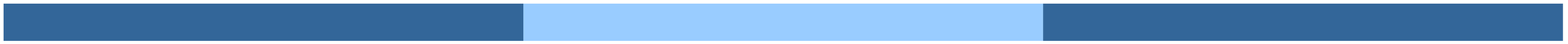
Systèmes de fichiers



DIRO

Menu

- Fichiers
- Répertoires
- Mount
- Partage et protection

Menu

- **Fichiers**
- Répertoires
- Mount
- Partage et protection

La notion de ficher

- Un **fichier** est une collection nommée d'informations connexes enregistrées sur le stockage secondaire
- Du point de vue de l'utilisateur, un fichier est le plus petit lot de stockage secondaire logique
- Types de fichiers:
  - Les données
    - numérique
    - caractère
    - binaire
  - Les programmes

Structure de fichier

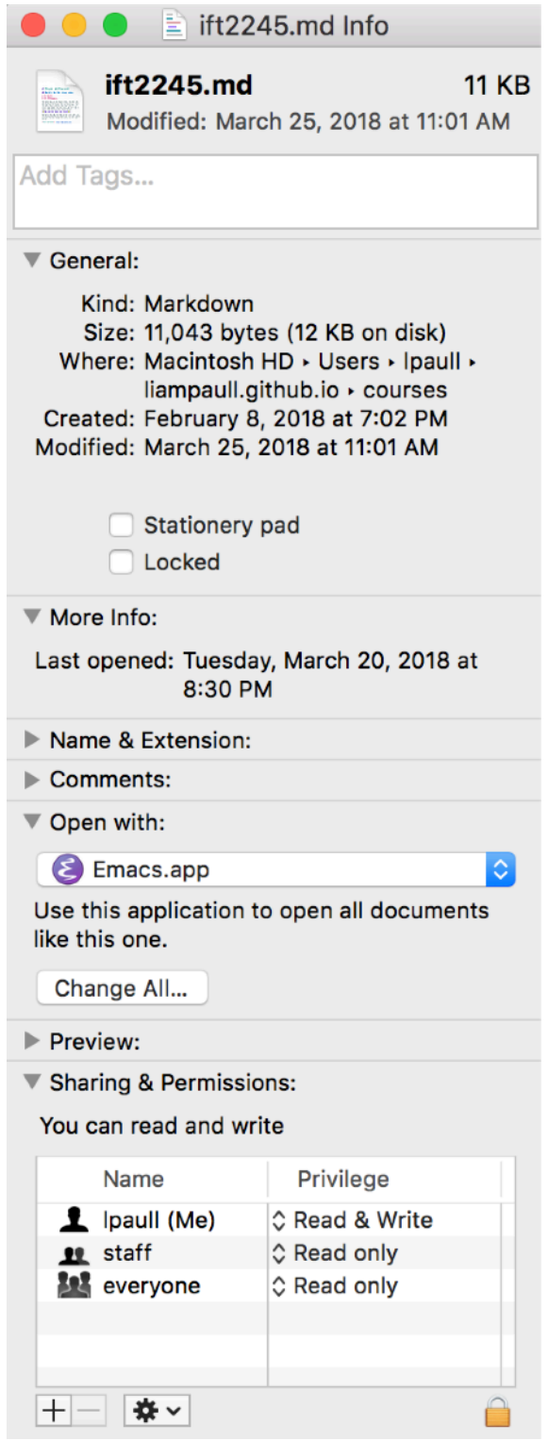
- Selon les systèmes, un fichier peut avoir différente structure
  - Aucune: un séquence de bytes
  - Une séquence d'enregistrements de taille fixe
  - Une séquence d'enregistrements de taille variable
- De nos jours, toujours le premier cas
- D'autres structure ajoutées par dessus (l'insu de SE) e.g. les bits parity

Attributs d'un fichier

- **Nom** - Seuls informations conservées sous une forme lisible par l'homme
- **Identifiant** - le tag (numéro) unique identifie le fichier dans le système de fichiers
- **Type** - nécessaire pour les systèmes prenant en charge différents types
- **Emplacement** - pointeur vers l'emplacement du fichier sur l'appareil
- **Taille** - taille actuelle du fichier (et taille sur disque = N*taille de bloc)
- **Protection** - contrôle qui peut lire, écrire, exécuter
- **Heure, date et identification de l'utilisateur** - données pour la protection, la sécurité et la surveillance de l'utilisation

- Les informations sur les fichiers sont conservées dans la structure du répertoire, qui est conservée sur le disque

