

La gestion des utilisateurs

# Table des matières

1. Généralités .....	2
2. Gestion des groupes .....	3
2.1. Commande groupadd .....	3
2.2. Commande groupmod .....	4
2.3. Commande groupdel .....	5
2.4. Fichier /etc/group .....	5
Fichier /etc/gshadow .....	6
3. Gestion des utilisateurs .....	7
3.1. Définition .....	7
3.2. Commande useradd .....	7
Valeur par défaut de création d'utilisateur. ....	9
3.3. Commande usermod .....	9
3.4. Commande userdel .....	11
3.5. Fichier /etc/passwd .....	12
3.6. Fichier /etc/shadow .....	12
4. Propriétaires des fichiers .....	14
4.1. Commandes de modifications : .....	14
Commande chown .....	14
4.2. Commande chgrp .....	15
5. Gestion des invités .....	16
5.1. Commande gpasswd .....	16
5.2. Commande id .....	16
5.3. Commande newgrp .....	17
6. Sécurisation .....	18
6.1. Commande passwd .....	18
7. Commande chage .....	20
8. Gestion avancée .....	22
8.1. Fichier /etc/default/useradd .....	22
8.2. Fichier /etc/login.defs .....	23
8.3. Fichier /etc/skel .....	23
9. Changement d'identité .....	24
9.1. Commande su .....	24
Chargement du profil .....	24

## Objectifs

- ☑ ajouter, supprimer ou modifier un **groupe** ;
- ☑ ajouter, supprimer ou modifier un **utilisateur** ;
- ☑ connaître la syntaxe des fichiers associés à la gestion des groupes et des utilisateurs ;
- ☑ changer le **propriétaire** ou le **groupe propriétaire** d'un fichier ;
- ☑ **sécuriser** les comptes utilisateurs ;
- ☑ changer d'identité.

# Chapitre 1. Généralités

Chaque utilisateur est membre d'au moins un groupe : **c'est son groupe principal**.

Plusieurs utilisateurs peuvent faire partie d'un même groupe.

Les utilisateurs peuvent appartenir à d'autres groupes. Ces utilisateurs sont **invités** dans ces **groupes secondaires**.



Chaque utilisateur possède un groupe principal et peut être invité dans un ou plusieurs groupes secondaires.

Les groupes et utilisateurs se gèrent par leur identifiant numérique unique **GID** et **UID**.

Les fichiers de configuration se trouvent dans **/etc**.

## UID

User IDentifier. Identifiant unique d'utilisateur.

## GID

Group IDentifier. Identifiant unique de groupe.



Il est recommandé d'utiliser les commandes d'administration au lieu de modifier manuellement les fichiers.

## Chapitre 2. Gestion des groupes

Fichiers modifiés, ajout de lignes :

- `/etc/group`
- `/etc/gshadow`

### 2.1. Commande groupadd

La commande `groupadd` permet d'ajouter un groupe au système.

*Syntaxe de la commande groupadd*

```
groupadd [-f] [-g GID] groupe
```

Exemple :

```
[root]# groupadd -g 512 GroupeB
```

*Table 1. Options de la commande groupadd*

Option	Description
<code>-g GID</code>	<code>GID</code> du groupe à créer.
<code>-f</code>	Le système choisit un <code>GID</code> si celui précisé par l'option <code>-g</code> existe déjà.
<code>-r</code>	Crée un groupe système avec un <code>GID</code> compris entre <code>SYS_GID_MIN</code> et <code>SYS_GID_MAX</code> . Ces deux variables sont définies dans <code>/etc/login.defs</code> .

Règles de nommage des groupes :

- Pas d'accents, ni caractères spéciaux ;
- Différents du nom d'un utilisateur ou fichier système existant.

Sous **Debian**, l'administrateur devrait privilégier, sauf dans des scripts ayant la vocation d'être portables vers toutes les distributions Linux, les commandes `addgroup/delgroup` comme précisé dans le man :



```
[root] # man addgroup
```

DESCRIPTION

`adduser` et `addgroup` ajoutent des utilisateurs ou des groupes au système en fonction des options fournies en ligne de commande et des informations contenues dans le fichier de configuration `/etc/adduser.conf`. Ce sont des interfaces plus conviviales que les programmes `useradd` et `groupadd`. Elles permettent de choisir par défaut des UID ou des GID conformes à la charte Debian, de créer un répertoire personnel configuré suivant un modèle (squelette), d'utiliser un script sur mesure, et d'autres fonctionnalités encore.

