CHAPITRE 7 : Administration

7.1. Introduction :

Unix est un système multi-utilisateurs. Plusieurs personnes peuvent l'utiliser de façon simultanée (dans le cas de configurations en réseau). Pour le système, un utilisateur n'est pas obligatoirement une personne physique. Un utilisateur peut détenir des fichiers, exécuter des programmes ou encore déclencher automatiquement des fonctions systèmes. Par exemple, un utilisateur peut être créé dans le seul but de détenir des fichiers publics. On parle alors de pseudo utilisateur.

Un utilisateur possède un nom d'utilisateur appelé aussi login lui permettant d'établir une connexion. Ce login est associé à un mot de passe personnel. Pour accéder aux ressources du système, l'utilisateur doit entrer la bonne paire login/mot de passe : c'est l'authentification (le login).

Les utilisateurs sont identifiés par le système grâce à un UID (identifiant d'utilisateur) unique. Cet identifiant est une valeur numérique.

7.2. Rôle de l'administrateur système :

Unix étant un système d'exploitation multi−utilisateurs, la gestion du système et des utilisateurs est confiée à un super−utilisateur nommé root ou racine.

- **Créer, modifier, supprimer un utilisateur**: y compris la modification de l'environnement de travail, avec la modification des droits, stratégie des mots de passe...
- **Gérer les fichiers et les disques**: intégrité du système de fichiers, organisation de l'arborescence, protection avec les droits, gestion des systèmes de fichiers (création, montage, démontage), gestion des disques physiques (installation, partitionnement, ...)
- **Surveiller l'espace disque**: taux d'occupation des disques, swapping, quotas, ...
- **Organiser les sauvegardes**: commandes, support, intervalles, stratégies, ...
- **Ajouter des périphériques**: nouveaux disques, cartes réseau, fichiers spéciaux, ...
- **Améliorer les performances du système**: gestion des ressources, mémoire, optimisation des paramètres du noyau, ...
- **Gérer les services et installer les nouveaux produits**: services au démarrage, crontab, procédures d'installation standard, impression, ...
- **Sécuriser le système**: sécurité de connexion, discipline des utilisateurs, éventuellement sécurité réseau.
- **connaître le processus d'arrêt/relance.**

7.2.1 Devenir root

Seul l'**administrateur**, ou éventuellement quelques utilisateurs privilégiés dans le cas de certains services, peut administrer un système Unix. Cet utilisateur particulier se nomme **root**, ou **super user**. Il porte l'**UID 0**. Son login est généralement root mais tout autre login ayant l'UID 0 est root.

Dans le cas d'une connexion root, l'invite du shell est le caractère « # ».

La commande « su » permet de prendre temporairement l'identité de l'administrateur. (su - change aussi l'environnement).
Il faut éviter de travailler directement et en permanence en tant que root.

7.2.2 Administrer

L'administration d'un système Unix s'effectue de quatre manières principales :

1. A l'aide des **commandes d'administration** : adduser/useradd, chown, date, ... C'est la méthode pour les plus expérimentés, pour ceux qui connaissent par coeur les multiples paramètres et options proposés par les commandes.
2. En **éditant les fichiers d'administration** : on peut ajouter un utilisateur avec vi à la main dans /etc/passwd. La grande majorité des fichiers de configuration d'un Unix est au format texte et très bien documentée.
3. En **exécutant des scripts** : on peut comme cela automatiser des séquences complètes d'administration et de configuration. Dans beaucoup de cas, les commandes d'arrêt/relance du système ou des services, ou d'ajout de d'utilisateurs, sont des scripts.
4. En **utilisant les outils intégrés** d'administration : les divers Unix proposent des outils d'administration intégrés, comme linuxconf ou Webmin sous Linux. L'avantage est la simplification et la centralisation.