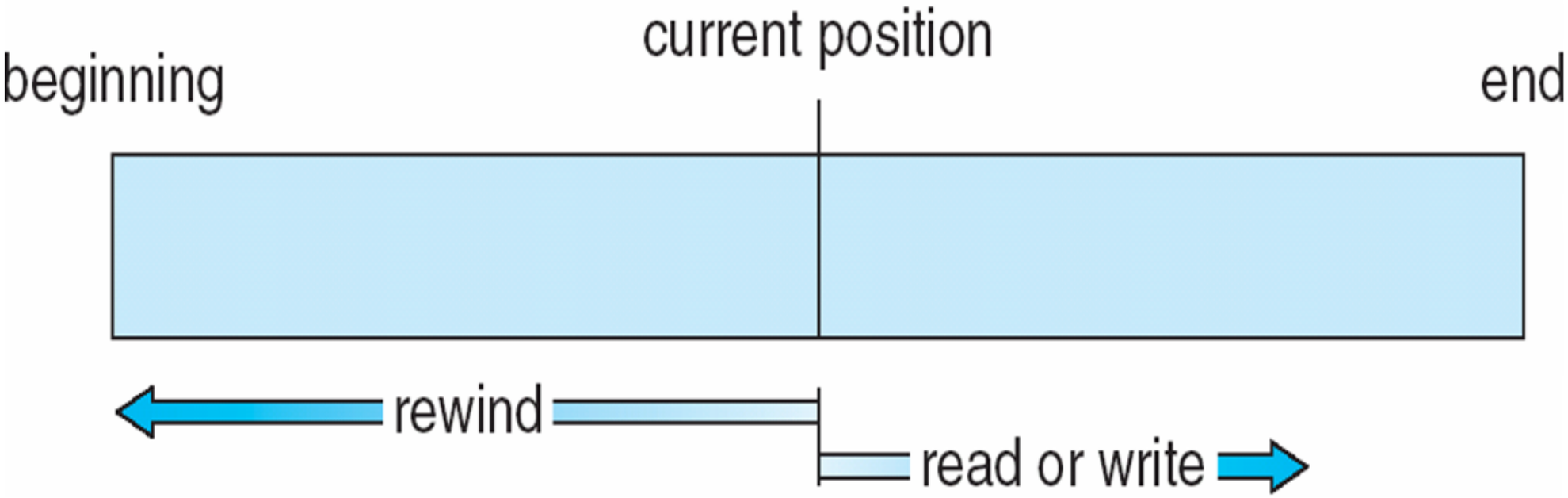
Attributs d'un fichier



Opérations sur fichier

- **Create:** généralement commence vide
  1. trouver de l'espace dans le système de fichiers
  2. faire une nouvelle entrée dans le répertoire
- **Open:** Trouver un fichier pour y opérer et le mettre dans le mémoire centrale
- **Read:** Lire un certains nombre de bytes d'un fichier
  - le **read pointer** indique l'emplacement dans le fichier où le lecture suivante doit avoir lieu
  - syscall: `read(pos,size)` (accès direct) ou `read(size)` (accès séquentielle)
- **Close:** Indiquer qu'on a finit d'opérer -
- **Write:** Écrire par dessus, ou étendre un fichier - déplacer le contenu en mémoire vers le disque (si cela a changé - rappelez-vous le dirty bit?)
  - le **write pointer** indique l'emplacement dans le fichier où l'écriture suivante doit avoir lieu
  - syscall: `write(pos,size,bptr)` (accès direct) ou `write(size,bptr)` (accès séquentielle)
- **Reposition within file:** les read and write pointers sont changer. Appelé une **seek**. syscall: `seek(pos)`.
- **Delete:**
  1. libère tout l'espace fichier
  2. effacer l'entrée du répertoire
- **Truncate:** Effacer une partie de la fin du ficher (ne doit pas le recréer)

Fichier d'accès séquentiel



Fichier d'accès directe

- utile pour un accès immédiat à de grandes quantités d’informations
  - e.g., des bases de données
- faire des accès séquentiels sur un système d'accès direct n'est pas très efficace

sequential access	implementation for direct access
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp* + 1;
write next	*write cp*; *cp* = *cp* + 1;

Pourquoi Open + Close

- Permet de séparer opérations coûteuses:
  - Recherche dans les répertoires
  - Trouver les méta-données du ficher
  - Verifications des droits d'accès
- Permet aussi de maintenir un **état** entre plusieurs opérations
  - Éviter d'effacer un ficher en cours d'usage
  - Garder un pointer sur la position courante (read et write)
  - Éviter l'accès concurrent (locking)

- Le SE maintient une **open file table** qui contient des informations sur tous les fichiers ouverts
- Quand une opération de fichier est demandée, le fichier est spécifié via un index dans cette table (pas besoin de chercher le répertoire)

Informations associées à un fichier ouvert

- Les **file pointers**: read et write
- **File-open count**: tenir un compte du nombre de processus qui ont ouvert le fichier
- **Emplacement du fichier sur le disque**: Les informations nécessaires pour localiser le fichier sur le disque sont conservées en mémoire afin que le système n'ait pas à le lire depuis le disque pour chaque opération
- **Des droits d'accès**:
  - NB: ceci doit être stocké pour chaque processus qui a le fichier ouvert

Synchronisation: file locking

- Deux approches:
  - Mandatory: accès interdit si on n'a pas le verrou
  - Advisory: Les verrous sont là, si vous voulez les utiliser

- shared lock = reader lock (plusieurs processus peuvent acquérir le verrou simultanément)
- exclusive lock = writer lock (un seul processus peut acquérir à la fois)
- Peut vérouiller:
  - le contenu d'un fichier
  - le nom d'un fichier (i.e. une entrée de répertoire)
  - une partie d'un ficher