## 2.2. Commande `groupmod`

La commande `groupmod` permet de modifier un groupe existant sur le système.

*Syntaxe de la commande `groupmod`*

```
groupmod [-g GID] [-n nom] groupe
```

Exemple :

```
[root]# groupmod -g 516 GroupeP
[root]# groupmod -n GroupeC GroupeB
```

*Table 2. Options de la commande `groupmod`*

Option	Description
<code>-g GID</code>	Nouveau <b>GID</b> du groupe à modifier.
<code>-n nom</code>	Nouveau nom.

Il est possible de modifier le nom d'un groupe, son **GID** ou les deux simultanément.

Après modification, les fichiers appartenant au groupe ont un **GID** inconnu. Il faut leur réattribuer le nouveau **GID**.

```
[root]# find / -gid 502 -exec chgrp 516 {} \;
```

## 2.3. Commande `groupdel`

La commande `groupdel` permet de supprimer un groupe existant sur le système.

*Syntaxe de la commande `groupdel`*

```
groupdel groupe
```

Exemple :

```
[root]# groupdel GroupeC
```



Pour être supprimé, un groupe ne doit plus contenir d'utilisateurs.

La suppression du dernier utilisateur d'un groupe éponyme entraînera la suppression de ce groupe par le système.



Chaque groupe possède un **GID** unique. Un groupe peut être dupliqué. Par convention, les **GID** des groupes systèmes vont de 0 (**root**) à 499.



Un utilisateur faisant obligatoirement partie d'un groupe, il est nécessaire de créer les groupes avant d'ajouter les utilisateurs. Par conséquent, un groupe peut ne pas avoir de membres.

## 2.4. Fichier `/etc/group`

Ce fichier contient les informations de groupes (séparées par `:`).

```
[root]# tail -1 /etc/group
GroupeP:x:516:stagiaire
(1) (2)(3) (4)
```

1

Nom du groupe.

2

Mot de passe (**x** si défini dans `/etc/gshadow`).

3

GID.

4

Membres invités (séparés par des virgules, ne contient pas les membres principaux).



Chaque ligne du fichier `/etc/group` correspond à un groupe. Les utilisateurs dont ce groupe est leur groupe principal ne sont pas listés à ce niveau.

Cette information d'appartenance est en fait déjà fournie par le fichier `/etc/passwd...`

## Fichier `/etc/gshadow`

Ce fichier contient les informations de sécurité sur les groupes (séparées par `:`).

```
[root]# grep GroupeA /etc/gshadow
GroupeA:$6$2,9,v...5Bn160:alain:stagiaire
(1)      (2)      (3)      (4)
```

1

Nom du groupe.

2

Mot de passe chiffré.

3

Administrateur du groupe.

4

Membres invités (séparés par des virgules, ne contient pas les membres principaux).



Pour chaque ligne du fichier `/etc/group` doit correspondre une ligne du fichier `/etc/gshadow`.

Un **!** au niveau du mot de passe indique que celui-ci est bloqué. Ainsi aucun utilisateur ne peut utiliser le mot de passe pour accéder au groupe (sachant que les membres du groupe n'en ont pas besoin).



# Chapitre 3. Gestion des utilisateurs

## 3.1. Définition

Un utilisateur se définit comme suit dans le fichier `/etc/passwd` :

- 1  
Login ;
- 2  
Mot de passe ;
- 3  
UID ;
- 4  
GID du groupe principal ;
- 5  
Commentaire ;
- 6  
Répertoire de connexion ;
- 7  
Interpréteur de commandes (`/bin/bash`, `/bin/nologin`,...).

Il existe trois types d'utilisateurs :

- **root** : Administrateur du système ;
- **utilisateur système** : Utilisé par le système pour la gestion des droits d'accès des applications ;
- **utilisateur ordinaire** : Autre compte permettant de se connecter au système.

Fichiers modifiés, ajout de lignes :

- `/etc/passwd`
- `/etc/shadow`

## 3.2. Commande useradd

La commande `useradd` permet d'ajouter un utilisateur.

