Projet Huffman

0.1

Généré par Doxygen 1.8.11

Table des matières

Chapitre 1

A faire

- corrigé docs
- netoyer l'archive
- les listings (fichiers sources) documentés (doxygen) PAPIER
  - doxygen
  - cd/latex
  - make
- Archive nommée par les noms du groupe
- Envoyer à `meynard@lirmm.fr` et `pompidor@lirmm.fr`

- Quel est le nombre maximum de caractères (char) différents ?
  - Le nombre maximum de caractères est 256.
- Comment représenter l'arbre de Huffman ? Si l'arbre est implémenté avec des tableaux (fg, fd, parent), quels sont les indices des feuilles ? Quelle est la taille maximale de l'arbre (nombre de noeuds) ?
  - L'arbre de huffman est représenté par une structure possédant les varriables `pere`,`fg`, `fd` et `frequences`.
  - Si l'arbre est implémenté avec des tableaux (fg, fd, parent), les indices des feuilles corresponde au code du carractère.
  - L'arbre peut avoir au maximum 256 Noeuds.
- Comment les caractères présents sont-ils codés dans l'arbre ?
  - il sont codé par leurs code ASCii.
- Le préfixe du fichier compressé doit-il nécessairement contenir l'arbre ou les codes des caractères ou bien les deux (critère d'efficacité) ?
  - Le préfixe du fichier compressé doit contenir soit l'arbre soit les codes des caractères.
  - Stocker l'arbre est plus efficace en terme de taille de stokage ainsi qu'en efficacité lors de la décomprétion.
- Quelle est la taille minimale de ce préfixe (expliquer chaque champ et sa longueur) ?
  - Le préfixe est composé de 3 octet magiques (Pas obligatoire) permétant d'identifié pouvant être décompréssé. 1 octet représentant le nombre de bits utile du dernier octet puis l'arbre.
- Si le dernier caractère écrit ne finit pas sur une frontière d'octet, comment le compléter ? Comment ne pas prendre les bits de complétion pour des bits de données ?
  - on le complete avc des 0 inutile.
  - Lors de la décomprétion on vérifie si on est sur le dernier octet, si c'est le cas on ne traite que les bits utile. le nombre de bits utile a été récupéré dans l'entête.
- Le décompresseur doit-il reconstituer l'arbre ? Comment ?
  - oui, a partir de l'entête. Dans notre cas l'arbre est reconstruit dans un tableau de `Noeuds`.

Chapitre 2

Compresseur de Huffman

Dans le cadre de l'UE Systèmes d'exploitation (HLIN303) nous avons réalisé un compresseur sans perte utilisant l'algorithme de Huffman.

État d'avancement du projet

Nous avons terminé le compresseur ainsi que le décompresseur. Nous n'avons pas pu réaliser l'archiveur Python demandé dans le sujet par manque de temps.

Si nous avions eu plus de temps, nous aurions stocké l'arbre bit par bit au lieu de octet par octet.

Cas testés

Nous avons testé avec succès notre compresseur sur les fichiers suivants :

- un fichier vide
- un fichier ne comportant qu'un seul type de caractère
- un fichier comportant tous les caractères
- fichiers textes sans particularités
- autres types de fichiers : images, vidéos, ...

Bug résiduels

Tous les bugs rencontrés ont été corrigés. Après de nombreux tests, nous n'en avons pas trouvé de nouveaux.

Chapitre 3

Index des structures de données

3.1 Structures de données

Liste des structures de données avec une brève description :

Chapitre 4

Index des fichiers

4.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

Chapitre 5

Documentation des structures de données

5.1 Référence de la structure noeud

Noeud d'un arbre binaire.

`#include <defNoeud.h>`

Champs de données

- unsigned int pere
- unsigned int fg
- unsigned int fd
- double frequences

5.1.1 Description détaillée

Noeud d'un arbre binaire.

