

Systeme de fichiers sous GNU/linux

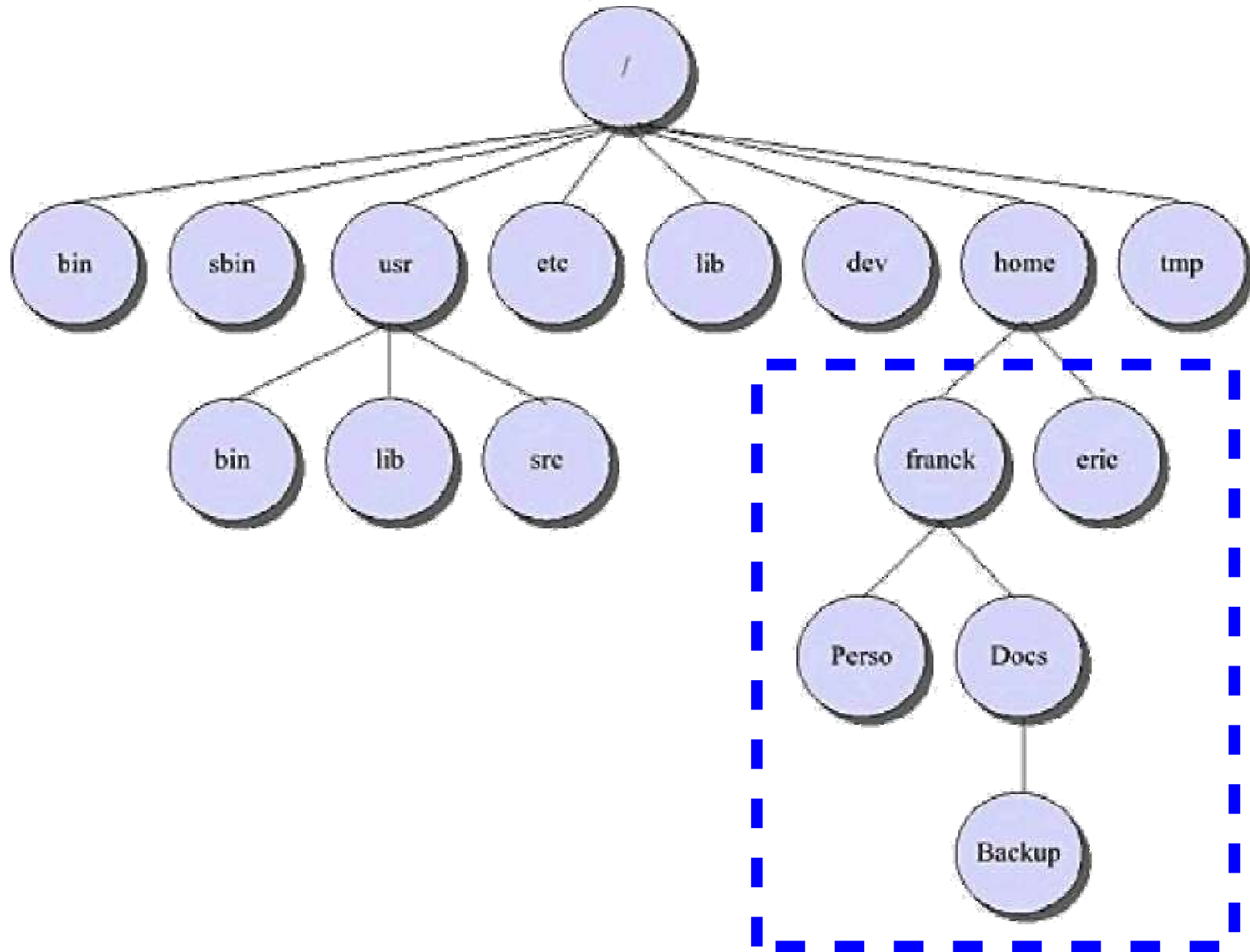
Systeme de fichiers

- File System ou FS, definit :
 - L'organisation des donnees sur un support de stockage.
 - Comment sont geres et organises les fichiers par le S.E.
- GNU/Linux est un S.E. entierement oriente fichiers. Tout est represente par des fichiers :
 - Donnees ;Peripheriques (terminaux, souris, clavier, carte son, etc.) ;Moyens de communication (sockets, tubes nommes, etc.)
- Le S.F. est hierarchique. Il decrit une arborescence de repertoires et de sous-repertoires, en partant d'un repertoire de base appele la **racine** ou **root directory**.

