

Licence Professionnelle

Remise à Niveau

N°1 : Introduction

- Introduction
 - Identification
 - Système de fichiers
 - Filtres

Remise A Niveau

- Objectif : Être capable d'exploiter un poste sous Linux
 - Identification
 - Système de fichiers
 - Filtres

- Supports : <http://moodle.univ-lille1.fr> (LP CGIR - RAN *clé : lpcgir*)
 - *proxy* : *cache-etu.univ-lille1.fr:3128*
 - *inscription* : *http://portail.univ-lille1.fr*

Unix/Linux

- 1970 : Création UNIX (*DARPA*) 1994 : Linux (*Linus Torvalds*)
- Caractéristiques du système :
 - Multi-tâches : plusieurs exécutions en parallèles (ordonnanceur)
 - Multi-utilisateurs : plusieurs utilisateurs sur un même poste
 - Multi-plate-formes : portabilité sur plusieurs architectures
 - Complet, ouvert, gestion **dynamique** des pilotes de périphériques
 - Support des systèmes de fichiers Windows et Macintosh
- Conforme à POSIX, Distribution GPL (*utiliser, modifier, distribuer...*)
- Interfaces utilisateur :
 - Mode commande : **CLI** Command Line Interface => interpréteur
 - Mode graphique : **X-Window** *Gnome, Mate, KDE, XFCE...*

L'interpréteur

- L'interpréteur fournit :
 - une interprétation ligne à ligne
 - un environnement adaptable via des variables d'environnement
 - un langage de programmation (**shell-script**)
 - structures conditionnelles, boucles, fonctions...
- Plusieurs interpréteurs "shell" => **bash** : (Bourne Again SHell)
 - édition de la ligne de commande,
 - historique des commandes (↑, history, fc -l, fc -s),
 - complétion (*touche de **tabulation***)
 - pour compléter l'argument : fichiers du répertoire courant

Les commandes

- Commandes **internes** (*built-in*)
 - implantées au sein de l'interpréteur (*de l'exécutable bash*)
 - liées au shell utilisé, pas de création de processus
 - 2 catégories
 - commandes simples : `cd` , `echo` , ...
 - commandes composées : `if ... fi`, `while ... done`, ...
- Commandes **externes**
 - le code se trouve dans un fichier
 - accès fonction de la variable d'environnement `PATH` (`/bin:/usr/bin`)
 - création d'un processus spécifique pour leur exécution
- **type** *commande* : indique si la commande est interne ou externe
- **which** *commande* : localise le fichier associé à la commande **externe**

Le manuel de référence

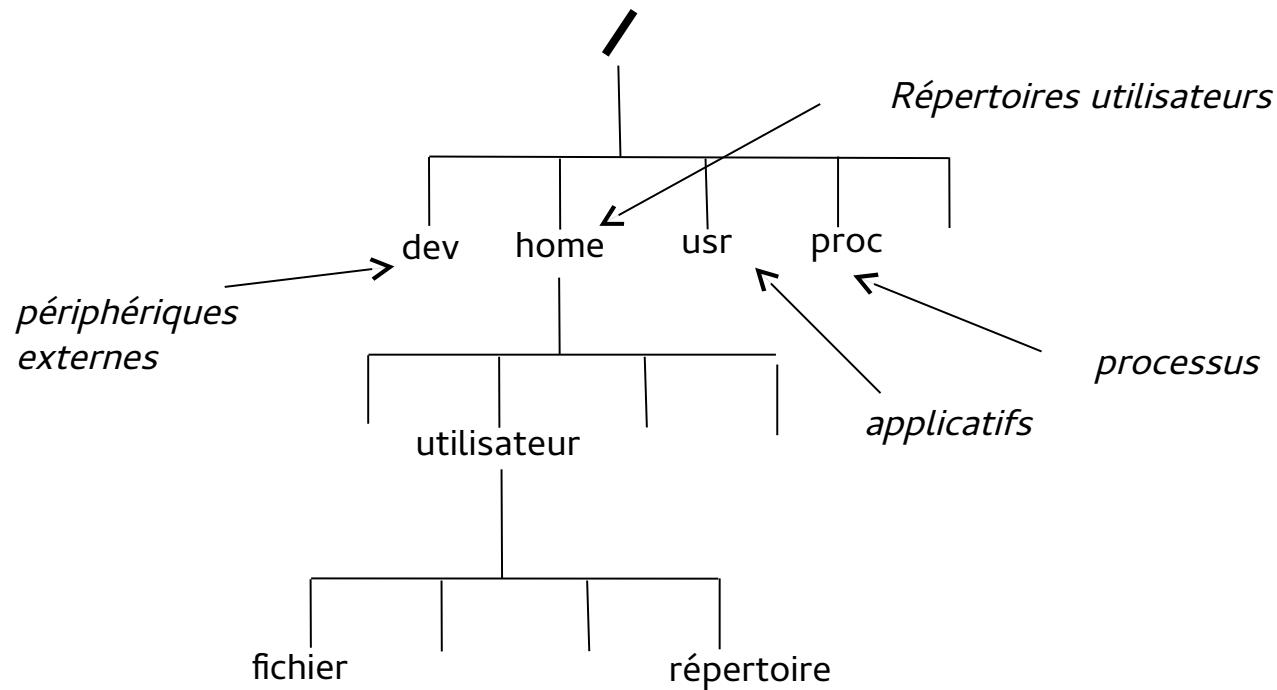
➤ Manuel : **man** :

