

Script de démonstration

Projet de Compression Huffman

EL MAZROUA Ahmed BENALI Abdellah

2024-2025

1 Préparation de la démonstration

1.1 Compilation des programmes

```
# Compilation du programme de compression
gnatmake compresseur.adb

# Compilation du programme de décompression
gnatmake decompresser.adb
```

1.2 Préparation des fichiers de test

```
# Création d'un fichier de test simple
echo "exemple_de_texte:
exemple_tempe_lexeme" > test1.txt

# Création d'un fichier avec répétitions
echo "aaabbbcccc
dddeeeeeeee" > test2.txt
```

2 Démonstration (4 minutes)

2.1 Étape 1 : Compression simple (1 minute)

```
# Démonstration de la compression avec mode bavard
./compresseur -b test1.txt

# Vérification de la création du fichier compressé
ls -l test1.txt test1.txt.hff

# Comparaison des tailles
echo "Taille originale:"
wc -c test1.txt
echo "Taille compressée:"
wc -c test1.txt.hff
```

Points à noter :

- Affichage de l'arbre de Huffman
- Table de codage générée
- Taux de compression obtenu

2.2 Étape 2 : Analyse du fichier compressé (1 minute)

```
# Affichage hexadecimal du fichier compressé
xxd test1.txt.hff | head -n 5

# Explication de la structure :
# - Entête avec les symboles
# - Structure de l'arbre
# - Données compressées
```

Points à souligner :

- Format du fichier compressé
- Position des métadonnées
- Organisation des données

2.3 Étape 3 : Décompression (1 minute)

```
# Décompression avec mode binaire
./decompresser -b test1.txt.hff

# Vérification de l'intégrité
diff test1.txt test1.txt.hff.d

# Si la différence est nulle, la décompression est réussie
echo $?
```

Observations :

- Reconstruction de l'arbre
- Processus de décodage
- Vérification de l'intégrité

2.4 Étape 4 : Cas particuliers (1 minute)

2.4.1 Test avec un fichier binaire

```
# Création d'un fichier binaire de test
dd if=/dev/urandom of=test_bin.dat bs=1024 count=1

# Compression et décompression
./compresser -s test_bin.dat
./decompresser -s test_bin.dat.hff

# Vérification
diff test_bin.dat test_bin.dat.hff.d
```

2.4.2 Gestion des erreurs

```
# Test avec un fichier inexistant
./compresser fichier_inexistant.txt

# Test avec un fichier non .hff
./decompresser test1.txt
```

2.4.3 Mode silencieux

```
# Compression en mode silencieux
./compresser -s test2.txt

# D compression en mode silencieux
./decompresser -s test2.txt.hff
```

3 Points clés à mettre en valeur

- Robustesse des programmes
- Performance de la compression
- Gestion des erreurs
- Options de verbosité
- Intégrité des données