Types de fichiers - Nom, extension

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

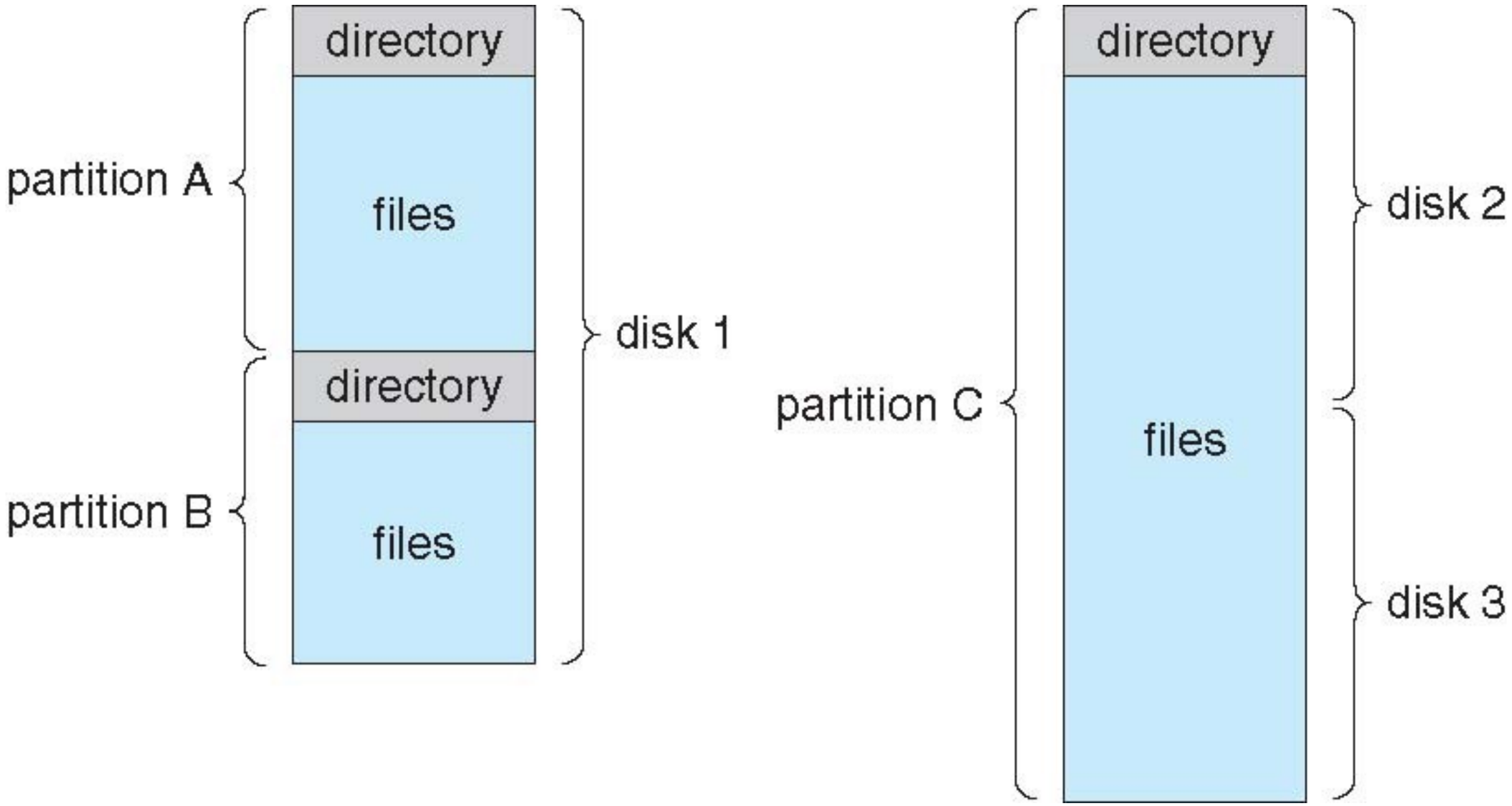
Menu

- Fichiers
- **Répertoires**
- Mount
- Partage et protection

Structure du disque

- Le disque peut être subdivisé en **partitions**
- Les disques ou partitions peuvent être protégés avec RAID
- Le disque ou la partition peut être utilisé **brut** - sans système de fichiers, ou **formaté** avec un système de fichiers
- Entité contenant un système de fichiers appelé **volume**
- Chaque volume contenant le système de fichiers suit également les informations de ce système de fichiers dans le **répertoire** (comme une table des matières du volume)

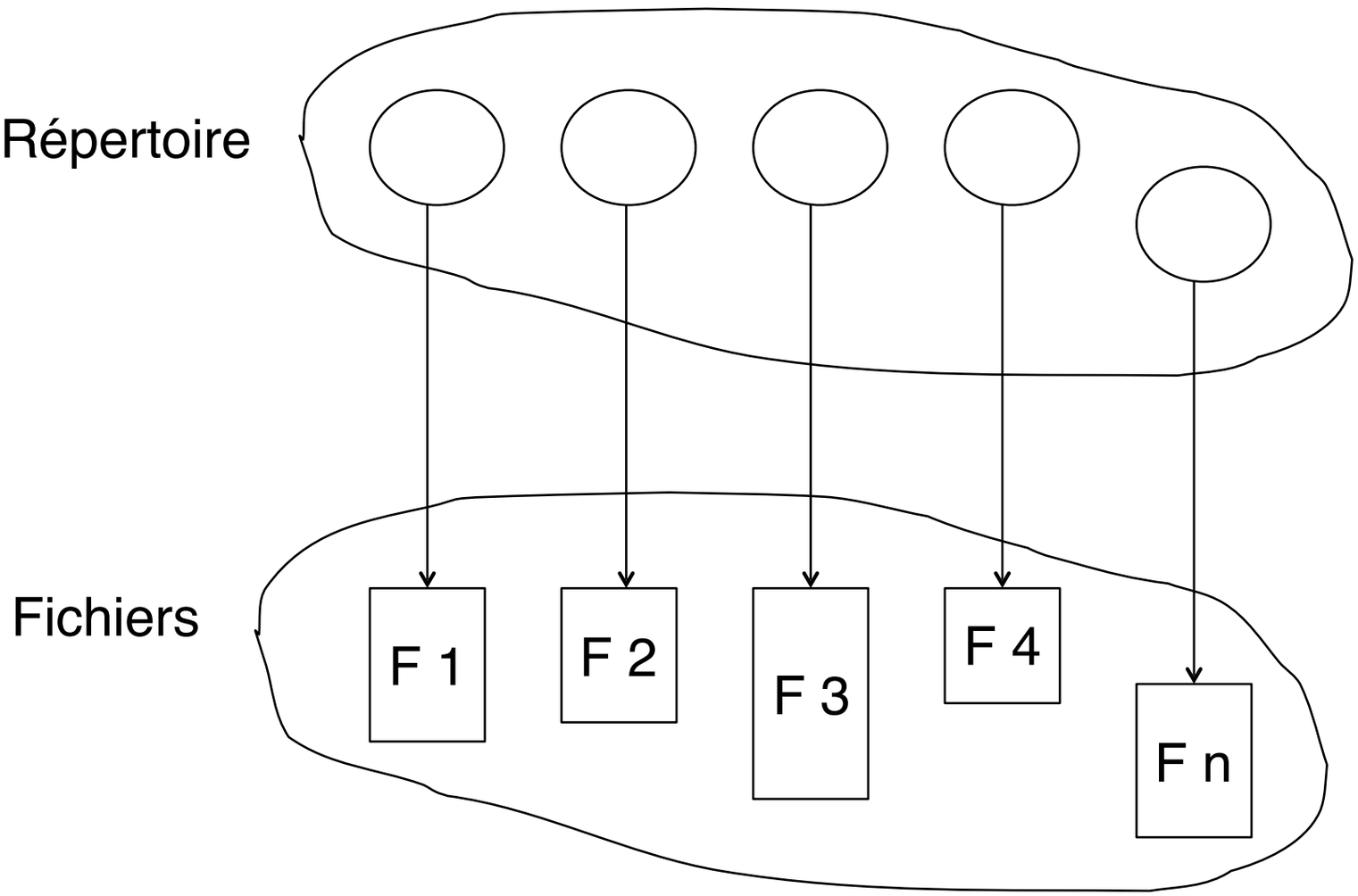
Organisation du système de fichiers



Un **volume** est une **partition** qui a été **formatée** avec un **système de fichiers**