- Les pages du manuel sont divisées en plusieurs **sections**
 - `man passwd` : manuel de la commande `passwd`
 - `man 5 passwd` : format du fichier `passwd`
- `man commande` informations sur les commandes ou fonctions disponibles
- `man man` informations sur l'usage du manuel et de ses sections
- `man bash` informations sur les commandes built-in du shell
- Parcours du manuel
 - **Espace** Avance d'une page entière
 - **Entrée** Avance d'une ligne
 - **b** Recule d'une page
 - **/mot** Recherche mot
 - **q** Quitte
- **info** `commande` informations sur la commande
- **help** : `commande` informations sur les commandes **internes**/intégrées
- **--help** : option pour les commandes externes

Le système de fichiers

- Les fichiers sont rangés dans une **structure arborescente**
- **Tout est fichier** sous UNIX, tout fichier se trouve dans l'**arborescence** unique
- 3 types de fichiers :
 - Les **répertoires** : permettent d'organiser l'arborescence (*liste de fichiers*)
 - Les **fichiers réguliers**/ordinaires : contiennent des données de tout type
 - Les **fichiers spéciaux** : désignent des éléments matériels, périphériques
- L'arborescence logique est construite par **montage** des partitions
 - une **partition** est une portion de disque (hda0, sdb1...) dotée d'un système de fichiers (une arborescence), *hormis la partition de swap*
 - **monter** une partition : « accrocher » la **racine** du système de fichiers de la partition, à un répertoire de l'arborescence logique

