Administration Systèmes & Réseaux

Administration de base d'un système

May 3, 2015

Houcemeddine HERMASSI

houcemeddine.hermassi@enit.rnu.tn

École Nationale d'Ingénieurs de Carthage ENI-CAR
Université Carthage
Tunisie

Plan de cour

Processus de démarrage

Le chargeur de démarrage

Administration des utilisateurs

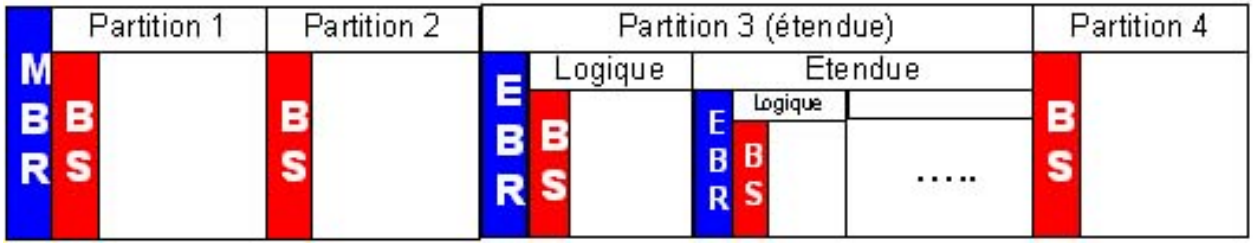
Rôle

Le BIOS (Basic Input Output System) est l'interface logicielle entre le matériel et le système d'exploitation. En d'autre terme il fournit le niveau d'interface le plus bas aux pilotes et périphériques.

Fontionnement

Le BIOS est présent sur une mémoire EEPROM alimentée par une batterie, quand l'ordinateur est démarré, un signal appelé **powergood** est envoyé au microprocesseur, celui-ci déclenche l'exécution du BIOS. Le BIOS doit vérifier le bon fonctionnement des composantes et doit déterminer le périphérique de démarrage.
Le BIOS lit et exécute le premier secteur physique du média de démarrage. Il s'agit généralement des 512 premiers octets du premier disque dur (MBR: Master Boot Record) ou de la partition active (PBR: Primary Boot Record)

Processus de démarrage
Le chargeur de démarrage



MBR & EBR

MBR: Le secteur de partition principale est le premier secteur d'un disque et comme tout secteur il fait une taille de 512 octets. Sa structure contient diverses informations sur le disque ainsi que sur les différentes partitions principales qui le composent. Il va également nous permettre de démarrer à partir du disque.
EBR: Extended Boot Record (fonction identique au MBR, partitions logiques).

BS

BS: Le secteur d'amorage détient les instructions permettant de charger le noyau en mémoire depuis son emplacement sur le disque. Le noyau dispose de ses propres fonctions de détection.

Linux

- LILO (Linux loader)
- GRUB (GRand Unified Bootloader)

Windows

- Ntldr
- WinLoad

Apple

- Boot Camp

Le chargeur de démarrage
GRUB(1/2)

Caractéristiques

- Hautement paramétrable
- Protectéger par mot de passe cryptée
- Un interpréteur de commandes
- Accès avec des graphiques

Configuration

Le fichier de configuration de GRUB est **/etc/grub.conf** ou **/boot/grub/menu.lst**, les principales options sont:

- **title**: nom de l'image
- **root(hdx,y)**: tous les accès fichiers spécifiés dessous le seront à partir de cette partition. hd0,0 représente le premier disque détecté (ici /boot) et la première partition
- **kernel**: chemin du noyau en partant de la racine (/boot/vmlinuz...)
- **initrd**: initial ramdisk. Le noyau va charger ce fichier comme disque en mémoire
- **rootnoverify**: la racine spécifié, à ne pas monter par GRUB (EX: FAT, NTFS)

Installation

Le chargeur de démarrage
GRUB(2/2)

Démarrage & édition

Au démarrage de GRUB, un menu s'affiche. Il peut être graphique ou textuel, selon la configuration. Vous devez choisir une image de démarrage avec les flèches de direction parmi celles proposées. En appuyant sur la touche [Entrée] vous démarrez l'image sélectionnée.
Vous pouvez éditer les menus directement pour modifier par exemple les paramètres passés au noyau Linux ou init. Dans ce cas, sélectionnez une entrée de menu et appuyez sur la touche **e** (edit). Ici, toutes les lignes de la section sont affichées. Vous pouvez appuyer sur :

- **e**: pour éditer la ligne (la compléter)
- **d**: pour supprimer la ligne
- **o**: pour ajouter une ligne
- **b**: pour démarrer l'image (booter).

