Algorithme de Compression LZW

Implémentation de l'algorithme Lempel-Ziv-Welch en C

Vue d'ensemble

Système de compression/décompression de fichiers développé en C, utilisant l'algorithme LZW (Lempel-Ziv-Welch), une méthode de compression sans perte particulièrement efficace pour les textes avec des motifs répétitifs.

Architecture du Projet

Programmes Principaux

1. **Compresseur** (`./compress`)
    a. Programme de compression
    b. Construction du dictionnaire
    c. Encodage des séquences
2. **Décompresseur** (`./decompress`)
    a. Programme de décompression
    b. Reconstruction du dictionnaire
    c. Décodage des séquences

Algorithme LZW

Principe de Fonctionnement

- **Compression adaptative**
    - Construction dynamique du dictionnaire
    - Apprentissage des motifs récurrents
    - Optimisation progressive
- **Dictionnaire**
    - Initialisation avec caractères ASCII (0-255)
    - Extension dynamique avec nouvelles séquences
    - Gestion efficace de la table de correspondance

Étapes de Compression

1. **Initialisation**
    a. Création du dictionnaire de base
    b. Allocation des structures de données
2. **Traitement**
    a. Lecture séquentielle du fichier
    b. Recherche des plus longues séquences
    c. Ajout de nouvelles entrées au dictionnaire
3. **Encodage**
    a. Conversion en codes de longueur variable
    b. Optimisation du stockage
    c. Écriture du fichier compressé

Caractéristiques Techniques

Performance

- **Taux de compression**
    - Texte : 40-60%
    - Code source : 50-70%
    - Données binaires : 30-50%
- **Vitesse de traitement**
    - Compression : ~10MB/s
    - Décompression : ~15MB/s

Limites

- Taille maximale du dictionnaire : 2^16 entrées
- Mémoire utilisée : ~64MB
- Taille maximale de fichier : 2GB

Points Forts

- Compression sans perte
- Adaptatif aux motifs
- Performance optimisée
- Robustesse
- Validation extensive