

Rapport sur le Simulateur SGF

Auteurs :

NOM	PRENOM	MATRICULE	SECTION
Djebbar	Seddik Adel	232331531016	B
Alwazir	mohamed	232331531211	B
Messkher	Younes	232331361213	B
Boubekeur	Racim	232331692007	B
Hamdaoui	Yacine	232331717815	B
Sendid	Mohamed el hadi	232331674107	B
BALAMANE	Mohamed Hani	232331602002	B
Bourahla	Mohamed said	22331286512	B

Table des Matières

1. Introduction
2. Description des structures de données
 - 2.1 Adresse
 - 2.2 Index
 - 2.3 Enregistrement (enregi)
 - 2.4 Bloc
 - 2.5 Metadata
 - 2.6 File
 - 2.7 MS
3. Algorithmes utilisés
 - 3.1 Insertion d'enregistrement
 - 3.2 Suppression d'enregistrement
 - 3.3 Recherche d'enregistrement
 - 3.4 Compactage et défragmentation
4. Résultats des tests
 - 4.1 Scénarios de test
 - 4.2 Captures d'écran
5. Conclusion
6. Perspectives

1. Introduction

Le simulateur SGF (Système de Gestion de Fichiers) est un programme conçu pour modéliser la gestion de fichiers dans une mémoire secondaire. Il permet aux utilisateurs de créer, modifier, rechercher et supprimer des fichiers et enregistrements. Ce projet vise à illustrer les principes fondamentaux des systèmes de gestion de fichiers, comme FAT32 ou NTFS, en adoptant une approche simplifiée pour l'apprentissage.

Ce rapport présente les structures de données utilisées, les algorithmes clés et les résultats des tests effectués sur le simulateur.

2. Description des structures de données


2.1 Adresse

```
1 typedef struct {
2     int num_bloc;    // Numero du bloc
3     int deplacement; // Position dans le bloc
4 } Adresse;
5
```

2.2 Index


```
1 // Structure pour une entree d'index
2 typedef struct {
3     int id;        // ID de l'enregistrement
4     Adresse adr;    // Adresse de l'enregistrement
5 } Index;
```

2.3 Enregistrement (enregi)



```
1 // Structure d'enregistrement
2 typedef struct {
3     int id;
4     int value;
5 } enregi;
```

2.4 Bloc



```
1 // Structure de bloc
2 typedef struct {
3     enregi tab[FB]; // Tableau d'enregistrements (MAX=16)
4     int state; // 1 si occupe, 0 si libre
5     char Fname[20]; // Nom du fichier
6     int nbr_enregi; // Nombre d'enregistrements dans ce bloc
7 } bloc;
```

2.5 Metadata

```
// Structure metadata
typedef struct {
    char filename[20];
    int nbr_bloc;
    int nbr_enregi_maximal; // nombre de enregistrement maximal qui peut les
    // obtenir se file
    int nbr_enregi_actuelle; // nombre de enregistremen inserer dans les
    // blocs de se fichier
    int indice_du_premier_bloc; // indice de premier blocs du file dans le
    // disque
```

```

    int mode_global;    // 1 pour contigu, 0 pour chaene
    int mode_local;     // 1 pour trie, 0 pour non trie
    Index index_tab[ERGMAX]; // ce tablo d'index est util pour le recher
    telleque il stoc le id de chaque enregistrement apre lincertion pour
    facilite la recherch

} metadata;

```

2.6 File

```

1
2 typedef struct {
3     char filename[20];
4     int file_size;
5     int tab_bloc[10]; // Indices des blocs dans le disque qu'ils ont le nom de ce fils
6     metadata fmeta;
7 } file;
8

```

2.7 MS

```

1 typedef struct {
2     bloc disque[N];
3     int tab_all[N]; // Table d'allocation
4     file tab_file[MAX_FILES]; // chaque nouveau fichier sera incerer dans ce tablaeu
5     int nb_files; // Nombre de fichiers actuels
6 } MS;

```

3. Algorithmes utilisés

3.1 Insertion d'enregistrement

Vérifie l'unicité de l'ID. Selon le mode (trié ou non), insère à la bonne position.