Par exemple, pour démarrer en mode urgence (emergency) :

- Allez sur la ligne Linux et appuyez sur **e**.
- Allez sur la ligne kernel et appuyez sur **e**.
- À la fin de la ligne rajoutez 1 ou Single et appuyez sur [Entrée].
- Appuyez sur **b**.

Vous pouvez aussi accéder à un interpréteur de commandes en appuyant sur [Echap]. Attention seules les commandes GRUB sont reconnues.

Processus de démarrage
Initialisation du noyau

Initialisation du noyau

- Le matériel est détecté et initialisé
- iIntird est chargé, les modules présents éventuellement chargés
- Le noyau monte le système de fichiers racine en lecture seule
- Il crée la première console
- Le premier processus est lancé (init)

Toutes les traces du noyau sont placées dans le fichier **/var/log/dmesg**

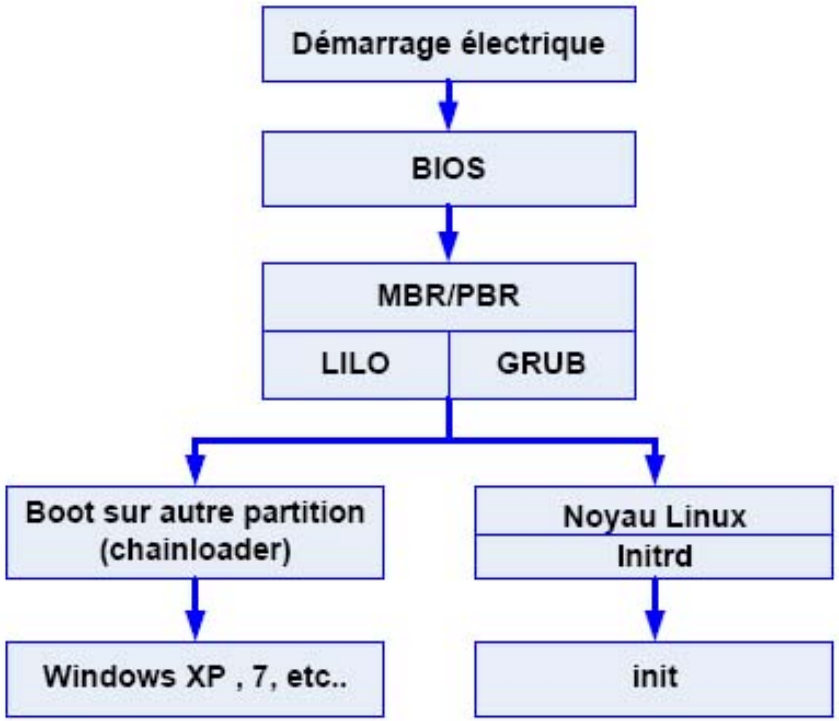


Schéma de la séquence de démarrage

Processus de démarrage
Le processus init

Rôle

- Premier programme démarré et dernier stoppé au sein du système
- Celui qui lance toutes les tâches
- Démarrer et d'arrêter tous les services
- **init** reste actif pour gérer les changements d'état des processus et les niveaux d'exécution
- Le processus **init** est le père de tous les processus, il a toujours le PID 1
- Le fichier de configuration est: **/etc/inittab**

Niveaux d'exécution

(runlevel) est un état dans lequel se trouve le système. Cet état est contrôlé par init.

- Le niveau d'exécution par défaut est positionné dans **/etc/inittab** sur la ligne initdefault Id:5:initdefault
- Le changement de niveau à la volée ce fait par la commande: telinit 5
- Le niveau d'exécution actuel est visible avec la commande `/sbin/runlevel runlevel"`

Niveau	Effet
0	Halt
1	Mode mono-utilisateur utilisé pour la maintenance, mode console
2	Multi-utilisateur, sans réseau, console
3	Multi-utilisateur, avec réseau, console
4	Idem que le 3, laissé à la convenance de l'administrateur
5	Multi-utilisateur, avec réseau, avec environnement graphique X. window
6	Reboot
S,s	Single user mode (urgence)

Méthodes d'arrêt

Pour arrêter le système proprement, la commande `shutdown"` peut être utilisée.

