Fonctionnalités de Linux: Applications (3/3)

- Pratiquement tout programme Unix diffusé sous forme de source peut être compilé sous Linux et s'exécute parfaitement, grâce à la compatibilité implémentée dans le noyau et dans les librairies.

Schéma d'exploitation de la machine

- **shell** : interpréteur de commandes Unix (vérifie, interprète les commandes, exécute et renvoie les réponses). Le Shell envoie des appels au noyau en fonction des requêtes des utilisateurs.
- **noyau :** couche logicielle la plus interne du S.E dédiée à la gestion des composants matériels : processeur, mémoire, périphériques, …
- Autour du noyau gravite un certain nombre d'utilitaires.

SHELL

- **Shell:** est le programme interface entre l'utilisateur et le noyau Unix.
- Il est indépendant du noyau.
- Vous pouvez choisir le shell que vous voulez lors de vos sessions de travail.
- Par contre, un shell vous est toujours affecté par défaut pour toutes vos sessions lors de la création de votre compte
- Il a plusieurs rôles:
  - Il interprète les commandes passées par l'utilisateur afin qu'elles soient traitées par le noyau;
  - Il a aussi la fonction de langage de programmation,

Les commandes

Première connexion/Déconnexion (1/2)

- Pour pouvoir travailler sous Unix il faut ouvrir une session, à l'aide d'un nom d'utilisateur et d'un mot de passe.

- On distingue les administrateurs, des utilisateurs normaux.
  - **L'administrateur** est appelé **root** ou **utilisateur privilégié** et dispose de tous les pouvoirs sur la machine et le système Unix.
  - **L'utilisateur normal** dispose de droits réduits et définis par l'administrateur.

Première connexion/Déconnexion (2/2)

- Pour se connecter :

  Login : <tapez ici votre nom d'utilisateur>

  Password : <tapez ici votre mot de passe>

- Le mot de passe n'apparaît pas en clair et doit être tapé en aveugle.
- En cas d'erreur, un message indiquera :

  Login incorrect

Changer son mot de passe

- On utilise la commande **passwd** pour modifier son mot de passe.

$ passwd
Old password:
New password:
Reenter password:

- Sur les Unix récents, l'administrateur peut définir des règles de sécurité comme le nombre minimum de caractères, contrôler le mot de passe depuis un dictionnaire s'il est trop simple, lui donner une date de péremption, ….

Format général des commandes (1/2)

- Unix fonctionne en mode ligne de commandes et non en mode Graphique ➔ permet des opérations plus complexes.
- Une commande est un programme. Pour l'exécuter ➔ taper son nom éventuellement suivi d'options et d'arguments.
- Syntaxe :

```
nom_commande [-liste_options][liste_arguments]
```

- Les options sont le plus souvent précédées par un tiret «-».
- L'ordre des options est le plus souvent indifférent et peuvent être regroupées derrière un seul tiret.

Format général des commandes (2/2)

- Les arguments peuvent être absents et, dans ce cas, ils prennent des valeurs par défaut.

- **Exemple : ls -l ↵**
- Lors de l'appui sur la touche Entrée, le shell analyse la ligne de commande et l'interprète.

- Différence entre majuscules et minuscules ➔ On dit que la console Unix est sensible à la casse.

Quelques commandes

- Pour se familiariser avec la saisie de commandes, nous pouvons tester quelques programmes d'information :
  - **date** :
    affiche la date et l'heure
  - **who** :
    liste des utilisateurs connectés (who am i : qui suis-je)
  - **cal** :
    affiche le calendrier (cal 12 1975)
  - **man** :
    mode d'emploi (man cal)

Pour *interrompre* une commande on peut utiliser la combinaison **Ctrl+C.**

Pour *lier* les commandes, on sépare les commandes par le caractère « **;** »
Exemple **who ; date**

Pour *se déconnecter*, il suffit de taper:

exit

Astuces pour se déplacer sur la ligne de commandes (1/2)

- Il existe certains raccourcis clavier utiles pour se déplacer sur la ligne de commandes ou pour rappeler une commande précédemment lancée.
- Voici une liste succincte de ces raccourcis:
  - **Ctrl+A:** revenir en début de ligne.
  - **Ctrl +E:** aller à la fin de la ligne.
  - **Ctrl +F:** se déplacer vers la droite.
  - **Ctrl +B:** se déplacer vers la gauche.
  - **Ctrl +D:** effacer le caractère à droite du curseur.
  - **Backspace:** effacer le caractère à gauche du curseur.
  - **Ctrl +K**: supprimer la fin de la ligne.
  - **Ctrl +P:** revenir à la commande précédente.