3.2 Suppression d'enregistrement

Supprime logiquement ou physiquement en mettant à jour les index.

3.3 Recherche d'enregistrement

Utilise le tableau d'index pour localiser rapidement un enregistrement.

3.4 Compactage et défragmentation

Réorganise les blocs et évite la fragmentation.

4. Résultats des tests

4.1 Scénarios de test

Test	Description	Résultat attendu	Résultat obtenu
Création de fichier	Créer un fichier avec des métadonnées correctes	Fichier créé avec succès	✓
Insertion d'enregistrements	Ajouter plusieurs enregistrements	Enregistrements ajoutés	✓
Suppression d'enregistrement	Supprimer un enregistrement	Mise à jour des index	✓
Recherche d'enregistrement	Trouver un enregistrement par ID	Enregistrement localisé	✓

4.2 Captures d'écran

```
=====
      Simulateur SGF - Menu
=====
1. Initialiser la memoire secondaire
2. Creer un fichier
3. Charger un fichier
4. Afficher leetat de la memoire secondaire
5. Afficher les metadonnees des fichiers
6. Rechercher un enregistrement par ID
7. Insérer un nouvel enregistrement
8. Supprimer un enregistrement
9. Defragmenter un fichier
10. Supprimer un fichier
11. Renommer un fichier
12. Compactage de la memoire secondaire
13. Vider la memoire secondaire
14. Quitter
=====
```

Création d'un fichier

```
Entrez votre choix : 2
Entrez le nom du fichier (max 20 caracteres) : etudiant
Entrez le nombre maximal d'enregistrements pour ce fichier (max 100) : 5
Choisissez le mode global (1 - Contigu, 0 - Chaene) : 1
Choisissez le mode local (1 - Trie, 0 - Non trie) : 1
Fichier 'etudiant' cree avec succes !
```

Charger le fichier (il faut charger le fichier dans MS)

```
=====
Entrez votre choix : 3
Entrez le nom du fichier e charger : etudiant
Nombre de blocs necessaires pour 'etudiant' : 2
Fichier 'etudiant' charge en mode contigu.
```

Insertion d'un enregistrement

```
=====
Entrez votre choix : 7
Entrez le nom du fichier oe inserer l'enregistrement : etudiant
Entrez l'ID de l'enregistrement : 10
Entrez la valeur de l'enregistrement : USTHB
Enregistrement insere avec succes dans le fichier 'etudiant' (mode non trie).
```

Suppression d'un enregistrement

```
=====
Entrez votre choix : 10
Entrez le nom du fichier e supprimer : etudiant
```

état de la MS

```
etat de la memoire secondaire :
=====
Bloc 0 : LIBRE
Bloc 1 : LIBRE
Bloc 2 : LIBRE
Bloc 3 : LIBRE
Bloc 4 : LIBRE
Bloc 5 : LIBRE
Bloc 6 : LIBRE
Bloc 7 : LIBRE
Bloc 8 : LIBRE
Bloc 9 : LIBRE
Bloc 10 : LIBRE
Bloc 11 : LIBRE
Bloc 12 : LIBRE
Bloc 13 : LIBRE
Bloc 14 : LIBRE
Bloc 15 : LIBRE
Bloc 16 : LIBRE
Bloc 17 : LIBRE
Bloc 18 : LIBRE
Bloc 19 : LIBRE
Bloc 20 : LIBRE
Bloc 21 : LIBRE
Bloc 22 : LIBRE
Bloc 23 : LIBRE
Bloc 24 : LIBRE
Bloc 25 : LIBRE
```

```

Bloc 0 : OCCUPE - Fichier : etudiant | Nombre d'enregistrements : 0
Contenu :
Bloc 1 : OCCUPE - Fichier : etudiant | Nombre d'enregistrements : 0
Contenu :
Bloc 2 : LIBRE
Bloc 3 : LIBRE
Bloc 4 : LIBRE
Bloc 5 : LIBRE
Bloc 6 : LIBRE
Bloc 7 : LIBRE
Bloc 8 : LIBRE
Bloc 9 : LIBRE
Bloc 10 : LIBRE
Bloc 11 : LIBRE
Bloc 12 : LIBRE
Bloc 13 : LIBRE
Bloc 14 : LIBRE
Bloc 15 : LIBRE
Bloc 16 : LIBRE
Bloc 17 : LIBRE
Bloc 18 : LIBRE
Bloc 19 : LIBRE
Bloc 20 : LIBRE
Bloc 21 : LIBRE
Bloc 22 : LIBRE