7.2.3. Dialogue avec les utilisateurs

L'administrateur se doit de dialoguer et de prévenir ses utilisateurs de l'état du système et des modifications à l'aide des moyens suivants :

- **messagerie** : mail
- **communication directe** : write, talk, wall
- **messages à la connexion**: /etc/issue (avant connexion), /etc/motd (après connexion)

7.3 Les principaux répertoires

- / ou root directory, le répertoire principal de tout Unix, généralement le premier monté juste après le chargement du noyau. Toute l'arborescence de base d'un Unix y est soit directement présente, soit reliée par montage. Cette partition étant la première accédée par le système elle doit contenir les fichiers de configuration, les binaires système et les bibliothèques de base, de manière à pouvoir démarrer correctement le système. La racine et son contenu de base prennent en général jusqu'à 200 Mo.
- **/sbin** contient généralement les commandes Unix systèmes de base telles que les commandes de montage, que changement de runlevel, de création de swap, bref toutes les commandes utilisées par le système au démarrage ou par l'administration. Il doit rester dans le root directory.
- **/lib** contient toutes les librairies (bibliothèques) de base, ainsi que parfois les modules du noyau. Comme /sbin il doit rester dans le root directory.
- **/etc** est un répertoire important vital et stratégique car il contient toute la configuration de l'Unix : runlevels, règles de montage, utilisateurs, groupes, quotas, configuration réseau, ... Il doit donc rester dans le root directory.
- **/tmp** contient les fichiers et répertoires temporaires créés par le système ou les services, les utilisateurs. Sa taille peut être variable. Un éditeur comme vi y stockera ses fichiers en cours, une messagerie POP aussi (avant de les envoyer ou durant sa composition). Ainsi sa taille conseillée de 100 Mo à la base peut monter à plusieurs gigaoctets. Il est conseillé de créer une partition spécifique.

- **/var** est particulier car comme son nom l'indique sont contenu est variable. Il va contenir les traces et journaux du système (noyau) et des services, les spools d'impression, la crontab, les spools et fichiers de messagerie, ... Sa taille de base de 200 Mo peut évoluer, en cas de non archivage, à plusieurs gigaoctets (envisageons une centaine d'utilisateurs imprimant de fichiers d'une centaine de mégaoctets, recevant ou envoyant des messages avec de fichiers joints...). Une partition distincte est à envisager.
- **/dev** contient l'ensemble des fichiers spéciaux représentant les périphériques. Il doit forcément rester dans le root directory.
- **/usr** contient généralement les applications accessibles par les utilisateurs (dans bin) ou des outils système complémentaires (sbin) ou des composants optionnels et rajoutés (local). Dans le cadre d'un serveur au rôle délimité, sa taille peut être fixe et dans ce cas placé dans le root directory. Dans le cas d'un serveur de test ou de développement, sa taille peut énormément varier (installation d'outils et de kits de développement, multiples versions de tests d'un logiciel,...) et dans ce cas mieux vaut lui donner une partition distincte. Au minimum 600 Mo, pas de maximum.
- **/proc** n'est pas un véritable répertoire. Son contenu est logique et pas physique, et donc ne prend aucune place sur le disque. Il contient en fait des informations liées au noyau, au réseau, aux pilotes et aux processus en cours.
- **/root** est en principe le répertoire par défaut de l'administrateur. Il est préférable de la placer dans le root directory s'il contient par exemple des scripts de son cru, en cas de changement d'un runlevel.
- **/home** contient l'ensemble des répertoires utilisateurs. On peut le laisser dans le root directory si le nombre d'utilisateurs est faible et la taille est limitée (quotas). Autrement il faudra envisager de le placer dans sa propre partition. Il faut compter 10 à 50 Mo par utilisateur.
- **/mnt** contient par convention les points de montages autres que ceux cités ci-dessus, points de montage annexes comme les partitions non Unix, les disquettes, les cdroms, lecteurs Zip, ...

