



Manuel utilisateur - Projet PIM

Codage de Huffman

François Lauriol, Yael Gras

15 janvier 2022

Table des matières

1	Introduction	2
2	Fichiers test	2
3	Compresser un fichier .txt au format .hff.	3
3.1	Créer l'exécutable de Compression	3
3.2	Lancer la compression	3
3.3	Profiter de l'option bavard	4
3.4	Différents cas d'erreurs	4
4	Décompresser un fichier .hff	5
4.1	Créer l'exécutable de Décompression	5
4.2	Lancer la décompression	6
4.3	Profiter de l'option bavard	6
4.4	Différents cas d'erreurs	7

1 Introduction

Ce manuel utilisateur facilitera toute volonté que vous aurez d'utiliser nos programmes de compression et de décompression utilisant le principe du Codage de Huffman.

2 Fichiers test

Pour vous montrer l'usage de nos algorithmes, nous vous proposons deux fichiers initiaux textes (figures 1 et 2) que nous allons tout d'abord compresser puis décompresser (en supprimant bien évidemment le fichier initial avant de décompresser sa version compressé pour vous montrer le bon fonctionnement de nos algorithmes).

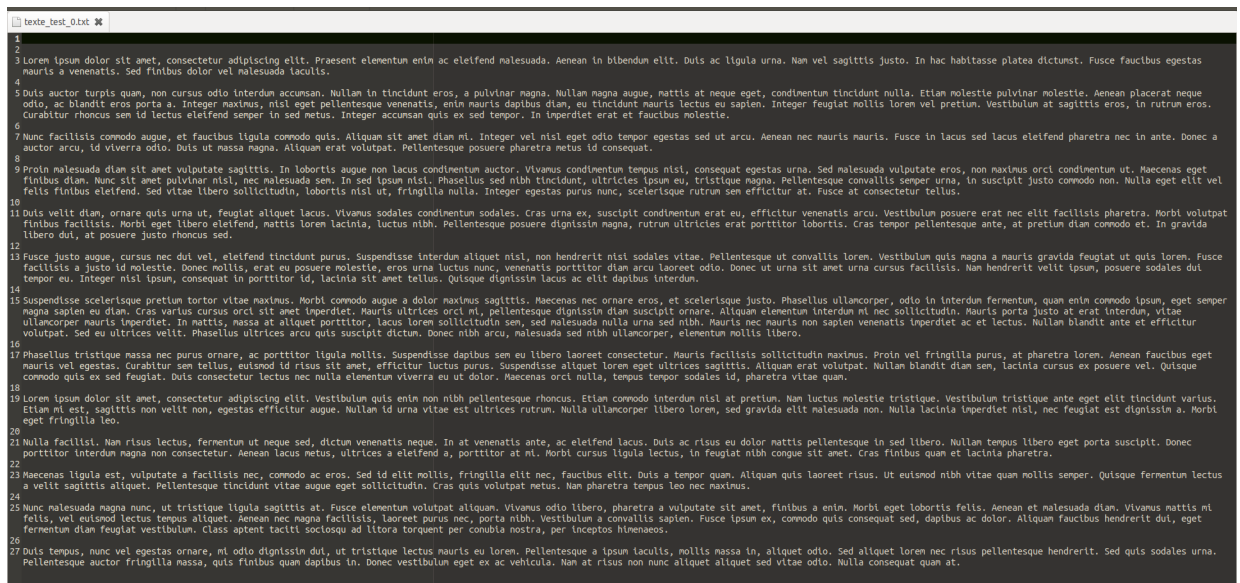


FIGURE 1 – Fichier exemple 1 (lorem ipsum de 1000 caractères)

