

Rapport détaillé du programme :

.1 Conception du programme

Le programme simule un système de gestion de fichiers à l'aide de structures en C, représentant les blocs de disque, les fichiers, les enregistrements, et leurs métadonnées. La conception est basée sur une hiérarchie bien définie et des fonctionnalités permettant la création, la manipulation et la gestion des données.

1.1 Objectifs principaux

- Créer un simulateur de gestion de fichiers en mémoire.
- Fournir des fonctionnalités pour gérer des enregistrements dans des blocs simulés.
- Implémenter des modes d'organisation (globale et interne) pour les fichiers.

1.2 Structures et relations

Structures principales

- Record : Représente les enregistrements individuels.
 - Attributs :
 - record_ID : Identifiant unique de l'enregistrement.
 - content : Contenu textuel.
- Block : Simule un bloc mémoire contenant un enregistrement.
 - Attributs :
 - is_empty : Indique si le bloc est libre.
 - file_name : Nom du fichier associé au bloc.
 - next_block : Pointeur pour le chaînage entre blocs.
 - record_data : Données de l'enregistrement.
- File : Représente un fichier logique.
 - Attributs :
 - name : Nom du fichier.
 - total_records : Nombre total d'enregistrements.
 - blocks : Tableau de pointeurs vers les blocs associés.
 - global_organ : Mode d'organisation global (contigu ou chaîné).
 - internal_organ : Mode interne (trié ou non trié).

- MetaInfo : Contient les métadonnées des fichiers.
- Attributs :
 - meta_file_name : Nom du fichier.
 - block_count : Nombre de blocs associés au fichier.
 - record_count : Nombre d'enregistrements dans le fichier.
 - first_block_addr : Adresse du premier bloc.
 - global_mode : Mode d'organisation global.
 - internal_mode : Mode interne.

Relations

- Chaque **fichier** peut contenir plusieurs **blocs**.
- Chaque **bloc** contient un **enregistrement**.
- Les métadonnées sont générées pour résumer les informations sur un fichier.

2. Fonctionnalités

Le programme offre une gamme de fonctionnalités interactives pour manipuler les fichiers et les enregistrements.

2.1 Création de fichiers et insertion d'enregistrements

- **Conception :**
- Un fichier est créé avec des options de mode global (contigu ou chaîné) et interne (trié ou non trié).
- Les enregistrements sont insérés dans les blocs disponibles, associés au fichier.
- **Résultats attendus :**
 - Le fichier est enregistré avec ses attributs et ses blocs sont initialisés.
 - Les enregistrements insérés sont immédiatement associés aux blocs et apparaissent lors de l'affichage du fichier.

2.2 Recherche et modification d'enregistrements

- **Conception :**
 - La recherche est basée sur l'identifiant unique de l'enregistrement (ID).
 - La modification remplace le contenu de l'enregistrement après une recherche réussie.
- **Résultats attendus :**
 - Les recherches réussies affichent l'index du bloc contenant l'enregistrement.
 - Les modifications sont visibles lors de l'affichage du fichier.

2.3 Suppression et tri des enregistrements

- **Conception :**
 - La suppression marque le bloc contenant l'enregistrement comme vide.
 - Le tri organise les enregistrements en fonction de leurs IDs (uniquement si le fichier est en mode trié).
- **Résultats attendus :**
 - Les enregistrements supprimés n'apparaissent plus lors de l'affichage.
 - Le tri réorganise les enregistrements dans l'ordre croissant de leurs IDs.

2.4 Défragmentation et compactage

- **Conception :**
 - La défragmentation réorganise les blocs d'un fichier en éliminant les espaces vides entre les blocs.
 - Le compactage regroupe tous les blocs utilisés dans la mémoire pour libérer les espaces inutilisés.
- **Résultats attendus :**
 - Les fichiers sont continus en mémoire après défragmentation.
 - La mémoire est compacte et les blocs vides sont regroupés.

2.5 Métadonnées et visualisation

- **Conception :**
 - Les métadonnées résument les informations essentielles sur un fichier (taille, adresse du premier bloc, modes d'organisation).
 - L'état du disque affiche si chaque bloc est occupé ou vide.
- **Résultats attendus :**
 - Les métadonnées reflètent précisément l'état actuel du fichier.
 - L'état du disque montre une vue d'ensemble claire des blocs.

3. Résultats observés

3.1 Cas de test : Gestion de fichiers

Entrées

- Création d'un fichier nommé test_file avec 3 enregistrements.
- Insertion d'enregistrements :
 - ID : 1, Contenu : "Record1".
 - ID : 2, Contenu : "Record2".
 - ID : 3, Contenu : "Record3".

Résultats

- L'affichage du fichier montre les trois enregistrements associés aux blocs.
- Les métadonnées indiquent un total de 3 enregistrements et le chaînage correct.

3.2 Cas de test : Suppression et tri

Entrées

- Suppression de l'enregistrement avec ID 2.
- Tri des enregistrements restants.

Résultats

- L'enregistrement avec ID 2 est supprimé.
- Le tri n'affecte pas les enregistrements restants car ils sont déjà dans l'ordre.

3.3 Cas de test : Compactage et visualisation

Entrées

- Ajout de 10 blocs sur disque, insertion de 5 enregistrements, puis compactage.

Résultats

- Les blocs utilisés sont regroupés en mémoire.
- Les blocs inutilisés apparaissent vides dans l'état du disque.