

Pascal Picard

Corto E.T.F., K&M

Sainte-Clotilde, Ile de la Réunion

Copyright © 2002-2005, 2006 Pascal PICARD, pascal@seth.homeunix.net

Permission to use, copy, modify, and distribute this documentation for any purpose with or without fee is here by granted, provided that the above copyright notice and this permission notice appear in all copies.

THE DOCUMENTATION IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS DOCUMENTATION INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS DOCUMENTATION.

1. Ajouter/Enlever un utilisateur
2. Analyse du fichier `/etc/passwd`
3. Analyse du fichier `/etc/group`
4. Sécurisation des comptes
 - 4.1. *GNU/Linux*
 - 4.2. *xBSD*
5. Commandes de gestion des utilisateurs
 - 5.1. Gérer les utilisateurs avec `user{add,mod,del}` sous *GNU/Linux*
 - 5.2. Gérer les groupes avec `group{add,mod,del}` sous *GNU/Linux*
 - 5.3. Cas particulier de *FreeBSD*
 - 5.4. Vérification
 - 5.5. Edition en ligne
6. Autres commandes
7. Exercices
- Références

Résumé

Nous abordons dans cette partie la gestion des comptes d'utilisateurs (création, modification et suppression) qui est une des premières tâches de l'administrateur système.

Un utilisateur désigne soit une personne réelle qui doit pouvoir se connecter au système, utiliser des ressources et se déconnecter ou bien un pseudo-utilisateur détenant certaines ressources et exécutant certains programmes.

Les informations concernant la définition d'un compte utilisateur sont principalement stockées dans les fichiers `/etc/passwd` et `/etc/group`, et selon qu'il s'agisse d'un système *GNU/Linux* ou d'un système *xBSD* dans les fichiers `/etc/shadow` et `/etc/master.passwd`. Avant d'en présenter la structure syntaxique, intéressons nous à déterminer ce qu'implique la création et la destruction d'un compte. Enfin, pour finir nous présenterons les principales commandes de gestion des comptes.

Pour nous joindre : [<pascal@seth.homeunix.net>](mailto:pascal@seth.homeunix.net)

1. Ajouter/Enlever un utilisateur

Créer un nouveau compte utilisateur requiert plusieurs étapes :

- Attribuer un nom de login, un `uid`, un groupe principal et un ou des groupes secondaires.
- Créer des enregistrements sur cet utilisateur dans les fichiers `/etc/passwd`, `/etc/group` et `/etc/master.passwd` ou `/etc/shadow`.
- Attribuer un (bon) mot de passe.
- Créer le repertoire de domiciliation de l'utilisateur (son `home`).
- Paramétrer le compte de l'utilisateur (durée de validité du mot de passe, date d'expiration du compte, limitation des ressources, privilèges ...).
- Copier les fichiers d'initialisation dans le repertoire de domiciliation de l'utilisateur.
- Rendre l'utilisateur propriétaire de son espace.
- Définir les paramètres supplémentaires, tels les *quotas disk*, l'impression, la messagerie...

Enlever un compte utilisateur nécessite les étapes suivantes :

- Désactiver le compte de l'utilisateur.
 - sur *GNU/Linux* en faisant `passwd -l <loginname>`
 - sur les *flavors xBSD* en préfixant le mot de passe du fichier `/etc/master.passwd` par le caractère `*` (cette technique marche aussi pour *GNU/Linux* dans le fichier `/etc/shadow`).
- Supprimer le compte de l'utilisateur.
 - Supprimer tous les processus de l'utilisateur.
 - Enlever les enregistrements concernant cet utilisateur des fichiers `/etc/passwd`, `/etc/group`, `/etc/shadow` ou `/etc/master.passwd`.
 - Retirer l'utilisateur des groupes secondaires.
 - Supprimer le fichier de courrier électronique de l'utilisateur.
 - Retirer ou rediriger les alias de courrier électronique de l'utilisateur.
 - Supprimer les tâches `cron` et/ou `at`.
 - Eventuellement faire une sauvegarde de l'espace de l'utilisateur puis le supprimer.
 - Supprimer les fichiers et/ou répertoires temporaires de l'utilisateur.
 - Mettre à zéro les quotas de cet utilisateur.

