Système de Fichiers

Le système de fichiers

Presque tous dans Unix est un fichier!

- Fichiers ordinaires
- Répertoires
  Les répertoires ne sont juste que des fichiers listant plusieurs fichiers
- Liens symboliques
  Fichiers faisant référence au nom d'un autre fichier
- Périphériques et dispositifs
  La lecture et l'écriture à partir d'un dispositif se fait comme un fichier
- Pipes
  Utiliser pour mettre en cascade plusieurs programmes

```
cat *.log | grep error
```

- Sockets
  Communication inter processus

Manipulation des fichiers

Stockage persistant, non volatile. Un système de fichiers est une structure de données permettant de stocker les informations et de les organiser dans des fichiers sur des mémoires secondaires.

Les fichiers sont gérés par le système d'exploitation. C.à.d:
La manière dont ils sont

- structurés
- nommés
- utilisés
- protégés

est à la charge du SE.

Manipulation des fichiers

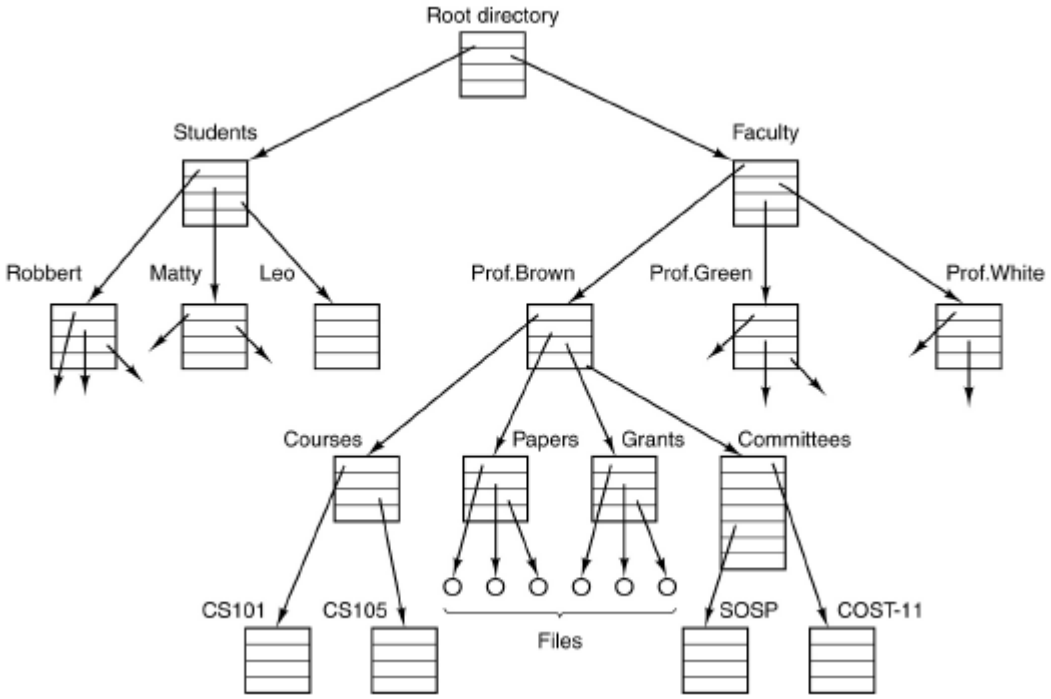
Depuis le début d'Unix, les noms de fichiers ont les caractéristiques suivantes:

- Sensibles aux majuscules / minuscules
- Pas de longueur limite évidente
- Peuvent contenir tous caractères (incluant l'espace, à l'exception de /).
  Les types de fichiers sont stockés dans un fichier (“nombre magique”).
  Les extensions d'un nom de fichier n'ont pas besoin et ne sont pas interprétés. Ils sont justes utilisés pour les utilisateurs .
- Exemples de noms de fichiers:

```
README          .bashrc         Windows Buglist
index.htm       index.html      index.html.old
```