*Syntaxe de la commande useradd*

```
useradd [-u UID] [-g GID] [-d répertoire] [-s shell] login
```

Exemple :

```
[root]# useradd -u 1000 -g 513 -d /home/GroupeC/carine carine
```

*Table 3. Options de la commande useradd*

Option	Description
<b>-u UID</b>	<b>UID</b> de l'utilisateur à créer.
<b>-g GID</b>	<b>GID</b> du groupe principal.
<b>-d répertoire</b>	Répertoire de connexion.
<b>-s shell</b>	Interpréteur de commandes.
<b>-c</b>	Ajoute un commentaire.
<b>-U</b>	Ajoute l'utilisateur à un groupe portant le même nom créé simultanément.
<b>-M</b>	Ne crée pas le répertoire de connexion.

À la création, le compte ne possède pas de mot de passe et est verrouillé. Il faut assigner un mot de passe pour déverrouiller le compte.

Règles de nommage des comptes :

- Pas d'accents, de majuscules ni caractères spéciaux ;
- Différents du nom d'un groupe ou fichier système existant ;
- Définir les options **-u**, **-g**, **-d** et **-s** à la création.



L'arborescence du répertoire de connexion doit être créée à l'exception du dernier répertoire. Le dernier répertoire est créé par la commande **useradd** qui en profite pour y copier les fichiers du **skel**.

**Un utilisateur peut faire partie de plusieurs groupes en plus de son groupe principal.**

Pour les groupes secondaires, il faut utiliser l'option **-G**.

Exemple :

```
[root]# useradd -u 500 -g GroupeA -G GroupeP,GroupeC albert
```



Sous **Debian**, il faudra spécifier l'option **-m** pour forcer la création du répertoire de connexion ou positionner la variable **CREATE\_HOME** du fichier **login.defs**. Dans tous les cas, l'administrateur devrait privilégier, sauf dans des scripts ayant la vocation d'être portables vers toutes les distributions Linux, les commandes **adduser/debuser** comme précisé dans le **man** :

```
[root] # man useradd
DESCRIPTION
    **useradd** is a low level utility for adding users. On Debian,
    administrators should usually use **adduser(8)** instead.
```

### Valeur par défaut de création d'utilisateur.

Modification du fichier **/etc/default/useradd**.

```
useradd -D [-b répertoire] [-g groupe] [-s shell]
```

Exemple :

```
[root]# useradd -D -g 500 -b /home -s /bin/bash
```

Table 4. Options de la commande **useradd** pour modifier les valeurs par défaut

Option	Description
<b>-D</b>	Définit les valeurs par défaut de création d'utilisateur.
<b>-b répertoire</b>	Définit le répertoire de connexion par défaut.
<b>-g groupe</b>	Définit le groupe par défaut.
<b>-s shell</b>	Définit le shell par défaut.
<b>-f</b>	Nombre de jours suivant l'expiration du mot de passe avant que le compte ne soit désactivé.
<b>-e</b>	Date à laquelle le compte sera désactivé.

### 3.3. Commande **usermod**

La commande **usermod** permet de modifier un utilisateur.

*Syntaxe de la commande **usermod***

```
usermod [-u UID] [-g GID] [-d répertoire] [-m] login
```

Exemple :

```
[root]# usermod -u 544 carine
```

Options identiques à la commande **useradd**.

Table 5. Options de la commande **usermod**

Option	Description
<b>-m</b>	Associé à l'option <b>-d</b> , déplace le contenu de l'ancien répertoire de connexion vers le nouveau.
<b>-l login</b>	Nouveau nom.
<b>-e AAAA-MM-JJ</b>	Date d'expiration du compte.
<b>-L</b>	Verrouille le compte.
<b>-U</b>	Déverrouille le compte.
<b>-a</b>	Empêche la suppression de l'utilisateur d'un groupe secondaire lors de l'ajout dans un autre groupe secondaire.
<b>-G</b>	Précise plusieurs groupes secondaires lors de l'ajout.

Avec la commande **usermod**, le verrouillage d'un compte se traduit par l'ajout de **!** devant le mot de passe dans le fichier **/etc/shadow**.



Pour être modifié un utilisateur doit être déconnecté et ne pas avoir de processus en cours.

Après modification de l'identifiant, les fichiers appartenant à l'utilisateur ont un **UID** inconnu. Il faut leur réattribuer le nouvel **UID**.

```
[root]# find / -uid 1000 -exec chown 544: {} \;
```

Il est possible d'inviter un utilisateur dans un ou plusieurs groupes secondaires avec les options **-a** et **-G**.