Structure du répertoire

- Le répertoire est une table de traduction nom => fichier



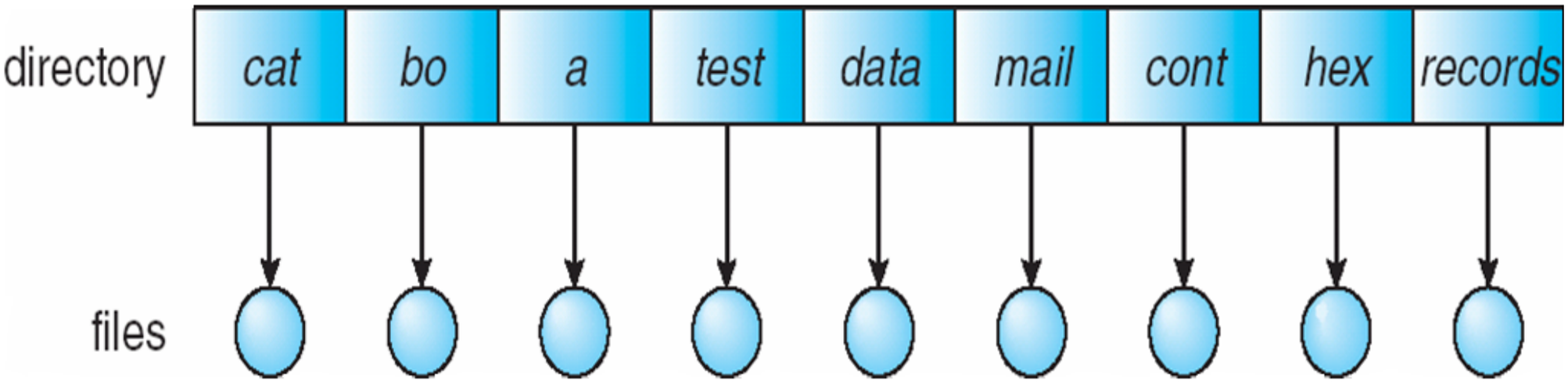
La structure du répertoire et les fichiers sont stockés sur le disque (un répertoire est un ficher!)

Opérations effectuées sur le répertoire

- **Search**: trouver un fichier ou un ensemble de fichiers (e.g. `ls *.tex`)
- **Create**: Lorsqu'un nouveau fichier est créé, une nouvelle entrée doit être ajoutée au répertoire
- **Delete**: Lorsqu'un fichier est supprimé, nous supprimons l'entrée du répertoire
- **List**: Enumérer les entrées de la table
- **Renommez un fichier**: Pour renommer un fichier, nous ne voulons pas créer un nouveau fichier
- **Traverse le système de fichiers**: par exemple pour trouver sa taille totale (e.g. `du -sh`)
- Généralement, un répertoire a une structure hiérarchique
  - **Make directory**: créer un nouveau sous-répertoire dans la hiérarchie (également appelé un dossier)

Répertoire à un niveau

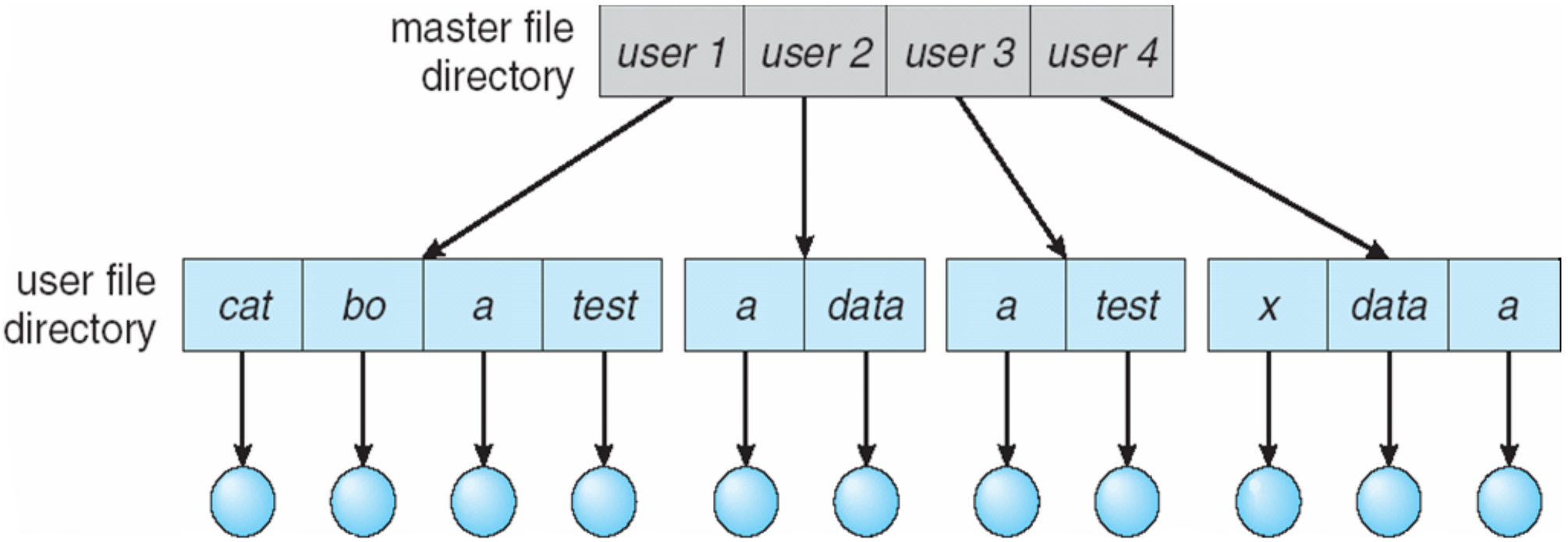
- Un répertoire unique pour tous les utilisateurs



