

Conversion Requêtes

Introduction :

Ce code est un programme écrit en langage C qui traite un fichier de requêtes, qui recherche des mots spécifiques dans d'autres fichiers, et qui génère un fichier de commandes en fonction des mots trouvés.

Nous allons maintenant voir en détail ce que réalise ce code.

1. Fonction `estPresentDansFichier`

Cette fonction prend deux arguments en entrée : un mot, qui se trouve être le verbe, le nom ou la couleur de la requête saisie et le nom d'un fichier. Elle ouvre le fichier en mode lecture ("r") et parcourt chaque ligne du fichier à la recherche du mot donné. Si le mot est trouvé, la fonction retourne 1, sinon elle retourne 0. La fonction utilise `fgets` pour lire chaque ligne et `strcmp` pour comparer le mot avec la ligne lue. Le fichier est fermé à la fin de la recherche.

2. Fonction `effacer_contenu_fichier`

Cette fonction prend en entrée le nom d'un fichier, l'ouvre en mode écriture ("w") pour effacer son contenu, puis le ferme immédiatement pour le sauvegarder. Elle est utilisée pour vider le contenu du fichier "commande.txt" au début de chaque exécution du programme.

3. Fonction `main`

La fonction principale du programme commence par appeler `effacer_contenu_fichier` pour vider le fichier "commande.txt". Ensuite, elle ouvre le fichier "requete_valide.txt" en mode lecture ("r") pour lire les mots de la requête.

Elle utilise ensuite une boucle `while` avec `fscanf` pour lire chaque mot de la requête. Pour chaque mot, elle ouvre le fichier "commande.txt" en mode ajout ("a") et vérifie si le mot est présent dans deux fichiers de synonymes ("synonyme_avance.txt" et "synonyme_tourne.txt") en utilisant la fonction `estPresentDansFichier`.

En fonction des résultats de ces vérifications, elle écrit différentes commandes dans le fichier "commande.txt". Si le mot est présent dans "synonyme_avance.txt", elle écrit `"Go_forward()"`. Si le mot est présent dans "synonyme_tourne.txt", elle écrit `"Turn()"`. Elle ajoute toujours un saut de ligne après chaque commande afin de rendre le fichier "commande.txt" plus lisible.

Le programme affiche ensuite un message indiquant que les commandes ont été enregistrées et se termine.

4. Remarques générale

Tout d'abord, les tailles des tableaux sont fixées à des valeurs spécifiques ici, cette valeur est égale à 100, ce qui devrait être largement suffisant pour l'utilisation de ce code. Cependant, si les lignes ou les mots peuvent dépasser ces tailles, il serait prudent d'ajuster ces valeurs ou d'implémenter une gestion dynamique de la mémoire.

Ensuite, les fichiers ne sont pas refermés explicitement en cas d'erreur lors de l'ouverture. Il serait utile d'ajouter des instructions `fclose` dans ces cas.

De plus, les messages d'erreur sont affichés avec `perror`, fournissant des informations détaillées sur l'erreur.

Enfin, le programme assure que les fichiers de requête et de synonymes sont correctement formatés et disponibles pour une nouvelle utilisation.

Conclusion :

Ainsi, ce programme offre une fonctionnalité de traitement de requêtes simple, utilisant des synonymes pour générer des commandes spécifiques. Il peut être étendu ou adapté en fonction des besoins spécifiques de l'application, que ce soit par un modérateur ou un simple utilisateur.