Astuces pour se déplacer sur la ligne de commandes (2/2)

- **Ctrl N:** passer à la commande suivante (dans la liste des commandes).
- **Ctrl R + début de la commande:** retrouver une commande dans la liste.
- **Ctrl S:** interrompre la transmission des caractères entre le programme et l'écran.
- **Ctrl Q:** libérer tous les caractères depuis le Ctrl S précédent.
- **Ctrl C:** interrompre l'exécution d'un programme.

Une première commande : « echo» (1/2)

- En principe cette commande n'est pas utile tout de suite, mais la première chose que l'on apprend généralement avec un shell ou un langage quelconque est d'afficher un message du genre « **Hello, world !** ».

- La commande **echo** est une commande centrale du shell.
- Elle transmet tous ses paramètres sur écran (ou canal de sortie standard).

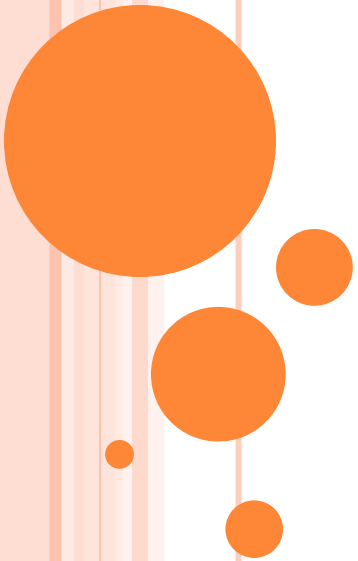
echo texte

- Le texte est quelconque mais peut aussi admettre quelques caractères de formatage.

Une première commande : « echo» (2/2)

Caractère	*Effet*
\n	Saut de ligne (newline)
\b	Retour arrière (backslash)
\t	Tabulation
\c	Pas de retour à la ligne (carriage)
\\	Affiche \
\$	Affiche $
\valeur	Affiche le caractère spécial de code octal valeur