Un système Unix™ propose généralement un ensemble d'utilitaires pour faciliter la gestion des comptes. De plus, rien n'empêche de créer ces propres scripts pour

automatiser au mieux cette tâche.

2. Analyse du fichier `/etc/passwd`

Ce fichier au format texte, référence l'ensemble des utilisateurs du système. Une ligne concerne un utilisateur et est structurée en sept champs :

```
loginname:password:uid:gid:gecos:homedir:shell
```

- `loginname` : le nom de login de l'utilisateur.
- `password` : le mot de passe hashé, initialisé au caractère `*`.
- `uid` : identifiant utilisateur. Entier non signé codé sur quatre octets [0, 4294967296]. La valeur 0 donne les privilèges d'administration système, les valeurs inférieures à 10 sont, par convention, réservées à des comptes systèmes. Généralement, un utilisateur standard possède un uid supérieur à 100. Pour des raisons d'interopérabilité, il convient de se limiter aux 65536 premières valeurs.
- `gid` : identifiant du groupe principal de l'utilisateur. Entier non signé codé sur quatre octets mais dont l'usage est limité, comme précédemment. Les valeurs inférieures à 10 sont réservées aux groupes systèmes .
- `gecos` : zone de commentaire. On y met généralement le nom réel de l'utilisateur. Ce champ est utilisé par les commandes `chfn`, `mail` et `finger`.
- `homedir` : répertoire de domiciliation de l'utilisateur.
- `shell` : nom de l'interpréteur de commandes. Modifiable par la commande `chfn`.

Voici un exemple de fichier `/etc/passwd` d'un système FreeBSD obtenu à partir du fichier `/etc/master.passwd` :

```
# $FreeBSD: src/etc/master.passwd,v 1.25.2.6 2002/06/30 17:57:17 des Exp $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:/bin/tcsh
daemon:*:1:1:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5:System &:/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/sbin/nologin
tty:*:4:65533:Tty Sandbox:/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/sbin/nologin
games:*:7:13:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8:News Subsystem:/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/sbin/nologin
smpsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/sbin/nologin
mailnull:*:26:26:Sendmail Default User:/var/spool/mqueue:/sbin/nologin
bind:*:53:53:Bind Sandbox:/sbin/nologin
uucp:*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
pop:*:68:6:Post Office Owner:/nonexistent:/sbin/nologin
www:*:80:80:World Wide Web Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534:Unprivileged user:/nonexistent:/sbin/nologin

pascal①:*②:666③:666④:Pascal P;⑤:/home/pascal⑥:/bin/tcsh⑦

postfix:*:1001:1001:Postfix Mail System:/var/spool/postfix:/sbin/nologin
pgsql:*:70:70:PostgreSQL Daemon:/usr/local/pgsql:/sbin/nologin
mysql:*:88:88:MySQL Daemon:/var/db/mysql:/sbin/nologin
```

- ① `login`.
- ② `password`.
- ③ `uid`.
- ④ `gid`.
- ⑤ `gecos`.
- ⑥ `homedir`.
- ⑦ `shell`.

Le fichier `/etc/passwd` stockent les mots de passe sous forme hashée. Une fonction de hachage a la propriété d'être difficilement inversible, i.e. s'il est relativement aisé, connaissant l'algorithme de hachage, de calculer à partir d'un mot de passe en clair son hashage correspondant, la réciproque est un problème très difficile.

A la création d'un nouveau compte une étoile est placée dans le deuxième champ (`passwd`) du fichier `/etc/passwd`, elle permet d'empêcher l'accès au compte tant qu'un mot de passe n'est pas défini. **Ce champ ne doit jamais être vide !**

Le standard d'encryptage, par défaut, des mots de passe est DES, lequel n'encode que les 8 premiers caractères d'un mot de passe. Il est désormais considéré comme insuffisant au regard de la puissance de calcul d'un simple PC de bureau et c'est pourquoi les distributions *GNU/Linux* propose une alternative : le hashage *MD5*. De fait, *MD5* n'est pas meilleur que DES (du point de vue cryptographique), mais *MD5* n'impose pas de limite sur la longueur du mot de passe. Un mot de passe long est plus dur à casser donc plus sûr. *FreeBSD* de son côté utilise le hashage *MD5* par défaut, on peut éventuellement le remplacer par un autre algorithme tel que *blowfish* (c.f. le fichier `/etc/login.conf`, ligne : `passwd_format= ...`).