7.4 Gestion des utilisateurs

Quand un utilisateur se connecte, il fournit un nom de connexion (**login name**) et un mot de passe (**password**). Si la connexion réussit, un **shell** est lancé et l'utilisateur se retrouve dans son répertoire de travail (**working directory**) qui est initialement son répertoire de connexion (**login directory**). Ces informations sont placées dans les fichiers /etc/passwd et /etc/group.

7.4.1. /etc/passwd

Les utilisateurs sont définis au sein du fichier /etc/passwd. En voici sa structure :

```
user:passwd:UID:GID:commentaire:homedir:commande
```

Champs	*Contenu*
user	nom de connexion (login) de cet utilisateur
passwd	mot de passe crypté. S'il est remplacé par une *, ou un x, son contenu est placée dans /etc/shadow. S'il est absent il n'y a pas de mot de passe. Si ! : compte verrouillé.

UID	User Identification, numéro d'identification de l'utilisateur. L'utilisateur root à l'UID 0, les numéros inférieurs à 100 sont des utilisateurs spéciaux. Généralement un Unix peut gérer plus de 60000 UID. Si l'affectation est généralement séquentielle, elle peut aussi être arbitraire. Plusieurs logins peuvent être associés à un seul UID. Dans ce cas en fait il n'existe qu'un seul utilisateur.
GID	Group Identification, numéro du groupe d'appartenance de l'utilisateur. Les groupes supplémentaires sont définis dans /etc/group
Commentaire	Zone libre, contenant généralement le nom et le prénom. Il faut éviter d'y placer de caractères spéciaux et accentués.
homedir	Répertoire de connexion, ou home directory / login directory. Si l'utilisateur est toto, le répertoire est généralement /home/toto
commande	Commande à exécuter lors de la connexion de l'utilisateur. C'est généralement le shell (/bin/sh, /bin/ksh, ...) mais aussi n'importe quelle autre commande.

7.4.2. /etc/group

Les groupes sont définis au sein du fichier /etc/group. Voici sa structure :

```
group:passwd:GID:liste_utilisateurs
```

Champs	*Contenu*
group	nom du groupe
passwd	mot de passe pour un utilisateur désirant changer de groupe sans en faire partie. Peu utilisé. (commande **newgrp**)
GID	Group Identification, numéro d'identification du groupe.
liste	liste des utilisateurs du groupe. Pour utilisation avec la commande newgrp.

7.4.3. Principe de l'ajout des utilisateurs

L'ajout d'un utilisateur consiste à :

- associer un mot de passe à l'utilisateur (ajout d'une entrée dans le fichier /etc/passwd);
- définir à quel groupe appartient l'utilisateur (ajout d'une entrée dans le fichier /etc/group) ;
- créer le répertoire personnel de l'utilisateur ;
- créer le fichier de configuration personnel du shell ;

Une entrée (c'est à dire une ligne) du fichier /etc/passwd est de la forme :
Nom :mot de passe :numéro d'utilisateur :numéro de groupe :champs spécial :répertoire personnel :shell de démarrage

Exemple : d'entrée du fichier /etc/passwd :
$ cat /etc/passwd | grep ali

ali:x:501:100:ali Benali:/home/ali:/bin/bash

Une entrée (c'est à dire une ligne) du fichier /etc/group est de la forme :
Nom de groupe : champs spécial : numéro de groupe : membre1 , membre2, etc..

Exemple : d'entrée du fichier /etc/group :

```
$ cat /etc/group | grep 100
users:x:100:
```

Pour ajouter un utilisateur ali, vous devez :

1. ajouter l'utilisateur philippe dans le fichier /etc/passwd
2. ajouter éventuellement un nouveau groupe dans /etc/group (si vous souhaitez créer un groupe spécifique pour ali)
3. créer le répertoire personnel du nouvel utilisateur (home directory), copier les fichiers de configuration du shell et changer les droits du répertoire ali afin que l'utilisateur ali devienne propriétaire de son répertoire personnel.
   $ mkdir /home/ali
   $ cp /etc/skel/* /home/ali
   $ chown ali /home/ali
   $ chgrp le_groupe_de_ali /home/ali
4. donner un mot de passe à l'utilisateur ali par la commande :
   $ passwd ali

7.4.4. Commandes

La commande **passwd** permet de changer le mot de passe, et dispose de plusieurs options.