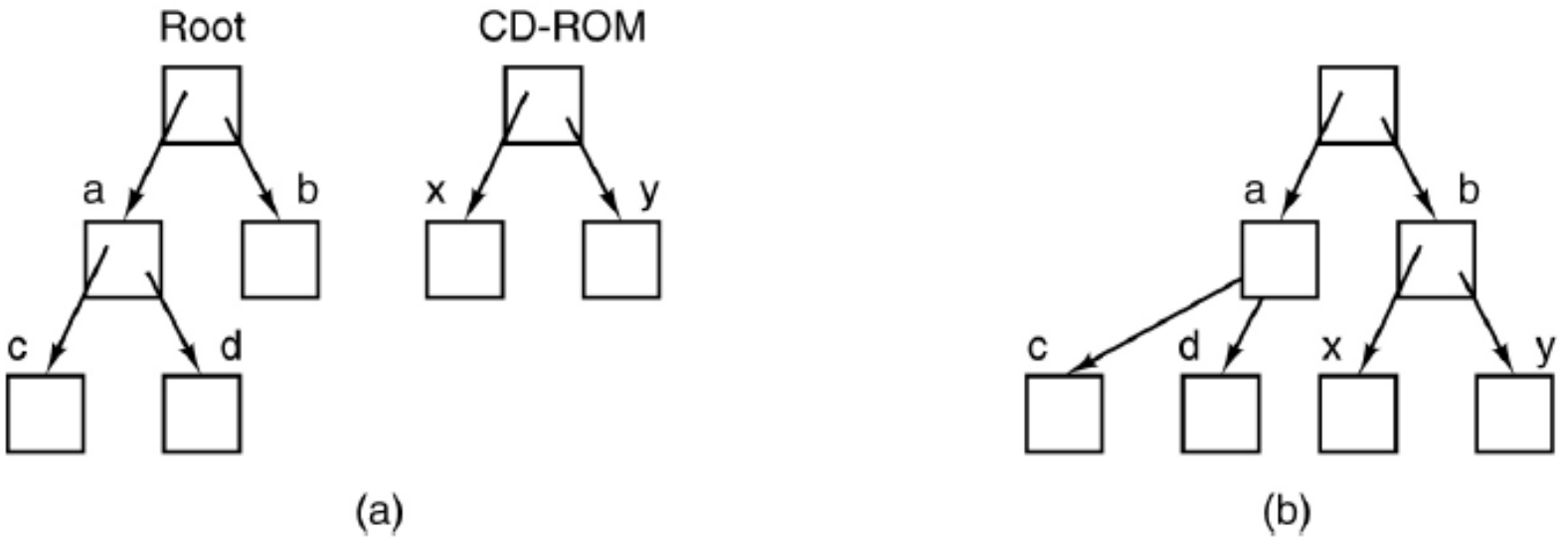
Manipulation des fichiers

- Le SE Linux masque les spécificités des disques.
- Offres des Appels systèmes : création, suppression, lecture écriture, ouverture, fermeture.
- Regroupe les fichiers en hiérarchie arborescente (répertoires + fichiers).
- Assure la protection des fichiers.



Manipulation des fichiers

- Montage d'un système de fichiers:
  - (a) Avant montage les fichiers du CD ne sont pas accessibles.
  - (b) Après montage, ils font partie de la hiérarchie des fichiers.



L’arborescence de fichiers

- Le système de fichier correspond à une arborescence que l’on parcourt de la racine (root) vers les feuilles
- La racine se note / (slash)
- Il s’agit d’un répertoire contenant les sous-répertoires suivants :

/bin exécutables essentiels pour le système, directement utilisable par les utilisateurs