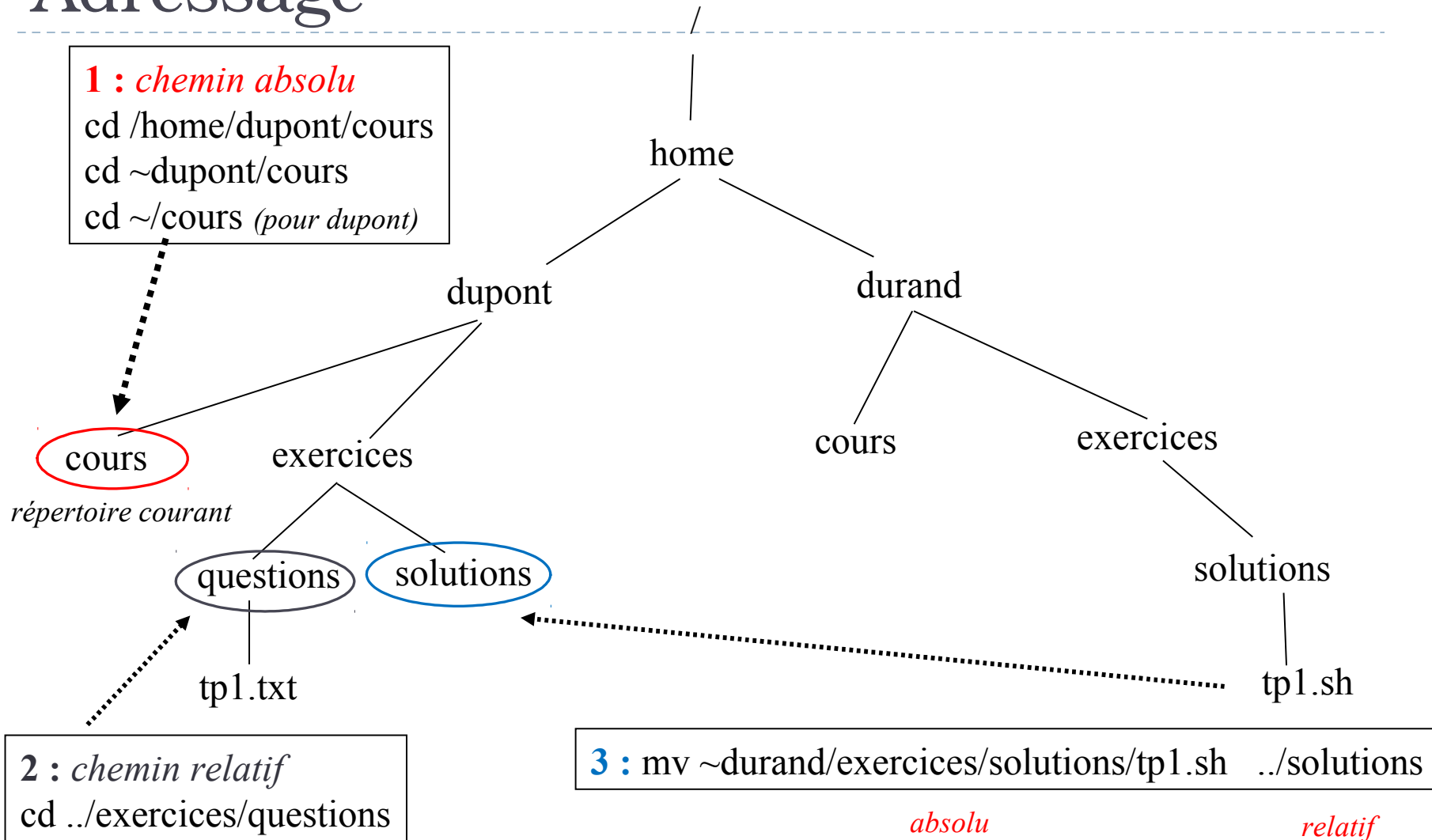
Arborescence



L'accès aux fichiers

- Le premier niveau de l'arbre est la **racine** représentée par le slash **"/"**
- répertoire **père** : répertoire au dessus du répertoire courant
- répertoire **fil** : répertoire en dessous du répertoire courant
- **.** : répertoire courant
- **..** : répertoire père du répertoire courant
- **~** : répertoire de connexion de l'utilisateur
- **~user** : répertoire de connexion d'un utilisateur *user*
- **Adressage absolu** : chemin depuis la racine (*commence par / ou ~*)
- **Adressage relatif** : chemin depuis le répertoire courant

Adressage



Editeur VI

- Editeur disponible sur les versions « *nues* » du système
 - Maîtrise indispensable pour l'administration d'un serveur

- Particularité : fonctionne selon **deux modes**
 - **insertion** : saisie
 - depuis le mode commande passage en mode insertion (i,a,o...)
 - **commande** : enregistrement, recherches...
 - depuis le mode insertion passage en mode commande Echap suivi de « : »

- vi s'ouvre en mode commande

Utilisateurs

- Les utilisateurs du système (invite **\$**):
 - Personnes physiques (*ou applicatif*)
 - Droits d'accès au système/aux ressources
 - Répertoire de connexion
 - Groupes d'utilisateurs
- Le super-utilisateur **root** (invite **#**):
 - Administrateur du système
 - Possède tous les droits sur le système, les utilisateurs et les fichiers
 - Commandes réservées (*invisibles des utilisateurs, pas de man*)
 - **Compte à n'utiliser que pour la configuration et l'installation**
- Les comptes et groupes sont désignés via des identifiants numériques :
 - **UID** : User Identifier **GID** : Group Identifier
 - La commande **id** permet d'afficher les ID d'un utilisateur