- passwd nom_user : changement du mot de passe de l'utilisateur donné
- passwd -d nom_user : supprime le mot de passe
- passwd -l nom_user : verrouille le compte
- passwd -f nom_user : force le changement de mot de passe à la prochaine connexion

Les autres commandes sont :

- **useradd** :ajout d'un utilisateur (adduser parfois sous Linux)
- **usermod** : modification d'un compte utilisateur
- **userdel** : supprime l'utilisateur (deluser parfois sous Linux)
- **groupadd** : ajout d'un groupe
- **groupmod** : modification d'un groupe
- **groupdel** : supprime un groupe
- **pwck** : vérification de la cohérence de /etc/passwd
- **grpck** : vérification de la cohérence de /etc/group
- **finger** : informations sur un utilisateur
- **su** : se connecter à un compte

- **id** : connaître son identité
- **chsh** (Linux) : changer de shell

On se reportera au manuel en ligne pour les options qui ne sont pas compliquées.

Sous Linux, la syntaxe de useradd est la suivante :

useradd [-c comment] [-d homedir] [-g initial_group] [-G group2,group3,...] [-m] [-s shell] [-u UID] [-o] [-r] [-e YYYY-MM-DD] [-f inactive_days]

- -c : saisie du commentaire. Un commentaire détaillé peut être inséré avec la commande chfn.
- -d : le répertoire personnel
- -g : le groupe de base
- -G : groupes supplémentaires pour utilisation avec la commande newgrp
- -m : recopie du /etc/skel dans le répertoire personnel.
- -s : le shell
- -u : choix d'un UID spécifique > 99
- -o : l'UID n'est pas unique
- -r : UID privilégié système <= 99
- -e : date d'expiration du login
- -f : délai avant verrouillage si mot de passe non changé
- -p : mot de passe déjà crypté avec crypt

On crée un utilisateur le plus simplement avec

useradd -m user
useradd -c "login de test" -d /home/logtest -s /bin/bash -g users -u 123-m logtest

La commande usermod modifie ces valeurs (mêmes paramètres) mais permet aussi de verrouiller un compte :

- -l login : modification du login
- -L : lock password (verrouillage)
- -U : unlock password (déverouillage

La commande userdel supprime le compte, avec son répertoire personnel si -r.

userdel -r logtest

La commande chsh modifie le shell de login.

- **-s** : spécifie le nouveau shell
- **-l** : donne la liste

```
chsh -s /bin/sh logtest
```

La commande **groupadd** crée un groupe :

```
groupadd [-g GID] [-o] [-r] [-p password] group
```

- -g : précise le GID > 99
- -o : le GID n'est pas unique
- -r : GID système <= 99
- -p password déjà crypté

```
groupadd -g 123 grptest
```

La commande **groupdel** supprime le groupe.

```
groupdel grptest
```

La commande **groupmod** modifie le groupe, -n renomme le groupe.

On peut changer de groupe (groupe secondaire) avec la commande **newgrp** :

```
newgrp [-] groupe [-c commande]
```

Ajout d'un utilisateur avec la commande useradd :
Syntaxe : **useradd nom−utilisateur −g groupe −d répertoire−personnel −m**

L'option −m permet de recopier les fichiers de configuration du shell. On peut remplacer le shell courant par un shell spécifique avec l'option −s (par exemple −s /etc/ftponly).
$ useradd ali −g users −d /home/ali −m
$ passwd ali
Changing password for user ali
New UNIX password :
Retype new UNIX password :
Passwd : all authentification tokens update successfully

Suppression d'un utilisateur avec la commande userdel :
$ userdel −r ali
L'option −r permet de supprimer le répertoire personnel de l'utilisateur à supprimer.