Note

Avec *Debian GNU/Linux* le système propose d'activer à l'installation l'encodage des mots de passe par *MD5*, valider toujours ce choix.

3. Analyse du fichier `/etc/group`

Ce fichier référence l'ensemble des groupes du système. Une ligne concerne un groupe et elle est structurée en quatre champs :

```
groupname:password:gid:member, ...
```

- `groupname` : le nom du groupe.
- `password` : le mot de passe hashé, initialisé au caractère `*`. En général, on n'utilise pas de mot de passe de groupe.
- `gid` : identifiant du groupe. Entier codé sur deux octets. Les valeurs inférieures à 10 sont réservées aux groupes systèmes .
- `member, ...` : liste des membres du groupe.

```
# $FreeBSD: src/etc/group,v 1.19.2.3 2002/06/30 17:57:17 des Exp $
#
wheel:*:0:root,pascal
```

```

daemon:*:1:daemon
kmem:*:2:
sys:*:3:
tty:*:4:
operator:*:5:root,pascal
mail:*:6:postfix
bin:*:7:
news:*:8:
man:*:9:
games:*:13:
staff:*:20:
sshd:*:22:
smb:*:25:
mailnull:*:26:
quest:*:31:
bind:*:53:
uucp:*:66:
xten:*:67:xten
dialer:*:68:
network:*:69:
www:*:80:
mysql:*:88:
pgsql:*:70:
nogroup:*:65533:
nobody:*:65534:
gnu:*:666:
postfix:*:1001:
maildrop:*:1002:
cvs:*:668:root

```

4. Sécurisation des comptes

Historiquement, les mots de passe étaient stockés dans le fichier `/etc/passwd` (universellement accessible en lecture)^[1], mais à mesure que les ordinateurs ont gagné en performance, il est devenu extrêmement dangereux de les laisser dans ce fichier.

4.1. GNU/Linux

GNU/Linux propose désormais de stocker les mots de passe dans un fichier à part : `/etc/shadow` (uniquement accessible au *super-utilisateur*. Ce mécanisme communément appelé shadow password est disponible sur toutes les distributions. En particulier, sur *Debian* cette option est explicitement proposée au moment de l'installation, il convient de l'adopter. La seule contre-indication à l'utilisation de cette stratégie concerne l'éventuel usage de votre machine comme serveur NIS/NFS.

Quand cette stratégie est activée, le champ mot de passe (deuxième champ) du fichier `/etc/passwd` est marqué par un x. Le fichier `/etc/shadow` contient donc l'encodage (ou hashage) des mots de passe et fournit des services supplémentaires non disponibles dans `/etc/passwd`. Attention, les deux fichiers sont nécessaires, l'un ne remplaçant pas l'autre !

Le fichier `/etc/shadow` est un fichier texte qui contient une ligne par utilisateur. Il est structuré en neuf champs, comme suit :

```
loginname:password:lcdate:mindays:maxdays:ndays1:ndays2:expiration:reserved
```

La sémantique associée à ces champs est la suivante :

- *loginname* : le nom de login de l'utilisateur, identique à celui du fichier `/etc/passwd` il permet ainsi d'établir le lien entre les deux fichiers.
- *password* : le hashage du mot de passe. S'il s'agit d'un hashage *MD5*, il commence par la séquence `1`.
- *lcdate* : date du dernier changement de mot de passe, exprimée relativement au temps Unix™, exprimé en jours, i.e. nombre de jours écoulés depuis le 1^{er} Janvier 1970. Généralement rempli par la commande `/usr/bin/passwd`
- *mindays* : Intervalle minimum de jours entre deux changements de mot de passe. Généralement positionné à 0, pour signifier qu'il n'y a pas d'intervalle.
- *maxdays* : Intervalle maximum de jours entre deux changements de mot de passe. Cela permet donc à l'administrateur de renforcer la stratégie de limitation dans le temps des mots de passe. Le délai maximum est obtenu comme somme de ce champ et du 7^e.
- *ndays1* : Nombre de jours, pour prévenir l'utilisateur, avant l'expiration du mot de passe.
- *ndays2* : Nombre de jours s'écoulant entre l'expiration du mot de passe et la désactivation du compte.
- *expiration* : Date d'expiration effective du compte, en nombre de jour depuis le 1^{er} Janvier 1970.
- *reserved* : Champ réservé pour un usage futur, vide pour le moment.