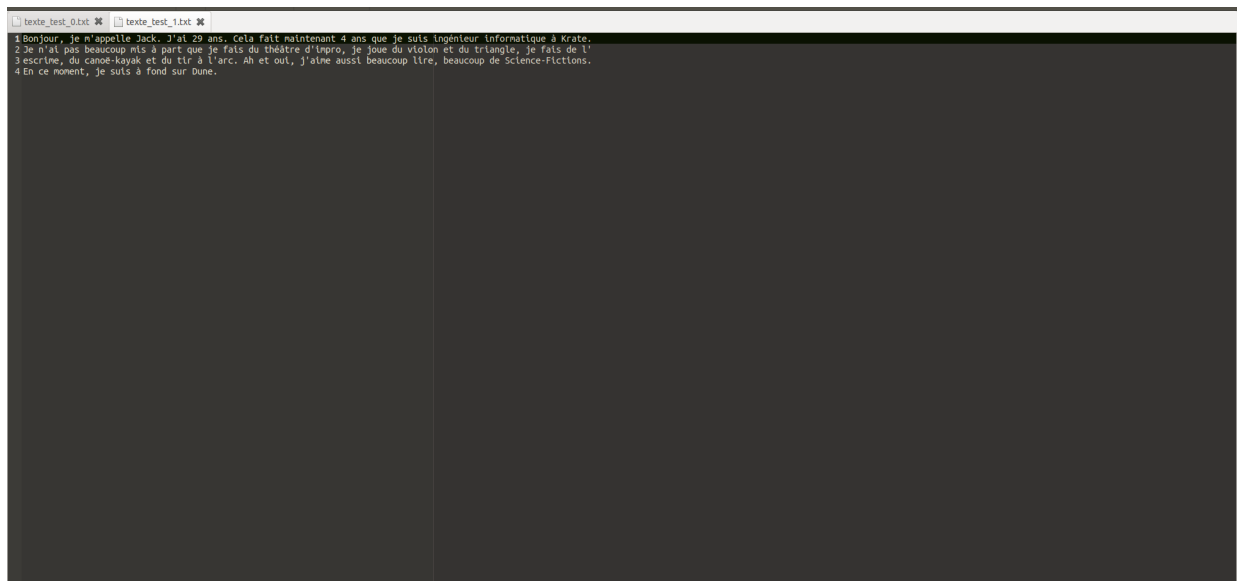


FIGURE 2 – Fichier exemple 2 (texte lambda)

Nous débuterons dans le cas de la figure 3 pour vous montrer comment compresser un fichier .txt à partir du fichier compression.adb (et le module arbre_de_Huffman) et décompresser un fichier .hff à partir du fichier decompression.adb (et le module arbre_de_Huffman).

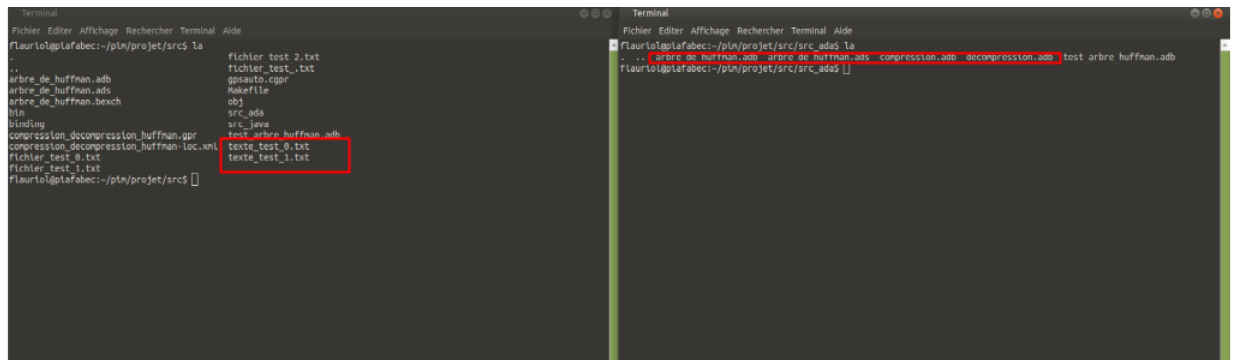


FIGURE 3 – Situation initiale

3 Compresser un fichier .txt au format .hff

3.1 Créer l'exécutable de Compression

Pour utiliser le programme compression, il faut créer son exécutable. Pour créer l'exécutable, on lance l'instruction avec gnatmake comme la figure 4.



FIGURE 4 – Situation avant de créer l'exécutable

On arrive alors au cas de la figure 5. On a bien l'exécutable compression.



FIGURE 5 – Situation après avoir créé l'exécutable

3.2 Lancer la compression

On revient alors au dossier dans lequel sont stockés nos fichiers et on lance la commande `./cheminpermettantd'accéderacompression/compression` suivi des noms des fichiers texte que l'on souhaite compresser (*texte_test_0.txt* par exemple). Comme on le montre sur les figures 6 et 7.

```

Flauriol@hydre:~/pim/projet/src$ la
..
arbre_de_huffman.adb      fichier_test_1.txt
arbre_de_huffman.ads      fichier_test_2.txt
arbre_de_huffman.bexch    fichier_test_.txt
bin                        makefile
binding                   obj
compression_decompression_huffman.gpr src_ada
compression_decompression_huffman-loc.xml src_java
fichier_test_0.txt        test_arbre_huffman.adb
fichier_test_1.txt        texte_test_0.txt
Flauriol@hydre:~/pim/projet/src$ ./src_ada/compression texte_test_0.txt
Flauriol@hydre:~/pim/projet/src$ la
..
arbre_de_huffman.adb      fichier_test_0.txt  fichier_test_.txt  src_ada      texte_test_0.txt
arbre_de_huffman.ads      fichier_test_1.txt  makefile          src_java     texte_test_0.txt.hff
arbre_de_huffman.bexch    compression_decompression_huffman.gpr  fichier_test_2.txt  obj          test_arbre_huffman.adb  texte_test_1.txt
bin                        compression_decompression_huffman-loc.xml
binding
compression_decompression_huffman.gpr
compression_decompression_huffman-loc.xml
fichier_test_0.txt
fichier_test_1.txt
Flauriol@hydre:~/pim/projet/src$

