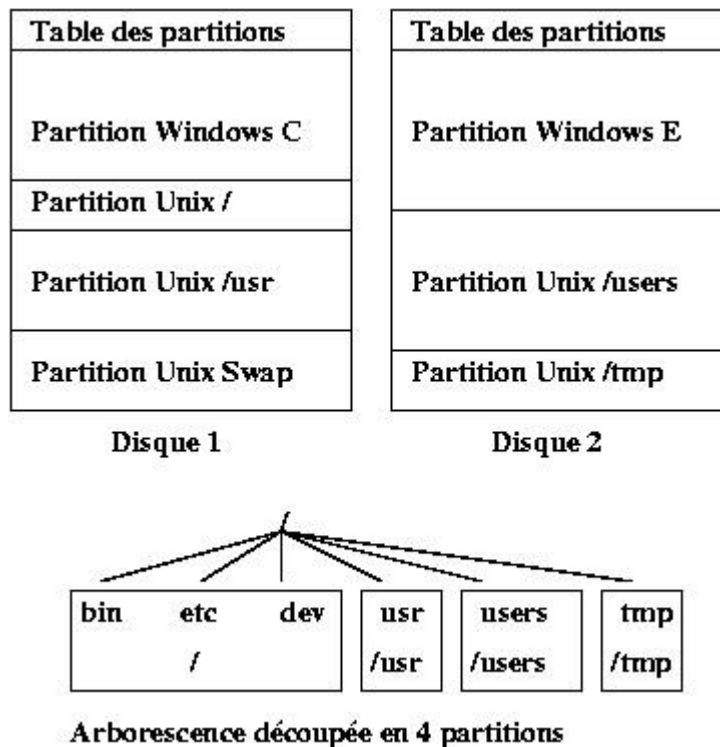


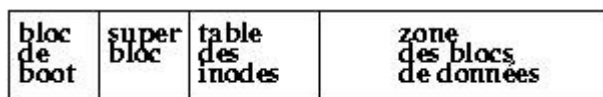
Notion sur le FileSystem

Le système de fichiers Unix est représenté par une arborescence. L'arbre est une abstraction dont la réalisation physique se présente sous la forme d'un ensemble de partitions de disques. Chaque partition est un système de fichiers à part entière. L'assemblage des partitions pour former l'arborescence complète se fait dynamiquement en mémoire et n'apparaît en aucune façon sur le disque. La figure ci-dessous illustre le mappage d'une arborescence en quatre partitions.



Dans l'exemple, les partitions Windows contiennent des structures Windows. La partition Swap contient la zone de swap qui sert de mémoire réelle des processus. Les autres partitions Unix sont formatées en système de fichiers Unix. La table des partitions qui figure en tête de chaque disque décrit chaque partition sous forme d'un triplet (début, longueur, type).

Une partition formatée en SGF Unix se compose de quatre parties: le bloc de boot, le super-bloc, la table des inodes et la zone des blocs de données.



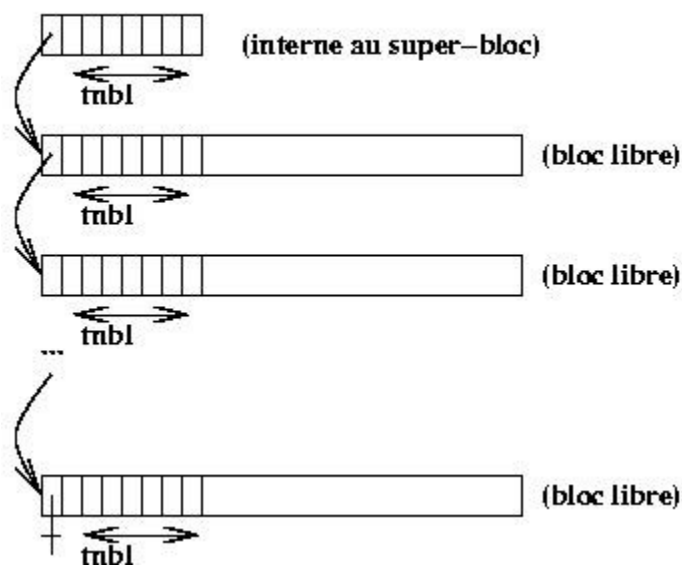
Structure d'une partition formatée

Le bloc de boot n'est rempli que si la partition est une partition de démarrage (partition qui contient une copie du noyau). Il contient un petit programme chargeant le noyau en mémoire et lui donnant le contrôle.

Le super-bloc contient la taille de la zone des blocs de données, le nombre de blocs libres dans cette zone, une table servant de tête de liste des blocs libres, la taille de la table des inodes, le nombre d'inodes libres, une table contenant des numéros d'inodes libres et enfin l'état du super-bloc.