Exemple :

```
[root]# usermod -aG GroupeP,GroupeC albert
```

La commande **usermod** agit en modification et non en ajout.

Pour un utilisateur invité dans un groupe par l'intermédiaire de cette commande et déjà positionné comme invité dans d'autres groupes secondaires, il faudra indiquer dans la commande de gestion de groupe tous les groupes dont il fait partie sinon il disparaîtra de ceux-ci.

L'option **-a** modifie ce comportement.

Exemples :

- Invite **albert** dans le groupe **GroupeP**

```
[root]# usermod -G GroupeP albert
```

- Invite **albert** dans le groupe **GroupeG**, mais le supprime de la liste des invités de **GroupeP**.

```
[root]# usermod -G GroupeG albert
```

- Donc soit :

```
[root]# usermod -G GroupeP,GroupeG albert
```

- Soit :

```
[root]# usermod -aG GroupeG albert
```

## 3.4. Commande userdel

La commande **userdel** permet de supprimer le compte d'un utilisateur.

*Syntaxe de la commande userdel*

```
[root]# userdel -r carine
```

*Table 6. Options de la commande userdel*

Option	Description
<b>-r</b>	Supprime le répertoire de connexion et les fichiers contenus.



Pour être supprimé, un utilisateur doit être déconnecté et ne pas avoir de processus en cours.

**userdel** supprime la ligne de l'utilisateur dans les fichiers **/etc/passwd** et **/etc/gshadow**.



Chaque utilisateur possède un **UID** unique. Par convention, les **UID** des utilisateurs 'système' vont de **0 (root)** à **499**.



Un utilisateur est obligatoirement membre d'un groupe. Il est donc nécessaire de créer les groupes avant d'ajouter les utilisateurs.

## 3.5. Fichier `/etc/passwd`

Ce fichier contient les informations des utilisateurs (séparées par `:`).

```
[root]# head -1 /etc/passwd
root:x:0:0:root:/root:/bin/bash
(1)(2)(3)(4)(5) (6) (7)
```

1

Login.

2

Mot de passe (x si défini dans `/etc/shadow`).

3

UID.

4

GID du groupe principal.

5

Commentaire.

6

Répertoire de connexion.

7

Interpréteur de commandes.

## 3.6. Fichier `/etc/shadow`

Ce fichier contient les informations de sécurité des utilisateurs (séparées par `:`).

```
[root]# tail -1 /etc/shadow
root:$6$. . . :15399:0:99999:7:::
(1) (2) (3) (4) (5) (6)(7,8,9)
```

1

Login.

2

Mot de passe chiffré.

3

Date du dernier changement.

4

Durée de vie minimale du mot de passe.

5

Durée de vie maximale du mot de passe.

6

Nombre de jours avant avertissement.

7

Délai avant désactivation du compte après expiration.

8

Délai d'expiration du compte.

9

Réservé pour une utilisation future.



Pour chaque ligne du fichier `/etc/passwd` doit correspondre une ligne du fichier `/etc/shadow`.

## Chapitre 4. Propriétaires des fichiers



Tous les fichiers appartiennent forcément à un utilisateur et à un groupe.

Le groupe principal de l'utilisateur qui crée le fichier est, par défaut, le groupe propriétaire du fichier.

### 4.1. Commandes de modifications :

#### Commande `chown`

La commande `chown` permet de modifier les propriétaires d'un fichier.

*Syntaxe de la commande `chown`*

```
chown [-R] [-v] login[:groupe] fichier
```

Exemples :

```
[root]# chown root fichier
[root]# chown albert:GroupeA fichier
```

*Table 7. Options de la commande `chown`*

Option	Description
<code>-R</code>	Modifie les propriétaires du répertoire et de son contenu.
<code>-v</code>	Affiche les modifications exécutées.

Pour ne modifier que l'utilisateur propriétaire :

```
[root]# chown albert fichier
```

Pour ne modifier que le groupe propriétaire :

```
[root]# chown :GroupeA fichier
```

Modification de l'utilisateur et du groupe propriétaire :

```
[root]# chown albert:GroupeA fichier
```

Dans l'exemple suivant le groupe attribué sera le groupe principal de l'utilisateur précisé.



```
[root]# chown albert: fichier
```

## 4.2. Commande chgrp

La commande **chgrp** permet de modifier le groupe propriétaire d'un fichier.