```

FIGURE 6 – Situation après avoir lancé la compression sur 1 fichier

```

Flauriol@hydre:~/pim/projet/src$ la
..
arbre_de_huffman.adb      fichier_test_1.txt
arbre_de_huffman.ads      fichier_test_2.txt
arbre_de_huffman.bexch    fichier_test_.txt
bin                        makefile
binding                   obj
compression_decompression_huffman.gpr src_ada
compression_decompression_huffman-loc.xml src_java
fichier_test_0.txt        test_arbre_huffman.adb
fichier_test_1.txt        texte_test_0.txt
Flauriol@hydre:~/pim/projet/src$ ./src_ada/compression texte_test_0.txt texte_test_1.txt
Flauriol@hydre:~/pim/projet/src$ la
..
arbre_de_huffman.adb      fichier_test_2.txt
arbre_de_huffman.ads      fichier_test_.txt
arbre_de_huffman.bexch    makefile
bin                        obj
binding                   src_ada
compression_decompression_huffman.gpr src_java
compression_decompression_huffman-loc.xml test_arbre_huffman.adb
fichier_test_0.txt        texte_test_0.txt
fichier_test_1.txt        texte_test_0.txt.hff
fichier_test_1.txt        texte_test_1.txt
fichier_test_1.txt        texte_test_1.txt.hff
Flauriol@hydre:~/pim/projet/src$

```

FIGURE 7 – Situation après avoir lancé la compression sur 2 fichiers

On remarque bien la création des fichiers .hff correspondant qui sont les versions compressées à l'aide de l'algorithme de Huffman des fichiers .txt .

3.3 Profiter de l'option bavard

Curieux de voir comment nos programmes considèrent et codifient les caractères que vous voulez compresser ? Profitez de l'option bavard !

Pour cela rien de plus simple, ajoutez la commande *-b* ou *--bavard* avant les noms des fichiers txt que vous souhaitez compresser pour faire afficher leurs arbre et table de Huffman correspondant ainsi que le rapport de compression. Par exemple au lieu d'écrire « *./src_ada/compression texte_test_0.txt texte_test_1.txt* » comme dans la figure 7, on écrira « *./src_ada/compression -b texte_test_0.txt texte_test_1.txt* » ou « *./src_ada/compression --bavard texte_test_0.txt texte_test_1.txt* ».

3.4 Différents cas d'erreurs

3.4.1 Pas de fichier écrit

```

Flauriol@neon:~/pim/projet/src$ ./src_ada/compression
Aucun fichier n'a été écrit dans la commande

Usage : ./src_ada/compression Option Nom_Fichier

Option (facultative) :
    -b ou --bavard : Affichage de l'Arbre de Huffman et de la table associée, ainsi que le rapport de compression

Nom_Fichier : Nom des fichiers à compresser en .txt

Flauriol@neon:~/pim/projet/src$

```

FIGURE 8 – Situation où on n'écrit pas d'argument après le lancement de l'exécutable

Si, comme nous, vous obtenez ce genre de message (figure 8), cela signifie que vous n’avez écrit aucun fichier à compresser. Et le message vous indique alors comment bien écrire la ligne de commande dans votre terminal.

3.4.2 Fichier non existant

```
flauriol@neon:~/pim/projet/src$ ./src_ada/compression aaa.txt
Le fichier aaa.txt n'existe pas
flauriol@neon:~/pim/projet/src$
```

FIGURE 9 – Situation où le fichier n’existe pas

Si, comme nous, vous obtenez ce genre de message (figure 9), cela signifie que le fichier .txt que vous avez demandé à compresser n’est pas répertorié là où vous le demandez ou qu’il n’existe pas.

