

## TP noté : Remplacement de pages

### Avant propos

- Ce TP comptera pour plus de 50% de votre note pratique.
- Le TP noté est à réaliser **individuellement**.
- Ce TP est à déposer sur Claroline **au plus tard** à le 7 mai à 8h00 (*deadline dure*).
- Votre rendu devra être de la forme d'une archive de type *tar.gz* et le nom du rendu sera `TPNOTE_NOM.PRENOM.tar.gz`.
- Votre archive devra contenir votre **code source**, un **makefile** pour compiler votre programme, un **script experience.sh**, un **pdf** contenant l'analyse des algorithmes et le fichier **latex** source (le dossier rendu doit être **propre**).
- **Un TP non rendu, vide ou rendu par mail impliquera la défaillance à l'UE..**
- Je vous rappelle que le plagiat est interdit et qu'il est soumis à de **lourdes sanctions disciplinaires** : Ne vous appropriez pas le TP/travail de l'un de vos camarades ou d'une source extérieure ! (Rappel : l'université dispose d'outils pour détecter automatiquement ce genre de fraudes)

Le but de ce TP est de simuler et d'analyser les algorithmes de remplacement de pages étudiés en cours et TD.

## Implémenter les algos classiques de remplacement de pages

La première partie du TP consiste à créer un programme C implémentant<sup>1</sup> les algorithmes étudiés en cours. Ce programme doit prendre en entrée 4 paramètres :

- un paramètre qui indique l'algorithme à simuler (1 : FIFO, 2 : LRU, 3 : horloge, 4 : optimal),
- un paramètre qui indique le nombre de cases dans la RAM,
- un paramètre qui indique le mode d'affichage (debug ou classique),
- le chemin d'accès à un fichier contenant l'ensemble des accès mémoire (**la première ligne contient le nombre d'accès au total, la deuxième ligne le nombre de page, puis un accès par ligne**).

Si le programme est appelé avec l'option -h ou -help, veuillez à ce que le programme affiche comment l'utiliser.

Vous pourrez utiliser les tableaux et listes suivantes :

- Un tableau `acces[]` qui permet de stocker les accès lus dans le fichier. `acces[i]` : numéro de la page accédée à la date `i`. Par exemple, si la liste des accès est 3 4 5 1 le tableau contiendra :  
`acces[0] = 3 , acces[1] = 4 , acces[2] = 5 , acces[3] = 1`
- Un tableau `valide[]` qui contient les bits de validité des pages (*i.e.*, indique si la page est en RAM) :  
si `valide[i] = 1` alors la page `i` est en RAM.
- Pour certains algos (FIFO et horloge) vous aurez sans doute besoin de mettre les numéros des pages RAM dans une liste.
- Un tableau `b_acces[]` qui contient les bits d'accès (pour l'algorithme de l'horloge).  
Rappel : `b_acces[i]` est le bit d'accès de la page `i`, `b_acces[i]` doit être mis à 1 à chaque

---

1. N'hésitez pas à réutiliser les fonctions sur les listes que vous avez probablement eu l'occasion d'utiliser en cours sur les graphes et/ou programmation.

accès à la page.

Écrire un programme qui simule les différents algorithmes de remplacement de pages vus en cours :

1. l'algorithme FIFO ;
2. l'algorithme LRU : pour cet algorithme il faut trouver la page la moins récemment utilisée (pour cela vous pourrez utiliser un tableau `dernier_acces[i]` qui contient la date du dernier accès à la page `i`) ;
3. l'algorithme de l'horloge ;
4. l'algo optimal.

Le programme doit lire la liste des accès aux pages dans le fichier passé en paramètre.

À la fin de l'exécution, le programme doit afficher la proportion de défauts de pages (en mode debug et classique). En mode debug, pour chaque accès, chacun de vos algorithmes doit afficher :

- s'il y a eu un défaut de page ;
- les numéros des pages en ram.

Une manière de tester le bon fonctionnement de vos algorithmes est de vérifier que vous obtenez les mêmes résultats qu'en TD (en mode debug).

## Comparer les algos

La seconde partie de ce TP consiste à effectuer une étude empirique du nombre de défauts de pages de chacun des algorithmes en fonction du nombre de pages et du nombre de cases. L'objectif est de pouvoir analyser dans quelle(s) situation(s), tel ou tel algorithme est le meilleur (en comparaison de l'algorithme optimal bien évidemment).

Pour ce faire, il vous faut

- définir un protocole expérimental (*profitez de la séance de TP restante pour me faire valider votre protocole*), n'oubliez pas de le décrire dans votre rapport ;
- créer un script bash **experience.sh** permettant de mettre en oeuvre le protocole (penser à utiliser gnuplot pour la visualisation de vos résultats) ;
- tirer des conclusions de votre analyse.

un rapport (rédigé en latex) de quelques pages est attendu. Votre rapport contiendra les deux sections suivantes :

1. Protocole expérimental ;
2. Analyse des résultats (pensez à insérer/sélectionner quelques graphiques pour illustrer vos conclusions).