`Shutdown param délai message"`

Paramètre	Action
k	N'effectue pas le shutdown mais envoi le message à tout les stations
r	C'est un reboot
h	Halt, c'est un arrêt
f	Empêche l'exécution de fsck au boot
F	Force l'exécution de fsck au boot
c	Annule le shutdown sans délai, mais un message est possible

Le délai peut être spécifié de différentes manières: ***hh:mm*** ou ***+m*** ou bien ***now***

- shutdown -r +10 "reboot pour maintenance"
- shutdown -c "maintenance annul
- shutdown -h équivalente de halt
- shutdown -r équivalente de reboot

La commande `dmesg"`

- Récupérer les messages du noyau émis au démarrage de la machine et par la suite
- Le tampon de **dmesg** est circulaire
- Les anciens traces sont sauvegarder dans **/var/log/messages**
- Récupérer les messages lors de la connexion à chaud de périphériques
- Récupérer les messages au chargement de certains modules
- Récupérer les messages lors des crashs ou une corruption du système de fichiers

La commande **Application**

Voir ce qui se passe à l'insertion d'une clé USB, après le boot au cours d'une utilisation normale, et à sa déconnexion.

`dmesg|grep CPU"`

Pour voir les messages spécifiques à l'utilisateur root

`,"tail -100 /var/log/messages |grep root"`

Kernel

Le noyau est:

- Le coeur du système d'exploitation
- Libre, possibilité de le recompiler, modifier et d'ajouter des extensions
- Famille des noyaux monolithiques (< 1.3) puis modulaires
- Présent dans /boot, son nom par convention **vmlinuz-X.Y.Z.p-Vtxt**
  - X: version majeure du noyau
  - Y: une valeur paire représente une branche stable du noyau.
  - Z: version mineure de noyau
  - P: version corrigée ou intermédiaire (correction des bugs)
  - V: version propre à l'éditeur de la distribution
  - txt: précession sur le noyau (Ex: smp indique un noyau multiprocesseur)

```
$ uname -r-p -a: voir la version du noyau"
```

Processus de démarrage
Services et modules noyau

Compilation du noyau (Avant la compilation)

Avant de compiler le noyau pour une éventuelle mise à jour du système, des informations nécessaires peuvent être contenues dans les fichiers du répertoire /proc comme sur les pilotes: /proc/devices ou sur les types de système de fichier dans le noyau: /proc/filesystems

Voir aussi

proc/cpuinfo, /proc/pci, /proc/interrupts, /proc/dma, /proc/ioports. Ces informations pourront être utiles si le nouveau périphérique n'est pas reconnu après recompilation du noyau.

Compilation du noyau et installation des modules

NB: la compilation se fait dans le répertoire /usr/src/linux-x.y.z

- make dep: vérifie les dépendances
- clean: Nettoie les fichiers résiduels d'une compilation antérieure
- bzimage: Compile le noyau
- modules: Compile et installe les modules (drivers)
- modules_install: Installe les modules dans le répertoire /lib/modules/x.y.z

Remarque: Dans le cas d'une recompilation (sans monter de version)
Renommer **/lib/modules/$(uname -r)** en **/lib/modules/old$(uname -r)**

Processus de démarrage
Services et modules noyau

Gestion des modules

Les pilotes de périphériques, systèmes de fichiers, protocoles réseaux peuvent être présents sous forme de modules, Les modules:

- communiquent avec le noyau via une API
- s'exécutent dans l'espace du noyau
- sont paramétrables
- peuvent être chargés et déchargés à la demande évitant ainsi un redémarrage

Les modules sont dans **/lib/modules/$(uname -r)**

Gestion des modules(affichage: lsmod)

lsmod liste les modules actuellement chargés avec leurs dépendances éventuelles
Exemple: le module vfat dépend du module fat
/proc/modules

```
Fichier  Édition  Affichage  Signets  Configuration  Aide
[root@oranix anis]# lsmod
Module                  Size  Used by
vfat                    6758  0
fat                    38062  1 vfat
usb_storage            35815  0
hidp                   13887  0
fuse                   51887  3
ebtable_nat             1431  0
ebtables               12146  1 ebtable_nat
ipt_MASQUERADE          1765  3
bridge                 57147  0
stp                     1438  1 bridge
llc                     3754  2 bridge,stp
rfcomm                 54295  6
```

Processus de démarrage

Services et modules noyau

Gestion des modules(dépendance: depmod)

dpmod met à jour l'arbre des dépendances entre les modules en modifiant le fichier /lib/modules/'uname -r'/modules.dep, contient le chemin vers le module ainsi que la liste des dépendances. Exemple: module fat

```
Fichier  Édition  Affichage  Signets  Configuration  Aide
[root@oranix anis]# grep fat /lib/modules/`uname -r`/modules.dep
kernel/fs/fat/fat.ko:
kernel/fs/fat/vfat.ko: kernel/fs/fat/fat.ko
kernel/fs/fat/msdos.ko: kernel/fs/fat/fat.ko
```

Le module vfat.ko dépend du module fat.ko, il faut que le module fat.ko soit chargé en premier.
depmod -a: Reconstruit les dépendances de tous les modules, cette action est exécutée à chaque démarrage du système.