/boot contient les fichiers permettant à Linux de démarrer

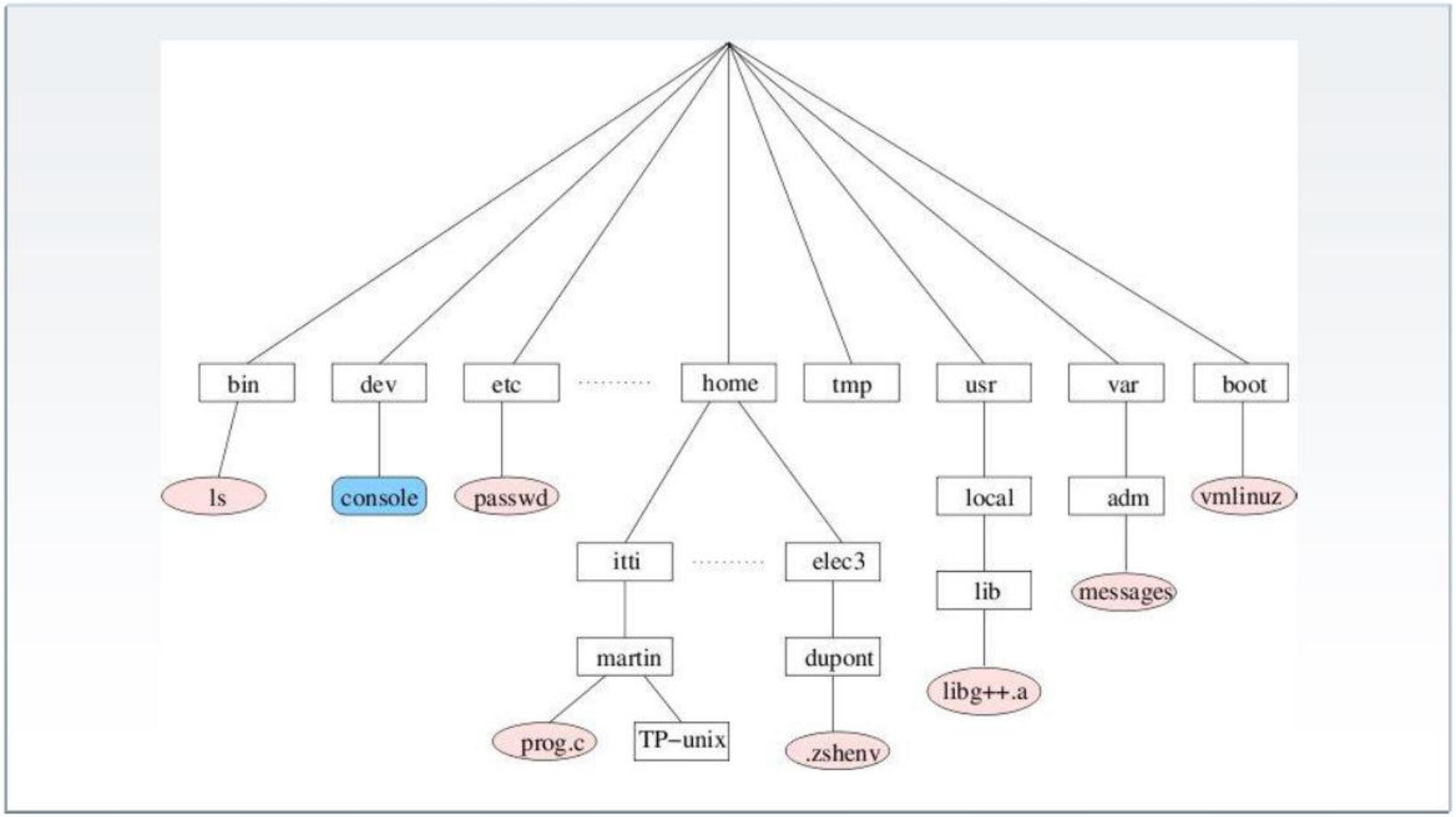
/dev contient les points d’entrée des périphériques (=device)

/etc configuration du réseau, contient les commandes et les fichiers nécessaires à l’administrateur du système (fichiers passwd, group, inittab, ld.so.conf, lilo.conf, ...)

L’arborescence de fichiers

Sous-répertoires de la racine (suite) :
/home répertoire personnel des utilisateurs
/lib contient des bibliothèques partagées essentielles au système lors du démarrage et pour les commandes dans /bin
/mnt contient les points de montage des partitions temporaires (cd-rom, disquette, ...), parfois nommé media
/opt contient des packages d’applications supplémentaires
/proc fichiers content des info sur la mémoire, E/S, périphérique, compatibilité pour le noyau, ...
/root répertoire de l’administrateur root
/usr hiérarchie secondaire (utilisateurs)
/var contient des données diverses (variables), telles que la boîte mail, des fichiers temporaires et des fichiers journaux
/tmp contient les fichiers temporaires
/usr/bin : commandes complémentaires de l'utilisateur
/usr/include : fichiers d'en-tête des langages