3.4.3 Fichier non conforme

```
flauriol@neon:~/pim/projet/src/src_ada$ ./compression aaa
aaa ne correspond ni a une option ni a un fichier dans le bon format pour cette commande
flauriol@neon:~/pim/projet/src/src_ada$
```

FIGURE 10 – Situation où le fichier n’est pas conforme

Si, comme nous, vous obtenez ce genre de message (figure 10), cela signifie que le fichier que vous avez demandé à compresser n’est pas au format .txt. Compression est un outil de compression qui n’opère que sur des fichiers au format .txt .

4 Décompresser un fichier .hff

4.1 Créer l’exécutable de Décompression

Pour utiliser le programme decompression, il faut créer son exécutable. Pour créer l’exécutable, on lance l’instruction avec gnatmake comme la figure 11.

```
flauriol@neon:~/pim/projet/src/src_ada$ ls
- arbre_de_huffman.adb  arbre_de_huffman.ads  compression.adb  decompression.adb  test_arbre_huffman.adb
flauriol@neon:~/pim/projet/src/src_ada$ gnatmake decompression.adb
```

FIGURE 11 – Situation avant de créer l’exécutable

On arrive alors au cas de la figure 12. On a bien l’exécutable décompression.

```
flaurto@neon:~/p1n/projet/src/src_ada$ la
.. arbre_de_huffman.adb arbre_de_huffman.ads compression.adb decompression.adb test_arbre_huffman.adb
flaurto@neon:~/p1n/projet/src/src_ada$ gnatmake decompression.adb
gcc -c decompression.adb
gcc -c arbre_de_huffman.adb
gnatbind -x decompression.all
gnatlink decompression.all
flaurto@neon:~/p1n/projet/src/src_ada$ la
.. arbre_de_huffman.adb arbre_de_huffman.ads arbre_de_huffman.all arbre_de_huffman.o compression.adb decompression decompression.adb decompression.all decompression.o test_arbre_huffman.adb
flaurto@neon:~/p1n/projet/src/src_ada$
```

FIGURE 12 – Situation après avoir créé l'exécutable

4.2 Lancer la décompression

On revient alors au dossier dans lequel sont stockés nos fichiers et on lance la commande `./cheminpermettantd'accéderadecompression/decompression` suivi des noms des fichiers texte que l'on souhaite décompresser (`texte_test_0.txt.hff` par exemple). Comme on le montre sur les figures 13 et 14.

```
flaurto@neon:~/p1n/projet/src$ la
.. arbre_de_huffman.adb binding compression_decompression_huffman.gpr fichier_test_0.txt fichier_test_1.txt src_ada 'texte_test_0 (copie).txt' texte_test_1.txt.hff
.. arbre_de_huffman.adb arbre_de_huffman.bexch compression_decompression_huffman.gpr fichier_test_1.txt Makefile src_java 'texte_test_0 (copie).txt' texte_test_1.txt.hff
.. arbre_de_huffman.adb bin compression_decompression_huffman.gpr fichier_test_2.txt Makefile test_arbre_huffman.adb 'texte_test_1 (copie).txt'
flaurto@neon:~/p1n/projet/src$ ./src_ada/decompression texte_test_0.txt.hff
.. arbre_de_huffman.adb binding fichier_test_0.txt fichier_test_1.txt src_ada 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
.. arbre_de_huffman.adb arbre_de_huffman.bexch compression_decompression_huffman.gpr fichier_test_1.txt Makefile src_java 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
.. arbre_de_huffman.adb bin compression_decompression_huffman.gpr fichier_test_2.txt Makefile test_arbre_huffman.adb 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
flaurto@neon:~/p1n/projet/src$
```

FIGURE 13 – Situation après avoir lancé la décompression sur 1 fichier

```
flaurto@neon:~/p1n/projet/src$ la
.. arbre_de_huffman.adb binding compression_decompression_huffman.gpr fichier_test_0.txt fichier_test_1.txt src_ada 'texte_test_0 (copie).txt' texte_test_1.txt.hff
.. arbre_de_huffman.adb arbre_de_huffman.bexch compression_decompression_huffman.gpr fichier_test_1.txt Makefile src_java 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
.. arbre_de_huffman.adb bin compression_decompression_huffman.gpr fichier_test_2.txt Makefile test_arbre_huffman.adb 'texte_test_1 (copie).txt'
flaurto@neon:~/p1n/projet/src$ ./src_ada/decompression texte_test_0.txt.hff texte_test_1.txt.hff
.. arbre_de_huffman.adb binding fichier_test_0.txt fichier_test_1.txt src_ada 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
.. arbre_de_huffman.adb arbre_de_huffman.bexch compression_decompression_huffman.gpr fichier_test_1.txt Makefile src_java 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
.. arbre_de_huffman.adb bin compression_decompression_huffman.gpr fichier_test_2.txt Makefile test_arbre_huffman.adb 'texte_test_0 (copie).txt' 'texte_test_1 (copie).txt'
flaurto@neon:~/p1n/projet/src$
```

FIGURE 14 – Situation après avoir lancé la décompression sur 2 fichiers

On remarque bien la création des fichiers `.txt` correspondant qui sont les versions décompressées à l'aide de l'algorithme de Huffman des fichiers `.hff`.