*Syntaxe de la commande chgrp*

```
chgrp [-R] [-v] groupe fichier
```

Exemple :

```
[root]# chgrp groupe1 fichier
```

*Table 8. Options de la commande chgrp*

Option	Description
<b>-R</b>	Modifie les groupes propriétaires du répertoire et de son contenu (récursivité).
<b>-v</b>	Affiche les modifications exécutées.



Il est possible d'appliquer à un fichier un propriétaire et un groupe propriétaire en prenant comme référence ceux d'un autre fichier :

```
chown [options] --reference=RRFILE FILE
```

Par exemple :

```
chown --reference=/etc/groups /etc/passwd
```

## Chapitre 5. Gestion des invités

### 5.1. Commande `gpasswd`

La commande `gpasswd` permet de gérer un groupe.

*Syntaxe de la commande `gpasswd`*

```
gpasswd [-a login] [-A login] [-d login] [-M login] groupe
```

Exemples :

```
[root]# gpasswd -A alain GroupeA
[alain]$ gpasswd -a patrick GroupeA
```

*Table 9. Options de la commande `gpasswd`*

Option	Description
<code>-a login</code>	Ajoute l'utilisateur au groupe.
<code>-A login</code>	Définit l'administrateur du groupe.
<code>-d login</code>	Retire l'utilisateur du groupe.
<code>-M login</code>	Définit la liste exhaustive des invités.

La commande `gpasswd -M` agit en modification et non en ajout.

```
# gpasswd GroupeA
New Password :
Re-enter new password :
```

### 5.2. Commande `id`

La commande `id` affiche les noms des groupes d'un utilisateur.

*Syntaxe de la commande `id`*

```
id login
```

Exemple :

```
[root]# id alain
uid=500(alain) gid=500(GroupeA) groupes=500(GroupeA),516(GroupeP)
```

## 5.3. Commande newgrp

La commande `newgrp` permet d'utiliser temporairement un groupe secondaire pour la création de fichiers.

*Syntaxe de la commande newgrp*

```
newgrp [groupesecondaire]
```

Exemple :

```
[alain]$ newgrp GroupeB
```



Après utilisation de cette commande, les fichiers seront créés avec le **GID** de son groupe secondaire.

La commande `newgrp` sans paramètre réaffecte le groupe principal.

## Chapitre 6. Sécurisation

### 6.1. Commande passwd

La commande `passwd` permet de gérer un mot de passe.

*Syntaxe de la commande passwd*

```
passwd [-d] [-l] [-S] [-u] [login]
```

Exemples :

```
[root]# passwd -l albert
[root]# passwd -n 60 -x 90 -w 80 -i 10 patrick
```

*Table 10. Options de la commande passwd*

Option	Description
<code>-d</code>	Supprime le mot de passe.
<code>-l</code>	Verrouille le compte.
<code>-S</code>	Affiche le statut du compte.
<code>-u</code>	Déverrouille le compte.
<code>-e</code>	Fait expirer le mot de passe.
<code>-n jours</code>	Durée de vie minimale du mot de passe.
<code>-x jours</code>	Durée de vie maximale du mot de passe.
<code>-w jours</code>	Délai d'avertissement avant expiration.
<code>-i jours</code>	Délai avant désactivation lorsque le mot de passe expire.

Avec la commande `passwd`, le verrouillage d'un compte se traduit par l'ajout de `!!` devant le mot de passe dans le fichier `/etc/shadow`.

L'utilisation de la commande `usermod -U` ne supprime qu'un seul des `!`. Le compte reste donc verrouillé.

Exemple :

- Alain change son mot de passe :

```
[alain]$ passwd
```

- root change le mot de passe d'Alain :

```
[root]# passwd alain
```



La commande **passwd** est accessible aux utilisateurs pour modifier leur mot de passe (l'ancien mot de passe est demandé). L'administrateur peut modifier les mots de passe de tous les utilisateurs sans restriction.

Ils devront se soumettre aux restrictions de sécurité.

Lors d'une gestion des comptes utilisateurs par script shell, il peut être utile de définir un mot de passe par défaut après avoir créé l'utilisateur.

Ceci peut se faire en passant le mot de passe à la commande **passwd**.