La zone des blocs de données est un mélange de blocs occupés et de blocs libres. Les blocs occupés sont accessibles à partir de l'inode auquel ils appartiennent. Les blocs libres sont chaînés les uns aux autres, avec une amorce de la liste dans le super-bloc. La figure ci-dessous montre la structure de la liste.

Structure de la liste des blocs libres



Dans chaque bloc de la liste, ainsi que dans le super-bloc, on trouve une table `tbl` de numéros de blocs libres (des blocs non chaînés dans la liste). Toutes ces tables ont la même taille. Chaque entrée de table contient soit un numéro de bloc libre, soit 0 (entrée vide).

Voici comment on alloue un bloc:

Si la table `tbl` du superbloc contient une entrée non nulle,

elle fournit directement le numéro du bloc alloué.
L'entrée correspondante est mise à zéro.

Sinon,
la table tnb1 du premier bloc de la liste est chargée
dans celle (vide) du super-bloc.
Le bloc libre ainsi vidé de sa table tnb1 est alloué.
La table tnb1 du super-bloc est rechaînée
au successeur du bloc alloué.

Voici comment on libère un bloc:

Si la table tnb1 du superbloc contient une entrée nulle,
le numéro du bloc libéré y est inscrit.

Sinon,
la table tnb1 du super-bloc (pleine) est copiée
dans le bloc libéré.
Le bloc libéré est inséré en tête.
La table tnb1 du super-bloc est vidée.
(elle pointe sur le bloc libéré).

La table des inodes est à accès direct (le numéro de l'inode désigne son entrée dans la table). Les deux premières entrées sont inoccupées (les numéros d'inode 0 et 1 sont inemployés). L'inode 2 (première entrée occupée de la table) est la racine du SGF figurant dans la partition.

Un inode est une structure de taille fixe représentant un noeud de l'arborescence. Un inode regroupe les champs suivants:

- l'identifiant du propriétaire (uid),
- l'identifiant du groupe propriétaire (gid),
- le type de l'inode (fichier, répertoire, spécial, tube, inode libre, ...),
- les droits (rwxrwxrwx),
- la date de dernière modification du fichier,
- la date de dernier accès au fichier,
- la date de dernière modification de l'inode,
- le nombre de liens,
- la table des blocs (13 entrées),
- la taille du fichier en octets.

Voici comment on alloue un inode:

Si la table tnil (table des numéros d'inodes libres)
du super-bloc contient une entrée non nulle,
le numéro de l'inode alloué y est inscrit.

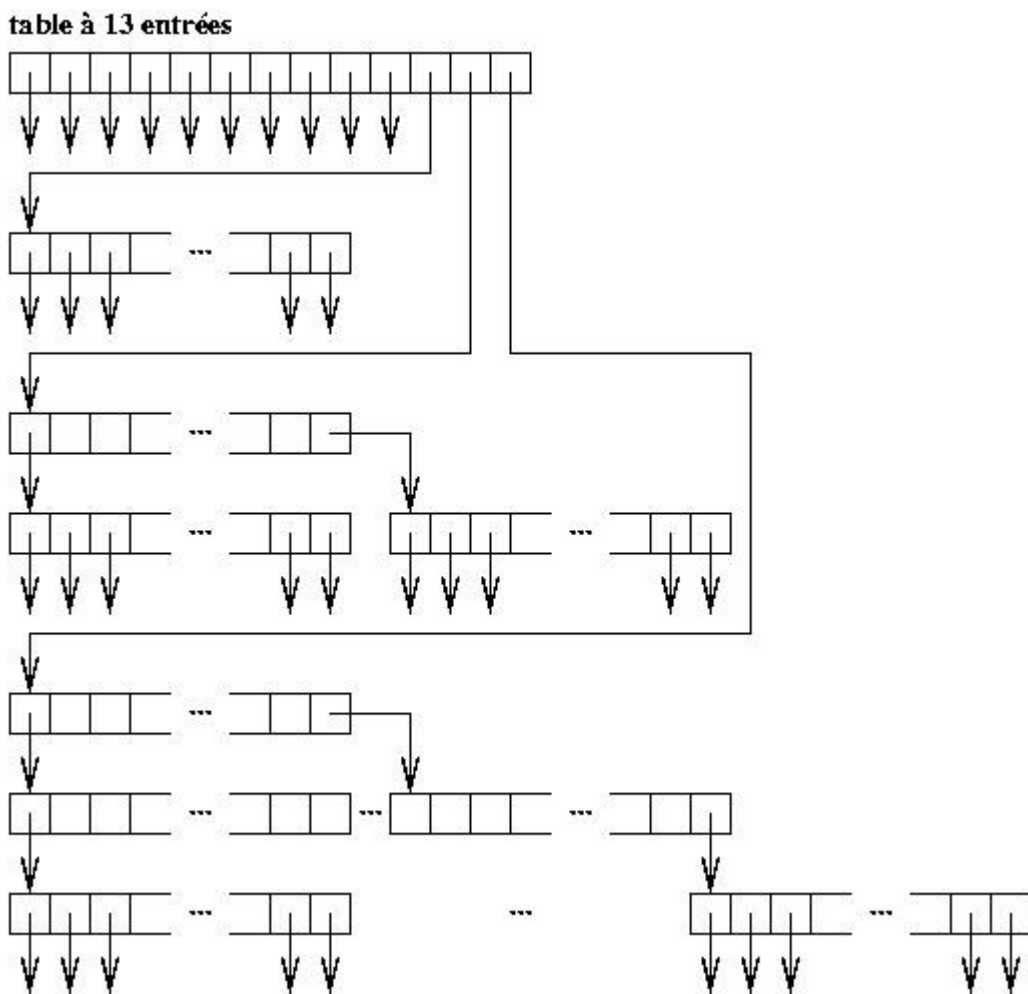
Sinon,
on parcourt la table des inodes pour remplir la table tnil du super-bloc.
Chaque inode dont le type est 0 est libre.
Son numéro est ajouté à la table tnil.
Quand la table est pleine, le parcours s'achève.