Un exemple typique de ligne extraite du fichier `/etc/shadow` :

```
pascal:$1$IzYw9H3u$Q7T2rxyPVKAlabue$.1n0:12144❶:0:180❷:7❸::12294❹:
```

- ❶ Le dernier changement de mot de passe a eu lieu le 2 avril 2003 (12144 jours depuis 01.01.1970) .
- ❷ Le mot de passe a une validité de 180 jours.
- ❸ Il y a un délai de 7 jours pour changer le mot de passe.
- ❹ Le compte expire le 30 août 2003 (12294 jours depuis le 01.01.1970).

Il existe un fichier `/etc/gshadow` dont le rôle est similaire à celui du fichier `/etc/shadow`, en pratique peu ou pas utilisé. On active rarement (voir jamais, par convention) les mots de passe de groupe.

4.2. xBSD

Les *flavors* xBSD stockent les comptes utilisateurs au sein du fichier `/etc/master.passwd`, uniquement accessible au *super-utilisateur*. Ce fichier est utilisé pour générer automatiquement le fichier `/etc/passwd`.

Ce fichier est au format texte et contient une ligne par utilisateur. Il est structuré en dix champs, comme suit :

```
loginname:password:uid:gid:userclass:change:expire:gecos:homedir:shell
```

La sémantique associée à ces champs est la suivante :

- *loginname* : le nom de login de l'utilisateur
- *password* : le hashage du mot de passe, préfixé par la séquence `1` s'il s'agit d'un hashage *md5* ou d'un `2` s'il s'agit d'un hashage *blowfish*.

- *uid* : identifiant utilisateur.
- *gid* : identifiant du groupe principal.
- *userclass* : classe d'appartenance de l'utilisateur telle que définie dans le fichier `/etc/login.conf`.
- *change* : exprimé en nombre de secondes à partir de l'origine des temps Unix (1^{er} Janvier 1970 [*Epoch*]), donne la date d'expiration du présent mot de passe. Si la valeur est à zéro, il n'y a pas d'expiration du mot de passe.
- *expire* : même unité que précédemment, mais exprimant la date d'expiration du compte. Si la valeur est à zéro, il n'y a pas d'expiration du compte.
- *gecos* : zone de commentaire, nom réel de l'utilisateur ...
- *homedir* : répertoire de domiciliation de l'utilisateur.
- *shell* : interpréteur de commande de l'utilisateur.

Un exemple typique de ligne extraite du fichier `/etc/master.passwd` :

```
pascal:$2a$15$WXXMWWxh1.4ZLw4DxnaquhJvv27WSkuRMnDBsZPzz11Fb0a686RS:666:666::1062450000❶:0❷:Pascal P;:/home/pascal:/bin/tcsh
```

- ❶ Expiration du présent mot de passe le 2 septembre 2003 à 01:00:00 (`date -r 1062450000`).
- ❷ Pas d'expiration du compte.

5. Commandes de gestion des utilisateurs

Dans ce qui suit les commandes sont préfixées par un # qui symbolise un accès *super-utilisateur* (*root*). Le mot qui précède ce caractère est le nom de machine (`tux` pour *GNU/Linux* et `chuck` pour *FreeBSD*).

Avertissement

Vous pouvez tester les commandes en utilisant l'accès étendu **sudo**, néanmoins je vous prie instamment de faire une copie initiale des fichiers que vous allez modifier. Puis quand vous aurez terminé ce paragraphe, de restaurer les fichiers dans leur état initiaux.