Systeme de fichiers GNU/Linux

- L'[extended file system](#) ou [ext](#), est le premier S.F. créé en avril 1992 spécifiquement pour GNU/Linux.
- 1993 : Version 2, [ext2](#), évolution de [ext](#).
- 2001 : Version 3, [ext3](#), ajoute la journalisation à [ext2](#). La journalisation améliore la récupération des données en cas d'arrêt brutal du système.
- 2008 : Version 4, [ext4](#), amélioration de [ext3](#).
- Futur : BtrFS (B-tree FS) : offre des améliorations en termes d'évolutivité, de fiabilité et de facilité de gestion.

Arborescence standard (1)



Arborescence standard (2)

- / : Répertoire racine
- /bin : Fichiers programmes exécutables binaires (commandes "essentielles").
 - /sbin : commandes pour l'administration du système.
 - /boot : Noyau du S.E. et les fichiers de démarrage.
 - /dev : Fichiers spéciaux qui servent comme canaux de communication avec les périphériques (disques, adaptateur réseau, cartes son, etc.)
 - /etc : Fichiers de configuration du S.E. et les principaux scripts de paramétrage.
 - /home : Répertoires personnels des utilisateurs.
 - /lib : Bibliothèques et modules du noyau.

Arborescence standard (3)

- `/mnt` : Points de montage des S.F. périphériques.
- `/media` : Points de montage des S.F. externes (CD, USB, etc.).
- `/opt` : Installation des applications supplémentaires.
- `/root` : Répertoire personnel du super-utilisateur root.
- `/tmp` : Stockage des fichiers temporaires.
- `/usr` : Programmes accessibles à tout utilisateur; sa structure reproduit celle de la racine /
- `/var` : Données variables liées à la machine (fichiers d'impression, traces de connexions http, smb .. dans `/var/log`)
- `/proc` : Contient une "image" du système (`/proc/kcore` est l'image de la RAM)

Types de fichiers

- Fichiers ordinaires ou réguliers :
 - Fichiers qui contiennent des données (texte, image, audio, etc.)
- Fichiers répertoires :
 - Un répertoire n'est rien d'autre qu'un fichier particulier contenant la liste des fichiers présents dans ce répertoire.
- Fichiers spéciaux :
 - Principalement des fichiers servant d'interface pour les divers périphériques. Se trouvent dans le répertoire /dev

Chemins

- permettent de définir et identifier un emplacement au sein du S.F.
- Chemin absolu représente l'arborescence complète à partir de la racine /
- Chemin relatif représente l'arborescence depuis le répertoire courant (répertoire de travail).
 - . Répertoire courant.
 - .. Répertoire père du répertoire courant.
- Un chemin relatif peut commencer par :
 - ~ Répertoire personnel de l'utilisateur (/home/user_login)

Nomenclature des fichiers

- On ne peut pas donner n'importe quel nom à un fichier.
- Longueur du nom jusqu'à 255 caractères (l'éventuelle extension comprise).
- GNU/Linux fait la distinction entre les minuscules et les majuscules.
- Quelques caractères sont à éviter car ils ont une signification particulière dans le Shell
- `& () ~ <espace> \ / | ` ? -` (en début de nom)

Commandes de base (1)

- Afficher le répertoire courant (print working directory)

pwd

- Changer de répertoire (change directory)

cd chemin

- Lister les fichiers et les répertoires

ls chemin

- Déterminer le type d'un fichier

file chemin

Commandes de base (2)

- Créer un fichier vide

touch chemin

- Créer un répertoire (make directory)

mkdir chemin

- Supprimer un fichier (remove)

rm chemin

- Supprimer un répertoire vide (remove directory)

rmdir chemin

Commandes de base (3)

- Copier un fichier (copy)

cp chemin_source chemin_destination

- Déplacer et/ou renommer un fichier (move)

mv chemin_source chemin_destination

- Remarques :

- La source est obligatoirement un fichier ordinaire.
- La destination peut être un fichier ordinaire ou un fichier répertoire.

Caractères de substitution (1)

- * : Remplace une chaîne de longueur variable, même vide.
- ? : Remplace un caractère unique quelconque.
- [...] : Une série ou une plage de caractères.
- [a-z] Un caractère parmi la plage indiquée (de a à z inclus).
- [!...] Inversion de la recherche.
- [^...] Inversion de la recherche.

Caractères de substitution (2)

- Exemples :
 - **ls a***
 - **ls a???**
 - **ls b??***
 - **ls *[12]**
 - **ls *(!3]**
 - **ls [A-Z]*[0-9]**
 - **ls [a-c]?*[12]**

Filtres

- Les filtres sont des commandes qui, à partir d'un flux d'entrées donné, effectuent des traitements avant d'afficher un résultat en sortie. On les nomme également commandes de **traitement de flux**.
- Le flux d'entrée est généralement un fichier. Il peut être aussi des caractères tapés à l'entrée standard (clavier).

cat

- Concaténer le contenu des fichiers et les afficher sur la sortie standard.

