

## TP de Système n° 5

Note : ce TP est à réaliser à l'aide des fonctions bas niveau sur les descripteurs de fichiers.

### Exercice 1 : Fusion de fichiers

Supposons que l'on ait deux fichiers, l'un contenant les notes de projet et l'autre les notes d'examen. On souhaite les fusionner en un seul fichier. Chaque ligne des fichiers d'entrée est de la forme `<nom> <note>` et les noms sont triés par ordre alphabétique. Le fichier de sortie aura la forme `<nom> <note1> <note2>`.

Écrivez un programme qui prend en argument trois noms de fichiers, lit les deux premiers dans le format indiqué ci-dessus, et écrit dans le troisième. Vous pourrez tester votre programme sur les fichiers d'exemple `projet.notes` et `examen.notes`.

Si vous avez le temps, modifiez votre programme pour qu'il prenne en plus deux arguments entiers et écrive sur chaque ligne la moyenne de `note1` et `note2` pondérée par ces coefficients.

### Exercice 2 : La commande « grep »

Écrire un programme « `mongrep` » tel que `mongrep motif fichier1 fichier2 ...` lise les fichiers donnés en paramètres et en affiche toutes les lignes contenant la chaîne `motif` sur la sortie standard, précédés du nom de fichier. Par exemple, après la commande :

```
$ mongrep toto42 monfichier.txt
```

si `monfichier.txt` contient une ligne "Cette ligne contient toto42.", alors `mongrep` déclenchera l'affichage suivant : `monfichier.txt: Cette ligne contient toto42.`

**Conseil :** on pourra commencer par programmer une fonction

```
void chercheFichier(char *nomFichier, char *motif)
```

qui cherche le motif dans les lignes de tout un fichier et affiche chaque ligne correspondante.

La fonction `int main(int argc, char **argv)` aura pour rôle d'analyser les paramètres de la ligne de commande et d'appeler la fonction `chercheFichier` autant de fois qu'il le faudra.

Vous pourrez avoir besoin de

- `char *strstr(const char *haystack, const char *needle);` (de `string.h`)
- et de `ssize_t getline(char **lineptr, size_t *n, FILE *stream);` (de `stdio.h`).

### Pour aller plus loin :

- Modifier le programme pour qu'il accepte l'option `-v` qui inverse le résultat de la commande. Par inverser, on entend afficher les lignes qui *ne contiennent pas* le motif (et donc masquer celles qui le contiennent).
- Modifier le programme pour qu'il accepte l'option `-c` qui remplace l'affichage des lignes sélectionnées par l'affichage du nombre de lignes total qui seraient affichées sans `-c`.
- Modifier le programme pour qu'il accepte l'option `-r` pour chercher récursivement dans les fichiers contenus dans l'arborescence des répertoires donnés en paramètre. Vous pourrez utiliser les fonctions de `dirent.h` comme :
  - `DIR *opendir(const char *name);`
  - `int closedir(DIR *dirp);`
  - `struct dirent *readdir(DIR *dirp);`

Vérifiez que toutes les combinaisons d'options fonctionnent.

Ex. : `mongrep -cvr Systeme /home/moi/Documents`