5.1. Gérer les utilisateurs avec `user{add,mod,del}` sous *GNU/Linux*

Première forme de la commande `useradd` : Invoquée sous cette forme, la commande crée un nouveau compte utilisateur en utilisant les valeurs spécifiées (sur la ligne de commande) et les valeurs système par défaut si les options ne sont pas précisées. Selon les options choisies, le nouveau compte sera créé et des fichiers initiaux seront copiés dans son espace de travail.

```
SYNOPSIS :
useradd [-c comment] [-d homedir] [-e expire_date] [-f inactive_date]
[-g initial_group] [-G group [ ,... ] ] [-m [-k skel_dir]]
[-p passwd] [-s shell] [-u uid [-o]] login

EXEMPLE : ajouter un utilisateur.

tux:/# useradd -c "Bernard's Account" -d /home/.bernard -e "2003-08-31" -s \
/bin/bash -u 668 -g gnu -G operator -m bernard
tux:/# passwd bernard
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
tux:/#
```

Deuxième forme de la commande `useradd` : Cette deuxième forme permet de lister les paramètres par défaut ou modifier les valeurs par défaut à partir des valeurs données en ligne de commande.

```
SYNOPSIS :
useradd -D [-g default_group] [-b default_home] [-f default_inactive]
[-e expire_date] [-s default_shell]

EXEMPLE : Lister les paramètres par défaut, puis modifier trois des paramètres.

tux:/# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel

tux:/# useradd -D -g 200 -s /bin/tcsh -e "2003-12-31"
tux:/# useradd -D
GROUP=200
HOME=/home
INACTIVE=-1
EXPIRE=2003-12-31
SHELL=/bin/tcsh
SKEL=/etc/skel
```

`usermod`. La commande `usermod` permet de modifier les caractéristiques d'un utilisateur.

```
SYNOPSIS :
usermod [-c comment] [-d homedir [-m]] [-e expire_date] [-f inactive_date]
[-g initial_group] [-G group [ ,... ] ] [-l loginname] [-p passwd]
[-s shell] [-u uid [-o]] [ -L | -U ] login

EXEMPLE : Ajouter l'utilisateur pascal dans le groupe operator
(il est déjà membre du groupe principal gnu et des groupes root et staff)
```

```
et lui mettre une date d'expiration au 31.12.2003 :
tux:/# usermod -G root,staff,operator -e 2003-12-31 pascal
```

userdel. La commande **userdel** permet d'enlever un compte utilisateur (s'il existe). L'option **-r** permet en plus la suppression du contenu et du répertoire de domiciliation de l'utilisateur, ainsi que son répertoire de mail.

```
SYNOPSIS :
userdel [-r] login

EXEMPLE : Enlever un utilisateur
tux:/# userdel -r bernard
```

5.2. Gérer les groupes avec group{add,mod,del} sous GNU/Linux

groupadd. La commande **groupadd** permet de créer un nouveau groupe.

```
SYNOPSIS :
groupadd [-g gid [-o]] login

EXEMPLE : Ajouter un groupe
tux:/# groupadd -g 666 gnu
```

groupmod. La commande **groupmod** permet de modifier les paramètres d'un groupe.

```
SYNOPSIS :
groupmod [-g gid [-o]] [-n group_name] group

EXEMPLE : Modifier le gid du groupe gnu.
tux:/# groupmod -g 667 gnu
```

groupdel. La commande **groupdel**, enlève un groupe (s'il existe).

```
SYNOPSIS :
groupdel group

EXEMPLE : Enlever le groupe gnu.
tux:/# groupdel gnu
```

5.3. Cas particulier de FreeBSD

La commande **pw** permet au *super*-utilisateur de gérer simplement les utilisateurs. Sa syntaxe est très proche de ces homologues « Linuxienne ». Cette commande fait les mises-à-jour dans les fichiers `/etc/passwd`, `/etc/master.passwd` et `/etc/group`. sous *OpenBSD*, par contre on utilise les mêmes commandes que pour *GNU/Linux*

- Première forme de la commande **pw useradd** :

```
SYNOPSIS :
pw [-V etcdir] useradd [ name | uid ] [-C config] [-q] [-n name] [-u uid]
[-c comment] [-d dir] [-e date] [-p date] [-g group] [-G grouplist]
[-m] [-k dir] [-w method] [-s shell] [-o] [-L class] [-h fd] [-N] [-P] [-Y]

EXEMPLE : Créer un nouvel utilisateur et lui attribuer un mot de passe
passwd :
chuck# pw useradd bernard -u 668 -c "Bernard's account" -d /home/.bernard -e \
"31-08-2003" -g gnu -G wheel,operator -m -s /bin/tcsh -N
chuck# passwd bernard
Changing local password for bernard.
New password:
Retype new password:
passwd: updating the database...
passwd: done
chuck#
```

