

Projet TP : Simulateur Simplifié d'un Système de Gestion de Fichiers (SGF)

1. Contexte

Un système de gestion de fichiers (SGF) est un élément central des systèmes d'exploitation, permettant le stockage, l'organisation et la manipulation des fichiers sur une mémoire secondaire. Ce projet a pour but de développer un simulateur simplifié de SGF afin de modéliser les principes fondamentaux de gestion des fichiers.

Le simulateur reproduira les mécanismes de base d'un SGF réel, tels que l'organisation de la mémoire secondaire en blocs, l'allocation de l'espace pour les fichiers, la gestion des métadonnées, et les opérations courantes sur les fichiers (création, insertion, recherche, suppression).

2. Objectifs

- Comprendre les concepts fondamentaux liés aux systèmes de gestion de fichiers.
- Apprendre à manipuler des structures de données (tableaux, listes chaînées) dans le contexte des structures de fichiers.
- Implémenter des politiques d'allocation de la mémoire secondaire.
- Développer des algorithmes pour effectuer des opérations sur les fichiers.
- Renforcer les compétences de programmation en langage C.
- Apprendre à rédiger un compte rendu de TP.

3. Description

3.1. Mémoire Secondaire (MS)

- **Structure de la mémoire secondaire :**

La mémoire secondaire (MS) est représentée par un disque virtuel constitué de blocs de taille fixe. Les blocs sont numérotés séquentiellement à partir de l'adresse logique 0. Le nombre de blocs et le facteur de blocage (taille maximale du bloc) sont définis lors de l'initialisation du disque.

Les blocs de la MS sont gérés via une table d'allocation qui indique si un bloc est libre ou occupé. La table d'allocation est stockée au début du disque virtuel dans le premier bloc.

- **Modes d'organisation :**

Le SGF implémente deux modes d'organisations des fichiers :

- Organisation contiguë : Les enregistrements du fichier sont stockés dans des blocs adjacents.
- Organisation chaînée : Les blocs du fichier peuvent être dispersés en MS et chaque bloc contient un pointeur vers le bloc suivant.

Le mode d'organisation est choisi par l'utilisateur lors de la création du fichier.

- **Fonctionnalités associées :**

- Initialisation du disque : Définir le nombre total de blocs et le facteur de blocage. Tous les blocs sont marqués comme libres au départ.
- Table d'allocation : Met à jour l'état des blocs après chaque opération (création, suppression, etc.).
- Compactage : Le compactage consiste à juxtaposer tous les fichiers sur le disque sans laisser d'espace libre entre eux-ci.
- Vider la MS : Effacer toutes les données du disque.
- Gestion de l'espace de stockage : Avant chaque opération (création, insertion) sur les fichiers, le SGF doit vérifier si suffisamment de blocs libres sont disponibles. Si l'espace est insuffisant, un compactage est proposé pour rassembler les blocs libres et éliminer la fragmentation. Si la MS est pleine, afficher un message.

3.2.Fichiers de données

Le SGF peut gérer plusieurs fichiers de données. Un fichier de données est une collection d'enregistrements stockés dans des blocs en MS selon une organisation globale (contiguë ou chaînée) et interne (triée ou non). L'organisation, globale et interne, du fichier est définie par l'utilisateur lors de la création du fichier. Les enregistrements sont de taille fixe, et chaque enregistrement contient un identifiant unique (ID) et plusieurs champs.

Pour faciliter la saisie des données, il faudra écrire une fonction qui remplit automatiquement les données des fichiers.