Ajout d'un groupe avec la commande groupadd :
$ groupadd ftpusers

Suppression d'un groupe avec la commande groupdel :
$ groupdel ftpusers

Gestion graphique des utilisateurs et des groupes :
Vous pouvez également ajouter un utilisateur graphiquement à partir de l'utilitaire de configuration DrakeConf fourni avec la distribution ou Mandrake (l'ajout d'un utilisateur s'accompagne de l'ajout d'un groupe du même nom) ou à partir de l'utilitaire linuxconf

disponible sur toutes les distributions linux (vous pouvez alors choisir le groupe d'appartenance de l'utilisateur).
Comme pour la plupart des tâches d'administration système, vous pouvez gérer les utilisateurs et les groupes très facilement avec Webmin :

7.5. La sauvegarde

7.5.1. Les outils de sauvegarde

La sauvegarde est un travail important de l'administrateur puisqu'en cas de gros problème, on passe généralement par une restauration du système depuis une sauvegarde ou une image du système lorsque celui-ci était encore intègre (bon fonctionnement, pas de corruption). Chaque Unix arrive avec des commandes et des procédures de sauvegarde qui lui sont propres. On distingue tout de même quelques outils communs.

7.5.1.1. Commandes, plans, scripts

- Pour la **sauvegarde de fichiers et d'arborescences**, on utilise les commandes **tar**, **cpio** et **pax**. Ces commandes sauvent une arborescence, et pas un système de fichiers. On peut faire coïncider les deux.
- Pour la **sauvegarde physique de disques et de systèmes de fichiers**, on utilise la commande **dd** et la commande **volcopy**.
- Une **sauvegarde incrémentale** consiste à sauvegarder une première fois la totalité des données, puis ensuite uniquement les fichiers modifiés. On utilise parfois les commandes **dump** et **restore**, mais là chaque OS vient avec ces propres outils. On trouve aussi en libre ou dans le commerce des solutions plus pointues.

L'administrateur aura parfois à définir des scripts de sauvegarde et de restauration adaptés au cas par cas (partition systèmes, données applicatives, ...) et à automatiser quand c'est possible l'exécution de ceux-ci en fonction de la date, l'heure ou la charge machine.

Il sera aussi très important de définir un plan de sauvegarde, en se posant les bonnes questions :

- Que faut-il sauvegarder ?
- Avec quelle fréquence ?
- Combien de temps conservera-t-on les sauvegardes, à quel endroit, en combien d'exemplaires ?
- A quel endroit sera stocké l'historique des sauvegardes ?
- Quel est le support le plus approprié ?
- Quels sont les besoins, en capacité, du support de sauvegarde ?
- Combien de temps prévoit-on pour un fichier, un système de fichier et est-ce raisonnable ?
- La sauvegarde doit-elle être automatique ou manuelle ?
- Quelle est la méthode de sauvegarde la plus appropriée ?

Voici quelques autres commandes :

- **mt** : contrôle d'une bande magnétique
- **touch** : met la date de dernière modification à l'heure actuelle, pour forcer une sauvegarde incrémentale
- **find** : sélectionne les fichiers à sauvegarder
- **compress** et **uncompress** : compression et décompression des fichiers
- **gzip**, **gunzip**, **zcat**, compression et décompression au format GnuZip.

7.5.1.2. Sauvegarde par tar

On emploie très souvent la commande tar car elle est simple et efficace. Elle sauvegarde des fichiers, y compris l'arborescence de fichiers. L'archive ainsi créée peut s'étendre sur plusieurs volumes : quand la bande ou la disquette est pleine, c'est à l'utilisateur d'en insérer une nouvelle et la sauvegarde/restitution continue.

