



Module : Structure de fichiers & Structure de données TP2 : Allocation & Indexation des fichiers

Note : Les programmes de cette série de travaux pratiques doivent être élaborés en langage C.

Exercice 1 : Allocation contiguë

Écrire un programme qui implémente l'allocation contiguë de fichiers dans un disque virtuel (tableau) de 100 secteurs (cases). Le programme affiche un menu principal qui permettra le lancement des sous-programmes suivants :

Menu :

1. Créer un fichier
2. Lire un fichier
3. Supprimer un fichier
4. Fragmenter le disque
5. Afficher l'état du disque
6. Quitter

Choix :

1. *Créer des fichiers* : L'utilisateur peut créer des fichiers en spécifiant le nom (2 caractères) et la taille en nombre de secteurs du fichier. Les fichiers doivent être sauvegardés sur le disque virtuel en marquant les secteurs occupés par chaque fichier.
2. *Lire un fichier* : Le programme doit permettre de lire un fichier existant et afficher sa taille ainsi que le premier secteur marquant le début du fichier.
3. *Supprimer des fichiers* : L'utilisateur peut supprimer des fichiers existants du disque virtuel tout en mettant à jour l'état des secteurs du disque.
4. *Défragmenter le disque* : Le programme doit offrir une option pour défragmenter le disque, optimisant ainsi l'utilisation de l'espace ; les secteurs vides seront décalés vers la fin du disque.
5. *Afficher l'état du disque* : L'état du disque doit être affiché sous forme d'une matrice de 10x10, où les secteurs vides sont marqués par "##", et les secteurs occupés sont marqués par les noms des fichiers qui les occupent.
6. *Quitter* : Terminer l'exécution du programme.

Pour tester le programme, suivez ce scénario :

1. Créez un fichier F1 de taille 5.
2. Créez un fichier F2 de taille 50.
3. Créez un fichier F3 de taille 15.
4. Créez un fichier F4 de taille 30.

5. Supprimez F1.
6. Supprimez F3.
7. Tentez de créer un fichier F5 de taille 20, le programme doit normalement signaler que l'espace disque est insuffisant, mais que l'ajout est possible après une défragmentation.
8. Défragmentez le disque.
9. Créez le fichier F5.

Assurez-vous que le programme affiche l'état du disque correctement, correspondant à chaque opération effectuée.

Exercice 2 : Allocation chaînée

Écrire un programme qui met en œuvre l'allocation chaînée de fichiers dans un disque virtuel représenté par un tableau de 100 secteurs. L'interface du programme offre un menu principal permettant d'accéder aux fonctionnalités suivantes :

1. *Créer des fichiers* : L'utilisateur peut générer des fichiers en spécifiant un nom (2 caractères) ainsi que la taille en nombre de secteurs du fichier. Les fichiers sont stockés sur le disque virtuel sous forme d'une liste chaînée, où chaque secteur comprend un espace pour les données et un espace contenant le numéro du secteur suivant.
2. *Supprimer des fichiers* : L'utilisateur peut effacer des fichiers existants du disque virtuel, tout en actualisant l'état des secteurs du disque. Chaque secteur vide est marqué comme suit : l'espace des données devient 0 et l'espace du secteur suivant devient -1.
3. *Afficher l'état du disque* : L'état du disque est affiché sous forme d'une matrice 10x10. Les secteurs vides sont marqués par '##|##', et les secteurs occupés sont marqués de la même façon en remplaçant les deux premières positions avant la barre par le nom du fichier occupant le secteur, et les deux positions après la barre par le numéro du secteur suivant sur le disque, ou -1 pour signaler la fin du fichier.
4. *Quitter* : Terminer l'exécution du programme.

Tester le programme avec le même scénario utilisé en Exercice 1.

Exercice 3 : Allocation chaînée indexée

De la même manière, écrire un programme qui met en œuvre l'allocation chaînée indexée de fichiers dans un disque virtuel représenté par un tableau de 100 secteurs.

Exercice 4 : Allocation par nœud d'information

Développer un programme pour gérer des fichiers sur un disque virtuel simulé en utilisant l'allocation par nœud d'information. Les structures de données suivantes seront utilisées :

1. *INode* : Représente les informations d'index d'un fichier, contenant la taille du fichier (métadonnées) et une allocation dynamique des secteurs où il est stocké sur le disque.
2. *Fichier* : Structure contenant le nom du fichier et sa référence à son INode associé.

Dans la continuité des exercices précédents, le programme se présente aussi avec un menu principal. Ce menu offre la possibilité de créer de nouveaux fichiers, de les supprimer et de visualiser l'état actuel du disque.