- **Fonctionnalités associées :**

- Création d'un fichier : Demander le nom, le nombre d'enregistrements, et les modes d'organisation (globale et interne).
- Charger le fichier : Allouer les blocs nécessaires en MS. Un bloc ne peut contenir des enregistrements de plusieurs fichiers.
- Insertion d'un nouvel enregistrement : L'opération d'insertion doit être effectuée conformément aux modes d'organisation (globale et interne) du fichier.
- Recherche d'un enregistrement par son ID : Retourner la position de l'enregistrement sous forme (numéro du bloc, déplacement) si l'enregistrement existe ; sinon afficher un message indiquant que l'enregistrement recherché n'existe pas. Là aussi, l'opération de recherche doit être réalisée selon les modes d'organisation (globale et interne) du fichier.
- Suppression d'un enregistrement :
 - o Logique : Marquer l'enregistrement comme supprimé.
 - o Physique : Réorganiser les blocs pour libérer l'espace.
- Défragmentation : Réorganiser les blocs d'un fichier pour récupérer l'espace inutilisé après des suppressions logiques.
- Renommer un fichier.
- Supprimer un fichier.

3.3.Fichiers de métadonnées

Chaque fichier de données est associé à un fichier de métadonnées qui contient les informations nécessaires pour le gérer. Les fichiers de métadonnées sont stockés dans des blocs séparés, distincts des données des fichiers. Les métadonnées sont mises à jour après chaque opération sur le fichier correspondant.

- **Contenu des métadonnées :**

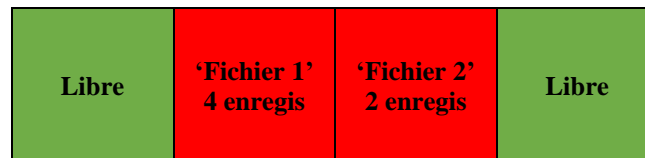
- Nom du fichier.
- Taille du fichier en blocs.
- Taille du fichier en enregistrements.
- Adresse du premier bloc.
- Modes d'organisation globale.
- Modes d'organisation interne.

3.4.Menu principal

Le simulateur devra proposer un menu interactif avec les options suivantes :

- Initialiser la mémoire secondaire.
- Créer un fichier (nom, nombre d'enregistrements, organisation globale, organisation interne) et le charger en MS selon les modes choisis.
- Afficher l'état de la mémoire secondaire de manière graphique :
 - o Blocs libres en vert, blocs occupés en rouge.
 - o Nom du fichier et nombre d'enregistrements pour chaque bloc occupé.

Par exemple :



- Afficher les métadonnées des fichiers de manière textuelle, sous forme d'un tableau descriptif.
- Rechercher un enregistrement par son ID dans un fichier spécifique.
- Insérer un nouvel enregistrement dans un fichier spécifique.
- Supprimer un enregistrement (logique ou physique) d'un fichier spécifique.
- Défragmenter un fichier.
- Supprimer un fichier.
- Renommer un fichier.
- Compactage de la mémoire secondaire.
- Vider la mémoire secondaire.
- Quitter le programme.

3.5.Étapes de développement

- Conception

- Définir les structures nécessaires pour représenter la MS, les fichiers de données, les enregistrements et les métadonnées.
- Élaborer les algorithmes pour gérer les opérations sur les fichiers.

- Implémentation

- Développer les fonctions indépendamment, avec des tests unitaires pour chaque fonctionnalité.
- Intégrer les fonctions dans un menu interactif.

- Validation
 - Tester le programme avec différents scénarios :
 - Création de fichiers, insertion, recherche, suppression.
 - Gestion des cas limites (MS pleine, fragmentation...).

3.6.Livrables

- Code source : Bien structuré et commenté, avec un fichier README expliquant comment utiliser le simulateur.
- Rapport (5 à 10 pages):
 - Description des structures de données et algorithmes utilisés.
 - Résultats des tests, avec captures d'écran illustrant le fonctionnement.

3.7.Critères d'évaluation

- Fonctionnalités implémentées (50%) : Qualité et complétude des fonctionnalités demandées.
- Structure du code (20%) : Lisibilité, modularité et bonne utilisation des structures.
- Interface utilisateur (10%) : Clarté et facilité d'utilisation du menu.
- Documentation (10%) : Rapport détaillant la conception et les résultats.
- Options supplémentaires (10%) : Git pour le suivi des modifications et la collaboration...

3.8.Remarques importantes

- Le projet doit être fait par groupe de 5.
- Le projet doit être implémenté en langage C.
- La date de remise du projet est le **Jeudi 02 Janvier 2025** via classroom.

Bonne chance.