Système d'Exploitation
Travaux Dirigés (4), Licence 2 Informatique
Gestion de Fichiers

On considère dans l'ensemble de ce TD une disquette formatée de 1.44 Mo divisée en blocs de 1Ko. Le stockage d'un fichier sur la disquette se fait en lui allouant un certain nombre de blocs qui peuvent ne pas être contigus. Le système doit donc gérer des structures de données lui permettant de savoir quels sont les blocs libres et les blocs occupés de la disquette.

1. Gestion d'Espace Libre

On suppose ici que les blocs 0 à 13 de la disquette sont occupés, ainsi que tous les blocs à partir de 20.
On considère dans un premier temps que le système gère l'espace libre au moyen d'un vecteur binaire (bitmap).

1.1.

> Quelle est la taille du vecteur binaire ? Combien de blocs occupe-t-il ? Donnez la valeur de ses 24 premières entrées.

Une autre stratégie consiste à gérer l'espace libre sous forme groupée : un bloc spécial (appelé superbloc) contient (entre autres) :

- En 1ère position : le numéro du prochain bloc contenant une liste de blocs libres,
- une liste de blocs libres.

Lorsqu'on utilise le dernier numéro de bloc qu'il contient, on recopie dans le bloc spécial le contenu du bloc correspondant à ce dernier numéro.

1.2.

> On se place dans le cas où les numéros de blocs sont codés sur 2 octets. Quelle est, initialement, la taille totale nécessaire pour gérer l'information sur les blocs libres de la disquette ?

1.3.

> Que pensez-vous de la place utilisée pour la gestion dans les deux stratégies ?

Mohamed Taghelit , taghelit@univ-tours.fr

2. Gestion d'Espace Occupé

Une première façon de gérer l'espace occupé par un fichier est de lui allouer un (ou plusieurs) bloc(s) d'index. Un répertoire est alors une table donnant pour chaque fichier les coordonnées de son bloc d'index. Le bloc d'index contient les numéros des blocs occupés par le fichier.

2.1.

> On suppose qu'un fichier F1 occupe sur disque les blocs 25, 26, 37 et 63. Son bloc d'index est stocké dans le bloc 13. Un fichier F2 occupe les blocs 19 et 43, son bloc d'index est stocké dans le bloc 16. Donnez le contenu du répertoire et des blocs d'index.

À l'ouverture d'un fichier, son bloc d'index est chargé en mémoire.

2.2.

> Quel est le nombre d'accès mémoire et le nombre d'E/S nécessaires pour lire le 4000ème octet du fichier F1 ?

Dans le système de gestion des fichiers d'UNIX, à chaque fichier est associée une structure de données appelée `inode`. Dans cette structure, on trouve (entre autres) 13 entrées :

- les 10 premières entrées référencent des blocs de données sur le disque,
- la 11ème référence un bloc de contrôle qui référence des blocs de données (simple indirection),
- la 12ème référence un bloc de contrôle qui référence des blocs de contrôle qui référencent des blocs de données (double indirection),
- la 13ème référence un bloc de contrôle qui référence des blocs de contrôle qui référencent des blocs de contrôle qui référencent des blocs de données (triple indirection).

Chaque bloc de contrôle permet de référencer au maximum 128 blocs (contrôle ou données).
On considère trois fichiers F1, F2 et F3 ayant les caractéristiques suivantes :

F1 7 blocs de données,
F2 11 blocs de données,
F3 139 blocs de données.

2.3.

> Donnez le nombre d'indirections utilisées pour le stockage des fichiers F1, F2 et F3.

2.4.

> Avec des blocs de 512 octets, quelle est la quantité maximum de données que l'on peut stocker dans un fichier par cette méthode ?

3. Gestion Mixte

La FAT (File Allocation Table) est une structure de données permettant de gérer à la fois l'espace libre et l'espace alloué. Il s'agit d'un tableau avec une entrée par bloc.
Dans un répertoire, on trouve pour chaque fichier le numéro du premier bloc occupé par le fichier. L'entrée correspondante de la FAT contient l'adresse du 2ème bloc occupé, etc… Le dernier contient une valeur spéciale "fin de fichier", les blocs vides sont à 0.

3.1.

On suppose que chaque numéro de bloc est stocké sur 2 octets. Quelle est la taille de la FAT pour une disquette de 1.44 Mo avec des blocs de 1 Ko ?

3.2.

On suppose qu'un fichier F1 occupe sur disque les blocs 25, 26, 37 et 63. Donnez le contenu du répertoire et des entrées de la FAT correspondant au stockage de ce fichier.

3.3.

Si la FAT est chargée en mémoire, quel est le nombre d'accès mémoire et le nombre d'E/S nécessaires pour lire le 4000ème octet du fichier F1 ?

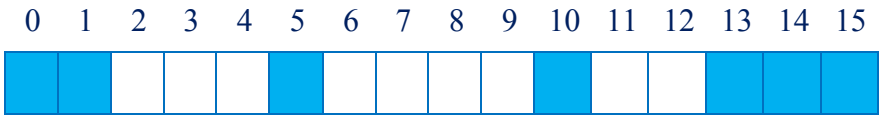
4. Allocation d'un Fichier sur Disque

On veut gérer l'allocation de données sur disque pour des fichiers utilisateurs. Le disque est structuré en **blocs**. On suppose que le système alloue à chaque fichier un bloc d'**index**.

4.1.

Quelle est la méthode de gestion d'espace libre qui vous semble la plus adaptée si l'on veut optimiser (en minimisant les déplacements de tête) l'accès séquentiel aux fichiers ?

La configuration de la piste 24 de la disquette est la suivante :



où les parties grisées représentent des blocs occupés, les parties blanches des blocs libres.
Un utilisateur réalise les allocations successives dans l'ordre suivant :

Création du fichier F1 avec 3 blocs de données
Création du fichier F2 avec 1 bloc de données
Création du fichier F3 avec 2 blocs de données

4.2.

En essayant d'optimiser l'accès séquentiel, faites un schéma des blocs du disque après l'allocation. Indiquez par un `I` les blocs d'index et par un `D` les blocs de données. Indiquez également le nom du fichier auquel le bloc appartient.