Gestion des modules(informations: modinfo)

modinfo fournit toutes les informations nécessaires sur un module:

- Le nom du fichier correspondant
- Une description du module
- L'auteur
- Les paramètres
- Les alias matériel
- Les dépendances

```
[anis@oranix ~]$ modinfo vfat
filename:       /lib/modules/2.6.35.14-96.fc14.i686/kernel/fs/fat/vfat.ko
author:         Gordon Chaffee
description:    VFAT filesystem support
license:        GPL
srcversion:     AF10F57347E6ECDFF6A8D57
depends:        fat
vermagic:       2.6.35.14-96.fc14.i686 SMP mod_unload 686
```

Gestion des modules(décharger: rmmod)

rmmod décharge le module donné, **rmmod** ne gère pas les dépendances

- Il n'est pas possible de décharger un module en cours d'utilisation
- Il n'est pas possible de décharger un module s'il est utilisé par un autre

```
[anis@oranix ~]$ mount |grep vfat
/dev/sdb1 on /media/KINGSTON type vfat (rw,nosuid,nodev,uhelper=udisks,uid=502,gid=502,sho
rtname=mixed,dmask=0077,utf8=1,showexec,flush)
[anis@oranix ~]$ rmmod vfat
ERROR: Module vfat is in use
```

Dans ce deuxième exemple, la clé est débranchée. Les modules fat et vfat sont devenus inutiles. On tente de supprimer le module fat. Le système retourne une erreur liée aux dépendances.

```
[anis@oranix ~]$ rmmod fat
ERROR: Module fat is in use by vfat
```

Dans ce dernier exemple, le module vfat est déchargé, puis le module fat, dans cet ordre:
rmmod vfat
rmmod fat

```
[root@oranix anis]# lsmod|grep fat
vfat                    6758  0
fat                    38062  1 vfat
[root@oranix anis]# rmmod vfat
[root@oranix anis]# rmmod fat
[root@oranix anis]# lsmod|grep fat
[root@oranix anis]#
```

Gestion des modules(charger: insmod)

insmod charge le module donné sans gérer les dépendances. C'est à vous de gérer les dépendances liées.

```
[root@oranix ~]# cd /lib/modules/$(uname -r)/kernel/fs/fat
[root@oranix fat]# lsmod|grep fat
[root@oranix fat]# insmod vfat.ko
insmod: error inserting 'vfat.ko': -1 Unknown symbol in module
[root@oranix fat]# insmod fat.ko
[root@oranix fat]# insmod vfat.ko
[root@oranix fat]# lsmod|grep fat
vfat                    6758  0
fat                    38062  1 vfat
[root@oranix fat]#
```

Gestion des modules(gérer les dépendances: modprob)

modprobe charge le module donné ainsi que ses dépendances.
Le fichier /etc/modprobe.conf fait référence.
Le chargement du module vfat à l'aide de modprobe va automatiquement charger le module fat.

```
[root@oranix anis]# lsmod|grep fat
[root@oranix anis]# modprobe vfat
[root@oranix anis]# lsmod|grep fat
vfat                    6758  0
fat                    38062  1 vfat
```

Décharger le module vfat et les modules dont il dépend (s'ils ne sont plus utilisés).

```
[root@oranix anis]# modprobe -r fat
FATAL: Module fat is in use.
[root@oranix anis]# modprobe -r vfat
[root@oranix anis]# lsmod|grep fat
[root@oranix anis]#
```

Administration des utilisateurs
Principe

Identification et authentification

L'identification, c'est savoir qui est qui, afin de déterminer les droits de la personne qui se connecte. Un utilisateur est identifié par un login.
L'authentification, c'est apporter la preuve de qui on est, par exemple via un secret partagé entre l'utilisateur et le système, et connus d'eux seuls. L'utilisateur est authentifié par un mot de passe.