- **$ echo "Bonjour\nComment ça va ?\c"**



Système de Fichiers

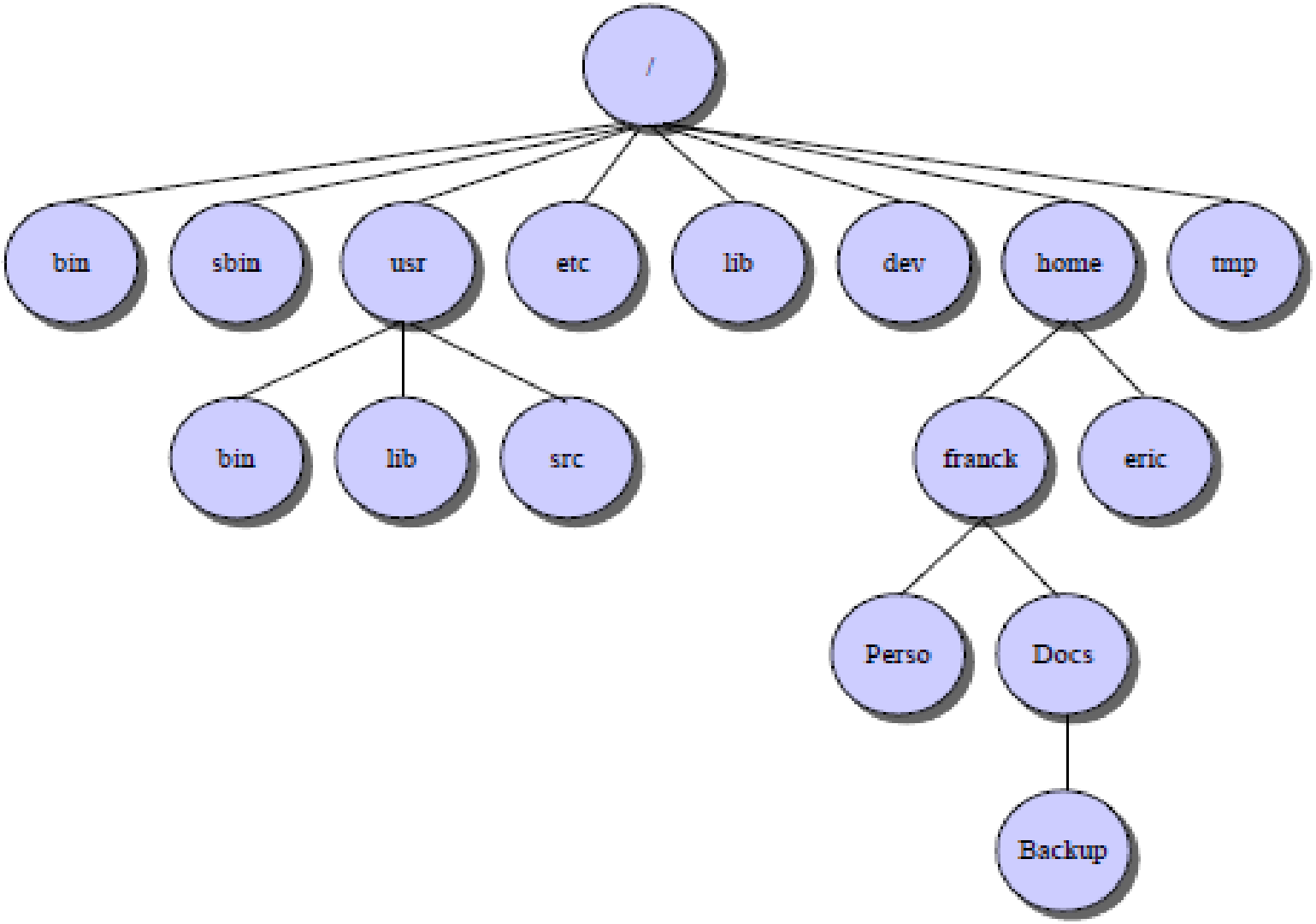
Le concept de fichier (1/2)

- L'unité logique de base de l'interface du Système de Gestion de Fichiers : le fichier.
- Un fichier Linux est une suite finie de bytes (octets) matérialisée par des blocs disques, et une inode qui contient les propriétés du fichier (mais pas son nom).
- Le contenu est entièrement défini par le créateur, la gestion de l'allocation des ressources nécessaires est la seule responsabilité du système.
- Les fichiers ne sont pas typés du point de vue utilisateur, le concept de fichier permet de proposer un type générique (polymorphe) aux programmeurs le système gérant la multiplicité des formats effectifs (différentes marques et conceptions de disques dur par exemple).

Le concept de fichier (2/2)

- L'inode définit le fichier, soit principalement les informations :
  - la localisation sur disque,
  - le propriétaire et le groupe propriétaire,
  - les droits d'accès des différents utilisateurs,
  - la taille,
  - la date de création.
- On trouvera sur d'autre systèmes d'autres structures d'information pour décrire les fichiers, par exemple NT utilise des "objets files records".
- *Un nom est lié à un fichier (une référence indique un fichier) mais un fichier n'est pas lié à une référence, un fichier peut exister sans avoir de nom dans l'arborescence*

La hiérarchie des fichiers (1/6)



La hiérarchie des fichiers (2/6)

- Pour assurer la compatibilité et la portabilité, les systèmes UNIX respectent la norme FHS (File Hierarchy Standard).
- La hiérarchie de base d'un système Unix est la suivante :

/				la racine, elle contient les répertoires principaux
	/bin			Contient les exécutables essentiels au système, employés par tous les utilisateurs.
	/boot			Contient les fichiers de chargement du noyau, dont le chargeur d'amorce.
	/dev			Contient les points d'entrée des périphériques.
	/etc			Contient les fichiers de configuration nécessaires à l'administation du système (fichiers *passwd, group, inittab, ld.so.conf, lilo.conf, ...*).