Session

- **Le login :**
 - identification de l'utilisateur
 - ouverture d'une session : accès aux ressources du système
 - accès en fonction des droits accordés par le super-utilisateur *root*

- **Le mot de passe :**
 - authentification de l'utilisateur
 - stocké sous forme cryptée
 - fixé par l'utilisateur et protège son compte
 - compte 6 caractères au moins
 - doit être changé fréquemment

Sessions

- Se déconnecter (close une session) : **logout, exit, Ctrl+D**
- **Ne jamais éteindre la machine brutalement**
 - commandes root : **shutdown, halt, reboot**
- **Ctrl+Alt+f#** permet de se connecter sur plusieurs terminaux
 - 1 à 6 mode console,
 - 7 mode graphique
- Avec la machine virtuelle **VMware**
 - NPO : sélectionner la fenêtre de la machine virtuelle avant de basculer,
sinon bascule sur la machine hôte

Le login

- Le répertoire de connexion:
 - répertoire où se trouve l'utilisateur en début de session
 - en général, sous-répertoire de **/home**, fixé par le super-utilisateur

- Le fichier des mots de passe **/etc/passwd** contient les informations sur l'utilisateur (*séparées par des « : »*) :
 - nom de login
 - mot de passe encrypté (*! = non défini*)
 - numéro d'utilisateur (UID)
 - numéro de groupe de connexion (GID)
 - commentaire (GECOS/finger)
 - répertoire initial de connexion
 - commande lancée après le login (*un shell en général pour un utilisateur*)

- **bash** ou **sh** : lance un shell (fils)

Sécurisation

- Le mécanisme **shadow** masque certaines informations sur les comptes
 - *mis en œuvre lors de l'installation*
 - shadowconfig, pwconv, grpconv
- Le fichier `/etc/passwd` reste visible, mais les mots de passes n'y sont plus *remplacés par des x*
- Le fichier **`/etc/shadow`** contient les mots de passe cryptés
- Des attributs complémentaires fixent leur durée de vie (*pour root*) :
 - options de la commande **passwd** (*pour root*)
 - commande **chage** pour une modification

Commandes

Principales commandes (root) de création/modification d'un utilisateur :

- **useradd** : commande de bas niveau ajout d'un utilisateur (*shell sh*)
- **passwd** : crée ou modifie le mot de passe
 - de l'utilisateur courant pour un utilisateur, contrôle sa longueur et sa "simplicité"
 - de l'utilisateur spécifié pour root, pas de contrôle
 - interactive : réclame l'ancien mot de passe sauf pour root
- **usermod** : modifie les caractéristiques d'un utilisateur
- **chfn** : change le GECOS/finger
- **chsh** : change le shell lancé au démarrage
- **vipw** : version spécifique de vi pour éditer le fichier passwd avec contrôle

Les groupes

- **Groupe principal** : groupe de connexion de l'utilisateur défini dans `/etc/passwd`
 - Groupe attribué aux fichiers lors de leur création par l'utilisateur
- **Groupes secondaires** : Pour assouplir la gestion des droits, un utilisateur peut appartenir à d'autres groupes, dits, **groupes secondaires** définis `/etc/group`
 - Structure d'une ligne (*séparés par des « : »*) :
 - Nom du groupe:Mot de passe :GID :liste des membres (*en tant que groupe secondaire*)
 - Ne contient pas les utilisateurs dont c'est le groupe de connexion
 - si shadow activé => **`/etc/gshadow`**
- **getent** retourne les entrées des tables d'administration (*les lignes du fichier*) :
 - `getent passwd user`
 - `getent group user`