Exemple :

```
[root]# echo "azerty,1" | passwd --stdin philippe
```



Le mot de passe est saisi en clair, **passwd** se charge de le chiffrer.

## Chapitre 7. Commande chage

La commande **chage** permet de gérer la stratégie de compte.

### Syntaxe de la commande chage

```
chage [-d date] [-E date] [-I jours] [-l] [-m jours] [-M jours] [-W jours] [login]
```

Exemple :

```
[root]# chage -m 60 -M 90 -W 80 -I 10 alain
```

Table 11. Options de la commande chage

Option	Description
<b>-I jours</b>	Délai avant désactivation, mot de passe expiré ( <b>i</b> majuscule).
<b>-l</b>	Affiche le détail de la stratégie ( <b>l</b> minuscule).
<b>-m jours</b>	Durée de vie minimale du mot de passe.
<b>-M jours</b>	Durée de vie maximale du mot de passe.
<b>-d AAAA-MM-JJ</b>	Dernière modification du mot de passe.
<b>-E AAAA-MM-JJ</b>	Date d'expiration du compte.
<b>-W jours</b>	Délai d'avertissement avant expiration.

La commande **chage** propose également un mode interactif.

L'option **-d** force la modification du mot de passe à la connexion.

Exemples :

```
[root]# chage philippe
[root]# chage -d 0 philippe
```



En l'absence d'utilisateur précisé, la commande concernera l'utilisateur qui la saisit.

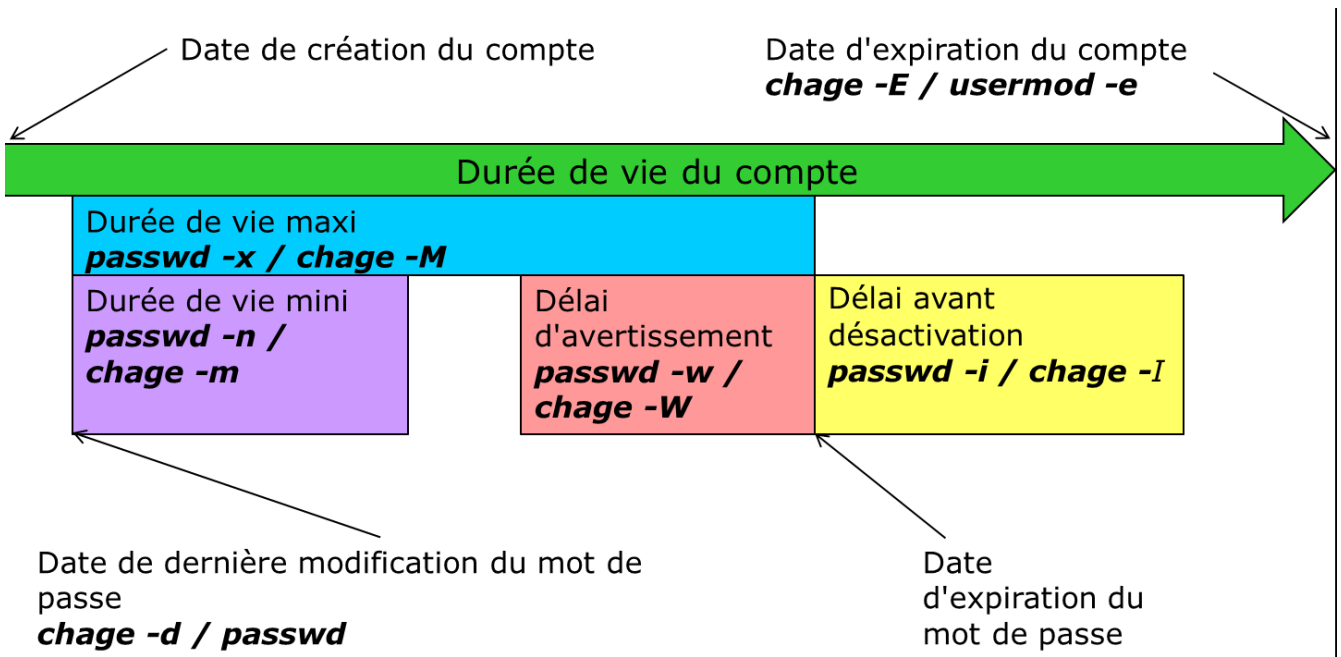


Figure 1. Gestion des comptes utilisateurs avec chage

## Chapitre 8. Gestion avancée

Fichiers de configuration :

- `/etc/default/useradd`
- `/etc/login.defs`
- `/etc/skel`



L'édition du fichier `/etc/default/useradd` se fait grâce à la commande `useradd`.