Les utilisateurs

Utilisateur=Nom de connexion + Login + UID (User ID) ou GID (Group ID)
L'utilisateur dispose des attributs de base suivants:

- Un nom de connexion appelé le login
- Un mot de passe
- Un UID
- Un GID correspondant à son groupe principal
- Un descriptif
- Un répertoire de connexion
- Une commande de connexion

Un login accepte la plupart des caractères. Il ne doit pas commencer par un chiffre. Il est possible de modifier la liste des caractères autorisés et de forcer la longueur et la complexité via les mécanismes d'authentification **PAM** et le fichier **/etc/login.defs**

Administration des utilisateurs
Principe

Les groupes

- **GID** du groupe accompagne toujours l'utilisateur pour le contrôle de ses droits.
- Un utilisateur peut faire partie de plusieurs groupes, auquel cas il faut distinguer son groupe primaire des groupes secondaires.
- Le groupe primaire est celui qui est toujours appliqué à la création d'un fichier. Si l'utilisateur **seb** a pour groupe primaire **users**, alors les fichiers créés par **seb** auront comme groupe d'appartenance **users**.
- La commande `id"` permet de connaître les informations essentielles sur un utilisateur : **uid**, **gid**, **groupes secondaires**.
- Un fichier est créé par **seb**. Son propriétaire est **seb** et son groupe est le groupe principal de **seb** : **users**.

Les mots de passes

Les mots de passe permettent d'authentifier les utilisateurs. Ils doivent être assez complexes pour ne pas être découverts facilement, mais assez intuitifs pour qu'ils s'en souviennent. Les mots de passe sont cryptés (MD5, DES par exemple) et ne sont pas directement lisibles sous leur forme cryptée par l'utilisateur afin que personne ne puisse tenter de le décrypter via un quelconque traitement.

/etc/passwd

Le fichier **/etc/passwd** contient la liste des utilisateurs du système local.

`Login:password:UID:GID:comment:homedir:shell"`

- Champ 1 : le login ou nom d'utilisateur.
- Champ 2 : sur les vieilles versions, le mot de passe crypté. Si un x est présent, le mot de passe est placé dans **/etc/shadow**. Si c'est un point d'exclamation le compte est verrouillé.
- Champ 3 : le User ID.
- Champ 4 : le GID, c'est-à-dire le groupe principal.
- Champ 5 : un commentaire ou descriptif. C'est un champ d'information.
- Champ 6 : le répertoire de travail, personnel, de l'utilisateur. C'est le répertoire dans lequel il arrive lorsqu'il se connecte.
- Champ 7 : le shell par défaut de l'utilisateur. Mais ce peut être toute autre commande, y compris une commande interdisant la connexion.

/etc/group

Le fichier **/etc/group** contient la définition des groupes d'utilisateurs et pour chacun la liste des utilisateurs dont il est le groupe secondaire. Chaque ligne est composée de quatre champs `Group:password:GID:user1,user2,.."`

- Champ 1 : le nom du groupe.
- Champ 2 : le mot de passe associé. Voyez l'explication juste en dessous.
- Champ 3 : le Group I d.
- Champ 4 : la liste des utilisateurs appartenant à ce groupe.

Un utilisateur a le droit de changer de groupe afin de prendre, temporairement tout du moins, un groupe secondaire comme groupe principal avec la commande **newgrp**.

Administration des utilisateurs

Les fichiers

/etc/shadow

Le fichier **/etc/shadow** accompagne le fichier **/etc/passwd**. C'est là qu'est stocké, entre autres, le mot de passe crypté des utilisateurs. Pour être plus précis il contient toutes les informations sur le mot de passe et sa validité dans le temps. Chaque ligne est composée de 9 champs séparés par des::

```
bean:$ 2a$10 $ AjADxPEfE5iUJcltzYA4wOZO.f2UZOqP/8EnOFY.P.m1OHifS7J8i:13"
913:0:99999:7:::"
```

- Champ 1 : le login..
- Champ 2 : le mot de passe crypté. Le $ xx $ initial indique le type de cryptage..
- Champ 3 : nombre de jours depuis le 1 e r janvier 1970 du dernier changement de mot de passe..
- Champ 4 : nombre de jours avant lesquels le mot de passe ne peut pas être changé (0 : il peut être changé n'importe quand).
- Champ 5 : nombre de jours après lesquels le mot de passe doit être changé.
- Champ 6 : nombre de jours avant l'expiration du mot de passe durant lesquels l'utilisateur doit être prévenu.
- Champ 7 : nombre de jours après l'expiration du mot de passe après lesquels le compte est désactivé.
- Champ 8 : nombre de jours depuis le 1 e r janvier 1970 à partir du moment où le compte a été désactivé.
- Champ 9 : réservé.