Noeud est une petite structure comportant les indices de son noeud père et de ses noeuds fils. Lui est également associé la fréquence d'apparation dans un fichier du caractère associé à ce noeud.

5.1.2 Documentation des champs

5.1.2.1 unsigned int noeud : :fd

Indices du fils droit

5.1.2.2 unsigned int noeud : :fg

Indices du fils gauche

5.1.2.3 double noeud : :frequences

Fréquence d'apparation du caractère lié au noeud

5.1.2.4 unsigned int noeud : :pere

Indices du noeud père

La documentation de cette structure a été générée à partir du fichier suivant :

- compresseur/include/defNoeud.h

Chapitre 6

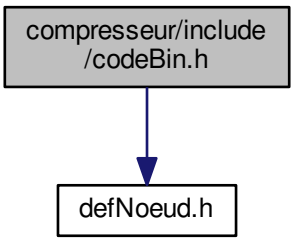
Documentation des fichiers

6.1 Référence du fichier compresseur/include/codeBin.h

Déclaration de la fonction qui va générer les codes binaires.

```
#include "defNoeud.h"
```

Graphe des dépendances par inclusion de codeBin.h :



Fonctions

— char ** CodeBin (Noeud *arbre, int racine)
Génère les codes binaires.

6.1.1 Description détaillée

Déclaration de la fonction qui va générer les codes binaires.

6.1.2 Documentation des fonctions

6.1.2.1 char** CodeBin (Noeud * *arbre,* int *racine*)

Génère les codes binaires.

Ces codes binaires sont associés à chaque caractère présent dans le fichier d'origine.

Paramètres

arbre	Arbre
racine	Indice de la racine de l'Arbre

Renvoie

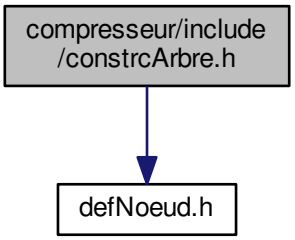
Index Tableau associant les codes binaires aux différents caractères

6.2 Référence du fichier compresseur/include/constrcArbre.h

Déclaration de la fonction qui va construire l'arbre.

```
#include "defNoeud.h"
```

Graphe des dépendances par inclusion de constrcArbre.h :



Fonctions

— int ConstrcArbre (double ∗tab_frequence, Noeud ∗arbre)
Construction de l'arbre.

6.2.1 Description détaillée

Déclaration de la fonction qui va construire l'arbre.

6.2.2 Documentation des fonctions

6.2.2.1 int ConstrcArbre (double ∗ *tab_frequence,* Noeud ∗ *arbre*)

Construction de l'arbre.

Paramètres

tab_frequence	Tableau des fréquences de répartition des caractères
arbre	Arbre préalablement initialisé

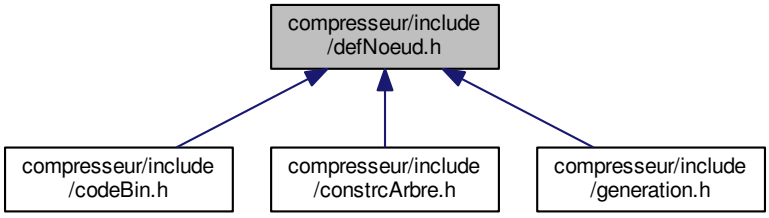
Renvoie

Ind_racine Indicide de la racine de l'arbre

6.3 Référence du fichier compresseur/include/defNoeud.h

Définit la structure Noeud.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Structures de données

— struct noeud
Noeud d'un arbre binaire.

Définitions de type

— typedef struct noeud Noeud
Noeud d'un arbre binaire.

6.3.1 Description détaillée

Définit la structure Noeud.

6.3.2 Documentation des définitions de type

6.3.2.1 typedef struct noeud Noeud

Noeud d'un arbre binaire.