L’arborescence de fichiers



Chemins d'accès

Notation absolue

- /usr/include/sys/home/elec3/dupont

Notation relative

- prog.c, adm/messages
- ./lib, ../../elec3

Nom spéciaux

- /La racine
- .Le répertoire courant
- .. Le répertoire père
- ~Le répertoire utilisateur(home)

Les Commandes

L'interpréteur de commande

- Shell : interface entre l'utilisateur et le système d'exploitation ("coquille")
- Application (fichier exécutable) chargée d'interpréter les commandes des utilisateurs et de les transmettre au système
- Différents types de shell, les principaux étant :
  - **sh** (Bourne shell)
  - **bash** (Bourne again shell)
  - **csh** (C shell)
  - **Tcsh** (Tenex C shell)
  - **ksh** Korn shell
  - **zsh** Zero shell
- Le nom du shell correspond généralement au nom de l'exécutable :

% /bin/bash

Utilisation du shell

- Le shell correspond à une fenêtre présentant un prompt, encore appelé invite de commande. Celle-ci est paramétrable et par défaut en **bash** se compose comme suit :

login@machine$

(suffixe **$** → utilisateur normal, suffixe **#** → super utilisateur/ administrateur)

On saisit les commandes à la suite du prompt

- Pour stopper la commande en cours : **Ctrl-C**
- Pour mettre en attente la commande en cours : **Ctrl-Z**
- Pour terminer l'entrée standard (les éventuelles paramètres données par l'utilisateur via le clavier) : **Ctrl-D**

Syntaxe des commandes

nom_commande [options] [arguments]

Exemple:

commande ls
options :

- ls -l -a
- ls -la
- ls -l --color (option à plusieurs caractères)

arguments : fichier, expression

- grep toto monFichier
- tar -cv -f archive.tar MonRepertoire

Méta-caractères

- * : suite de caractères
- ? : un seul caractère
- [ ] : un des caractères dans les crochets
- un ensemble : [hg]
- un intervalle : [a-k]

On verra ça en détaille par la suite

ls *

afficher le contenu du répertoire courant

ls *.exe

afficher tous les fichiers se terminant par .exe

ls ????

afficher tous les fichiers dont le nom est composé de 4 caractères exactement

ls [ct]*

afficher tous les fichiers dont le nom commence par c ou par t

Aide

- **man** commande

obtenir le manuel d'une commande

- **info** commande

obtenir de l'aide (renvoie souvent à man)

- commande **--help**

afficher une aide succincte (aide mémoire) et liste les arguments qui peuvent être passés à commande

Manipulation des fichiers : chemin

- **pwd**

Afficher le répertoire courant

Exemple:

yannick@nausicaa:~/toto $ pwd

/home/yannick/toto

- **cd** [chemin]

Changer le répertoire courant, se déplacer dans l'arborescence

sans argument : retour au répertoire de connexion

Alias :

. : répertoire courant

.. : répertoire parent

Exemples :

$ pwd → /home/yannick/toto

$ cd .. → /home/yannick/

$ cd projet → /home/yannick/projet

$ cd /usr/local → /usr/local

Manipulation des fichiers : Listing

ls [option] [chemin]

Liste le contenu d'un répertoire avec plus ou moins de détails

Remarques:

- fichier : afficher description
- répertoire : afficher contenu

Exemples :

$ ls l* → liste tous les fichiers commençant par l

$ ls -l → liste tous les fichiers du répertoire courant, en donnant les attributs des fichiers (droits, taille, etc)

$ ls -a → liste tous les fichiers du répertoire courant (y compris les fichiers cachés dont le nom commence par un ".")

