

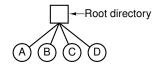
Systèmes d'exploitation Systèmes de fichiers / Répertoires

Jacques Supcik | 2018/2019 | T-2adfg I08-handouts | 2018-12-07





Les systèmes à répertoire hiérarchique



Système de répertoire à un seul niveau contenant quatre fichiers.

- Il n'y a que le «root directory» (pas de sous-répertoire).
- Utilisé par des systèmes tels que le CDC 6600 ou les premiers PC (TRS 80).



Système de répertoire à un seul niveau



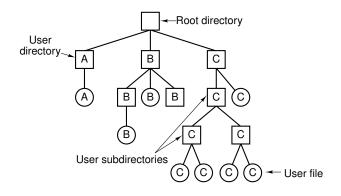
CDC 6600 (1965)



TRS 80 (1977)



Les systèmes à répertoire hiérarchique



Système de répertoire hiérarchique.



Les chemins d'accès (path names)

#### Chemin d'accès absolu

- Commence à la racine (root).
- Utilise un symbole défini comme séparateur («\» pour Windows, «/» pour UNIX, «>» pour MULTICS).
- Le premier caractère du chemin est le séparateur.

#### Chemin d'accès relatif

 Défini par rapport au répertoire de travail ou répertoire courant («working directory» ou «current directory» ou «current working directory»).

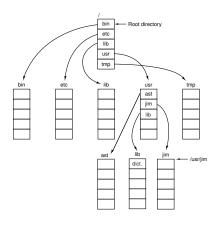


Les chemins d'accès (path names)

- Chaque processus a son propre répertoire de travail.
- Répertoires spéciaux («.» et «..») pour représenter le répertoire courant et le répertoire parent.
- Sous UNIX, les fichiers commençant par un point sont cachés. Pour les afficher, utilisez la commande «ls -a»<sup>1</sup>.



Les systèmes à répertoire hiérarchique



Arborescence UNIX.



Les opérations sur les répertoires

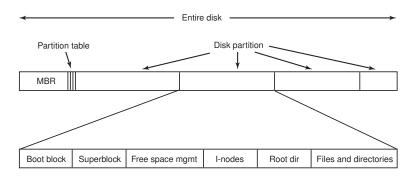
## Appels systèmes pour la gestion des répertoires :

- Create
- Delete
- Opendir
- Closedir

- Readdir
- Rename
- Link
- Unlink



#### L'organisation du système de fichiers



Organisation possible d'un système de fichiers.



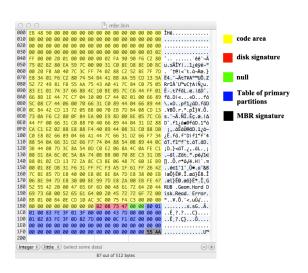
#### L'organisation du système de fichiers

Le «superblock» du système Ext2 contient, entre autres, les informations suivantes :

- Magic Number (0xEF53)
- Revision Level
- Mount Count and Maximum Mount Count
- Block Size
- Free Blocks
- Free Inodes
- First Inodes

Utilisez la commande «dumpe2fs» pour afficher les paramètres d'un système de fichier EXT2 (ou EXT3, ou EXT4).

#### Le MBR

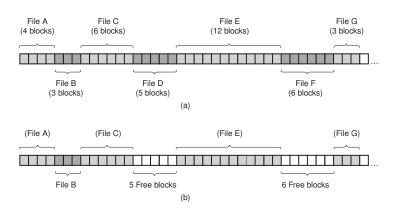


#### Le MBR (wikipedia)



- Il y a 440 Byte de disponible pour mettre le code de démarrage (lu par le BIOS).
- Partitions «primaires» et «étendues».

#### L'allocation contiguë



- Allocation contiguë de l'espace disque pour 7 fichiers.
- L'état du disque après la suppression des fichiers D et F.



L'allocation contiguë

## **Avantages**

- Simple à implémenter.
- Excellentes performances en lecture (1 recherche et une lecture).

## Désavantages

- Fragmentation (et compactage trop lent).
- Les fichiers ne peuvent pas «grandir» et on ne sait pas toujours à l'avance la taille que prendra un fichier.



L'allocation contiguë

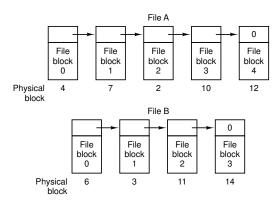
### Histoire

- Cette technique était utilisée par les premiers systèmes d'exploitation.
- Elle a été laissée de côté pendant de nombreuses années.
- Elle est revenue avec les CD-ROM.