La hiérarchie des fichiers (3/6)

	/home			Contient les répertoires personnels des utilisateurs. Dans la mesure où les répertoires situés sous */home* sont destinés à accueillir les fichiers des utilisateurs du système, il est conseillé de dédier une partition spécifique au répertoire */boot* afin de limiter les dégats en cas de saturation de l'espace disque.
	/lib			Contient les bibliothèques standards partagées entre les différentes application du système.
	/mnt			Permet d'accueillir les points de montage des partitions temporaires (cd-rom, disquette, ...).

La hiérarchie des fichiers (4/6)

	/proc			Regroupe un ensemble de fichiers virtuels permettant d'obtenir des informations sur le système ou les processus en cours d'exécution.
	/root			Répertoire personnel de l'administrateur root. Le répertoire personnel de l'administrateur est situé à part des autres répertoires personnels car il se trouve sur la partition racine, afin de pouvoir être chargé au démarrage, avant le montage de la partition */home*.

	/sbin			Contient les exécutables système essentiels (par exemple la commande **adduser**).
	/tmp			contient les fichiers temporaires
	/usr			Hiérarchie secondaire

La hiérarchie des fichiers (5/6)

		/usr/bin		contient la majorité des fichiers binaires et commandes utilisateurs
		/usr/include		contient les fichiers d'en-tête pour les programmes C et C++
		/usr/lib		contient la plupart des bibliothèques partagées du système
		/usr/local		contient les données relatives aux programmes installés sur la machine locale par le root
			/usr/local/bin	Binaires des programmes locaux
			/usr/local/include	Fichiers d'en-tête C et C++ locaux
			/usr/local/lib	Bibliothèques partagées locales
			/usr/local/sbin	binaires système locaux
			/usr/local/share	hiérarchie indépendante
			/usr/local/src	Fichiers sources locaux

La hiérarchie des fichiers (6/6)

		/usr/sbin		contient les fichiers bianires non essentiels au système réservés à l'administrateur système
		/usr/share		réservé aux données non dépendantes de l'architecture
		/usr/src		contient des fichiers de code source

	/var			contient des données versatiles telles que les fichiers de bases de données, les fichiers journaux (logs), les fichiers du spouleur d'impression ou bien les mails en attente.

Les divers types de fichiers (1/4)

On distingue principalement trois types de fichiers: ordinaires, catalogue, spéciaux.

Fichiers ordinaires (ordinary files)

- Ce sont soit des fichiers **contenant du texte**, soit des **exécutables (ou binaires),** soit des fichiers de **données**.
- Par défaut, rien ne permet de différencier les uns des autres, sauf à utiliser quelques options de certaines commandes (**ls -F** par exemple) ou la commande **file.**

$ file nom_fic

nom fic : 32 Bits ELF Executable Binary

Les divers types de fichiers (2/4)

Catalogues (les répertoires ou directory)

- Les répertoires permettent d'organiser le disque dur en créant une hiérarchie.
- Un répertoire peut contenir des fichiers normaux, des fichiers spéciaux et d'autres répertoires, de manière récursive.

Fichiers spéciaux

- Ce sont le bien souvent des fichiers servant d'interface pour les divers périphériques.
- Ils peuvent s'utiliser, suivant le cas, comme des fichiers normaux.
- Un accès en lecture ou écriture sur ces fichiers est directement dirigé vers le périphérique (en passant par le pilote Unix associé s'il existe)

Les divers types de fichiers (3/4)

Nomenclature des fichiers

- On ne peut pas donner n'importe quel nom à un fichier,
- Il faut pour cela suivre quelques règles simples.
- Ces règles sont valables pour tous les types de fichiers.
- Sur les anciens systèmes un nom de fichier ne peut pas dépasser 14 caractères.
- Sur les systèmes récents, on peut aller jusqu'à 255 caractères.
- Il est possible d'utiliser des extensions de fichiers mais cela ne modifie en rien le comportement du système (un exécutable n'a pas besoin d'une extension particulière).

