

# 310 Manipulation des fichiers Unix

INF3173

Principes des systèmes d'exploitation

Jean Privat

Université du Québec à Montréal

Hiver 2021



## Chemins vs. descripteurs

- **Chemin** désigne un fichier par un emplacement
- **Descripteur** désigne un fichier ouvert (on y reviendra...)

## Appels système

- `open(2)` (et `creat(2)`) : prennent un **chemin** et donnent un **descripteur**
- `read(2)`, `write(2)`, `close(2)` : manipulent le **descripteur**
- D'autres opérations utilisent un **chemin**

Supprimer (`unlink(2)`), renommer (`rename(2)`), état (`stat(2)`), exécuter (`execve(2)`), etc.

- Opérations  $\pm$  uniformes quelque soit le système de fichiers
- Les détails internes ne sont pas exposés

# Variations d'appels système de fichiers

## Appels système « f\* »

- Travaille sur un fichier déjà ouvert au lieu d'un chemin  
Exemple `fstat(2)`
- Attention: ne pas confondre avec les fonctions de `stdio(3)`  
Comme `fopen(3)` ou `fread(3)`

## Appels système « \*at »



- Prend un descripteur + un chemin + des flags  
Exemple `fstatat(2)` ou `openat(2)`
- Variation qui généralise les autres
- Permet de partir d'ailleurs que du répertoire courant
- Évite des **situations de compétition** (*race condition*)
- Les flags permettent du comportement spécial



- Modèles de **programmation** spécifiques
- Fonctionnalités offertes par le système d'exploitation

## Multiplexage

- Boucle événementielle : un seul point bloquant en général
  - `select(2)`, `poll(2)` (et `epoll(7)` sous Linux)
- Très utilisé

## Non-bloquant

- Rien ne bloque
  - `O_NONBLOCK` (`open(2)`...) : les accès au fichier sont non bloquants
  - `EWOULDBLOCK` ou `EAGAIN` retourné ensuite au lieu de bloquer
- Besoins très spécifiques
- **Bloquer** c'est la bonne chose par défaut



## Une entrée = un fichier

- numéro d'inode (inœud ou numéro d'index)
- type de l'inode (fichier standard, répertoire...)
- propriétaire (uid, gid)
- droits (utilisateur, groupe, autre)
- taille du fichier en octets
- dates (plusieurs sortes)
- nombre de liens durs
- pointeurs vers blocs de données

## Question

- Il manque un truc, non ?

# Table des inodes

## Stockage

- Dans l'espace de gestion d'un système de fichiers
- Une table par périphérique
- Le détail du contenu et de l'implémentation dépend du type du système de fichiers
- Une copie (partielle) en mémoire du noyau (cache)

## Accès

- `ls(1)` (avec options `-il`) et `stat(1)`
- `stat(2)`, `lstat(2)` (et `xstat(2)` sous Linux)
- `inode(7)`

## Plus de métadonnées

- Attributs étendus: `xattr(7)`



# Types de fichiers Unix

## Fichiers réguliers

- Textes, exécutables, code source, images...
- Contenu décidé par l'utilisateur

## Fichiers spéciaux

- Répertoires, fichiers physiques (dans `/dev`), liens symboliques, tubes nommés, etc.
- Manipulation par des appels système spécifiques
- Règles au cas par cas

# Répertoires

- Fichier spécial `d` (`S_IFDIR`)
- Représente la hiérarchie des fichiers
- `mkdir(2)` (création), `rmdir(2)` (suppression)
- On y reviendra...

# Liens symboliques

- Fichier spécial 1 (S\_IFLNK)
- Représente un autre fichier (via son chemin)
- `symlink(2)` et `ln -s` (création), `readlink(2)` (lecture)
- Documentation `symlink(7)`
- On y reviendra pas...

## Questions

- Quelle est la taille d'un lien symbolique ?
- Peut-on savoir si un fichier a des liens symboliques ?
- Un fichier lié doit-il exister ?
- Quels sont les droits pour suivre un lien symbolique ?
- Que faire en cas de cycle de liens symboliques ?

# Fichiers périphériques

- Fichiers spéciaux `c` et `b` (`S_IFCHR` et `S_ISBLK`)
- Traditionnellement dans `/dev` (*device*)
- Type caractère (`c`) envoie et/ou reçoit des séquences d'octets
- Type blocs (`b`) écrit et/ou lit dans un bloc d'octets
- Pas de taille : numéro majeur (le type de périphérique) et mineur (un périphérique spécifique)
- `mknod(2)` (création)

# Exemples de fichiers périphériques

- `/dev/nvme0n1` : le premier disque dur
- `/dev/tty1` : un terminal
- `/dev/input/mice` : les souris
- `/dev/null` : la poubelle

## Autres fichiers intéressants

- `/dev/zero`, `/dev/full`, `/dev/mem`, `/dev/kmem`, `/dev/random`,  
`/dev/urandom`, `/dev/tty`

## Questions

- Pour chacun des fichiers ci-dessus, bloc ou caractères ?
- Qu'est-ce que `/dev/stdout` (déroutez les liens symboliques) ?

# Tubes nommés et sockets

- Fichiers spéciaux `f` et `s` (`S_ISFIFO` et `S_ISSOCK`)
- Pour de la communication inter-processus
- Documentation `fifo(7)` et `unix(7)`
- On y reviendra...

# Dates (Unix)

## Trois types de dates

- mtime : date de dernière modification du fichier
- ctime : date de dernière modification des métadonnées (entrée dans la table des inodes)
- atime : date de dernier accès au fichier (lecture)

## Représentation

- Stockées en temps Unix  
Temps écoulé depuis le 1er janvier 1970 UTC
- En secondes ou en nanosecondes  
Ça dépend du type du système de fichiers
- `touch(1)`, `utime(2)`, `utimensat(2)`

## Questions

- Il manque pas une date ?
- Que se passe-t-il en 2038 ?