Gestions des groupes

- Lors de la connexion l'utilisateur appartient à son groupe de connexion et aux groupes dans lesquels il a été déclaré
- **groups** : afficher les groupes auquel l'utilisateur appartient
- **addgroup** : créer/gérer les groupes
 - les groupes sont créés initialement avec un mot de passe non défini
 - impossible d'y adhérer durant une session
- **gpasswd** *groupe* : fixer un mot de passe pour un groupe (root)
- **newgrp** *groupe* : adhérer *temporairement* à un groupe en fournissant son mot de passe (nouveau shell)
 - devient le groupe principal (utilisé pour la création des fichiers)
 - desadhésion par Ctrl+D (arrêt du shell fils)
- **id** permet d'afficher les ID d'un utilisateur

Changement d'identité

- La commande **su** permet de changer temporairement d'identité utilisateur
 - retour à l'identité d'origine par déconnexion (Ctrl+D ou exit)
- **su** : (*sans utilisateur spécifié*) prend l'identité **root**
- **su user** : prend l'identité de l'utilisateur *user*
- **su -c commande user** : exécuter une seule commande sous l'identité spécifiée
- **who am i** : identifiant pour la session d'origine
- **whoami** : identifiant pour la session effective (*suite à un su*)
- **who** : utilisateurs connectés
- **last** : dernières connexions utilisateurs
- La commande **sg** permet de changer temporairement d'identité de groupe

Syntaxe des commandes

- Format des commandes :
- **commande** **[(-/--)option(s)]** **argument(s)** => *cat -n toto.txt titi.txt*
 - **Respecter la casse** (majuscules/minuscules)
 - Respecter les espaces séparateurs
 - ";" : séparateur quand plusieurs commandes sur une même ligne
- **mkdir** : Créer un répertoire
- **cd** : Se déplacer dans les répertoires
- **pwd** : Affiche le répertoire courant
- **touch** : Créer un fichier vide ou mettre à jour sa date de modification
- **ls** : Lister les fichiers
- **more** / **less** : Afficher un fichier avec pagination
- **cat** : Concaténer et afficher des fichiers
- **cp** : Copier un fichier
- **rm** : Supprimer un fichier
- **mv** : Déplacer ou renommer un fichier

Globbering

- La plupart des commandes acceptent plusieurs fichiers en argument
- Le **Globbering** consiste à utiliser des **meta-caractères** pour définir un modèle pour désigner un ensemble de fichiers
 - recherche, depuis le répertoire courant, les noms de fichiers correspondant au modèle
- L'expansion des meta-caractères (i.e. remplacement par les noms de fichiers correspondant) a lieu avant d'exécuter la commande
- L'étoile : ***** interprétée comme toute suite de caractères alphanumériques
 - ne remplace pas le . comme 1^{er} caractère
 - **cat *.txt** : *affiche le contenu de tous les fichiers d'extension txt*

Meta-caractères

- Le point d'interrogation : **?** : remplace un et un seul caractère alphanumérique
 - **grep echo ??? .txt** : *recherche la chaîne echo dans tous les fichiers txt dont le nom possède 3 caractères*
- L'énumération : **[]** : remplace un caractère choisi parmi ceux entre crochets
 - les caractères entre crochets n'ont pas à être ordonnés
 - **ls *.v[123]** : *fichiers d'extension v1 ou v2 ou v3*
- L'intervalle : **[-]** : remplace un caractère choisi parmi ceux de l'intervalle (ordre ASCII)
 - **ls *[a-z].sh** : *fichiers .sh dont le nom se termine par une minuscule*
- L'énumération : **[^...]** : remplace tout caractère autre que ceux qui suivent
 - **ls a*[xyz].?[^0-9]** : *commence par a, se termine par x ou y ou z, possède une extension sur 3 caractères ne se terminant pas par un chiffre*

Inhiber les meta-caractères

- Certains caractères ont une signification particulière pour le shell :
`! ^ * ? [] [-] \ ; & | < > >> <<`
- Pour pouvoir les utiliser **sans interprétation** il faut les protéger :
 - Protection d'un caractère par l'anti-slash : `\`
 - inhibe la signification spéciale du meta-caractère qui le suit
 - Protection d'une chaîne : `'chaîne'` `"chaîne"`
 - les `'` et `"` inhibent la substitution des meta-caractères
- Exemple
 - `touch question\?` : crée le fichier *question?*
 - `ls ***` : liste les fichiers dont le nom comporte une *