Voici comment on libère un inode:

Si la table tnil contient une entrée nulle,

le numéro de l'inode libéré y est inscrit.
Le champ type de l'inode est mis à 0.

La figure suivante montre comment sont organisés les blocs de données d'un inode. Rappelons que chaque inode contient dans la structure qui le décrit une table des blocs à 13 entrées. Les 10 premières entrées donnent accès directement aux dix premiers blocs de données. La onzième entrée donne accès à un bloc prolongeant les dix premières entrées de la table. La douzième entrée donne accès à un bloc prolongeant la onzième entrée. Enfin, la treizième entrée donne accès à un bloc prolongeant la douzième entrée. Si un bloc peut contenir n numéros de blocs, la table à 13 entrées permet de désigner au maximum $(10 + n + n^2 + n^3)$ blocs. Les 10 premiers sont accédés directement, les n suivants avec un niveau d'indirection, les n^2 suivants avec deux niveaux et les n^3 derniers avec trois niveaux.



La structure abstraite de l'arborescence qui combine plusieurs partitions physiques est constituée dynamiquement par le système (cette arborescence abstraite n'est donc pas écrite sur le disque; elle est

reconstituée à chaque fois que le système est démarré et peut évoluer dynamiquement). Au démarrage du système, l'arborescence se réduit à une partition: la partition de démarrage. Par la suite, divers scripts shell de configuration montent (greffent) de nouvelles partitions soit sur la partition de démarrage, soit sur les partitions greffées. Chaque greffe ajoute une ligne à une table des points de montage. La ligne précise la partition greffée (disque, bloc de début), l'inode correspondant au point de greffe (un répertoire de la partition sur laquelle s'opère la greffe) et la partition du point de greffe (disque, bloc de début). A chaque déplacement descendant dans l'arborescence, le noyau cherche l'inode de destination dans la table des points de montage. S'il y est trouvé, le noyau change de partition. De même, à chaque déplacement ascendant, si la racine de partition est atteinte (inode 2), la partition courante est cherchée dans la table des points de montage, ce qui permet de retrouver la partition ascendante (on est à la racine absolue si la partition courante est celle de démarrage). Par exemple, la table des points de montage de l'arborescence abstraite de la figure 1 pourrait être (en supposant que les répertoires `usr`, `users` et `tmp` aient les inodes 140, 220 et 300 dans la partition de démarrage (partition `p2` du disque `d1`)):

partition greffée	inode	partition du point de greffe
<code>usr (d1, p3)</code>	<code>/usr (140)</code>	<code>(d1, p2)</code>
<code>users (d2, p2)</code>	<code>/users (220)</code>	<code>(d1, p2)</code>
<code>tmp (d2, p3)</code>	<code>/tmp (300)</code>	<code>(d1, p2)</code>

En remontant à la racine (inode 2) de la partition `(d2, p2)` (i.e. `users`), on se retrouve dans la partition `(d1, p2)` (i.e. `/`), inode 220. En descendant dans le répertoire `/tmp` (inode 300 de `(d1, p2)`), on se retrouve à la racine (inode 2) de `(d2, p3)`.