problèmes de nommage et de regroupement

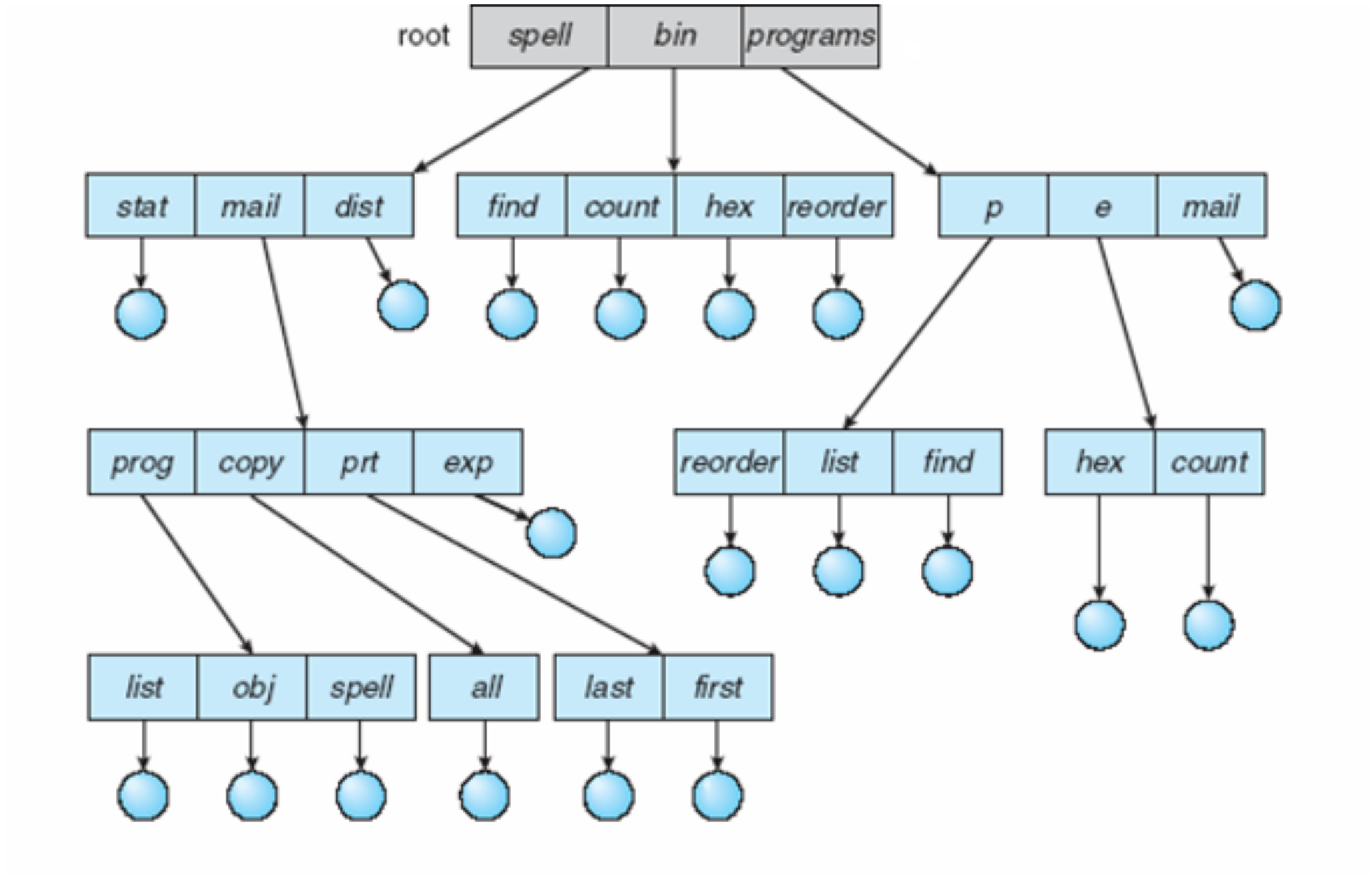
Répertoire à deux niveaux

- Séparer le répertoire pour chaque utilisateur



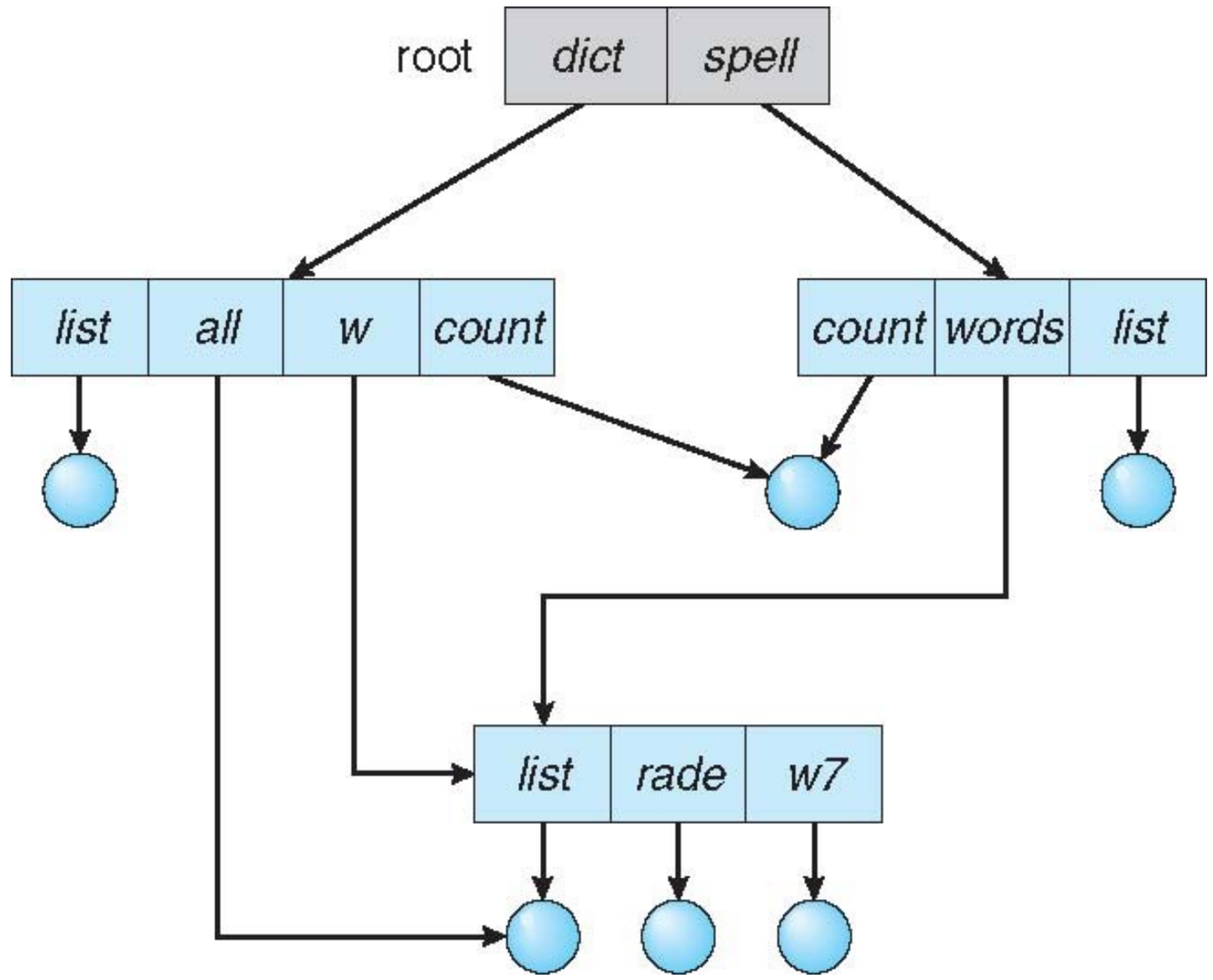
- le nom du fichier contient maintenant un **chemin**
- Peut avoir le même nom de fichier pour un utilisateur différent
- Impossible de grouper les fichiers de manière logique