$ man ls → affiche la page de manuel de la commande ls

Manipulation des fichiers : Visualisation

cat [option] [chemin vers le fichier1, fichier2, etc]
affiche le contenu d'un fichier

Exemples :
$ cat .bash_profile → affiche le contenu du fichier caché **.bash_profile**
$ cat toto > tata → écrit le contenu du fichier toto dans un fichier nommé tata
(> on verra ça plutard)

more [fichier]
Visualiser le contenu d'un fichier page à page

less [fichier]
Visualiser le contenu d'un fichier dans un flux

Manipulation des fichiers : Edition

wc [option] [chemin vers le fichier]
Obtenir des statistiques sur le contenu d'un fichier (affiche le nombre de mots / lignes / caractères d'un Fichier)

Exemples :
$ wc -l toto → affiche le nombre de lignes du fichier toto
$ wc -c toto → affiche le nombre de caractères du fichier toto
$ ls | wc -l → affiche le nombre de fichiers dans le répertoire courant

Comment éditer un fichier ?
emacs [fichier]
vim [fichier]
gedit [fichier]
. . . (On verra ça par la suite)

Manipulation des fichiers : copie

- **cp [-ipr] source dest**
- **cp** [option] [chemin vers fichier source] [chemin vers fichier destination]

➢ **Source** = **fichier**

➢ **Dest** = **fichier :** copie un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier

➢ **Dest** = **répertoire :** recopier dans dest

options

❖ **-i :** confirmation en cas d'écrasement

❖ **-p** : préserve les attribus (propriétaire, groupe, date de création

❖ **-r :** copie récursive (pour les répertoires imbriqués)

Exemples :

$ cp toto /tmp/ → copie le fichier local toto dans /tmp (toujours nommé toto)

$ cp toto /tmp/tata → copie le fichier local toto dans /tmp en le nommant tata

$ cp -r projet /tmp → copie le contenu du répertoire projet dans le répertoire /tmp/projet

Manipulation des fichiers : Déplacement

mv [option] [chemin vers fichier source] [chemin vers fichier destination]

Déplace un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier

Exemples :

$ mv toto /tmp/ → déplace le fichier local toto dans /tmp (toujours nommé toto)

$ mv toto /tmp/tata → déplace le fichier local toto dans /tmp en le nommant tata

$ mv -i toto /tmp → déplace le fichier toto dans /tmp en prévenant l'utilisateur s'il existe déjà un fichier /tmp/toto

Manipulation des fichiers : Suppression

rm [option] [chemin vers fichier]

supprime un fichier

Exemples :

$ rm toto → supprime le fichier toto

$ rm -i toto → supprime le fichier toto en demandant confirmation à l'utilisateur

$ rm -f toto* → supprime les fichiers dont le nom commence par toto, sans demander confirmation à l'utilisateur

$ rm -r projet → efface récursivement le contenu du répertoire projet non vide

Manipulation des fichiers Créer / supprimer un répertoire

mkdir [chemin vers répertoire]

créer un répertoire

rmdir [chemin vers répertoire]

Supprimer un répertoire **vide**

Sécurité: ne fonctionne que quand les répertoires sont vides

Alternative: rm -r

Exemples :

$ mkdir toto → crée le répertoire toto

$ rmdir toto → supprime le répertoire vide toto

$ rmdir projet → rmdir: projet/: Directory not empty

Commandes diverses

- who

lister des utilisateurs connectés au système

- date

afficher date et heure

- file fichier

déterminer le type du fichier

- head [-n] fichier

afficher les n premiers lignes du fichier

- tail [+n|-n] fichier

  +n : afficher à partir de la ligne numéro n

  -n : afficher le n dernières lignes

- more fichier

afficher le fichier page par page

- sort fichier

Trier le contenu d'un fichier

- wc [-cwl] fichier

  -c : nombre de caractères

  -w : nombre de mots

  -l : nombre de lignes

Gestion de processus

- **Ps** [options]

Afficher les informations sur les processus en cours d'exécution

Exemple: **ps** ux

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME %COMMAND
yannick   6316  0.0  0.0  13272  1728 ?        SL   09:26   0:00 /bin/echo
```

- **top**

Afficher les processus les plus actifs en temps réel, donne des informations sur l'activité du système (ressources occupées, etc)

- **Kill** [option] PID

Envoyer un message à un processus donné, généralement pour y mettre fin

  - **signal SIGTERM (15)** par défaut : arrêter le processus proprement
  - **signal SIGKILL (9)** : terminer brutalement un processus