- **cat chemin**

- Exemples :

- **cat /etc/passwd**

- **cat /etc/group /etc/passwd**

- **tac /etc/passwd**

cut

- Afficher que certaines colonnes (champs) d'un fichier. Cette commande utilise comme options :
 - Le séparateur de champs. S'il n'est pas spécifié, le séparateur de champs par défaut est le caractère de tabulation.
 - Le numéro du champ à afficher pour chaque ligne.

• **cut -d séparateur -f numéros chemin**

- Exemples :
 - **cut -d : -f 3 /etc/passwd**
 - **cut -d : -f 1,3 /etc/passwd**
 - **cut -d : -f 1,3-7 /etc/passwd**

grep

- Afficher les lignes contenant une chaîne de caractères donnée.
 - **grep chaîne chemin**
- Exemples :
 - **grep etudiant /etc/passwd**
 - **grep -v etudiant /etc/passwd**
 - **grep ^root /etc/passwd**
 - **grep bash\$ /etc/passwd**

head

- Afficher les premières lignes d'un fichier.
 - **head -n nombre chemin**
 - **head -nombre chemin**
- Exemples
 - **head -n 4 /etc/passwd**
 - **head -4 /etc/passwd**
 - **head /etc/passwd**

tail

- Afficher les dernières lignes d'un fichier.
 - **tail -n nombre chemin**
 - **tail -nombre chemin**
- Exemples
 - **tail -n 6 /etc/passwd**
 - **tail -6 /etc/passwd**
 - **tail /etc/passwd**

sort

- Trier le contenu d'un fichier par ligne.
 - **sort chemin**
- Options
 - n : pour effectuer un tri numérique.
 - r : pour effectuer un tri décroissant.
- Exemples
 - **sort /etc/passwd**

WC

- Word count : Affiche des statistiques sur un fichier, nombre de caractères, nombre de mots et nombre de lignes.

- **wc chemin**

- Options
 - b : affiche le nombre de caractères.
 - w : affiche le nombre de mots.
 - l : affiche le nombre de lignes.
- Exemples
 - **wc -l /etc/passwd**

Redirection

- GNU/Linux fonctionne avec trois types de flux de données :
 - Entrée standard identifiée par le descripteur 0 (clavier).
 - Sortie standard identifiée par le descripteur 1 (écran),
 - Sortie d'erreur standard identifiée par le descripteur 2 (écran).
- La redirection permet de changer l'entrée, la sortie ou la sortie d'erreur d'une commande.

Redirection de sortie

- Le caractère **>** est utilisé pour changer la sortie standard.

- **commande > chemin**

- Si le fichier de redirection n'existe pas encore il sera créé. S'il existe, son contenu sera écrasé.

- **commande >> chemin**

- Si le fichier de redirection n'existe pas encore il sera créé. S'il existe, la sortie de la commande est ajoutée au contenu du fichier (sans écrasement).

Redirection de l'entrée

- Le caractère **<** permet de spécifier une autre entrée que l'entrée standard.
 - **commande < chemin**
- Les caractères « **<<** » permettent de lire l'entrée jusqu'à ce que la commande rencontre une certaine chaîne de caractères donnée.
 - **commande << chaîne**

Redirection de la sortie d'erreur

- On utilise le descripteur 2 de la sortie standard.
 - **commande 2> chemin**
- Les caractères `>&` permettent de rediriger la sortie erreur et la sortie standard.
 - **commande > chemin 2>&1**
 - **commande >& chemin**

Exemples de redirection

- **echo 'Bonjour' > /home/etudiant/message**
- **ls >> liste.txt**
- **cat /etc/passwd > ~/passwd**
- **gcc prog.c 2> erreurs.txt**
- **gcc prog.c > compilation 2>&1**
- **gcc prog.c >& compilation**

Tubes

- Le tube (pipe) permet de faire en sorte que la sortie d'une commande devienne l'entrée d'une autre. Les tubes utilisent le caractère |
 - **commande1 | commande2**
- Tubes et redirections peuvent être enchaînés indéfiniment sur une ligne de commande selon les résultats que l'on veut obtenir.

Exemples

- **ls -a | wc -l**
- **head /etc/passwd | tail -5 > sortie.txt**
- **cat /etc/passwd | cut -d: -f3 | sort -n**
- **cat /etc/passwd | cut -d: -f7 | sort | uniq | wc -l**