Arbre de fichiers

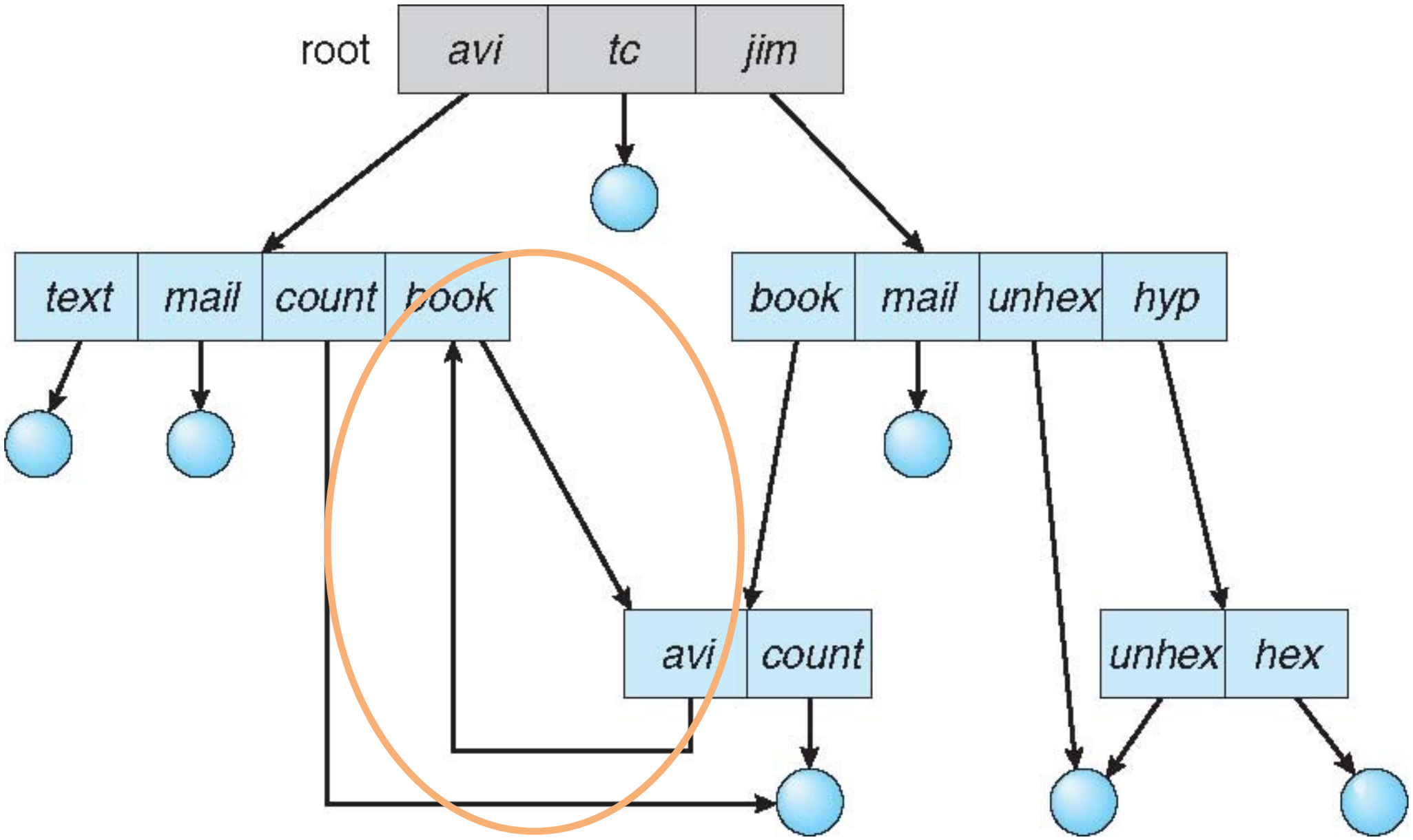


Graphe acyclique de répertoires

- Un fichier peut être référencé par plusieurs noms (e.g. avec `ln`)



Graphe générale de répertoires



Les liens

- Hard link: donner un autre nom à un fichier - maintenant le fichier a deux noms (aliasing)
- Symbolic link: ne contient pas les données dans le fichier cible seulement un pointeur

1. touch file1 2. ln file1 hardlink 3. rm file1 4. open hardlink fonctionne parce que le fichier existe toujours	1. touch file1 2. ln **-s** file1 symlink 3. rm file1 4. open symlink erreur car le fichier n'existe plus

- Peut pas créer un hard link vers un répertoire mais vous pouvez créer un symlink

Problèmes de graphes

- Problème 1: lors d'un DELETE, faut verifier si un objet devient inaccessible. En générale, c'est un propriété globale alors il faut chercher le disque en complet
- Solution 1: Chaque fichiers/répertoire à un compteur de référence

- Problème 2: Solution 1 ne fonctionnera pas dans le cas de cycle (nous pouvons avoir des auto-références)
- Solution 2: Interdire les cycles
  - il y a d'autres bonnes raisons d'éviter les cycles. Par exemple, la recherche dans le répertoire pourrait provoquer une boucle infinie

- Problème 3: Détecter un cycle en générale est très coûteux
- Solution 3: Une seule référence par répertoire

- Problème 4: Parfois nous voulons nous référer à un autre répertoire
- Solution 4: Liens symboliques

