

TD 7 : Fichiers

Programmation en C (LC4)

Semaine du 10 mars 2008

Exercice 1 Écrivez les fonctions suivantes :

1. `char *prochain_mot (FILE *f)` ; qui renvoie la chaîne de caractères contenue entre l'offset courant et le premier caractère qui n'est ni une lettre ni un trait d'union.
2. `void decoupe_fichier (FILE *f1, FILE *f2)` ; qui lit le fichier texte passé en argument et l'écrit dans `f2` en mettant un mot par ligne.
3. `void decoupe_stdout (FILE *f)` ; qui lit le fichier texte passé en argument et affiche le texte sur la sortie standard en mettant un mot par ligne.
4. `char **decoupe_tab (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 pointera vers `NULL`.

Exercice 2 Écrivez les fonctions suivantes :

1. `void renverse (FILE *f)` ; qui lit le texte contenu dans `f` et l'affiche à l'envers caractère par caractère sur la sortie standard.
2. `FILE *intervertit (FILE *f)` ; qui renvoie un nouveau fichier où les lignes de `f` auront été interverties deux à deux.

Exercice 3 Écrivez les fonctions suivantes :

1. Déclarez un type `mot` qui corresponde à une structure contenant un champs `chaine` de type `char *`, un champs `occurrence` de type `int`, et un champs `int *positions` contenant la liste des positions de chacune des occurrences du mot.
2. Déclarez un type `liste_mots` correspondant à une structure contenant un tableau de mots, la taille de ce tableau et le nombre de mots que contient effectivement le tableau. En effet, on ne suppose pas que le tableau soit toujours entièrement rempli.
3. Vous coderez une fonction `ajout_mot (char *m, int pos, liste_mots *l)` qui ajoute le mot `m` de position `pos` à la liste de mots `l`. Si `l` contient déjà le mot, il suffira d'ajouter une occurrence de plus avec la nouvelle position, sinon il faudra ajouter un nouveau mot à la liste.
4. `liste_mots decoupe_mot (FILE *f)` ; qui lit le fichier texte passé en argument et renvoie un tableau de mots contenant chaque mot avec le nombre d'occurrences de ce mot et la liste de leur position.
5. `void decoupe_fichier (char **tab, int t)` ; qui affiche sur la sortie standard, le texte constitué des chaînes de caractères contenues dans `tab` de taille `t`.
6. `void reconstruit_fichier (liste_mots l, FILE *f)` ; qui écrit dans `f` le texte constitué des mots contenus dans `l` remis dans le bon ordre.