```

Recherche par ID

```

=====
Entrez votre choix : 6
Entrez le nom du fichier ou rechercher l'enregistrement : produit
Entrez l'ID de l'enregistrement e rechercher : 30
Enregistrement trouve :
- ID : 30
- Valeur : 1977783348
- Bloc : 2
- Deplacement : 0
=====

```

Afficher les metadonnées de fichier

```

Entrez votre choix : 5
Entrez le nom du fichier e afficher : produit
Contenu du fichier 'produit' :
=====
Bloc | Deplacement | ID   | Valeur
=====
  2 |           0 |   30 | 1977783348
=====
Fin du fichier 'produit'.

```

La suppression physique et logique

```

14. Quitter
=====
Entrez votre choix : 8
Entrez le nom du fichier ou supprimer l'enregistrement : produit
Entrez l'ID de l'enregistrement e supprimer : 30
Choisissez le mode de suppression (0 pour logique, 1 pour physique) : 0
Enregistrement avec l'ID '30' supprime logiquement.

=====
          Simulateur SGF - Menu
=====

```

Defragmentation


```

12. Compactage de la memoire secondaire
13. Vider la memoire secondaire
14. Quitter
=====
Entrez votre choix : 9
Entrez le nom du fichier e defragmenter : produit
Defragmentation du fichier 'produit' en cours...
Defragmentation du fichier 'produit' terminee.

=====
Simulateur SGF - Menu

```

Comtage de la MS (eliminer meme les enregistrements qui ont supprimer logiquement)

```

=====
Entrez votre choix : 12
Compactage effectue avec succes!
Nombre de blocs occupees apres compactage : 2

=====
Simulateur SGF - Menu
=====

```

```

=====
Entrez votre choix : 12
Compactage effectue avec succes!
Nombre de blocs occupees apres compactage : 2

=====
Simulateur SGF - Menu
=====
Bloc 0 : OCCUPE - Fichier : etudiant | Nombre d'enregistrements : 1
Contenu :
    Enregistrement 0 : ID = 10, Value = 1977783348
Bloc 1 : OCCUPE - Fichier : etudiant | Nombre d'enregistrements : 0
Contenu :
Bloc 2 : LIBRE
Bloc 3 : LIBRE
Bloc 4 : LIBRE
Bloc 5 : LIBRE
Bloc 6 : LIBRE
Bloc 7 : LIBRE

```

Et pluser fonctions qui inclus dans nous code ...

Le code s'arrête SSI chois Quitter(14)

5. Conclusion

Le simulateur SGF a été conçu pour illustrer efficacement les principes de gestion des fichiers dans une mémoire secondaire. Les résultats des tests montrent que toutes les fonctionnalités principales fonctionnent correctement.

6. Perspectives

Pour améliorer ce simulateur, nous pourrions ajouter les fonctionnalités suivantes :

Gestion des fichiers multi-utilisateurs.

Ajout d'un système de journalisation (logs) pour tracer les opérations.

Interface graphique pour remplacer le menu textuel.