4.3 Profiter de l'option bavard

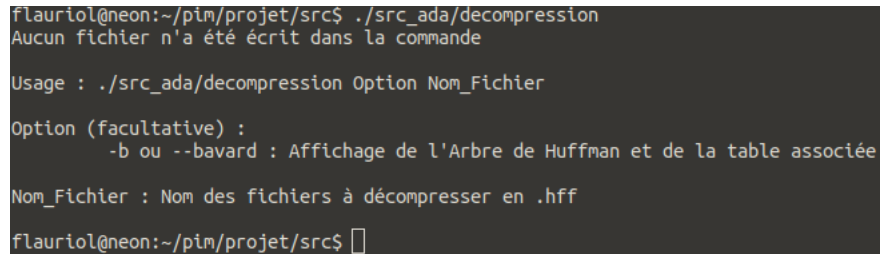
Curieux de voir comment nos programmes considèrent et codifient les caractères que vous voulez décompresser ? Profitez de l'option bavard !

Pour cela rien de plus simple, ajoutez la commande `-b` ou `--bavard` avant les noms des fichiers `txt` que vous souhaitez décompresser pour faire afficher leurs arbre et table de Huffman correspondant. Par exemple au lieu d'écrire « `./src_ada/decompression texte_test_0.txt.hff texte_test_1.txt.hff` » comme dans la figure 14, on écrira « `./src_ada/decompression -b`

texte_test_0.txt.hff *texte_test_1.txt.hff* » ou « *./src_ada/decompression --bavard*
texte_test_0.txt.hff *texte_test_1.txt.hff* ».

4.4 Différents cas d'erreurs

4.4.1 Pas de fichier écrit



```
flauriol@neon:~/pim/projet/src$ ./src_ada/decompression
Aucun fichier n'a été écrit dans la commande

Usage : ./src_ada/decompression Option Nom_Fichier

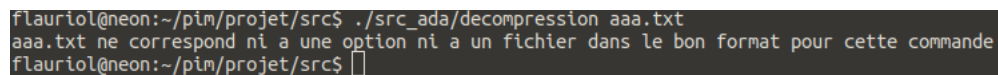
Option (facultative) :
    -b ou --bavard : Affichage de l'Arbre de Huffman et de la table associée

Nom_Fichier : Nom des fichiers à décompresser en .hff
flauriol@neon:~/pim/projet/src$
```

FIGURE 15 – Situation où on n'écrit pas d'argument après le lancement de l'exécutable

Si, comme nous, vous obtenez ce genre de message (figure 15), cela signifie que vous n'avez écrit aucun fichier à décompresser. Et le message vous indique alors comment bien écrire la ligne de commande dans votre terminal.

4.4.2 Fichier non existant

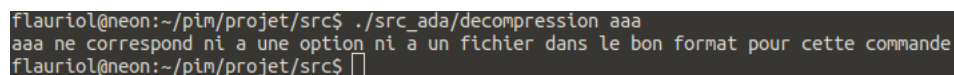


```
flauriol@neon:~/pim/projet/src$ ./src_ada/decompression aaa.txt
aaa.txt ne correspond ni a une option ni a un fichier dans le bon format pour cette commande
flauriol@neon:~/pim/projet/src$
```

FIGURE 16 – Situation où le fichier n'existe pas

Si, comme nous, vous obtenez ce genre de message (figure 16), cela signifie que le fichier .txt que vous avez demandé à décompresser n'est pas répertorié là où vous le demandez ou qu'il n'existe pas.

4.4.3 Fichier non conforme



```
flauriol@neon:~/pim/projet/src$ ./src_ada/decompression aaa
aaa ne correspond ni a une option ni a un fichier dans le bon format pour cette commande
flauriol@neon:~/pim/projet/src$
```

FIGURE 17 – Situation où le fichier n'est pas conforme

Si, comme nous, vous obtenez ce genre de message (figure 17), cela signifie que le fichier que vous avez demandé à décompresser n'est pas au format .hff. Decompression est un outil de décompression qui n'opère que sur des fichiers au format .hff .