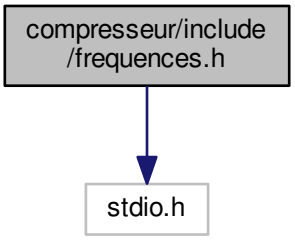
Noeud est une petite structure comportant les indices de son noeud père et de ses noeuds fils. Lui est également associé la fréquence d'apparation dans un fichier du caractère associé à ce noeud.

6.4 Référence du fichier compresseur/include/frequences.h

Définit la fonction de génération des fréquences.

```
#include <stdio.h>
```

Graphe des dépendances par inclusion de frequences.h :



Fonctions

— double ∗ CalculFrequencesCaractere (FILE ∗fichier)
Fonction de calcul de la répartition des fréquences dans un fichier.

6.4.1 Description détaillée

Définit la fonction de génération des fréquences.

6.4.2 Documentation des fonctions

6.4.2.1 double∗ CalculFrequencesCaractere (FILE ∗ *fichier*)

Fonction de calcul de la répartition des fréquences dans un fichier.

Paramètres

fichier	Fichier où les fréquences des caractères doivent être calculées

Renvoie

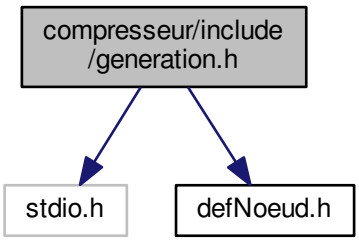
tab_frequence Un tableau contenant les fréquences des caractères contenus dans le fichier. L'indice d'une case correspond au code ASCII du caractère correspondant

6.5 Référence du fichier compresseur/include/generation.h

Déclaration de la fonction de génération du fichier compressé

```
#include <stdio.h>
#include "defNoeud.h"
```

Graphe des dépendances par inclusion de generation.h :



Fonctions

— void Generation (char ∗∗index, FILE ∗entre, FILE ∗sortie, int racine, Noeud ∗arbre)
Ecrit dans le fichier "compressé" les codes binaires des caractères.

6.5.1 Description détaillée

Déclaration de la fonction de génération du fichier compressé

6.5.2 Documentation des fonctions

6.5.2.1 void Generation (char ∗∗ *index,* FILE ∗ *entre,* FILE ∗ *sortie,* int *racine,* Noeud ∗ *arbre*)

Ecrit dans le fichier "compressé" les codes binaires des caractères.

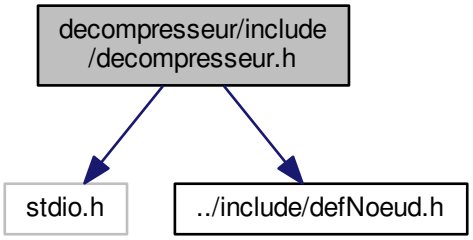
Paramètres

index	Tableau de codes binaires associés aux différents caractères
entre	Fichier d'entrée
sorti	Fichier de sortie (fichier compressé)
racine	Indice de la racine de l'arbre
arbre	Arbre

6.6 Référence du fichier decompresseur/include/decompresseur.h

```
#include <stdio.h>
#include "../include/defNoeud.h"
```

Graphe des dépendances par inclusion de decompresseur.h :



Macros

— #define ERR_FORMAT 1
Définit ERR_FORMAT = 1.

Fonctions

— int Decompression (FILE ∗Entree, FILE ∗Sortie)
Fonction permettant de décompresser un fichier.

6.6.1 Description détaillée

Déclaration de la fonction de décompression

6.6.2 Documentation des fonctions

6.6.2.1 int Decompression (FILE ∗ *Entree*, FILE ∗ *Sortie*)

Fonction permettant de décompresser un fichier.

Paramètres

Entree	Fichier compressé d'entrée
Sortie	Fichier de sortie décompressé

Renvoie

Retourne ERR_FORMAT si ce n'est pas un fichier qui peut être décompressé, retourne 0 sinon

Index