Rapport : projet LDP parti 2.

Talhaoui Ahmed, Matricule : 568936

April 21, 2023

1 Lancement du code :

Pour lancer le programme, il vous suffit de lancer la commande « make » sur un terminal linux, le lancement avec les fichiers « utils.cpp » et « utils.hpp » est impératif au bon fonctionnement du programme, car ces fichiers ont été modifié et certaines fonctions on été ajouté dans ce fichiers afin de ne pas nuire à la lisibilité du programme et ainsi facilité les future modifications potentiel du code.

Ainsi vous génèrerai, deux fichiers "encodeur" et "decodeur" que vous pourrez exécuter avec la commande suivante :

./« nomDuFichier » .

veuillez également a mettre les fichiers .txt dans le bonne ordre a l'execution du programme :

- pour l'encodeur : message.txt arbre.txt messageCoder.txt
- pour le decodeur: massageCoder.txt arbre.txt message.txt

2 Nouvelles fonctionalité :

La plus grosse diffrence est la fonction "huffman" ajouter dans le fichier "ulbzip.cpp", sont intitialisation des variables utile a la fonction huffman a été ajouté dans la fonction main() du fichier parmis elles compte :

-le tableau parent : initialiser à avec un charactere
-le tableau enfant gauche : initialiser à - 1
-le tableau enfant droite : initialiser à - 1
-le tableau de fréquence de mots : initialiser à 0
-le tableau de symbole

basiquement en premier lieu nous initialisons un tabeau contenant aucun symbole en doublons (grace à la fonction isIn), ensuite ce tableau est passer à une fonction avec le message original afin de compter le nombre de symbole. finalement tout cela est passer a la fonction "huffman" qui s'ocupe d'ajouter des élément (en choisissant les 2 minimum du tableau avec un parent = à -1 afin de ne pas recupérer de noeux non-orphelins) et de modifier les lien de parenté entre noeuds.

parlons maintenant de la gestions du stockage dans la variable "messageEncode", plus bas dans la fonction "huffman", le code le code binaire sous forme ASCCI composé de '0' et '1' est decouper sur 8 "bits" (ASCCI) et convertis. C'est 8 "bits" sont ensuite conevrtis en int puis stocker sous forme de unsigned char dynamiquement.

le fichier "Ulbunzip" a également été modifier afin d'allouer la variable "message" dynamiquement et convertir les unsigned char en binaire.

ce code est assez different et plus complexe que le code python de manièere générale.

3 Variable par référence et copie :

le language de programmation python lui est exclusivement par réference, c'est le fait d'avoir accés a la case mémoire contenant la valeur voulue, le fait d'assigner une variable par copie est tout simplement le fait dde copier une valeur dans une autres variable. c++ plus fonctionne par copie et éggalement par pointeur.

4 efficacité du code:

mon code me parait moyennement efficace, plusieur chose tel que la supressions de certains pointeurs aurait pu etre rajouter. la taille de certaine variable aurait également pu etre raccourcit .