Pour une sauvegarde :

```
tar cvf nom_archive Fichier(s)
tar cvf archive1.tar /home/*
```

- **c**: création d'archive
- **v** : mode bavard 'verbose' : dit ce qui se fait
- **f** : le paramètre suivant est le nom de l'archive

Pour lister le contenu de l'archive :

```
tar tvf nom_archive
tar tvf archive1.tar
```

- **t** : liste le contenu de l'archive

Pour une restauration :

```
tar xvf nom_archive fichiers
tar xvf archive1.tar
```

- **x** : extraction de l'ensemble des fichiers de l'archive, ou du ou des fichiers spécifiés.

Autres clés et paramètres :

- **r** : les fichiers sont ajoutés à la fin de l'archive. Ne fonctionne pas avec les bandes et cartouches.
- **k** : spécifie la taille en ko du support d'archivage, pour faire du multi-volumes
- **l** : on sauvegarde les fichiers et pas les liens symboliques (le fichier pointé est sauvé)
- [**LINUX**] **z** : l'archive est compressée au format gzip.

- [**LINUX**] **Z** : l'archive est compressée au format compress.
- [**LINUX**] **j** : l'archive est compressée au format bzip2.

7.5.1.3. Sauvegarde par cpio

La commande cpio sauvegarde sur la sortie standard les fichiers dont on saisit les noms sur l'entrée standard, par défaut le clavier et l'écran. On utilisera donc les redirections. Voici les options :

- -v : mode bavard « verbose », informations détaillées
- -c : sauvegarde des attributs des fichiers sous forme ASCII (pour l'échange entre divers OS)
- -B : augmente la vitesse d'exécution en utilisant une mémoire tampon (5120 octets soit 10 blocs)

Pour une sauvegarde :

```
cpio -oL
```

- o : output, creation de la sauvegarde en sortie
- L : sauve les fichiers liés et pas les liens symboliques

Pour lister le contenu de l'archive :

```
cpio -it
```

- i : lecture de l'archive en entrée
- t : comme pour tar, liste le contenu de l'archive

Pour une restauration :

```
cpio -i[umd]
```

- u : restauration inconditionnelle, avec écrasement des fichiers qui existent déjà. Par défaut les fichiers ne sont pas restaurés si ceux présents sur le disque sont plus récents ou du même âge.
- m : les fichiers restaurés conservent leur dernière date de modification
- d : cpio reconstruit l'arborescence des répertoires et sous-répertoires manquants.

Exemples :

- Sauvegarde de l'arborescence courante sur une disquette avec compression :
  ```
  find . print | cpio -ocvB | compress > /dev/fd0
  ```
- Restauration
  ```
  cat /dev/fd0 | uncompress | cpio -iuvBd
  ```

7.5.1.4. Sauvegarde par dd

La commande « dd » (device to device) est destinée à la copie physique, bloc à bloc, d'un fichier périphérique vers un fichier périphérique. A l'origine on l'utilisait pour la lecture et l'écriture sur bande magnétique, mais elle peut être employée avec n'importe quel fichier. La commande dd permet de réaliser des copies physiques de disques et de systèmes de fichiers.

Argument	***utilisation***
if=fichier	« fichier » désigne le ficher à copier, à défaut l'entrée standard.
of=fichier	« fichier » désigne le résultat de la copie, à défaut la sortie standard.
bs=valeur	« valeur » désigne la taille commune du bloc pour les fichiers d'entrée et de sortie, par défaut à 512 octets.
skip=n	nombre de blocs qu'il faut sauter au début du fichier d'entrée.
seek=n	nombre de blocs à sauter au début du fichier de sortie
count=n	nombre de blocs à copier
conv=...	Conversion lors de la copie (lcase : minuscule, ucase : majuscule, ascii, abcdic, swab : permutation des octets)

Exemple sous Linux :

Ici on va placer le secteur de boot de la partition (ou est installé lilo ou grub) dans un fichier. Le fichier ainsi créé pourra être utilisé avec le chargeur de NT/2000/XP pour démarrer sous Linux.

```
dd if=/dev/hda5 of=boot.lnx bs=512 count=1
```