Il n'est donc pas complètement inutile d'étudier les «anciens» systèmes!



#### Allocation par liste chaînée



Stockage d'un fichier à l'aide d'une liste chaînée de blocs de disque.



#### Allocation par liste chaînée

## **Avantages**

- Pas de perte de capacité, pas de gaspillage (pas de «trous»).
- Lecture séquentielle facile.

## Désavantages

- Lecture aléatoire difficile («seek»). Pour lire un bloc, n on doit lire les n-1 blocs précédents.
- La taille des blocs de données n'est plus une puissance de 2, car le pointeur fait partie du bloc.

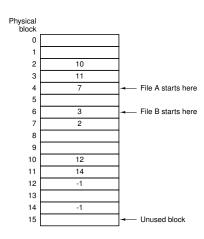


#### Allocation par liste chaînée avec table en RAM

- Idée : sortir les pointeurs des blocs.
- Garder cette structure en RAM.



#### Allocation par liste chaînée avec table en RAM



Allocation par liste chaînée utilisant une table en mémoire principale.



Allocation par liste chaînée avec table en RAM

## **Avantages**

- Accès aléatoire plus performant, car la table est en mémoire.
- Les blocs peuvent à nouveau avoir des tailles avec des puissances de 2.

## Désavantages

- La table utilise de la RAM.
- La table doit être régulièrement copiée sur le disque.
- Problème avec les très gros disques.

Ce système est utilisé par MS-DOS (FAT = «File Allocation Table»)



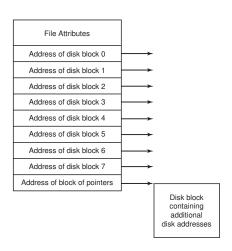
#### Allocation par liste chaînée avec table en RAM

Le problème avec les gros disques :

- Supposons que nous avons un disque de 1 TByte (courant de nos jours).
- On souhaite utiliser des blocs de 4 kByte pour limiter le gaspillage.

On va alors avoir 256 millions de blocs et avec des adresses sur 32 bit (4 Byte), on aura alors besoin de 1 GByte de RAM pour la FAT!

Les i-nodes



## Exemple d'i-node.



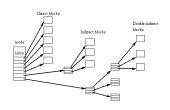
Les i-nodes

## **Avantages**

 Seul les tables des fichiers ouverts sont en RAM (et non la table du disque entier).



#### Les i-nodes / Ext2



- 12 pointeurs directs.
- 1 pointeur indirect.

- 1 pointeur «double» indirect.
- 1 pointeur «triple» indirect.

Avec des blocs de 2 kByte et des pointeurs sur 32 bit, on a une taille maximum pour les fichiers de :

$$(12 + 512 + 512^2 + 512^3) \cdot 2048 \approx 256 \,\mathrm{GByte}$$



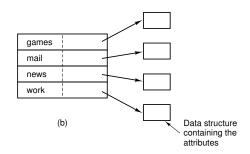
Que devons-nous stocker dans chaque entrée du répertoire?

- Le nom du fichier
- L'adresse (dans le cas de l'allocation contiguë), le premier bloc, ou le premier i-node.
- Des attributs tels que le propriétaire, les droits d'accès, la date de création, ...



#### Où stocker les attributs?





- Répertoire simple contenant des entrées de taille fixe avec les adresses disque et les attributs.
- Répertoire dans lequel chaque entrée fait référence à un i-node.

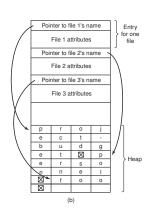


Longueur des noms de fichiers

- Longueur fixe courte (8.3): facile.
- Longueur fixe longue (250 caractères): gaspillage.
- Longueur variable : plus difficile à implémenter.







Deux façons de traiter les noms de fichiers longs dans un répertoire :

- En ligne (in-line).
- Dans un tas (heap).



Longueur des noms de fichiers variable

- En-ligne: la suppression d'un fichier reste assez difficile («trous» de taille variable dans la structure).
- Dans les deux cas, la recherche d'un fichier peut prendre beaucoup de temps (recherche séquentielle).
- On peut améliorer la vitesse de la recherche avec des techniques telles que «hash table», «balanced tree», «cache».