Menu

- Fichiers
- Répertoires
- **Mount**
- Partage et protection

Mount

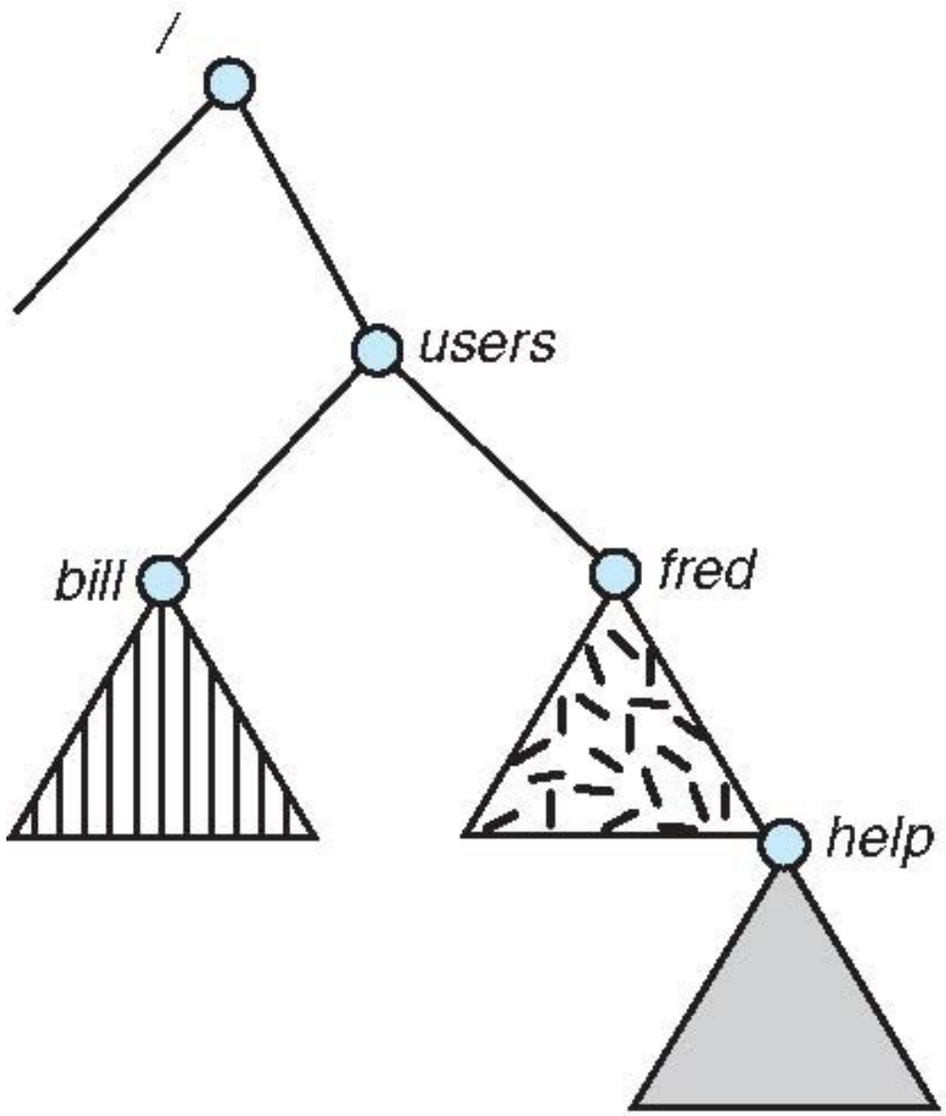
- Système de fichiers: arbre (graph) de fichiers sur un disque appliqué à un disque logique (e.g. partition)

- **Mount**: Placer un système de fichier dans un autre arbre pour le rendre accessible depuis un autre système de fichiers
  - Action purement en mémoire, pas d'effet sur le disque

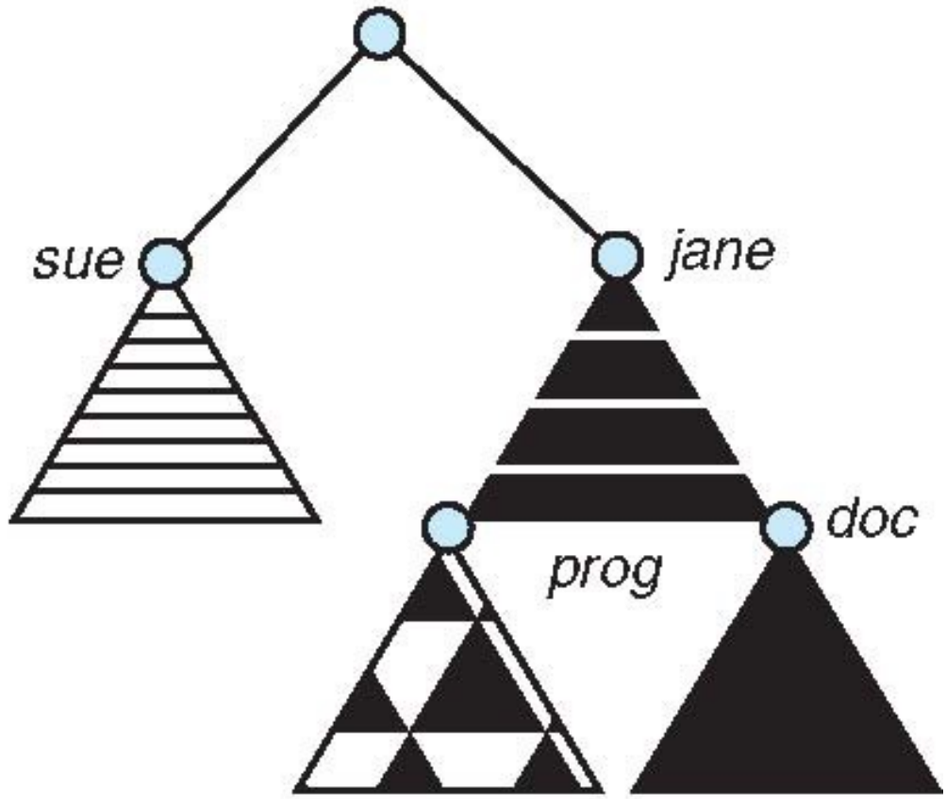
- La **mount table** associe **mount points** => système de fichiers

- Lors de l'ouverture d'un fichier:
  1. Consulter la mount table pour trouver le système de fichiers
  2. Consulter le système de fichiers (répertoire) pour trouver le fichier

