

## TD de Système n° 3

### Exercice 1 : La commande “cat”

Nous nous proposons de réécrire ici la commande “cat”, qui recopie son entrée standard sur sa sortie standard. Cette commande peut éventuellement recevoir un ou plusieurs noms de fichiers en paramètre : ce sont alors ces fichiers (et non plus l’entrée standard) qui sont copiés séquentiellement sur la sortie standard.

Vous pouvez utiliser les fonctions de `stdio` comme :

- `FILE *fopen(const char *path, const char *mode);`
- `size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);`
- `size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);`
- `char *fgets(char *s, int size, FILE *stream);`
- `int fputs(const char *s, FILE *stream);`

1. Écrire une fonction qui prend en argument un flot et le copie sur la sortie standard (en utilisant un tampon).
2. Écrire la fonction `main` correspondante, qui appelle la fonction précédente sur chacun des fichiers listés sur la ligne de commande ou l’entrée standard si aucun argument n’est fourni.
3. Modifier la première fonction pour qu’elle puisse numéroter chaque ligne (c’est-à-dire préfixer chaque ligne de son numéro et d’une espace).
4. Modifier la fonction `main` pour qu’elle analyse les options de la ligne de commande et accepte l’option `-n`.

### Exercice 2 : La commande “tr”

Écrire un programme “montr” tel que `montr fichier1 fichier2 a b` lise un fichier *fichier1* et le recopie dans le fichier *fichier2* en remplaçant tous les caractères *a* par des caractères *b*.

### Exercice 3 : La fonction `fseek`

On s’intéresse maintenant à la fonction `fseek` qui a pour signature `int fseek(FILE *flot, long decalage, int origine);` et qui nous permet de nous positionner dans le fichier en fonction de la variable `origine` pouvant prendre les valeurs `SEEK_SET`, `SEEK_CUR`, `SEEK_END`.

1. Écrire un programme “del100” tel que `del100 fichier1 fichier2` lise un fichier *fichier1* et le recopie sans les 100 premiers caractères dans *fichier2*.
2. Écrire un programme “keep100” tel que `keep100 fichier1 fichier2` lise un fichier *fichier1* et recopie les 100 derniers caractères dans *fichier2*.
3. Écrire un programme “fusion” tel que `fusion fichier1 fichier2` lise un fichier *fichier1* et le recopie à la fin du fichier *fichier2*.

### Exercice 4 : La commande “cp” à l’envers

Écrire un programme permettant de copier le contenu d’un fichier dans un autre en inversant l’ordre des caractères. Vous aurez besoin des fonctions `size_t fread(void *tampon, size_t taille, size_t nombre, FILE *flot)` et `size_t fwrite(const void *tampon, size_t taille, size_t nombre, FILE *flot)` qui permettent de travailler sur un flot binaire.