Les autres fichiers sont à modifier avec un éditeur de texte.

### 8.1. Fichier `/etc/default/useradd`

Ce fichier contient le paramétrage des données par défaut.



Lors de la création d'un utilisateur, si les options ne sont pas précisées, le système utilise les valeurs par défaut définies dans `/etc/default/useradd`.

Ce fichier est modifié par la commande `useradd -D` (`useradd -D` saisie sans autre option affiche le contenu du fichier `/etc/default/useradd`).

Table 12. Contenu du fichier `/etc/default/useradd`

Valeur	Commentaire
<code>GROUP</code>	Groupe par défaut.
<code>HOME</code>	Chemin dans lequel le répertoire de connexion du nom de l'utilisateur sera créé.
<code>INACTIVE</code>	Nombre de jours suivant l'expiration du mot de passe avant que le compte ne soit désactivé.
<code>EXPIRE</code>	Date d'expiration du compte.
<code>SHELL</code>	Interpréteur de commandes.
<code>SKEL</code>	Répertoire squelette du répertoire de connexion.
<code>CREATE_MAIL_SPOOL</code>	Création de la boîte aux lettres dans <code>/var/spool/mail</code> .



Sans l'option `-g`, la commande `useradd` crée un groupe du nom de l'utilisateur et l'y place.

Pour que la commande `useradd` récupère la valeur du champ `GROUP` du fichier `/etc/default/useradd`, il faut préciser l'option `-N`.

Exemple :



```
[root]# useradd -u 501 -N GroupeA
```

## 8.2. Fichier `/etc/login.defs`

Ce fichier contient de nombreux paramètres par défaut utiles aux commandes de création ou de modification d'utilisateurs. Ces informations sont regroupées par paragraphe en fonction de leur utilisation :

- Boîtes aux lettres ;
- Mots de passe ;
- UID et GID ;
- Umask ;
- Connexions ;
- Terminaux.

## 8.3. Fichier `/etc/skel`

Lors de la création d'un utilisateur, son répertoire personnel et ses fichiers d'environnement sont créés.

Ces fichiers sont copiés automatiquement à partir du répertoire `/etc/skel`.

- `.bash_logout`
- `.bash_profile`
- `.bashrc`

Tous les fichiers et répertoires placés dans ce répertoire seront copiés dans l'arborescence des utilisateurs lors de leur création.

# Chapitre 9. Changement d'identité

## 9.1. Commande su

La commande **su** permet de modifier l'identité de l'utilisateur connecté.

*Syntaxe de la commande su*

```
su [-] [-c commande] [login]
```

Exemples :

```
[root]# su - alain
[albert]$ su -c "passwd alain"
```

*Table 13. Options de la commande su*

Option	Description
-	Charge l'environnement complet de l'utilisateur.
-c commande	Exécute la commande sous l'identité de l'utilisateur.

Si le login n'est pas spécifié, ce sera **root**.

Les utilisateurs standards devront taper le mot de passe de la nouvelle identité.



Il y a création de 'couches' successives (un empilement d'environnement **bash**). Pour passer d'un utilisateur à un autre, il faut d'abord taper la commande **exit** pour reprendre son identité puis la commande **su** pour prendre une autre identité.

### Chargement du profil

**root** endosse l'identité de l'utilisateur **alain** avec **su** :

```
...
/home/GroupeA/alain/bash_rc
/etc/bashrc
...
```

**root** endosse l'identité de l'utilisateur **alain** avec **su -** :

```
...  
/home/GroupeA/alain/bash_profile  
/home/GroupeA/alain/bash_rc  
/etc/bashrc  
...
```

Un utilisateur peut endosser temporairement (pour une autre commande ou une session entière) l'identité d'un autre compte.

Si aucun utilisateur n'est précisé, la commande concernera **root** (**su -**).

Il est nécessaire de connaître le mot de passe de l'utilisateur dont l'identité est endossé sauf si c'est **root** qui exécute la commande.

Un administrateur peut ainsi travailler sur un compte utilisateur standard et n'utiliser les droits du compte **root** que ponctuellement.