page la plus ancienne est remplacée.

Avec cet algorithme, nous obtenons les résultats suivants avec les tests mis à notre disposition :

N Taille Matrice	Algo Système Nanvix (ms)	Notre Algo (ms)
100	1 600	1 700
500	217 900	220 700
600	376 300	380 500

Suite à nos résultats, nous avons constaté que les performances de notre algorithme ne répondaient pas à nos attentes. Les résultats restent tout de même proches mais non intéressants pour en conclure que notre algorithme est plus performant que celui de base.

Il y a trois possibilités à cela :

- La première peut provenir de l'implémentation de notre algorithme qui n'est peut-être pas optimale. En particulier, nous aurions aimé pouvoir mettre les scores à jour plus souvent et pas seulement lors d'un remplacement de page.
- La deuxième pourrait provenir de la valeur fixée pour nos paramètres tels que le score de départ et les valeurs d'incrément ou de
- La troisième est tout simplement que notre algorithme n'est pas adapté à ce genre de situation. En effet, nous avons envisagé la possibilité que la multiplication de matrices soit un cas particulier dans lequel notre algorithme offre des performances suboptimales