- Deuxième forme de la commande **pw useradd** (sémantique équivalente à son homologue *Linux*) :

```
SYNOPSIS :
pw [-V etcdir] useradd [ name | uid ] -D [-C config] [-q] [-b dir]
[-e days] [-p days] [-g group] [-G grouplist] [-k dir] [-u min,max]
[-i min,max] [-w method] [-s shell] [-y path]
```

- **pw usershow**

```
SYNOPSIS :
```

```
pw [-V etcdir] usershow [ name | uid ] [-n name] [-u uid] [-F] [-P]
[-7] [-a]
```

EXEMPLE : Vérifier la présence de l'utilisateur *bernard* :

```
chuck# pw usershow bernard -P
Login Name: bernard          #668          Group: gnu          #666
Full Name: Bernard's account
Home: /home/.bernard        Class:
Shell: /bin/tcsh            Office: [None]
Work Phone: [None]          Home Phone: [None]
Acc Expire: Sun Aug 31 00:00:00 2003 Pwd Expire: [None]
Groups: wheel,operator
chuck#
```

- La commande **pw usernext** permet d'obtenir le prochain couple uid/gid disponible (séparé par les « : ») :

SYNOPSIS :

```
pw [-V etcdir] usernext [-C config] [-q]
```

EXEMPLE :

```
chuck# pw usernext
1002:1003
chuck#
```

- La commande **pw usermod** :

SYNOPSIS :

```
pw [-V etcdir] usermod [ name | uid ] [-C config] [-q] [-n name] [-u uid]
[-c comment] [-d dir] [-e date] [-p date] [-g group] [-G grouplist]
[-l name] [-m] [-k dir] [-w method] [-s shell] [-L class] [-h fd] [-N] [-P] [-Y]
```

EXEMPLE :

```
chuck# pw usermod -G wheel,operator,daemon,sys -n bernard
```

- La commande **pw userdel** :

SYNOPSIS :

```
pw [-V etcdir] userdel [ name | uid ] [-n name] [-u uid] [-r] [-Y]
```

EXEMPLE :

```
chuck# pw userdel -n bernard
```

- La première forme de la commande **pw groupadd** :

SYNOPSIS :

```
pw [-V etcdir] groupadd [ group | gid ] [-C config] [-q] [-n group] [-g gid]
```

EXEMPLE :

```
chuck# pw groupadd -g 666 -n gnu
```

- La deuxième forme de la commande **pw groupadd** :

SYNOPSIS :

```
pw [-V etcdir] groupadd [ group | gid ] [-C config] [-q] [-n group] [-g gid]
[-M members] [-o] [-h fd] [-N] [-P] [-Y]
```

- La commande **pw groupdel** :

SYNOPSIS :

```
pw [-V etcdir] groupdel [ group | gid ] [-C config] [-q] [-n group] [-g gid]
```

EXEMPLE :

```
chuck# pw groupdel gnu
... ou bien ...
chuck# pw groupdel -g 666
```

- La commande **pw groupmod** :

SYNOPSIS :

```
pw [-V etcdir] groupmod [ group | gid ] [-C config] [-q] [-n group] [-g gid]
```

```
[-l name] [-M members] [-m newmembers] [-h fd] [-N] [-P] [-Y]
```

- La commande **pw groupshow** :

SYNOPSIS :

```
pw [-V etcdir] groupshow [ group | gid ] [-g name] [-g gid] [-F] [-P] [-a]
```

EXEMPLE :

```
chuck# pw groupshow wheel
wheel:*:0:root,pascal
```

- La commande **pw groupnext** qui retourne le prochain `gid` disponible :

SYNOPSIS :

```
pw [-V etcdir] groupnext [ group | gid ] [-q]
```

EXEMPLE :

```
chuck# pw groupnext
1001
```

- La commande **pw lock** qui permet de verrouiller l'accès à un compte :

