

Projet Langage C

Projet d'IN301 - Franck Quessette

Projet fait par Ines Lebib et Nicolas Fond-Massany

Canevas

/* A remplir */

Du .txt au tableau

Le but est de connaître mettre des caractères dans un tableau de caractères et de savoir à quelle est le premier caractère de chaque ligne.

On va créer 2 tableaux: `T[]` de type *char* qui contiendra tous les caractères du fichier et `L[]` de type *int* qui contiendra la position du premier caractère de chaque ligne de façon à savoir que, par exemple, c'est le caractère 243 qui commence la ligne 4. On notera alors `L[4] = 243`.

Pour ce faire on va utiliser la librairie *stdio.h* qui va nous permettre de prendre le fichier pris en argument dans la fonction *main* de *principal.c* pour ensuite l'ouvrir en mode écriture et se servir de `fgetc(F)` pour mettre dans notre tableau de caractère `T` tous les caractères du fichier `F`. Dans un même temps on mettra dans le tableau d'entier `L` le caractère de début de chaque ligne. Si on a un retour chariot, c'est qu'on est sur une nouvelle ligne, en partant de ce principe on peut mettre un simple `if` mais attention, il ne faut pas mettre `fgetc(F)` car cette fonction fait avancer la lecture des caractères de un à chaque appel. On mettra donc simplement `if (T[i] == '\n')`. On vérifie aussi si le fichier est bien présent dans le dossier où on exécute le programme en vérifiant `if (!F)` (traduit si le fichier `F` n'existe pas) pour éviter les erreurs et stoppé le programme si jamais c'est le cas avec `exit(2)`.

On déclarera les variable `T` et `L` comme des pointeurs car on aura besoin de mémoire pour chaque éléments des deux tableaux. on aura donc deux déclarations de variables qui pointe vers de la mémoire. `char *T = malloc(taille_fic*(sizeof(char)));` et `int *L = malloc(nbr_lignes*(sizeof(int)));`. On rajoute juste après un `if` pour vérifier si l'allocation a bien fonctionné avec un message d'erreur et un `exit`.

On veux aussi connaître le nombre de caractère et le nombre de ligne afin d'allouer la mémoire comme il faut. On travail ici sur un fichier texte simple, donc le nombre d'octets est strictement égale au nombre de caractères. On peut alors utiliser la librairie *stat.h* qui nous permet de savoir la taille du fichier avec la fonction *stat*. Pour le nombre de ligne, on fait une boucle qui va parcourir le fichier de caractère en caractère et qui, dès qu'il voit un `'\n'`, va faire +1 à notre variable qui compte le nombre de ligne `nbr_lignes`. Il ne faut pas oublié qu'à chaque appel de `fgetc`, on avance d'un caractère, comme on utilisera cette fonction après, il faut créer une autre variable temporaire `FILE tmp` qui va nous permettre d'utiliser le fichier *text* sans gêner l'autre variable `FILE F`.

Pour le moment on affiche `L` et `T` avec des `printf` pour voir si notre programme fonctionne, on les enlèvera par la suite.

```
void text_to_tab (char *text) {
    FILE *F = fopen(text, "r");           // Ouverture du fichier texte
    FILE *tmp = fopen(text, "r");         // en mode "read"
```

```

if (!F || !tmp) { // Si le fichier n'existe pas
    printf("\nFichier inexistant \n\n");
    exit(2); // On arrête le programme si le
} // fichier n'existe pas

struct stat buf;
stat(text, &buf);
int nbr_char = buf.st_size; // On met dans nbr_char (qu'on
//aurait put appeler taille_fic)

int nbr_lignes = 1; // le nombre de caractères du texte
int k = 0;
while (k<=nbr_char) { // On compte ici les lignes pour
    if (fgetc(tmp) == '\n') nbr_lignes++; // avoir un malloc qui aloue le
    k++; // bon nombre de mémoire
}

fclose(tmp);

char *T = malloc(nbr_char*(sizeof(char))); // On aloue ici la mémoire pour
int *L = malloc(nbr_lignes*(sizeof(int))); // pouvoir ensuite la pointer

int i = 0;
int y = 1;

while (i<=taille_fic) {
    T[i] = fgetc(F); // On prend le premier caractère
    printf("T[%d] = %c\n", i, T[i]); // On vérifie si T[i] à la bonne valeur
    if (T[i] == '\n') { // Si il y a un retour à la ligne
        L[y] = i+1; // c'est que le prochain caractère
        printf("L[%d] = %d\n", y, L[y]); // sera le premier caractère de la ligne
        y++; // On vérifie que L[y] à la bonne valeur
    }
    i++;
}

fclose(F);
}

```