Dans l'exemple de la ligne bean, le mot de passe a été changé 13913 jours après le 01/01/1970. Le mot de passe doit être changé avant 0 jours mais il est toujours valide car le champ suivant

Administration des utilisateurs
Gestion des utilisateurs

Ajout

La création dun utilisateur consiste à :

- rajouter une ligne dans **/etc/passwd**
- rajouter une ligne dans **/etc/shadow**
- rajouter d'éventuelles informations dans **/etc/group**
- créer le répertoire personnel et mettre à jour son contenu avec **/etc/skel**
- changer les permissions et le propriétaire du répertoire personnel
- changer le mot de passe (encodé)

la commande `vipw"` met à jour les divers caches associés à la gestion des comptes. La commande `vipw"` admet trois arguments :

- -p : édition de **/etc/passwd**.
- -g : édition de **/etc/group**.
- -s : édition de **/etc/shadow**.

Administration des utilisateurs
Gestion des utilisateurs

Ajout(useradd)

`useradd"` ajoute un nouveau compte et effectue les principales opérations :

- création de lutilisateur et remplissage des fichiers,
- création dun groupe privé dutilisateur (de même nom que celui-ci),
- création du répertoire personnel, remplissage et modification des droits.

`useradd <options> login"`

Options	Rôle
-m	Crée aussi le répertoire personnel
-u	Précise l'UID numérique de l'utilisateur,
-g	Précise le groupe principal de l'utilisateur,
	par GID ou par son nom (variable GROUP).
-G	Précise les groupes additionnels (secondaires, de l'utilisateur) séparés par des virgules (variable GROUPS).
-d	Chemin du répertoire personnel. Généralement /home/<login>,
-c	Un commentaire associé au compte.
-k	Chemin du répertoire contenant le squelette de l'arborescence du répertoire utilisateur. C'est généralement /**etc/skell** (variable SKEL).
-s	Shell (commande de connexion) par défaut de l'utilisateur (variable SHELL).
-p	Le mot de passe de l'utilisateur.

Administration des utilisateurs
Gestion des utilisateurs

Ajout(useradd): exemple

La commande suivante crée le compte robert avec la plupart des options de base précisées. C'est juste un exemple car, sauf parfois le **-m**, si vous ne précisez rien ce sont les options par défaut par rapport à celles précisées dans le fichier **/etc/defaults/useradd**.

```
 useradd -m -u 1010 -g users -G video,dialout,lp -s /bin/bash -d"
/home/robert -c "Compte de Robert" robert"
 grep robert /etc/passwd"
robert:x:1010:100:Compte de Robert:/home/robert:/bin/bash"
```

La commande ne crée pas de mot de passe. Il faut le faire à la main avec la commande `passwd`".

```
 passwd robert"
Changing password for robert."
Nouveau mot de passe :"
Retaper le nouveau mot de passe :"
Mot de passe changé."
```

Administration des utilisateurs
Gestion des utilisateurs

Changer des mots de passes

La commande **passwd** permet de gérer les mots de passe mais aussi les autorisations de connexion et la plupart des champs présents dans **/etc/shadow**.

```
$ id"
uid=1000(seb) gid=100(users)"
$ passwd"
Changing password for seb."
Ancien mot de passe :"
Nouveau mot de passe :"
Retaper le nouveau mot de passe :"
Mot de passe changé."
```

Les modules **PAM** (Pluggable Authentification Module) peuvent imposer des contraintes plus ou moins sévères pour le choix du mot de passe : de telle longueur, pas basé sur un mot du dictionnaire, etc. Voyez ce qu'il se passe en tentant dutiliser successivement toto (trop court), azerty (trop long) et Martine (dictionnaire) :

L'utilisateur root a le droit de modifier les mots de passe de tous les utilisateurs du système, sans avoir à connaître le précédent mot de passe. Mieux : il peut forcer lutilisation dun mot de passe même si celui-ci n'est pas validé par **PAM** :

```
 passwd seb"
Changing password for seb."
Nouveau mot de passe :"
Mot de passe incorrect : basé sur un mot du dictionnaire"
Retaper le nouveau mot de passe :" Mot de passe changé."
```

Administration des utilisateurs
Gestion des utilisateurs

Gérer les informations de validité

Tous les champs de **/etc/shadow** peuvent être modifiés par la commande `passwd"`. Voici quelques options disponibles:

Options	Rôle
-l	Lock : verrouille le compte en rajoutant un ! devant le mot de passe crypté.
-u	Unlock : déverrouille le compte.
-d	(root) Supprime le mot de passe du compte.
-n < j>	(root) Durée de vie minimale en jours du mot de passe.
-x < j>	(root) Durée de vie maximale en jours du mot de passe.
-w < j>	(root) Nombre de jours avant avertissement.
- i < j>	(root) Délai de grâce avant désactivation si le mot de passe est expiré.
-S	(root) Statut du compte.