Les divers types de fichiers (4/4)

- Unix fait la distinction entre les minuscules et majuscules. Toto, TOTO, ToTo et toto sont des noms de fichiers différents.
- La plupart des caractères (chiffres, lettres, majuscules, minuscules, certains signes, caractères accentués) sont acceptés, y compris l'espace (très déconseillé).
- Cependant quelques caractères sont à éviter :

  & ; ( ) ~ <espace> \ | ` ? - (en début de nom)

- Quelques noms valides :
  - Fichier1
  - Paie.txt
  - 123traitement.sh
  - Paie_juin_2002.xls

Chemins

Structure et nom de chemin

- Les chemins permettent de se déplacer dans le FileSystem (FS).
- Un nom de fichier est ainsi généralement complété de son chemin d'accès.
- C'est ce qui fait que le fichier « toto » du répertoire « rep1 » est différent du fichier « toto » du répertoire « rep2 ».
- Le FS d'Unix étant hiérarchique, il décrit une arborescence.

Chemins

- Le schéma représente une arborescence d'un FS Unix.
- Le « **/** » situé tout en haut s'appelle la **racine** ou **root** directory (**à ne pas confondre avec le répertoire de l'utilisateur root**).
- Le nom de chemin ou **path name** d'un fichier est la concaténation, depuis la racine, de tous les répertoires qu'il est nécessaire de traverser pour y accéder, chacun étant séparé par le caractère « **/** ».

➔ **C'est un chemin absolu.**

/home/franck/Docs/Backup/fic.bak

Chemins

Chemin relatif

- Un nom de chemin peut aussi être relatif à sa position courante dans le répertoire.
- Le système (ou le shell) mémorise la position actuelle d'un utilisateur dans le système de fichier, le répertoire actif.
- On peut accéder à un autre répertoire de l'arborescence depuis l'emplacement actuel sans taper le chemin complet.
- Pour se déplacer dans les répertoires, on utilise la commande **cd**.
- Le « **..** » permet d'accéder au répertoire de niveau supérieur.
- Le « **.** » définit le répertoire actif (répertoire courant).

Répertoire

- La commande **ls** permet de lister le contenu du répertoire.
- La commande **pwd** (print working directory) affiche le chemin complet du répertoire actif.

```
$ cd /
$ ls
bin
sbin
usr
etc
lib
dev
home
tmp
$ cd /usr/lib
$ pwd
/usr/lib
$ cd ../bin
$ pwd
/usr/bin
$ cd ../../etc
$ pwd
/etc
```

Répertoire

Répertoire personnel

- Lors de la création d'un utilisateur, l'administrateur lui alloue un répertoire utilisateur.
- Après une connexion, l'utilisateur arrive directement dans ce répertoire, qui est son **répertoire personnel.**
- C'est dans ce répertoire que l'utilisateur pourra créer ses propres fichiers et répertoires.
- La commande **cd sans argument** permet de retourner directement dans son répertoire utilisateur.

Login : toto
Password :
$ cd
/home/toto

Répertoire

- La commande **ls** permet de lister le contenu d'un répertoire (catalogue) en lignes ou colonnes. Elle supporte plusieurs options.

Option	*Signification*
-l	Sortie de chaque information des fichiers
-F	Ajoute un « / » au nom d'un répertoire, un « * » au nom d'un exécutable, un « | » pour un tube nommé et « @ » pour un lien symbolique.
-a	Affiche toutes les entrées, y compris « . », « .. » et les fichiers cachés (qui commencent par un .)

Répertoire

-d	affiche le nom (et les attributs) des répertoires et pas leur contenu.
-i	Affiche les numéros d'inode.
-R	Mode récursif. Rentre dans les répertoires et affiche leur contenu.
-r	Inverse l'ordre du tri (à l'envers)
-t	Tri par date de modification
-c	Affiche la date de création (si -l) ou tri par date de création (si -t)
-C	Les noms sont affichés sur plusieurs colonnes
-u	Affiche la date d'accès (-l) ou tri par date d'accès (-t)
-1	Liste sur une seule colonne.