

300 Systèmes de gestion de fichiers

INF3173

Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Hiver 2021

Mémoire de masse

Objectif : stocker des données

- Sur des périphériques
- De manière persistante (non volatile)
- En grande quantité (gros volumes)

Problèmes

- Technologies physiques variées
 - Temps d'accès varié aussi (mais plus lent que la RAM)
- Responsabilité du système d'exploitation

Disque : abus de langage

Il n'y a pas forcément de **disque physique**

- Disque SSD (*solid-state drive*)
- Espace disque : `df(1)`, `du(1)`

Temps d'accès

- Registres CPU, ko, $\approx .25\text{ns}$
- Cache CPU, Mo, $\approx 10\text{ns}$
- RAM, Go, $\approx 100\text{ns}$
- Disque SSD, To, $\approx 25\,000\text{ns}$ ($25\mu\text{s}$)
- Disque magnétique, To, $\approx 5\,000\,000\text{ns}$ (5ms)

Gestion de l'espace disque et des fichiers

Gestion de l'espace disque

- Répondre aux demandes d'allocation de libération de l'espace disque
 - Retrouver les fichiers et répertoires
 - S'assurer de la fiabilité
- le tout, efficacement

Abstraction pour l'utilisateur

- Abstraction de la gestion de l'espace
 - Cohérente et indépendante
- Fichiers (et répertoires)

Notion de base : le fichier

Système de gestion de fichiers (SGF)

- La partie du SE qui s'occupe des fichiers

Ubiquitaire et requis

- L'utilisateur (ou le logiciel) veut enregistrer des données
→ Il doit utiliser un fichier

Questions

- Y a-t-il des alternatives aux fichiers pour stocker des données ?
- Est-ce que le concept de fichier a tendance à être moins important de nos jours ?

Les fichiers pour l'utilisateur

Besoins de l'utilisateur (et des logiciels)

- Nombreux (plusieurs millions, voire milliards)
- Contenu défini par l'utilisateur
- Fichiers nommés (plutôt que numérotés)
- Organisés pour les retrouver facilement
- Notion de propriétaire et droits d'accès
- Indépendants du matériel

Les fichiers dans INF3173

Nombreux points de vu : utilisateur, programme, bibliothèque, processus, noyau, contrôleur, périphérique, format de système de fichiers, etc.

Niveau **utilisateur**

- Les fichiers que l'humain « voit » et manipule sur le disque
- Inclut aussi le niveau programmeur et processus

Niveau **disque**

- Matériel : ce qui est physiquement stocké (ou simulé)
- Persistant : existe même quand l'ordinateur est éteint
- Inclue aussi tout ce qui est format et type de système de fichiers

Niveau **noyau** du système d'exploitation

- Ce qui est nécessaire à la gestion globale des fichiers

Terme ambigu : fichier

On va essayer d'être rigoureux. Termes définis dans la suite...

- **Inode** (ou juste *fichier*) : utilisateur, noyau et disque
Données réellement sur le disque* (données et métadonnées)
- **Entrée** (ou *dentry*) : utilisateur, noyau et disque
Un nom de fichier dans un répertoire
- **Chemin** : utilisateur et noyau
Chaîne de caractères qui désigne un fichier (ou pas)
- **Fichier ouvert** (le nom est pas super) : noyau
Un fichier* en cours de lecture et/ou écriture (par le noyau)
- **Descripteur de fichier** : utilisateur et noyau
Numéro (par processus) qui désigne un fichier ouvert du noyau
- **Flux** (*stream*) : utilisateur
Structure programmatique désignant un fichier ouvert*
(FILE*, fstream, InputStream, etc.)

*Ou un machin proche.

« Tout est fichier »

Philosophie importante Unix

- Pseudo-systèmes de fichiers, comme `proc(5)`
- Périphériques vus comme des fichiers spéciaux (on y reviendra)
- Descripteurs de fichiers pour ce qui peut être lu et écrit :
 - Tubes, sockets, etc. (on y reviendra)
 - Mais aussi pour de l'évènementiel :
`eventfd(2)`, `signalfd(2)`, `inotify(7)`, etc.

Avantages

- De nombreuses combinaisons : ex. entrée standard
- Réutilisation d'appels système : ex. `read(2)`, `write(2)`
- Réutilisation de politiques : ex. chemins et droits des fichiers



Racines

- Unix: la racine s'appelle / (slash) et elle est unique
- Windows: plusieurs racines possibles (C:, etc.)

Chemins

- Absolus : commencent par un / et partent de la racine
- Relatif : partent du répertoire courant du processus
Et non du répertoire où est stocké le binaire, etc.

Répertoire courant

- Un par processus
`pthread(7)` partagent, `fork(2)` hérite, `execve(2)` préserve
- `chdir(2)` et `getcwd(3)`
- **Question** Pourquoi `cd` est une commande interne du shell ?

Résolution de chemins

- Partir d'une chaîne de caractère
- Trouver un fichier
- En étant le plus performant possible

Pas si facile

- Trouver le répertoire de départ (racine, répertoire courant, etc.)
- Se promener (droits, liens symboliques, points de montages, etc.)
- Trouver et valider le dernier élément
- `path_resolution(7)`
- On y reviendra...

Systèmes de fichiers

Organisation

- **Système de fichiers** = ensemble **autonome** de fichiers
 - Chaque système de fichiers est **indépendant** et **cohérent**
- Mais fait partie d'un grand tout : la hiérarchie des fichiers

Caractéristique d'un système de fichiers

- Le **périphérique: emplacement** où sont stockées les données
- Le **type: format** de stockage des données

Note: un pseudo système de fichiers comme `proc(5)` n'a pas de périphérique associé

Type de système de fichiers

- Il s'agit du **format** utilisé pour représenter un système de fichiers
- FAT32, NTFS, HFS+, [ext4\(5\)](#), [btrfs\(5\)](#), [xfs\(5\)](#), ZFS...
- Attention : dans la plupart des contextes, « type » est implicite. Ne pas confondre un « système de fichiers » et « type de systèmes de fichiers »
- Chaque système d'exploitation peut supporter différents types [/proc/filesystems](#) donne une liste sous Linux

Contenu d'un système de fichiers

- Espace de donnée : les données des fichiers
 - Espace de gestion : les métadonnées des fichiers, leur organisation et celle de l'espace libre
- C'est habituellement persistant
- Les détails dépendent **grandement** du type du système de fichiers

Périphérique

- Les **systèmes de fichiers** résident (habituellement) sur des **périphériques** (*device*)
- Exemple: disques, partitions, etc. (des fichiers de type bloc)
- `lsblk(8)`, `blkid(8)`

Abus de langage

- « Périphérique » est utilisé de façon libérale et ne correspond pas forcément à un dispositif physique distinct
- « Partition » s'utilise parfois à la place de « périphérique » (qui n'est pas forcément une vraie partition),
Voire désigne le système de fichiers qui y est stocké

Montage et démontage

- **Point de montage**: répertoire où est accroché un système de fichiers
- Pour monter: `mount(8)`, `mount(2)`
- Pour démonter: `umount(8)` et `umount(2)`
- Pour voir l'arborescence: `findmnt(8)`

Question

- Pourquoi c'est des commandes de l'administrateur ?