Exemple: Dans l'exemple suivant le compte bean est modifié comme ceci :

- Il doit attendre 5 jours après saisie d'un nouveau mot de passe pour pouvoir le changer,
- Son mot de passe est valide 45 jours,
- Il est prévenu 7 jours avant qu'il doit changer de mot de passe,
- S'il ne change pas de mot de passe après 45 jours, il dispose encore de 5 jours avant d'être désactivé.

```
 passwd -n 5 -x 45 -w 7 -i 5 bean"
Password expiry information changed."
```

Voici la ligne de **/etc/shadow** associée:

```
bean:$2a$10$dwbuGrC76bs3162V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHuCR2:13
```

Administration des utilisateurs
Gestion des utilisateurs

La commande **chage**

La commande `chage"` permet de faire à peu près la même chose. Elle n'est accessible que par *root*. Lancée sans autre argument que le login de l'utilisateur, elle est interactive.

```
 chage bean"
Changing aging information for bean."
Minimum Password Age [7]:"
Maximum Password Age [40]:"
Password Expiration Warning [10]:"
Password Inactive [ 5 ]:"
Last Password Change (YYYY-MM-DD) [2008-04-10]:"
Account Expiration Date (YYYY-MM-DD) [1969-12-31]: 2010-01-01"
Aging information changed."
```

Voici la ligne **/etc/shadow** résultante :

```
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:13"
```

Options	Rôle
-l	Lock : verrouille le compte en rajoutant un ! devant le mot de passe crypté.
-m	Mindays : équivaut à passwd - n .
-M	Maxdays : équivaut à passwd -x.
-d	Date de derni modification du mot de passe (depuis le 01/01/1970).
-E	Date dexpiration du mot de passe (depuis le 01/01/1970).
-i	Inactive : équivaut à passwd - i .
-w	Warndays : équivaut à passwd -w.
-l	List : affiche tous les détails.

Administration des utilisateurs
Gestion des utilisateurs

Modification

Utilisez la commande `usermod"` pour modifier un compte. Elle prend la même syntaxe et les mêmes options que `useradd"` mais dispose aussi d'une syntaxe complémentaire qui nécessite quelques précisions.

Options	Rôle
-l	Lock du compte, comme passwd - l .
-u	Unlock du compte, comme passwd - u .
-e <n>	Expire : le mot de passe expire n jours apre 01/01/1970.
-u <UID>	Modifie l'UID associé au login.
-l <login>	Modifie le nom de login.
-m	Move : implique la présence de -d pour préciser un nouveau répertoire personnel.

Supression

Supprimez un utilisateur avec la commande `userdel"`. Par défaut le répertoire personnel n'est pas supprimé. Vous devez pour ceci passer l'option -r.

```
userdel -r bean"
```

Administration des utilisateurs
Gestion des groupes

Ajout

Vous pouvez créer un groupe directement dans le fichier **/etc/group** ou bien passer par les commandes associées. Si vous éditez le fichier à la main, utilisez la commande `vigr"` (ou **vipw -g**). La commande **groupadd** permet de créer un groupe. Sa syntaxe simple accepte largument **-g** pour préciser un GID précis.

```
 grep amis /etc/group"
amis:!:1234:"
```

Modification

La commande `groupmod"` permet de modifier un groupe. Ses paramètres sont les suivants :

Options	Rôle
-n <nom>	Renomme le groupe.
-g <GID>	Modifie le GID.
-A <user>	Ajoute l'utilisateur spécifié dans le groupe (groupe secondaire).

```
 groupmod -R seb amis"
 grep amis /etc/group"
amis:!:1234:"
```

Supression

La commande `groupdel"` supprime un groupe. La commande vérifie d'abord si le groupe que vous voulez supprimer est le groupe principal d'un utilisateur. Dans ce cas le groupe ne peut pas être supprimé.

```
 groupdel amis"
```

Administration des utilisateurs
Commandes additionnelles

Conversion des fichiers

Quelques systèmes Unix n'utilisent pas par dut (il faut l'activer après) la gestion des comptes avec les fichiers **shadow**. Dans ce cas il peut être nécessaire de convertir les fichiers **/etc/shadow** et **/etc/passwd** en un seul et unique **/etc/passwd**. C'est le rôle de la commande `pwunconv"`.
Dans l'exemple suivant, le fichier **/etc/passwd** est converti. Une fois la commande exécutée, toute trace de **/etc/shadow** a disparu.

```
 pwunconv"
 grep bean /etc/passwd"
bean:$2a$10$dwbUGrC75bs3l52V5DHxZefkZyB6VTHsLH5ndjsNe/vF/HAzHOcR2:1001:100:toto:..."
 ls -l /etc/shadow"
ls: ne peut accr /etc/shadow: Aucun fichier ou rrtoire de ce type"
```

