COMPTE RENDU APNÉE 3-4 ALGO

✔ Nous avons pu terminé huff_encode.c et huff_decode.c :

Ainsi nous les avons testé(on utilisant un fichier de type shell pour automatiser les testes) avec des fichiers de taille différente notamment avec:

test1:fichier vide,
test2:fichier avec un seul caractere,
test3:un fichier assez court (avec des accents et des caractères spéciaux),
test4:un fichier de taille plus grande,
et aussi avec **les fichiers fournis** (candide.txt et gargantua.txt).

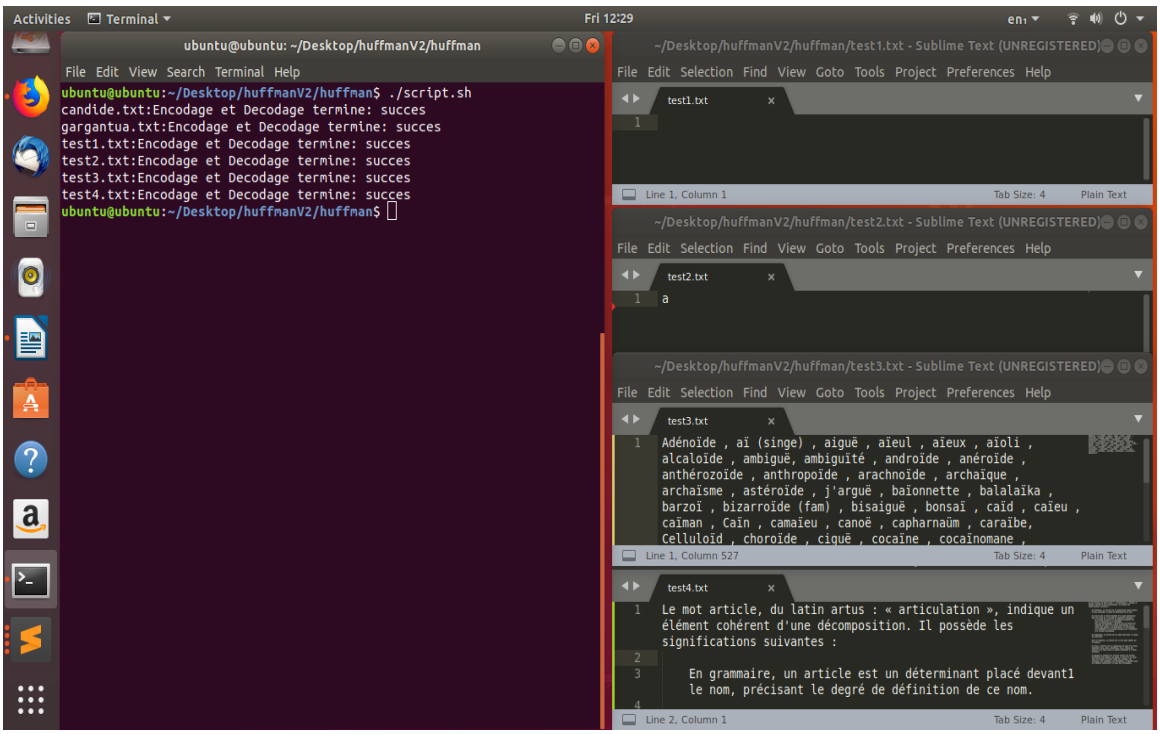
Script de test

./script : pour lancer le script.

```
#/bin/sh
fileout="out.txt"
for file in *.txt
do
        make clean > tmp1.txt
        make > tmp1.txt
        ./huff_encode $file $fileout  > tmp1.txt
        ./huff_decode $fileout > tmp.txt
        diff tmp.txt $file -i > tmp1.txt
        NB_LIGNES=$(wc -l tmp1.txt | cut -f1 -d' ')
        if [ $NB_LIGNES -eq 0 ]
        then
                echo $file:"Encodage et Decodage termine: succes"
        else
                echo $file:"Encodage et Decodage termine: echec"
        fi
        rm tmp.txt
        rm tmp1.txt
        rm out.txt
done
```

✔ Tests

Trace du script



Nous avons aussi pris en compte les temps d'exécution pour l'encodage et le décodage de chaque fichier:

Fichiers	**Temps d'exécution encodage**	**Temps d'exécution décodage**
test1.txt (fichier vide)	0.000129 seconds	0.000069 seconds
test2.txt (1 caractère)	0.000153 seconds	0.000075 seconds
test3.txt (accents + caractère spéciaux)	0.000248 seconds	0.000188 seconds
test4.txt (grande taille)	0.000312 seconds	0.000428 seconds
candide.txt	0.022288 seconds	0.029566 seconds
gargantua.txt	0.022701 seconds	0.041183 seconds

✔ Les modules fournis ont été utilisés comme suit:

- fap.c et fap.h

L'initialisation et la création de l'arbre de Huffman : On crée les feuilles (Chaque feuille est ajoutée dans la FAP avec comme priorité l'occurrence des caractères). On extrait les éléments de la FAP pour créer les nœuds constituant les sous-arbres, puis on insère ces sous-arbres dans la FAP. On refait les mêmes opérations jusqu'il nous reste qu' un seul élément dans la FAP qui est le racine et l'arbre de Huffman.

- arbrebin.c et arbrebin.h

La création de l'arbre de Huffman.
Pour le parcours de l'arbre de Huffman soit pour créer le tableau de code (codeHuffman) ou la partie du décodage.
Pour lire ou écrire l'arbre de huffman dans un fichier.

- bfile.c et bfile.h

La lecture bit a bit d'un fichiers encodé (étape de décodage).
L'écriture bit a bit dans un fichier (étape de l'encodage).

✔ Conclusion

Grace aux tests effectués, il est possible de dire que la correction et l'efficacité du programme ont été vérifiés.
On a utilisé:
Deux fichiers test1 et test2 pour s'assurer que notre programme ne bug pas.
Un fichier test3 pour s'assurer qu' il accepte tout type de fichier texte.
Un fichier assez court pour tester le bon fonctionnement de notre programme.
Un fichier de taille plus grande, pour tester la rapidité de notre programme et s'il y a des problème de type (ex: erreur de segmentation).