

GIS 3 - T.P. Système

Manipulation de fichiers en C

1 ANSI C

Dans cette section, vous utiliserez les fonctions de la norme ANSI C (`stdio.h`).

1.1 Echauffement

On considérera qu'un mot saisi au clavier a une taille inférieure à `TAILLE_MAX`.

1. Ecrivez une fonction `void saisie(FILE* fic, int n)` permettant de saisir n noms de personnes au clavier et de stocker ces noms, en les séparant par des tabulations (`\t`), dans le fichier `fic`;
2. Testez l'utilisation de cette fonction dans une fonction `main`, puis vérifiez le contenu de ce fichier en l'ouvrant avec votre éditeur de texte préféré;
3. Dans un autre programme, écrivez une fonction `void affichage(FILE* fic, int n)` permettant de lire n noms de personnes à partir du fichier `fic` et de les afficher à l'écran;
4. Testez l'utilisation de cette fonction dans une fonction `main`.

1.2 Mini base de données

On souhaite réaliser une mini base de données permettant de gérer une classe. La base est constituée d'un unique fichier. Chaque ligne du fichier correspond à un élève. Pour chaque élève, on stocke son nom et sa moyenne.

1. Ecrire une fonction permettant d'ajouter les informations d'un élève dans la base (dans le fichier);
2. Ecrire une fonction permettant d'afficher tout le contenu de la base;
3. Ecrire une fonction calculant la moyenne de la classe;
NB : On considère que cette fonction est indépendante des précédentes, dans le sens que vous ne devez pas calculer la moyenne durant la saisie d'un élève.

1.3 fgetc

A l'aide de la fonction `fgetc` (`man fgetc`), réalisez un programme qui compte le nombre d'occurrences d'un caractère donné, dans un fichier.

2 API POSIX

Dans les exercices de cette section, pour les opérations sur les fichiers vous utiliserez les fonctions système POSIX (`open`, `read`, `write`, ...).

N'oubliez pas de tester les valeurs de retour des appels système pour vérifier que tout s'est bien passé!

2.1 Echauffement

Dans un but de simplification, pour cet exercice les tampons de lecture/écriture pourront avoir la taille du texte écrit (votre nom).

1. Ecrivez un programme qui :
 - Crée un fichier `mon_nom.txt` ;
 - Ecrit votre nom dans ce fichier.
2. Vérifiez le contenu de ce fichier en l'ouvrant avec votre éditeur de texte préféré ;
3. Ecrivez un deuxième programme qui :
 - Ouvre le fichier `mon_nom.txt` créé précédemment ;
 - Lit le nom contenu dans le fichier et l'affiche à l'écran.

2.2 Echantillonnage

Ecrivez un programme qui (bien lire l'énoncé jusqu'au bout) :

- Etant donnés deux noms de fichiers passés en argument de votre exécutable ;
- Recopie le contenu du premier dans le second ;
- Avec pour contrainte : les 5 premiers caractères sont recopiés, mais pas les 5 suivants, et ainsi de suite jusqu'à avoir parcouru tout le contenu du (premier) fichier.

3 Bonus

3.1 POSIX

A l'aide de l'API POSIX, programmez votre clone de la commande `tee` : cette commande lit sur l'entrée standard et écrit sur la sortie standard ainsi que dans un fichier passé en paramètre du programme (cf le `man` de `tee`).

NB : Rappelez vous que l'entrée standard et la sortie standard sont accessibles via des descripteurs de fichiers (voir le cours pour le numéro de ces descripteurs).

3.2 ANSI

1. Faites en sorte que votre fonction programmée en section 1.2 calcule maintenant aussi le nombre d'élèves dont la note est supérieure à la moyenne de la classe.

NB : Vous ne devez pas fermer le fichier entre les calculs de ces deux questions (cf `fseek`) ;

2. A l'aide de l'API ANSI, programmez votre propre version de la commande `cp`, en utilisant les fonctions `fgets/fputs`.