Vérification de la cohérence

La commande `pwck"` effectue une vérification des fichiers /**etc/passwd** et /**etc/shadow** et reporte les erreurs.

```
 pwck"
Checking ‘/etc/passwd"
User ‘suse-ncc: directory ‘/var/lib/YaST2/suse-ncc-fakehome does"
not exist."
User ‘bean: unknown group ‘14400"
User ‘bean: shell ‘/bin/bashr is not executable."
Checking ‘/etc/shadow."
```

La commande `grpck"` fait la même chose pour les groupes.

Véfification de la connexion

La commande `lastlog"` se base sur le contenu de **/var/log/lastlog**. Elle accepte les paramètres **-u** (précision d'un utilisateur) et **-t** pour rechercher les connexions des **n** derniers jours.
La commande `last"` fait à peu près la même chose, mais se base sur **/var/log/wtmp** qui fournit des informations supplémentaires

/etc/default/useradd

Le fichier **/etc/default/useradd** contient un certain nombre de variables définissant les règles par défaut à appliquer à la création d'un utilisateur:

- son groupe
- la racine de son répertoire personnel (là où celui-ci sera situé)
- s'il est actif ou non
- le shell
- son ou ses groupes secondaires
- l'endroit où est situé le squelette des comptes (structure de base d'un répertoire utilisateur)
- la création ou non d'un spool (dépôt) de courrier

```
$ cat /etc/defaults/useradd"
GROUP=100"
HOME=/home"
INACTIVE=-1"
EXPIRE="
SHELL=/bin/bash"
SKEL=/etc/skel"
GROUPS=video,dialout"
CREATE_MAIL_SPOOL=no"
```

/etc/default/passwd

Le fichier **/etc/default/passwd** contient quelques règles utilisées par la commande `passwd"` pour le cryptage des mots de passe. Il est possible de définir des règles de cryptage globales, mais aussi par type de fichier, et de passer quelques options selon la méthode.

```
$ cat /etc/default/passwd"
 Cryptage par defaut"
CRYPT=md5"
 Cryptage pour les fichiers (/etc/shadow)"
CRYPT_FILES=blowfish"
 option pour blowfish"
BLOWFISH_CRYPT_FILES=10"
 Pour NIS"
CRYPT_YP=des"
```

/etc/default/su

Le fichier **/etc/default/su** permet de configurer le fonctionnement de la commande `su"`. Par défaut **su** avec le paramètre **-m** et en place un nouveau PATH car il charge l'environnement de l'utilisateur ciblé. Vous pouvez modifier ceci et mettre en place votre propre PATH, ou conserver lancien.

```
 Change le PATH meme sans le tiret"
ALWAYS_SET_PATH=no"
 Path par défaut"
PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin"
 Path par défaut pour root"
SUPATH=/usr/sbin:/bin:/usr/bin:/sbin:/usr/X11R6/bin"
```

/etc/issue

Lorsqu'un utilisateur se connecte depuis la console, un message est généralement affiché juste avant l'invite de saisie de son login . Ce message est contenu dans le fichier **/etc/issue**. C'est un message d'accueil et à ce titre il peut contenir tout ce que vous voulez. Par défaut, il contient généralement le nom de la distribution Linux et le numéro de version du noyau.

```
$ cat issue"
Mandriva Linux release 2008.1 (Official) for i586"
Kernel 2.6.24.4-desktop-1mnb on an i686 / ł"
```

/etc/issue.net

Le message d'accueil peut être différent lorsqu'un utilisateur se connecte depuis une console distante (telnet, ssh, etc.). C'est souvent le m mais sans les caractères de contrôles liés à un shell donné. Pour modifier ce message spécifique, éditez le contenu du fichier **/etc/issue.net**.

/etc/motd.

Motd signifie Message of the day, le message du jour. Une fois l'utilisateur connecté depuis une console (locale ou distante), un message peut être affiché. L'administrateur peut modifier ce message en éditant le fichier **/etc/motd**. Par défaut il est vide. Vous pouvez par exemple modifier ce fichier pour prévenir vos utilisateurs qu'un reboot de maintenance aura lieu tel jour à telle heure, ceci évitant d'envoyer **n** mails

Merci pour votre attention!