Avant mount

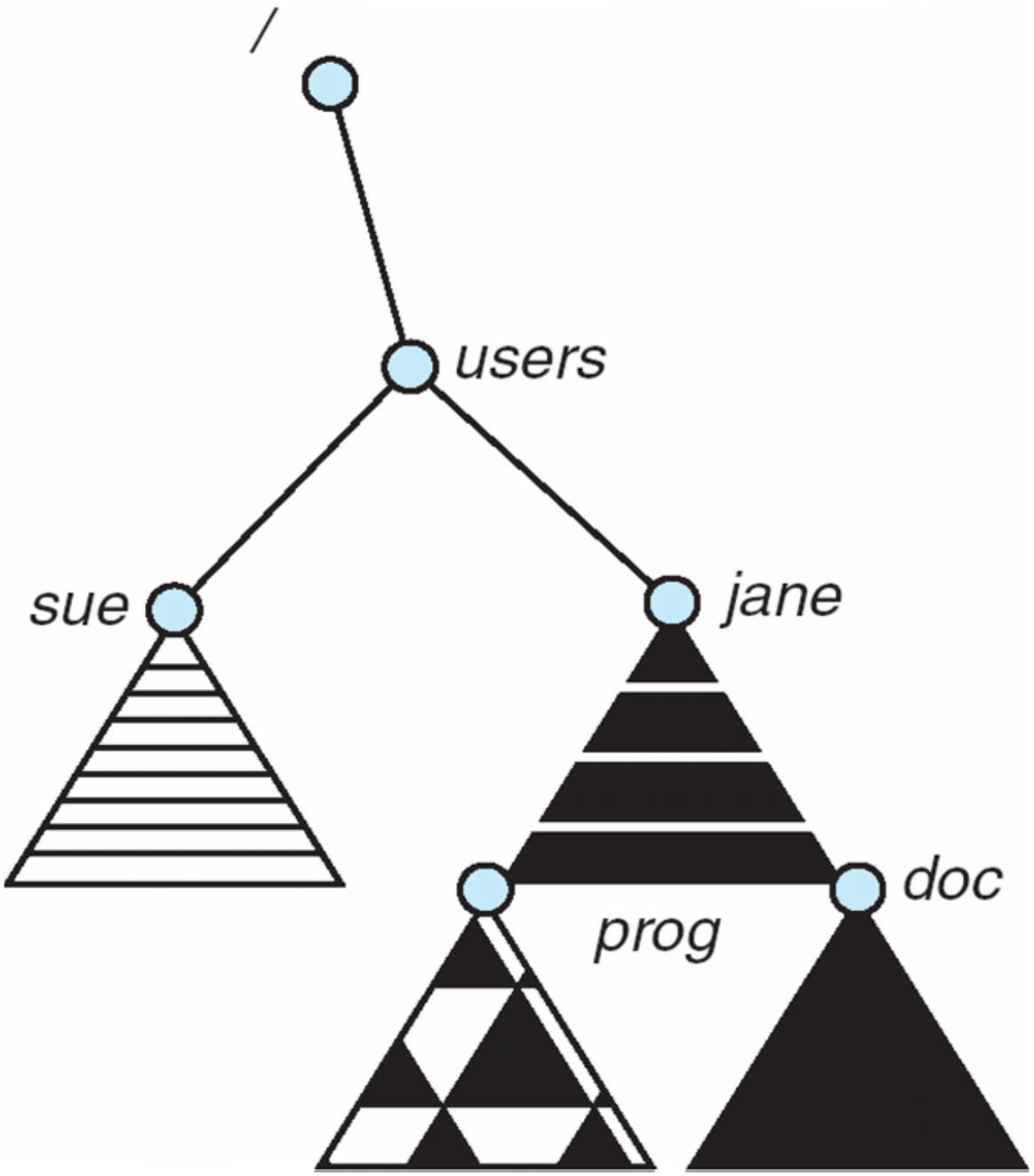


(a)



(b)

Après mount à /users



Menu

- Fichiers
- Répertoires
- Mount
- **Partage et protection**

Partager les fichiers

- Plupart de SE ont les concepts d'utilisateur (propriétaire) et group pour gérer le partage de fichiers entre utilisateurs
  - **User IDs** identifier les utilisateurs, en permettant aux autorisations et aux protections d'être par utilisateur
  - **Group IDs** permettre aux utilisateurs d'être dans des groupes, permettant des droits d'accès de groupe
- Les systèmes de fichiers distants permettent à un ordinateur de "mount" un système de fichiers à partir d'une machine distante (sur un réseau)
  - network file system NFS
  - difficile d'assurer une authentification correcte
- **Consistency semantics** spécifier comment plusieurs utilisateurs doivent accéder simultanément à un fichier partagé
  - Algorithme de synchronization

Protection

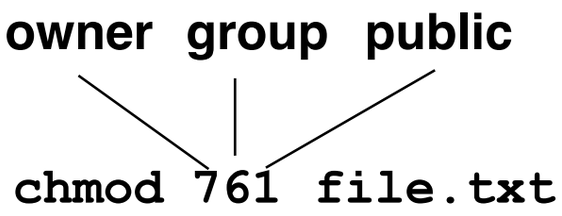
- Contrôle d’accès:
  - Certains utilisateur (user ID)
  - Certains groupes
- Types d’accès:
  - Lecture, écriture, exécution du fichier/répertiore
  - Ajout d’une référence dans un répertoire
  - Enlever référence
  - Changer les droits d’accèes

Permissions

- 3 bits: read, write, execute, pour 3 catégories = 9 bits totales

			RWX
a) **owner access**	7	⇒	1 1 1
b) **group access**	6	⇒	RWX 1 1 0
c) **public access**	1	⇒	RWX 0 0 1

- Un administrateur doit créer les utilisateurs et les groupes
- Pour un fichier, peut changer l'accès avec `chmod`



Associer un fichier à un groupe avec `chgrp`: `chgrp G game`

ls -al

```
-rw-rw-r--     1 pbg    staff      31200   Sep 3 08:30      intro.ps
drwx------     5 pbg    staff        512   Jul 8 09.33      private/
drwxrwxr-x     2 pbg    staff        512   Jul 8 09:35      doc/
drwxrwx---     2 pbg    student      512   Aug 3 14:13      student-proj/
-rw-r--r--     1 pbg    staff       9423   Feb 24 2003      program.c
-rwxr-xr-x     1 pbg    staff      20471   Feb 24 2003      program
drwx--x--x     4 pbg    faculty      512   Jul 31 10:31     lib/
drwx------     3 pbg    staff       1024   Aug 29 06:52     mail/
drwxrwxrwx     3 pbg    staff        512   Jul 8 09:35      test/
```

Sommaire

- Un fichier est une entité créée par le système d'exploitation sous la forme d'une séquence d'enregistrements logiques
- Le système d'exploitation mappe les fichiers logiques sur des périphériques de stockage physiques
- Chaque système de fichiers a un répertoire qui ressemble à une table des matières
- Ces répertoires peuvent prendre la forme d'arbres ou même de graphes
- Un système de fichiers doit être mounted pour devenir accessible
- Les permissions sur les fichiers sont utilisées pour le partage et la protection