

TP n°2

Règles du jeu :

- on écrit en langage C uniquement (donc on compile avec **gcc**)
- on gère les erreurs avec **perror()** si on peut
- on peut afficher du texte avec **printf()** et on lit au clavier avec **scanf()**
- pour connaître la taille que prend une variable d'un type X, on peut utiliser **sizeof(X)**
ex: **sizeof(int)** ou **sizeof(struct contact)**

attention, si **x** est un pointeur (ex : un tableau), **sizeof(x)** retournera toujours la même valeur (4 ou 8 selon les processeurs et les systèmes)

L'idée de l'exercice est de vous faire manipuler un fichier binaire (non ASCII) : lecture, écriture, déplacement, et d'introduire les notions de verrouillages de fichiers.

Vous devez télécharger le fichier lockfile.c (disponible sur ma page). Celui-ci contient le squelette (main, nom des fonctions et arguments) permettant de gérer un répertoire de noms. Commencez par lire attentivement les commentaires qui précèdent chaque procédure/fonction afin de comprendre leur rôle.

1. Modifiez/complétez les fonctions ajout, affichage et afficheEntree.
2. Modifiez/complétez la fonction modificationEntree. Dans un premier temps, ne vous souciez pas du verrouillage. Implémentez le verrouillage une fois que votre code fonctionne
3. Modifiez les fonctions verrouillage et deverrouillage(). Vous devrez utiliser la fonction de la libc lockf(). Celles ci doivent être utilisées dans les fonctions ajout() et modificationEntree() de manière à garantir qu'il ne pourra pas y avoir d'accès concurrent en écriture sur les parties du fichier concernées. Vous testerez un verrouillage bloquant et un verrouillage non bloquant. Pour ce dernier, si un verrou est en cours, le programme devra afficher un message d'erreur et revenir au menu principal.

Question :

1. Le verrouillage fonctionne-t-il pour des accès en lecture ?
2. Quelle est la différence entre les opérations F_LOCK, F_TLOCK et F_TEST ?

Amélioration :

- faites en sorte de stocker les contacts dans l'ordre des identifiants.