

# TD8 - Manipulation de fichiers

Langage C (LC4)

Semaine du 22 mars 2010

## 1 Découpage d'un fichier mot par mot

Vous pourrez utiliser `int fscanf(FILE *stream, const char *format, ...)` avec le format `%s`, qui récupère une chaîne de caractères jusqu'à un espace ou un saut de ligne, et avec l'option `a` qui alloue la mémoire nécessaire pour stocker la chaîne trouvée. La valeur renvoyée est le nombre de motifs trouvés.

**Question 1.** Écrivez une fonction `char *prochain_mot(FILE *f)` qui renvoie la chaîne de caractères contenue entre la position courante dans `f` et le premier espace ou saut de ligne. Renvoyez `NULL` à la fin du fichier.

**Question 2.** Écrivez une fonction `void decoupe_et_reconstruit(FILE *f1, FILE *f2)` qui lit le fichier texte de `f1` et l'écrit dans `f2` en mettant un mot par ligne.

**Question 3.** Écrivez une fonction `void decoupe_et_affiche(FILE *f1)` qui lit le fichier texte de `f1` et l'écrit sur la sortie standard en mettant un mot par ligne.

**Question 4.** Écrivez une fonction `int main(int argc, char *argv[])` qui appelle la fonction `decoupe_et_reconstruit` sur deux fichiers dont les noms sont passés en argument au programme.

## 2 Découpage d'un fichier mot par mot et stockage dans un tableau

**Question 5.** Écrivez une fonction `char **decoupe_en_tableau(FILE *f)` qui lit le fichier texte passé en argument et renvoie un tableau de chaînes de caractères dont chaque case contient un mot du texte. La dernière case du tableau contiendra `NULL`.

**Question 6.** Écrivez une fonction `void reconstruit_depuis_tableau(char **tab, FILE *f)` qui écrit dans `f` les mots contenus dans `tab` séparés par des tabulations. On suppose que le dernier élément de `tab` est `NULL`.

## 3 Déplacement dans un fichier

Les fonctions utilisées jusqu'ici parcourent un fichier séquentiellement du début à la fin, en déplaçant implicitement un curseur dans le fichier. On peut jouer avec ce curseur grâce à la fonction `int fseek(FILE *f, long decalage, int origine)`. La valeur renvoyée

est 0 en cas de succès, autre chose sinon. Cette fonction permet de se positionner en `origine + decalage`. Le paramètre `decalage` est exprimé en octets et `origine` peut prendre les 3 valeurs suivantes :

`SEEK_SET` : début du fichier ;

`SEEK_CUR` : position courante ;

`SEEK_END` : fin du fichier ;

Pour connaître la position courante à partir du début du fichier, on utilise `long ftell(FILE *f)`.

**Question 7.** Écrivez une fonction `void moities(FILE *f1, FILE *f2, FILE *f3)` qui écrit la première moitié de `f1` dans `f2` et la seconde moitié de `f1` dans `f3`.

**Question 8.** Écrivez une fonction `int main(int argc, char *argv[])` qui appelle la fonction `moities` sur trois fichiers dont les noms sont passés en argument au programme.

**Question 9.** Écrivez une fonction `void renverse(FILE *f1, FILE *f2)` qui lit le texte contenu dans `f1` et l'affiche à l'envers caractère par caractère sur `f2`.

On définit un type liste de caractères ainsi :

--