SYNOPSIS :

```
pw [-V etcdir] lock [ name | uid ] [-C config] [-q]
```

- La commande **pw unlock** qui a un effet symétrique à la commande précédente :

SYNOPSIS :

```
pw [-V etcdir] unlock [ name | uid ] [-C config] [-q]
```

5.4. Vérification

Sous *GNU/Linux* deux commandes permettent de vérifier l'intégrité des fichiers de gestion des utilisateurs. Ce sont la commande `/usr/sbin/pwck`, qui vérifie les fichiers `/etc/passwd` et `/etc/shadow` et la commande `/usr/sbin/grpck` qui vérifie les fichiers `/etc/group` et `/etc/gshadow`.

5.5. Edition en ligne

Elle se font par la commande **vipw** pour le fichier `/etc/passwd` (et respectivement `/etc/master.passwd` pour *FreeBSD* et `/etc/shadow` pour *GNU/Linux*) ; elle synchronise ces deux fichiers. La commande **vigr** agit de même sur les fichiers `/etc/group` et `/etc/gshadow` pour *GNU/Linux* uniquement.

6. Autres commandes

Tableau 1. Autres commandes

commande	sémantique
id	Imprime l'uid, le nom de login, le gid, le nom du groupe : pascal@chuck:C-AdmSys > id uid=666(pascal) gid=666(gnu) groups=666(gnu), 0(wheel), 5(operator)
groups	Affiche les groupes de l'utilisateur : pascal@chuck:C-AdmSys > groups gnu wheel operator
passwd	Permet à l'utilisateur de changer son mot de passe (moyennant la connaissance de l'ancien). Invoqué sous l'identité du <i>super</i> -utilisateur avec pour argument un nom d'utilisateur, elle permet de changer inconditionnellement son mot de passe.
su	Permet de changer d'identité. Typiquement permet à un utilisateur de devenir <i>super</i> -utilisateur à condition de connaître son mot de passe. Sur les <i>flavors</i> BSD, seuls les utilisateurs membre du groupe <i>wheel</i> sont autorisés à utiliser cette commande. Par défaut, les seules variables d'environnement modifiées sont : <code>USER</code> , <code>LOGNAME</code> , <code>HOME</code> et <code>SHELL</code> . La commande su - permet l'exécution d'un <i>login shell</i> et par conséquent le changement d'environnement.

7. Exercices

Les exercices suivants sont à pratiquer sous *FreeBSD* puis sous *Debian*. Indiquer sur papier votre réponse puis tester-là.

Exercice 1

A l'aide des utilitaires **grep** et **awk** déterminer si l'utilisateur *bin* existe sur votre système. Si oui, donner son uid.

Bien s'assurer de ne renvoyer au plus qu'une seule réponse !

Exercice 2

Quels sont les `gid` de vos différents groupes ?

Exercice 3

Quels sont les membres du groupe `operator` ?

Utiliser `grep` et `awk`

Exercice 4

[nécessite l'accès `sudo`] Créer l'utilisateur `test` avec la commande `useradd` en utilisant toutes les valeurs par défaut. Vérifier que le compte est bien créé.

Ajouter cet utilisateur au groupe `operator` et vérifier ensuite ces groupes d'appartenance

Se connecter à ce compte via le login puis via la commande `su`. Expliquer la différence.

Que faut-il faire pour que la connexion au login fonctionne pour cet utilisateur ?

Références

[1] MAN. *Manual pages*.

[2] NEMETH Evi, SNYDER Garth, et HEIN Trent H.. *Linux Administration Handbook*. Prentice Hall PTR. 2002.

^[1] Le « jeu » consistait à récupérer une copie du fichier `/etc/passwd` puis à faire tourner un programme de « crack ». Ce dernier passe en revue un dictionnaire de mots de passe qu'il a hashé un à un et compare au hashage présent dans le fichier des mots de passe. En cas d'égalité on a retrouvé le mot de passe (la fonction de hashage est injective). Ce genre de programme parcourt donc l'espace des mots de passe possible ; il lui faut donc de la puissance de calcul et du temps. Or les PC de bureau d'aujourd'hui ont une puissance de calcul non négligeable !