Amine Bourennane
Univeristé Gustave Eiffel
Thomas Berthereau

Rapport TP6

Amine Bourennane
Univeristé Gustave Eiffel
Thomas Berthereau

But :

Créer un utilitaire opérant sur des fichiers textes en manipulant des chaînes de caractères des fichiers textes, l'entrée et la sortie standard, les arguments d'un programme, les structures de données dynamiques, modularisation d'un projet relativement abstrait et l'écriture des tests .

Compilation :

Pour compiler il suffit de lancer dans le répertoire du projet la commande 'make' dans le terminal les règles 'clean' et 'mrproper' sont présent dans le **Makefile** ainsi que les règles 'install' et 'uninstall' qui met l'exécutable dans un répertoire binaire ou le supprime.

On exécute le programme avec . /clm nom_fichier -option où option peut être soit -a qui fait tri lexicographique de la sortie, -n qui fait un tri décroissant des occurrences, -e N qui affiche des groupes de mots par groupe de N mot, -s MOT qui n'affiche que les mots succédant à MOT, -p MOT qui n'affiche que les mots précédent MOT et -test qui affiche les tests des fonctions.

Il y a des combinaisons d'option tel que -a, -n, -e peuvent se lancer ensemble ainsi que -s ou -p. Le programme autorise de lancer cette appelle : ./clem nom_fichier -a -n -e N -p MOT -s, il affichera tous les groupes de N mots succédant à MOT trier par ordre lexicographique puis par ordre décroissant des occurrences.

Description du programme :

Le programme possède un nombre de 6 modules :

- un module *Parseur* qui s'occupe des arguments du programme

- un module *Affichage* qui affichera tous ce qu'il y aura à afficher sur la sortie standard ou d'erreur

- un module *Ecriture* qui crée un fichier nom_fichier.clm et qui écrit la sortie dedans

- un module *Traitement* qui va lire dans le fichier mot à mot et qui va appeler les fonctions de création de la liste de mots

- un module *ListeMot* qui va stocker toutes les infos en rapport avec les mots contenus dans le fichier

- un module *Test* qui va tester les fonctions importantes du programme.

Conclusion :

On peut conclure que les tests sont primordial lors de la conception d’un programme car ils nous permettent d’avoir une idée de ce qu’il se passe dans notre code, ainsi que de voir si lors d’une implémentation nous n’avons rien cassé. Le tp nous a beaucoup fait travailler sur la gestion des erreurs, les allocations mémoires et la gestion des différents cas du programme.